Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 46 No. 3 (2025)

Ddos Attack Detection in Ciciot2023 Dataset Using Two Dimensional Convolutional Neural Network

Gowsalya M[1*], Suganya S[2], Shunmuga Karpagam N[3]

[1, 2, 3] *Department of Computer Science and Engineering, Er.Perumal Manimekalai College of Engineering, Hosur, Tamil Nadu, India*

Abstract: *The way devices interact and communicate has been completely transformed by the Internet of Things (IoT), which has resulted in an exponential rise in data output. However, this surge in data also brings difficulties* in regard to privacy and classification, particularly in identifying malicious activities within IoT networks. The CICIoT 2023 dataset provides a comprehensive framework for evaluating deep learning models in both multi-class and binary classification tasks. This study employs a 2D Convolutional Neural Network (CNN) to classify IoT traffic, leveraging its Capability for collecting spatial hierarchies in the data. The goal of this study is to add to the increasing amount of research on IoT security by demonstrating the efficacy of 2D CNNs in classifying the CICIoT 2023 dataset, while also exploring the broader landscape of deep learning techniques used in this domain.

Keywords: *DDoS attack detection, deep learning, 2D CNN, IoT*

1. Introduction

The term "Internet of Things" (IoT) refers to a network of tangible items, or "things," that are equipped with software, sensors, and various other technologies that connect and share information with other systems and devices over the internet. [1].

Such connected objects may be whatever from modern household gadgets and appliances to automotive products, manufacturing facilities, or even whole towns.[2].

Innovations in cloud computing, artificial intelligence, and wireless connectivity are driving the Internet of Things' explosive growth. These technologies make it easy to collect, process, and automate data in a variety of fields. Because it increases productivity and lowers operating costs, IoT is essential to smart homes, healthcare, manufacturing, agriculture, and transportation. However, security, privacy, and data management issues are also brought up by the vast interconnectedness of gadgets. IoT ecosystems are at serious risk from data breaches, illegal access, and cyber-attacks. Strong encryption, authentication procedures, and intrusion detection systems are necessary to lessen these risks.

In order to maliciously disrupt the normal functioning of a targeted server, service, or network, a distributed denial-of-service (DDoS) attack consists of flooding the target or the surrounding infrastructure with Internet traffic[1]. DDoS attacks achieve effectiveness by utilizing multiple compromised computer systems as sources of attack traffic. Exploited machines can include computers and other networked resources such as IoT devices.

The malware types in the botnet, which are developed from Mirai and Bashlite, infect devices by taking advantage of security flaws and weak passwords. Malware payloads are downloaded and run during infection stages, connecting to command-and-control servers to receive assault instructions.

To overcome this kind of problems Comprehensive security measures must be put in place by business organizations to counter the growing threat of DDoS attacks. To find and fix possible flaws before they may be exploited, ongoing DDoS vulnerability testing is crucial.

To resolve this kind of problems Deep learning algorithms have been used more and more in recent years for tasks concerning the classification of IoT data. IoT devices are becoming more successful in a variety of applications like convolutional neural networks (CNNs), which enable them to identify intricate patterns and anomalies in data streams. CNNs have been used, to evaluate sensor data for predictive maintenance, environmental monitoring, and activity recognition. Furthermore, to improve the accuracy of time-series predictions, temporal dependencies in IoT data using recurrent neural networks (RNNs) and their variations, such as long short-term memory (LSTM) networks. The creation of intelligent IoT systems with the ability to make decisions on their own and react adaptively is greatly aided by these developments in deep learning.

2. Objectives

This research presents an advanced approach for DDoS attack detection using a **2D Convolutional Neural Network (2D CNN)** on the **CICIOT 2023** dataset for both **binary and multi-class classification**.

The key contributions of this study are as follows:

- We propose a **2D CNN-based deep learning model** for classifying cyberattacks in IoT networks, demonstrating its effectiveness for both **binary and multi-class attack detection**.
- We preprocess and transform the CICIOT 2023 dataset into a structured format suitable for deep learning, addressing challenges related to feature extraction and dimensionality reduction.
- We evaluating the proposed approach with different performance metrics as accuracy, precision, recall, and F1-score, false alarm rate.
- We analyze the impact of different hyper parameter configurations on classification performance, optimizing model depth, filter size, and activation functions for improved results.
- We provide insights into the real-world applicability of CNN-based DDoS detection in IoT environments, highlighting its potential advantages over conventional approaches.

These innovations open the door for better real-time attack detection and mitigation techniques by providing a scalable and efficient cyber security solution for IoT networks.

3. Methods

3.1 Pre-processing

The pre-processing is a crucial step for creating an efficient a robust approach. A CSV file, a popular format for tabular data storage, is used to load the dataset. To prevent the data loading process from failing because of small inconsistencies, the on_bad_lines="skip" argument is utilized for handling any malformed lines in the file. When working with huge datasets, when it is impractical to manually verify every line, this is crucial for resilience. The effectiveness of machine learning models can be greatly impacted by imperfections and discrepancies in the dataset, which must be eliminated or corrected through data cleaning. Incomplete data collection or mistakes in data input are just two of the many causes of missing values. To eliminate rows that contain missing values, utilize the drop function.

Label Encoder: The Label Encoder for transforming categorical labels into numerical format, utilize Scikit-learn's Label Encoder class. Since categorical data cannot be immediately processed by DL algorithms, this step is crucial. The model can more efficiently learn the connections between features and labels if labels are converted into numerical representation.

Standard Scaler: The Standard Scaler is used to standardize the features to have a mean of 0 and a standard deviation of 1. This process ensures that all features contribute equally to the model's learning process.

Dimensionality Reduction: Reducing the number of attributes in a dataset while keeping the majority of its variation is known as dimensionality reduction. This stage is crucial for lowering the chance of overfitting and increasing the effectiveness of the algorithm's training phase.

Principal Component Analysis: PCA, is a prevalent dimensionality reduction method that preserves the majority of the variation while converting the data into a lower-dimensional space. This code uses PCA to cut the feature count from 46 to 40. For high-dimensional datasets, where there may be significantly more features than samples, this phase is essential. The framework can learn more effectively and generalize to new data more effectively by lowering the dimensionality.

Splitting Data: Train_test_split is used to divide the dataset into training and testing sets with an 80-20 split ratio. In order to assess the model's performance on unknown data, this stage is essential. The model is trained on the training set, and its performance is assessed on the testing set. An objective assessment of the model's efficacy can be obtained by employing an independent testing set.

Handling Class Imbalance: When there are fewer samples in each class, there is a class imbalance problem occurs. Poor model performance may result from this, particularly for the minority class.An advanced approach for addressing class imbalance in datasets, especially when dealing with binary classification issues, is borderline-SMOTE. It focuses on creating synthetic samples close to the class boundary and is an extension of the Synthetic Minority Over-Sampling Technique (SMOTE).it reduces overfitting and also better for handling noisy data.

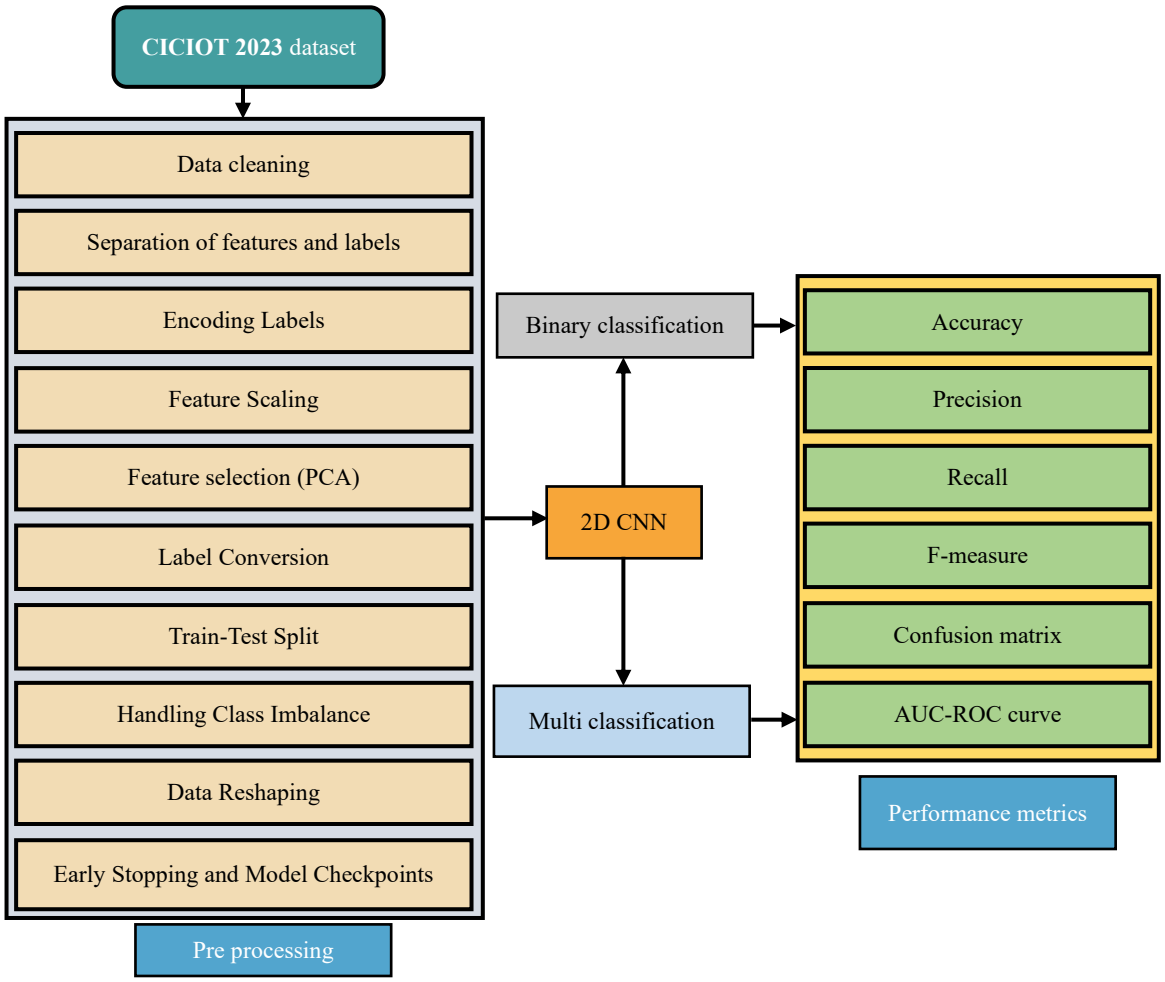

Figure 1: workflow diagram

CNN reshaping: CNNs require input data to have a certain shape, usually consisting of channels, width, and height. This function transforms the features into a CNN-compatible 2D format. CNNs are strong models for image classification problems, and this step is essential for training them.so the process of reshaping is very essential for CNNs.
Normalization: To guarantee that the pixel values fall within the anticipated range for CNNs, the reshaped data is normalized to the interval [0, 1]. One crucial preprocessing step that may result in a big impact on the model's performance is normalization. The model could find it difficult to identify the underlying patterns in the data without normalization.

Preventing Over fitting: The over fitting issue arises when neural networks are being trained. In this study, early stopping and dropout layers are successful strategies for preventing over fitting. In order to prevent over fitting of the training data, early stopping was employed at the beginning, allowing the algorithms to run for two more rounds before stopping. Additionally, dropout layers were employed, which randomly remove specific neurons during training to stop them from controlling the learning process.

Normalization of batches: Using batch normalization layers lowers the chance of over fitting by stabilizing and speeding up training and serving as a regularizer.

Activation functions: An essential part of neural networks are activation functions, which decide the network's output based on an input or collection of inputs. Without activation functions, neural networks would simply be linear models that could only learn linear relationships. They allow the model to learn and represent complex patterns by introducing non-linearity. A neuron's input signal is converted into the output signal by activation functions. The degree to which a neuron must be stimulated is determined by this transition. Depending on the objective, multiple activation functions can be helpful in limiting the outcome to different ranges. For instance, sigmoid is appropriate for probability estimation since it produces results in the range of 0 and 1.

ReLU: One widespread activation function in neural networks, especially in deep learning models, is the Rectified Linear Unit (ReLU). The ReLU activation function is included in the hidden layers in order to add non-linearity and facilitate the model's effective learning of intricate patterns. The mathematical definition of the ReLU function is:

$$ReLU(x) = \max(0, x) \quad (1)$$

This indicates that the output of the ReLU function for each given input, x is 0 else x , if x is greater than zero.

Sigmoid: the sigmoid activation function utilized to generate a probabilistic score between 0 and 1, which represents the possibility of the positive class, in the output layer of the binary classification model. The mathematical definition of the sigmoid function is:

$$\sigma(x) = \frac{1}{1+e^{-x}} \quad (2)$$

x is input function, e is the base of the natural algorithm

Softmax: The Softmax activation function utilized to generate a probability distribution across several classes in the output layer of the multi-class classification model, enabling the machine to categorize inputs into a number of groups. The mathematical definition of the softmax function is:

$$softmax\ (z_i) = \frac{e^{z_i}}{\sum j e^{z_j}} \quad (3)$$

z_i Is the ith element of input vector z, e is the base of the natural logarithm, all of the input vector's components are added up in the denominator.

4. Methods & Pre-Processing

4.1 Proposed method

Deep learning has developed as a high-performance, effective machine learning technique for handling extremely challenging classification and prediction issues. Convolutional Neural Networks (CNNs) are now the preferred architecture for tasks like segmentation, object identification, and image classification, having completely transformed the fields of computer vision and image processing.

Online services' dependability and availability are seriously threatened by Distributed Denial of Service (DDoS) assaults. These attacks cause a target system to become unreachable to authorized users by flooding it with traffic from various sources. Conventional DDoS detection techniques frequently find it difficult to stay up with the changing strategies used by attackers. In order to detect these harmful behaviors, 2D Convolutional Neural Networks (CNNs) provide a strong and flexible method. It created for image processing applications, still 2D CNNs have demonstrated incredible promise in the field of cyber security, especially in the detection of DDoS attacks. CNNs are very good at identifying hierarchical structures and geographical patterns in data. These patterns may match unusual traffic patterns suggestive of an assault in the context of DDoS detection. CNNs are capable of adaptively learning from data, which allows them to spot intricate and complex patterns that static rules might detect. This is in contrast with typical rule-based systems. It is possible to convert network traffic data into a 2D grid-like structure that resembles an image. This change enables CNNs to examine traffic patterns by utilizing their advantages in spatial data processing.in 2D CNN the filter moves in two directions but in CNN the filter only moves in single direction.

4.2 Evaluation Matrix

One of the criteria used to evaluate the models' performance is a confusion matrix. False positive (FP), false negative (FN), true positive (TP), and true negative (TN) are the four factors that make up the confusion matrix.

Accuracy: The frequency with which the trained models accurately identify the intended attacks is shown by accuracy.

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FN} + \mathrm{FP} + \mathrm{TN}} \quad (4)$$

Precision: which indicates the TP recommended by the classifier, characterizes the model's performance. The precision is calculated by dividing the total number of positive predictions by the number of TP. It is computed using

$$precision = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \quad (5)$$

Recall: In classification problems, recall also referred to as sensitivity or true positive rate It assesses a model's capacity to locate every significant instance in a dataset. Recall is defined mathematically as:

$$recall = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \quad (6)$$

F-measure: A measure of effectiveness called the F1 score is used to assess a classification model's accuracy. It is especially helpful in situations when you have an unequal distribution of classes and need to strike a balance between recall and precision. A statistic that takes into account both false positives and false negatives is the F1 score, which is calculated as the harmonic mean of precision and recall.

$$F - measure = 2 \times \frac{(\mathrm{precision} \times recall)}{(\mathrm{precision} + recall)} \quad (7)$$

FAR (false alarm rate): it is a performance metric applied to tasks involving binary categorization. It calculates the percentage of negative cases that the model mistakenly classifies as positive. The False Alarm Rate can be expressed mathematically as follows:

$$FAR = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \quad (8)$$

4.1 Result analysis

An essential component of a research study is this section. It offers a thorough analysis of the main conclusions, making use of tables and textual explanations to ensure clarity and conciseness. The suggested methodology's total performance is graphically depicted through graphs, making it possible to spot trends in accuracy and loss across many test situations.

Experimental setup

The Python programming language was used in this work to implement the models. The experiment was carried out using a Jupyter Notebook. The DL models were implemented using the DL application programming interface (API) libraries sci-kit-learn, matplotlib, pandas, Keras, and scipy, tensorflow, numpy.

4.4.1. Classification of CICIoT2023 dataset

The CICIoT2023 dataset is used in the experiments for both binary and multi-class classification.

4.4.2. Binary Classification

The CICDDoS2023 dataset produces robust performance in binary classification. The performance measures shown in table 1.

Table 1: Performance analysis for binary classification using CICIoT2023 dataset

Method	Accuracy	Precision	Recall	F-measure	FAR	Testing time /s
2D CNN	99.44	99.13	99.76	99.44	0.86	62

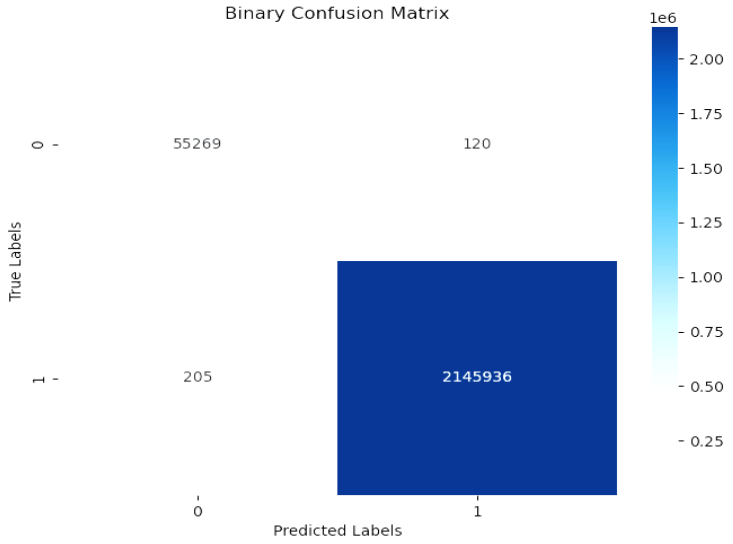

Figure 2: Confusion matrix for binary classification using CICIoT2023 dataset

The performance of our binary classification model is thoroughly assessed in the confusion matrix shown in Figure 2. By contrasting the real labels with the predicted labels, this matrix is an essential tool for evaluating the model's correctness and predictability. The matrix provides a clear visual representation of the classification results, with the true labels on the vertical axis and the predicted labels on the horizontal axis.

TN: This value, which is located in the upper-left cell, shows that 55,269 cases were accurately classified as negative (class 0) by the model. This shows that the model can correctly identify negative situations, which is essential to comprehending its specificity.

FP: This result, which is shown in the upper-right cell, indicates that 120 cases were mistakenly classified by the model to be positive (class 1) whereas they actually proved negative. Monitoring false positives is crucial, particularly in situations where making the wrong positive prediction could result in needless expenses or actions.

FN: This value, which is located in the bottom-left cell, shows that 205 cases were mistakenly labeled as negative when they were actually positive. False negatives are especially important in situations like medical diagnosis when failing to detect a positive case might have severe consequences.

TP: This value, which is located in the bottom-right cell, shows that 2,145,936 cases were correctly classified as positive by the model. The high percentage of true positives highlights how well the program can identify positive cases.

The model's ability to detect positive cases is demonstrated by the high number of true positives, and its balanced performance is suggested by the comparatively small amount of false alarms and false negatives.

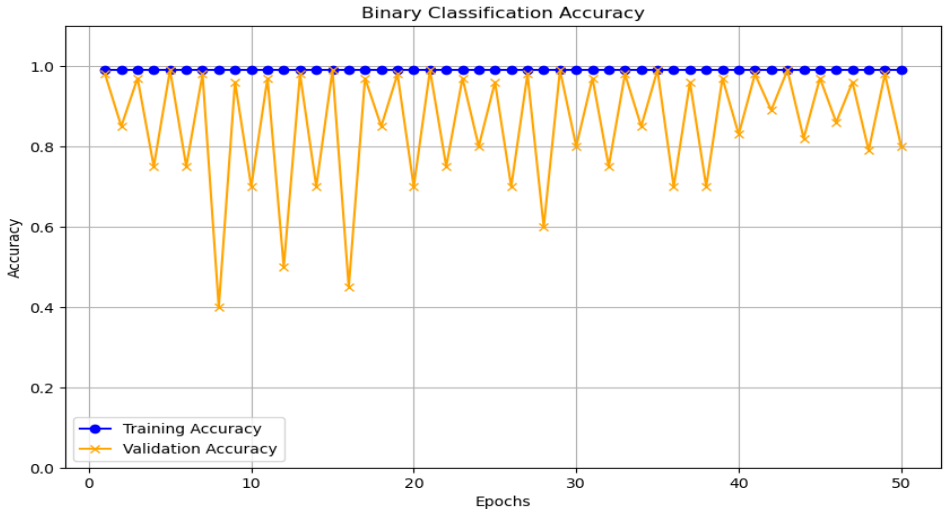

Figure 3: Training and validation accuracy for binary classification using CICIoT2023 dataset

The figure 3 shows the training and validation accuracy chart for binary classification, during each of the 50 epochs, the training accuracy stays high, near 1.0. This shows that the model has successfully learned from the training data, as evidenced by the dataset's almost flawless classification accuracy. At regular intervals, the validation accuracy fluctuates, ranging from roughly 0.8 to 1.0. This pattern indicates that there is little variation in the model's predicted accuracy and that it performs well on the validation data. The model's excellent learning capability is demonstrated by the high training accuracy, which shows that it has successfully minimized errors on the training dataset. Although there are minor variations, the validation accuracy is still quite high and consistent. This implies that the model performs well when applied to unknown data, which is a sign of its dependability and resilience. The model's performance throughout training and validation datasets is shown in detail in the graph, which shows a typically good and reliable performance. In order to create reliable binary classification models, the model exhibits strong learning and generalization skills. The model's ability to produce accurate and dependable predictions in real-world situations is demonstrated by this strong performance.

4.4.3. Multi-Classification

CNNs, have shown themselves to be quite successful at classification problems in the field of deep learning. However, conventional CNN architectures could find it difficult to attain high accuracy when datasets became increasingly varied and complicated. Due to a complication in the dataset the 2D CNN model produced less accuracy to over comet hat issues, Combining Residual Networks (ResNet) with 2D CNNs provides a solid solution to this problem by utilizing the advantages of both architectures.

The CICDDoS2023 dataset produces better performance in multi-classification. The performance measures shown in table 2.

Table 2: performance analysis for Multi classification using CICIoT2023 dataset

Method	Accuracy	Precision	Recall	F-measure	FAR	Testing time /s
2D CNN	95.45	94.13	96.69	95.39	5.71	183

The 2D CNN model exhibits good classification accuracy across most classes, as can be shown by looking at the confusion matrix. With only 120 cases of Class 0 being mistakenly categorized as Class 1 and 50 cases from other classes being mistakenly labeled as Class 0, the model demonstrates a high degree of precision. Interestingly, Class 0 shows strong performance with 122,629 true positives. Class 1, which scores an impressive 982,563 true positives with little misclassifications (only 50 cases incorrectly categorized as Class 0 and 2,571 as Class 3), further demonstrates the resilience of the model. This excellent accuracy also holds true for other classes. Only 22 cases are incorrectly classified as Class 5, whereas Class 2 achieves 827,225 true positives. Class 3 exhibits significant overlap with Class 4, with 244 occurrences misclassified, although it records 552,726 genuine positives.
Having 608,955 true positives, Class 4 performs well; nevertheless, there are a few minor misclassifications, including 129 cases that were misclassified as Class 6.
Class 5 produces 105,826 genuine positives and only a few misclassifications, such as 35 cases that were mistakenly classified as Class 2. Having 882,725 true positives and only 2,736 cases incorrectly categorized as Class 0, Class 6 performs quite well. Despite the model's remarkable overall performance, there is room for improvement in handling the few misclassifications that were noticed, like the 244 cases in which Class 3 was incorrectly categorized as Class 4. These findings imply that specific improvements could improve the accuracy and dependability of the model even further.

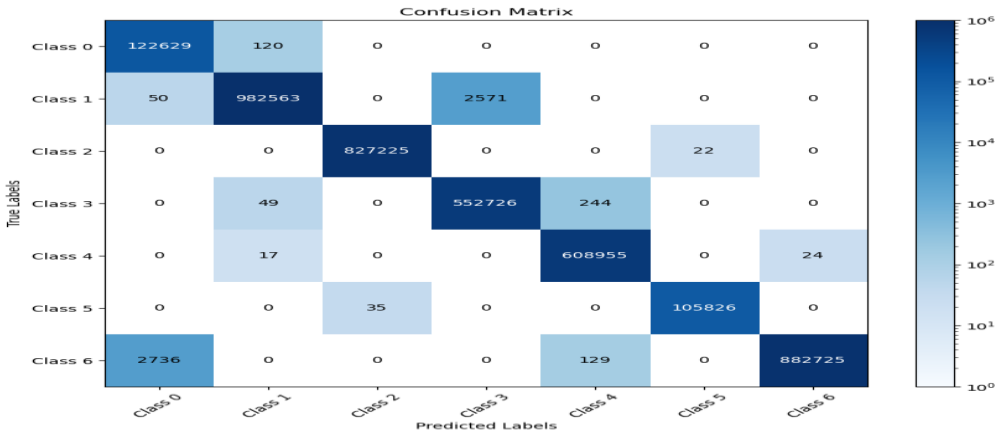

Figure 4: confusion matrix for Multi-classification using CICIoT2023 dataset

The figure 5 shows the training and validation accuracy chart for multi classification, Over the course of the 50 epochs, the training accuracy stays comparatively constant after starting out high, near 1.0. This implies that the model can fit the training data quite effectively right away. Around 0.2 is the initial validation accuracy, which is lower than the training accuracy. As the number of epochs increases, the validation accuracy progressively rises. Particularly after about 20 epochs, there is a discernible variation in the validation accuracy, suggesting some instability in the model's performance on unseen data. Although it occasionally peaks, the validation accuracy never quite reaches the training accuracy level. The model may be overfitting to the training data, as indicated by the notable discrepancy between training and validation accuracy, particularly in the early epochs. When a model learns the training data including its noise and outliers too well and performs badly on fresh, unknown data, this is known as overfitting.
Given that the model's performance on validation data is inconsistent, the variation in validation accuracy lends more credence to the idea of overfitting.

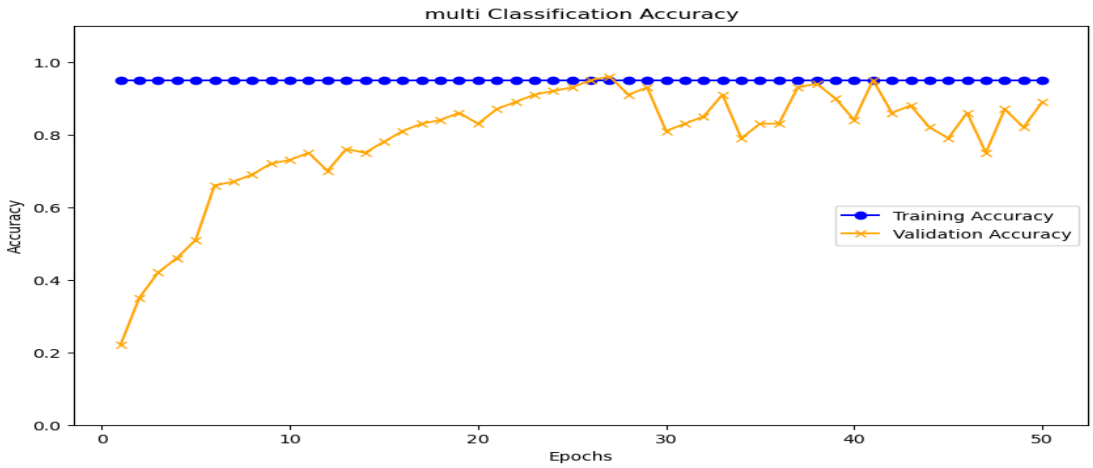

Figure 5: training and validation accuracy for Multi-classification using CICIoT2023 dataset

The model's validation accuracy is lower and varies, suggesting possible overfitting, even though the training accuracy is good. Techniques like cross-validation, regularization, or changing the model architecture could be taken into consideration to enhance the model's performance on unknown data.

5. Related works

Deep learning-based DDoS attack detection has gained significant attention due to the increasing frequency and complexity of cyber-attacks. This section examines previous research on deep learning-based ddos attack detection, with an emphasis on model designs, evaluation metrics, and dataset usage. Although various approaches have been proposed, challenges remain in handling imbalanced datasets, real-time detection, and model interpretability.

J. Zhao et al. (2023) [3] proposed new approach as self-attention mechanism with CNN-BiLSTM, the RF algorithm and Pearson correlation analysis are combined to select key features as model inputs to reduce the redundancy of input data. 1D CNN and BiLSTM are then used to extract spatial and temporal features, respectively.

V. R. S. Dora et al. (2022) [4] CNN-O-LSTM is the suggested hybrid algorithm. With the goal of decreasing the correlation between the features, the GWO is utilized for the best feature selection. The detection performance is improved when properly chosen features are combined with CNN features.

H. Kumar et al. (2022) [5] the CNN-RF revealing model was then employed to detect minor DDoS attacks at the gateway. Within a 120-second window, it can detect four different types of low-rate DDoS attacks.

A. A. Najar et al.[6] High performance in binary and multi-classification is attained by the suggested model, which additionally provides comprehensive information to a specified email address. Additionally, they conducted a number of experiments in three different scenarios to assess the efficacy and efficiency of the suggested DDoS mitigation system.

M. A. Setitra et al. (2023) [7] The detection efficiency is increased by combining CNN and MLP. The significance of the approach is further reinforced by the employment of a Bayesian optimizer for hyper parameter tuning and SHAP for feature contributions

C. Padmavathy *et al.* (2024) [8] 1D CNN, which focuses on connectivity of brain systems from a range reliant factors serves as an example of these mathematical functions, which are structured to represent intricate connections between reliant factors. They predict three different kinds of attacks.

B. Goparaju et al. (2023) [9] the Features are extracted from the source network activity data using the suggested 1D CNN framework. Additionally, by decreasing the data's dimension, this process eliminates the data's redundancy.

B. A. Alabsi et al. (2023) [10] strategy that combines CNN and CNN. From the raw network traffic data, the initial CNN model is used to identify the important aspects that aid in the identification of IoT attacks. To create a strong detection model that reliably identifies IoT assaults, the second CNN makes use of the attributes found by the first CNN.

B. B. Gupta et al. (2023) [11] provides a novel Deep CNN-based architecture for detecting DDoS attacks in IoT networks. Through the collection of geographical and temporal patterns in data, the methodology fully utilizes CNNs' inherent capabilities in detecting DDoS attacks.

M. B. Anley et al. (2024) [12] suggested method for DDoS detection that makes use of transfer learning techniques, CNN, and adaptive architectures. According to experimental findings, the suggested adaptive transfer learning approach successfully recognizes harmful and benign activity as well as certain attack types.

P. Shorubiga et al. (2023) [13] revolved around DDoS attacks using HTTP flooding. The suggested approach identifies the HTTP flooding attack early on by putting into practice a model for detection based on 1-D CNN.

P. Gahelot et al. (2023) [14]demonstrated the CNN method for identifying botnet activity in IoT networks and devices. Because of its improved precision, this model has a low loss. CNN has the ability to automatically learn the most pertinent elements.

S. Yaras et al. (2024) [15]The model is trained and tested using the "CICIoT2023" and "TON_IoT" datasets. the correlation approach is used to decrease the characteristics in the datasets. LSTM and 1D CNN are used in the building of a hybrid deep learning algorithm.

A. Gueriani et al. (2024) [16] By utilizing CNN's spatial feature extraction capabilities for pattern recognition and LSTM's sequential memory retention for identifying intricate temporal dependencies.

A. Gueriani et al. (2024) [16] provides a hybrid model that uses SVM as a classifier and CNN for feature extraction. This approach categorizes attacks as either benign or DDoS by using a CNN to extract key components from the data. The findings, which were trained on the actual CICIoT2023, demonstrate the potential performance.

Z. S. Mahdi et al. (2024) [17] suggested a hybrid feature selection technique that combines a correlation coefficient and a sequential feature selector. a hybrid model is proposed by combining a cascaded LSTM and a Naive Bayes classifier.CIC-DDoS2019, CIC-IoT2023, and CIC-IoV2024 were used for training and performance evaluation; these data were also balanced to produce useful outcomes.

S.-M. Tseng et al. (2024) [18] Substantial data on IoT environments can be found in the CIC-IoT-2023 dataset. They evaluated the network traffic characteristics using seven deep learning models, such as Transformer, and use binary and multivariate classifications to spot unusual activity and possible intrusions.

S. Abbas *et al.* (2024) [19] Used three models as RNN, CNN ,DNN For every model, three variations were tested. Each of these variations is tailored and adjusted in a unique way to assess the effectiveness of the recommended approach. The importance of matching training and validation accuracy curves that show controlled overfitting is also highlighted by this noteworthy observation.

C. Ejikeme et al. (2024) [20] suggested model performs better than RNN, BiLSTM, and Transformer RNN models, according to an experimental evaluation conducted utilizing the CICDDoS2019 and CICIoT2023 datasets, which cover a variety of DDoS attack types and network conditions.

N. Pandey et al. (2024) [21]suggested a two-tier hybrid strategy for Internet of Things networks that uses entropy variation to separate attack traffic from first-tier benign traffic. The analysis's conclusions were confirmed using the CICIoT2023 dataset, yielding comparable results.

A. D. Aguru et al. (2024) [22] This study's novel anomaly-based IDS framework uses suggested stacked mGRU to detect and identify multi-vector DDoS attacks in mobile healthcare informatics systems. The BCE and the MCE, two examples of IDS, have been developed to generate results that are customized for each user.

6. Materials

The CICIoT2023 [23] dataset is used as a benchmark dataset to assess cybersecurity models in Internet of Things (IoT) environments . The CICIoT2023 was created by the Canadian Institute for Cyber security and provides a realistic representation of attacks in an IoT topology composed of 105 devices. This dataset includes 33types of attacks, divided into groups like web, brute force, spoofing, DDoS, DoS, recon, and Mirai. With an incredible record of 46,686,579 occurrences overall, the CICIoT2023 offers 47 unique attributes. It supports both binary and multi-class classification tasks in DDoS attack detection by offering a sizable real-time dataset with a variety of assault scenarios. Deep learning models are frequently trained and tested using this dataset, especially for detecting DDoS attacks and other cyber threats in Internet of Things networks. CICIoT2023 uses actual IoT devices to carry out attacks, in contrast to many datasets that either mimic attacks or utilize non-IoT devices to carry them out. Because it offers a more realistic depiction of how assaults could materialize in real-world IoT settings, the dataset is extremely useful for creating workable security solutions. the figure 6 represents what are types attack types are in CICIoT2023 dataset. The dataset, which is accessible in both pcap and csv formats, provides features that were taken from network traffic. This adaptability enables scientists to use the data in different ways based on their needs for analysis. Additionally, the feature extraction procedure is flexible enough to allow for the insertion of new characteristics as needed.

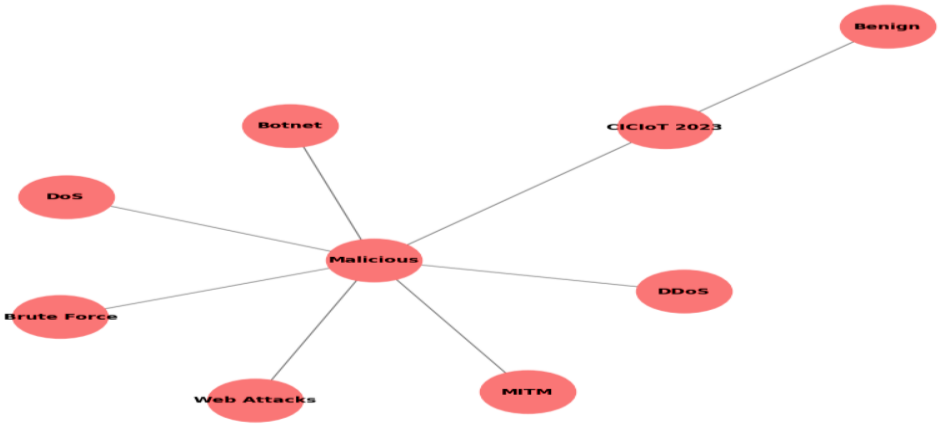

Figure **6** : Attack types

In order to remain relevant as IoT technology advances, the dataset takes consideration of potential protocols and attack methods. CICIoT2023 is a significant long-term resource for scientists and programmers because of its progressive methodology.

7. Conclusions

By utilizing cutting-edge deep learning methods, particularly 2D Convolutional Neural Networks (CNNs), the study described in this paper solves the crucial problem of Distributed Denial-of-Service (DDoS) attack detection in Internet of Things (IoT) networks. By proving that deep learning models are effective at detecting and preventing DDoS attacks, which severely compromise the dependability and accessibility of online services, the study's conclusions provide a fundamental contribution to the field of IoT security. The study clearly illustrates how 2D CNNs may be used for binary and multi-class classification tasks in order to identify DDoS attacks in Internet of Things environments. The model is especially well-suited for examining network traffic patterns suggestive of DDoS attacks due to its capacity to capture spatial hierarchies in data.

The suggested model was trained and assessed using the CICIoT 2023 dataset, which offers a thorough and accurate depiction of IoT network threats. The size and diversity of this dataset made it possible to thoroughly test the model's performance in a range of assault scenarios.

In both binary and multi-class classification tasks, the 2D CNN model demonstrated remarkable accuracy, as evidenced by performance indicators including precision, recall, and F1-score. The resilience and generalization properties of the model are demonstrated by its ability to maintain high accuracy on both training and validation datasets.
The model's ability to accurately detect true positives and minimize false alarms is demonstrated by the confusion matrices for both binary and multi-class classification. This is important for real-world implementation.

The quality and consistency of the incoming data were guaranteed by the use of efficient preparation procedures, such as data cleaning, label encoding, and standardization. In order to improve model performance and efficiency, dimensionality reduction techniques like Principal Component Analysis (PCA) were used to shrink the feature space while keeping important information. By using methods like SMOTE, which helps in creating synthetic samples to balance the class distribution, the study tackles the problem of class imbalance in the dataset. This method improves overall detection performance by strengthening the model's capacity to identify instances of the minority class.

In order to avoid overfitting and ensure the model performs well when applied to new data, techniques like batch normalization, dropout layers, and early stopping were used. In real-world applications, these methods support the stability and dependability of the model. The architecture and training techniques of the suggested model can be expanded and modified to handle bigger datasets and changing cyber threats, guaranteeing consistent performance in dynamic IoT contexts.

To improve DDoS detection capabilities, the results of this study can be incorporated into current cybersecurity frameworks. Working together with industry stakeholders can make it easier to use these models in practical situations, enhancing IoT networks' security posture. Still the model falls under overfitting and under fitting problem in future work these queries are resolved.

In order to capture both spatial and temporal relationships in network traffic data, future research should investigate hybrid models that combine CNNs with other deep learning architectures, such as Transformers or Recurrent Neural Networks (RNNs). Furthermore, examining the effects of various hyper parameter setups and optimization strategies might improve model performance even further.

References

[1] https://www.oracle.com/in/internet-of-things/ (accessed.

[2] M. Tealab, A. Hassebo, A. Dabour, and M. AbdelAziz, "Smart cities digital transformation and 5G–ICT architecture," in *2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, 2020: IEEE, pp. 0421–0425.

[3] J. Zhao, Y. Liu, Q. Zhang, and X. Zheng, "CNN-AttBiLSTM Mechanism: A DDoS Attack Detection Method Based on Attention Mechanism and CNN-BiLSTM," *IEEE Access,* vol. 11, pp. 136308–136317, 2023.

[4] V. R. S. Dora and V. N. Lakshmi, "Optimal feature selection with CNN-feature learning for DDoS attack detection using meta-heuristic-based LSTM," *International Journal of Intelligent Robotics and Applications,* vol. 6, no. 2, pp. 323–349, 2022.

[5] H. Kumar, Y. Aoudni, G. G. R. Ortiz, L. Jindal, S. Miah, and R. Tripathi, "Light weighted CNN model to detect DDoS attack over distributed scenario," *Security and Communication Networks,* vol. 2022, no. 1, p. 7585457, 2022.

[6] A. A. Najar and S. M. Naik, "Cyber-secure SDN: A CNN-based approach for efficient detection and mitigation of DDoS attacks," *Computers & Security,* vol. 139, p. 103716, 2024.

[7] M. A. Setitra, M. Fan, B. L. Y. Agbley, and Z. E. A. Bensalem, "Optimized MLP-CNN model to enhance detecting DDoS attacks in SDN environment," *Network,* vol. 3, no. 4, pp. 538–562, 2023.

[8] C. Padmavathy *et al.*, "1D CNN Based Model for Detection of DDoS Attack," in *2024 2nd International Conference on Device Intelligence, Computing and Communication Technologies (DICCT)*, 2024: IEEE, pp. 1–6.

[9] B. Goparaju and B. S. Rao, "Distributed Denial-of-Service (DDoS) Attack Detection using 1D Convolution Neural Network (CNN) and Decision Tree Model," *Journal of Advanced Research in Applied Sciences and Engineering Technology,* vol. 32, no. 2, pp. 30–41, 2023.

[10] B. A. Alabsi, M. Anbar, and S. D. A. Rihan, "CNN-CNN: dual convolutional neural network approach for feature selection and attack detection on internet of things networks," *Sensors,* vol. 23, no. 14, p. 6507, 2023.

[11] B. B. Gupta, A. Gaurav, V. Arya, and P. Kim, "A deep CNN-based framework for distributed denial of services (DDoS) attack detection in internet of things (IoT)," in *Proceedings of the 2023 international conference on research in adaptive and convergent systems*, 2023, pp. 1–6.

[12] M. B. Anley, A. Genovese, D. Agostinello, and V. Piuri, "Robust DDoS attack detection with adaptive transfer learning," *Computers & Security,* vol. 144, p. 103962, 2024.

[13] P. Shorubiga and R. Shyam, "Cnn-based model for the http flood attack detection," in *2023 International Conference for Advancement in Technology (ICONAT)*, 2023: IEEE, pp. 1–6.

[14] P. Gahelot, P. K. Sarangi, and L. Rani, "Intelligent detection of ddos attack in IOT Network," in *Mobile Radio Communications and 5G Networks: Proceedings of Third MRCN 2022*: Springer, 2023, pp. 173–184.

[15] S. Yaras and M. Dener, "IoT-Based Intrusion Detection System Using New Hybrid Deep Learning Algorithm," *Electronics,* vol. 13, no. 6, p. 1053, 2024.

[16] A. Gueriani, H. Kheddar, and A. C. Mazari, "Enhancing IoT Security with CNN and LSTM-Based Intrusion Detection Systems," in *2024 6th International Conference on Pattern Analysis and Intelligent Systems (PAIS)*, 2024: IEEE, pp. 1–7.

[17] Z. S. Mahdi, R. M. Zaki, and L. Alzubaidi, "Advanced hybrid techniques for cyberattack detection and defense in IoT networks," *Security and Privacy,* p. e471, 2024.

[18] S.-M. Tseng, Y.-Q. Wang, and Y.-C. Wang, "Multi-Class Intrusion Detection Based on Transformer for IoT Networks Using CIC-IoT-2023 Dataset," *Future Internet,* vol. 16, no. 8, p. 284, 2024.

[19] S. Abbas *et al.*, "An efficient deep recurrent neural network for detection of cyberattacks in realistic IoT environment," *The Journal of Supercomputing,* pp. 1–19, 2024.

[20] C. Ejikeme, N. Kahani, and S. A. Ajila, "Optimizing DDoS Detection with Time Series Transformers," in *2024 34th International Conference on Collaborative Advances in Software and COmputiNg (CASCON)*, 2024: IEEE, pp. 1–6.

[21] N. Pandey and P. K. Mishra, "Devising a hybrid approach for near real-time DDoS detection in IoT," *Computers and Electrical Engineering,* vol. 118, p. 109448, 2024.

[22] A. D. Aguru and S. B. Erukala, "A lightweight multi-vector DDoS detection framework for IoT-enabled mobile health informatics systems using deep learning," *Information Sciences,* vol. 662, p. 120209, 2024.

[23] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, "CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment," *Sensors,* vol. 23, no. 13, p. 5941, 2023.