Journal of Network and Systems Management (2023) 31:2
https://doi.org/10.1007/s10922-022-09697-x

Efficient Intelligent Intrusion Detection System for Heterogeneous Internet of Things (HetIoT)

Shalaka Mahadik[1] · Pranav M. Pawar[1] · Raja Muthalagu[1]

Received: 18 February 2022 / Revised: 2 September 2022 / Accepted: 26 September 2022

Abstract

Moving towards a more digital and intelligent world equipped with internet-of-thing (IoT) devices creates many security issues. A distributed denial of service (DDoS) attack is one of the most formidable and challenging security threats that has taken hold with the emergence of the heterogeneous IoT (HetIoT). The massive DDoS attacks have exhibited their impact by continuously destroying a variety of infrastructures, resulting in huge losses, and endangering the overall availability of the digital world. The emphasis of this research is to identify and mitigate various DDoS attacks for HetIoT. The research proposes an intelligent intrusion detection system (IDS) using a convolutional neural network (CNN), i.e., HetIoT-CNN IDS, a novel deep learning-based convolutional neural network for the HetIoT environment. The proposed intelligent IDS successfully identifies and mitigates various DDoS attacks in the HetIoT infrastructure. The feasibility of the new proposed HetIoT-CNN IDS is assessed by considering binary and multi-class (8- and 13-classes) classification. The performance of the proposed intelligent IDS is compared with two state-of-the-art deep learning approaches for HetIoT, and the results reveal that the proposed HetIoT-CNN IDS outperforms it. The proposed HetIoT-CNN IDS successfully identifies various DDoS attacks with an accuracy rate of 99.75% for binary classes, 99.95% for 8-classes, and 99.99% for 13-classes. The work also compares the individual accuracy of binary classes, 8-classes, and 13-classes with state-of-the-art work.

Keywords Internet-of-Things · Security · Distributed denial-of-service · Deep learning · 1D-convolutional neural network

✉ Pranav M. Pawar
pranav@dubai.bits-pilani.ac.in

Extended author information available on the last page of the article

Published online: 06 October 2022

1 Introduction

The digitization of the world has led to extensive research into the IoT [1–4], which is a collection of many devices that are interlinked and communicate via the internet. If we look at IoT devices today, there are a massive set of diversified applications. This broad-scale IoT is used in almost every domain with distinct functional areas, e.g., smart cameras, smartphones, smart glasses, smartwatches, and many more [5]. The IoT network uses different protocols, different architectures like wireless sensor network (WSN), wireless mesh network (WMN), cellular network, etc., different designs, patterns, and standards [6, 7]. Therefore, the heterogeneity of devices, protocols, and network architecture makes wider deployment of IoT networks challenging to manage and operate. IoT systems also produce a number of heterogeneous data streams. Consequently, such a complex and large IoT system is also referred to as HetIoT [7].

The growth of IoT networks shows that the use of several HetIoT networks is increasing daily, and by 2025 it will go beyond 17 billion [8–12]. It is projected that this large-scale HetIoT will involve billions of heterogeneous devices in the near future. As technology advances, the number of security vulnerabilities also continues to increase [13]. The security of HetIoT is a significant challenge owing to its complexity, heterogeneity, and many interconnected resources. Also, it is evident from the discussion that the issue of protecting HetIoT devices is significantly intensified by their resource-constrained design [14]. As a result, attack mitigation and privacy protection measures used in conventional networks cannot be used effectively on HetIoT networks [15].

Major research studies concentrate on different security threats and defense mechanisms in HetIoT. The DDoS attack is one of the most powerful attacks that have taken hold with the emergence of HetIoT [16–18]. According to Corero Network, Security's study [19], the probability of DDoS attacks almost doubled in 2017 compared to 2016 due to the rising number of HetIoT devices. The Mirai Botnet infected millions of HetIoT devices and targeted DNS servers to break Internet connections to major websites [20–22]. The year 2020 has become noteworthy in several ways, especially when cyber-attacks are on the rise. The Covid-19 pandemic offered cybercriminals a great opportunity to hack and dismantle the IT infrastructure of any organization. The work-from-home system adopted by such organizations has been credited to the increase in cyber-attacks [23, 24].

1.1 Motivation

A security mechanism can be formulated to overcome security threat measurements. Deep learning IDS are well-suited for classification and prediction because of their conscious architecture [25, 26]. Deep learning (DL) will deliver promising results for HetIoT networks. An immense amount of data is produced by the HetIoT system, where learning techniques can be used for better and wise decision-making. Furthermore, the security solution can be enhanced by incorporating intelligence using

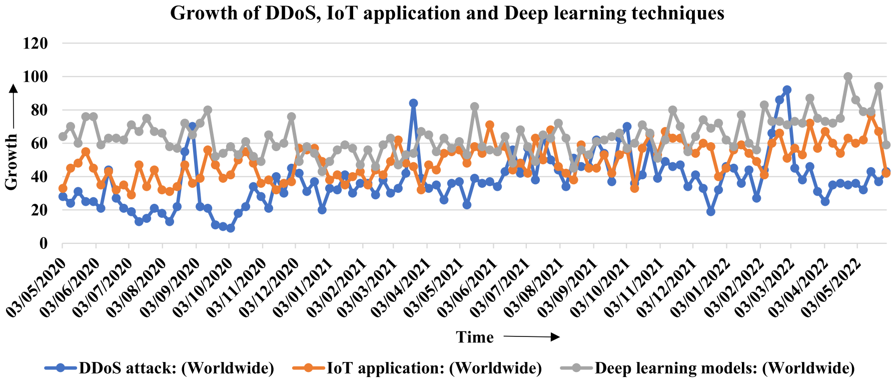


Fig. 1 Growth of DDoS attack, IoT applications and DL techniques (*Source* Google trend)

learning-based approaches. The Fig. 1 below shows the impact of DDoS attacks on various IoT applications. The trend of using different DL models is booming, which motivates the research to focus on developing an IDS for DDoS attacks using DL techniques. The graph (2020–2022) demonstrates that, despite extensive research being conducted to defend HetIoT infrastructure, DDoS attacks continue.

With the motivation of all of the aforementioned, this research proposes a novel CNN-based IDS called the HetIoT-CNN IDS to anticipate various types of DDoS attacks. The proposed approach is capable of identifying DDoS attacks with high accuracy. The dataset considered in this research is well-known, recent, and extensively used in the research world for creating IDS for HetIoT networks. The research also provides a survey of several DL-based models by considering binary and multi-class classification. The proposed HetIoT-CNN IDS performance is compared with two recent state-of-the-art techniques; namely, DL-based IDS model [27] and Flowguard model [28]. Furthermore, the research examined a time complexity analysis in comparison to the DL-based IDS model [27]. In terms of performance, the proposed HetIoT-CNN IDS outperforms the state-of-the-art intelligent IDS. The following are the significant contributions made in this research work.

1.1.1 Contributions

- The research provides a comprehensive insight into the current state-of-the-art work, focusing mainly on DL techniques that have used the CICDDoS2019 dataset [29] for IDS development.
- The research presents detailed data pre-processing steps using feature selection technique, memory optimization, data cleaning, and feature scaling after thorough exploratory data analysis.
- Deep learning-based convolutional neural network for detecting DDoS attack on HetIoT, the HetIoT-CNN IDS.
- Performance analysis with binary and multi-class classification (8-class and 13-class) using HetIoT security datasets, namely, CICDDoS2019. The perfor-

mance is evaluated against the following parameters, Precision, F1-score, Recall, and Accuracy.

- The proposed HetIoT-CNN IDS effectively identifies DDoS attacks for binary and multi-class (8- and 13-class) classification with good accuracy, i.e., 99.75% for binary class classification, 99.95% for 8-class classification, and 99.99% for 13-class classification.
- The individual accuracy of each class of attack is also measured and compared with state-of-the-art work.
- The research also examined state-of-the-art DL-based IDS [27] and compared it with the proposed HetIoT-CNN IDS using asymptotic time complexity analysis. The result shows that the proposed HetIoT-CNN IDS is lightweight, simple, and less complex in terms of computation time and the number of layers used.

The remaining content of the research paper is structured as follows. Section 2 reviews the related work with a comparative review. Section 3 presents the proposed HetIoT-CNN IDS architecture and algorithm in detail. Also, this section articulated CICDDoS2019 dataset pre-processing, i.e., feature selection, memory optimization, data cleaning, and feature scaling. Section 4 presents an asymptotic time complexity analysis of the proposed intelligent IDS. Section 5 provides experimental set-up, performance metrics, and performance results with discussion. Lastly, Sect. 6 presents conclusions and future work.

2 Related Work

This section aims to review the related work in HetIoT DL based intelligent security. The review mainly focused on the state-of-the-art work from 2018 to 2021.

The study [30] suggested an approach for detecting DDoS attacks in a software-defined network (SDN) environment entitled DDoSNet, which is based on a recurrent neural network (RNN) with an autoencoder. This approach concentrates on the binary classification of attacks with an accuracy of 99%. A CNN based real-time SDN security system is created to combat DDoS attacks [31]. This approach shows a 95.4% accuracy rate for binary classification of DDoS attacks. The study [28] focuses on IoT-DDoS defense approaches and offers FlowGuard, an edge-centric IoT defensive strategy. The CNN model's accuracy rate is 99.9% for multi-class classification whereas, for binary classification, long-short-term memory (LSTM) measures 98.9%.

In an SDN context, Assis et al. [32] proposed a gated recurrent unit (GRU) model for detecting DDoS attacks, which is for binary classification of attacks, and it achieves an accuracy of 99.6%. The study in [33] proposed the energy-based flow classifier, a binary classification model with an accuracy of 97%. A hybrid deep learning approach for the detection of DDoS attacks for the SDN environment is proposed in [34]. The author considered eight different DDoS attacks and achieved a 99.74% accuracy.

In tandem with Edge computing, Nie et al. [35] presented a deep learning-based IDS model for social-IoT (SIoT), which is based on the generative adversarial

Table 1 DDoS attack binary class classification

References	Model name	DL techniques	No. of features	Accuracy %
[27]	DL-based IDS model	CNN	67	99.95
[28]	Flowguard model	LSTM	40	98.9
[30]	DDoSNet model	RNN with autoencoder	77	99
[31]	IoT based SDN environment	CNN	87	95.4
[32]	GRU deep learning model	GRU-model	83	99.6
[33]	Energy based flow classifier model	EFC model	84	97

Table 2 DDoS attack multi-class classification

References	Model name	DL techniques	No. of features	Accuracy %	No. of classes
[27]	DL-based IDS model	CNN	67	95.12	13
				95.90	7
[28]	Flowguard model	CNN	40	99.9	12
[34]	Hybrid DL approach in SDN	CuDNN-LSTM, CuDNNGRU	80	99.74	8
[35]	SIoT with edge computing	GAN-algorithm	10	98.53	12
[36]	Efficient framework for B5G network	DNN	10	99.66	10
[37]	DIDDoS model	GRU	82	99.69$^{\times}$	9
				99.94*	3
[38]	Cuda base-LSTM model	LSTM	–	99.6	4

*Exploitation attack

$^{\times}$Reflection attack

network (GAN) strategy. The overall accuracy rate of this multiple attack based GAN model is 98.53%. The deep learning-based IDS for agriculture 4.0 is proposed in [27] to detect different classes of DDoS attacks. The research proposed three different models using RNN, CNN, and dense neural network (DNN) for 13 and 7 class classification of attack. The CNN approach shows the highest accuracy of 95.12% for 13-class and 95.90% for 7-class classification.

Beyond 5th generation (B5G) architecture is suggested for DDoS attack detection using multilayer DNN [36]. The pearson correlation coefficient (PCC) method is employed for feature selection, and the model is formulated for multi-class classification. It shows an accuracy of 99.66% for 10 class classification. GRU based method for recognizing DDoS attacks is suggested in [37]. The accuracy of the models is 99.69% for reflection-type DDoS attacks and 99.94% for exploitation DDoS attacks. The model's average performance rate is 99.7%. This model achieves the highest accuracy for SSDP attacks, which is 99.91%. The cuda-base LSTM (cu-LSTM) model proposed in [38], is used for multi-class classification of DDoS attacks and achieved a 99.6% accuracy rate.

Tables 1 and 2 below presents a comparative summary of reviwed DDoS attack detection techniques using DL. Table 1 shows binary class classification techniques for DDoS attack and Table 2 shows multi-class classification techniques for DDoS attack. According to the literature, existing models address either binary class classification or multi-class classification, excluding DL-based IDS model [27], and the Flowguard model [28], which addresses both classification. The CICDDoS2019 dataset includes all recent types of DDoS attacks. However, the existing state-of-the-art models fail to address these attacks. Even though existing state-of-the-art models using binary class classification achieve good accuracy; still, binary classification will only help to detect if a DDoS attack is present or not without providing specifics about the type of DDoS. Such specific information is essential for the good setup of any infrastructure. Hence the research further studied existing multi-class classification models. It is observed that existing state-of-the-art models didn't focus on all the various types of DDoS attacks except [27].

A DL-based HetIoT-CNN IDS is developed in this research to safeguard against DDoS attacks in a heterogeneous environment. Furthermore, the Portmap DDoS attack, part of the CICDDoS2019 dataset, receives minimal attention from the researchers. This attack is addressed by [34, 35]. The model developed by [35] detects Portmap attacks with an accuracy of 98.34%, whereas the proposed HetIoT-CNN IDS detects the Portmap with an accuracy of 100%. The strength of the proposed HetIoT-CNN IDS is lightweight, simple, and less complex, i.e., the number of layers used in the model is less when compared to the existing state-of-the-art models and can be easily processed at the network layer. Also, the proposed HetIoT-CNN IDS considers all the recent DDoS attacks present in the CICDDoS2019 dataset. This research employed three types of classification, binary and multi-class (8- and 13-class) classification, to identify DDOS attacks and obtain higher accuracy than previous techniques reported in the literature. The implementation of the proposed HetIoT-CNN IDS is addressed in detail in the next section.

3 Research Methodology

3.1 Proposed HetIoT-CNN IDS Architecture

CNN is a unique kind of artificial neural network used for pattern detection. It has proven to produce promising results in a wide range of domains, including image classification, computer vision, image and video recognition, and many others [39–41]. The potential of applying these methods in the security realm has captured many researchers' interest. As mentioned by [42, 43], CNN utilizes various building blocks like convolution layers, pooling layers, and fully connected layers to acquire spatial hierarchies of features automatically and constructively through the training process.

This research focused on deep learning-based IDS premised on the CNN model. To reduce the dimensionality and evaluate essential features, the proposed HetIoT-CNN IDS considers two 1D-convolution layers, two 1D-max-pooling layers, flattened, and one fully connected dense layer as an output layer with the SoftMax activation function to give an adequate classification performance. In addition to this, two dropout layers are included exclusively for binary classification to avoid over-fitting and achieve better accuracy results. The model extracts important features, i.e., feature learning, and uses them to classify various DDoS attacks correctly during each processing layer. The following are the functions of each layer present in the proposed HetIoT-CNN IDS:

(i) Convolution layer: Convolution is a linear operation used for feature extraction that takes the input, processes it, and obtains the feature map, also known as a convoluted map. Stride is used to move the kernel, whereas padding is used to preserve the information while changing the kernel size.
(ii) Pooling layer (max): This layer extracts both strong and fine features after convolution. It is also utilized to cut down the computation time and errors. The feature map will be either max-pooled or avg-pooled to extract the maximum value or average value, respectively. The proposed HetIoT-CNN IDS considers max-pooled to extract maximum activated features since it is the most commonly used method, as well as less complex than other methods [40, 44].
(iii) Dropout layer: This layer prevents the model from over-fitting during the training process. The research considers two dropout layers for binary classification to regularise and improve the proposed HetIoT-CNN IDS performance. The datasets used for binary classification exhibit data distribution and behavior differences. As a result, two dropout layers with a value of 0.5 are employed during the training process for binary classification [31, 45].
(iv) Flatten layer: This layer aids in the flattening of all information into a format appropriate for use by the subsequent layer. It converts any dimensional data into one-dimensional data, subsequently sent to the fully connected dense layer.
(v) Fully connected dense layer: This layer works the same as an artificial neural network. The proposed HetIoT-CNN IDS considers one fully connected dense layer as an output layer that gives the SoftMax activation function input to detect and classify the various DDoS attacks.

After conducting empirical research and applying the RandomSearch-hyperparameter tuning technique [45], the following hyperparameters are adopted for the proposed HetIoT-CNN IDS: 32 and 64 filters with the kernel of size 5, Stride is set to 2, and padding is set as 'same.' The max-pooling of size two is considered. The Sigmoid and SoftMax activation functions are employed. The Sigmoid is a non-linear activation function used after each convolution layer to provide the weighted sum of inputs to the subsequent layer. Another activation function (also known as the classification function) called SoftMax is adopted to classify the various DDoS attacks. The proposed HetIoT-CNN IDS architecture is shown in Fig. 2.

3.2 HetIoT-CNN IDS Algorithm

The working principle of the proposed HetIoT-CNN IDS for multi-class classifications (refer to Fig. 3a) and binary classification (refer to Fig. 3b) along with

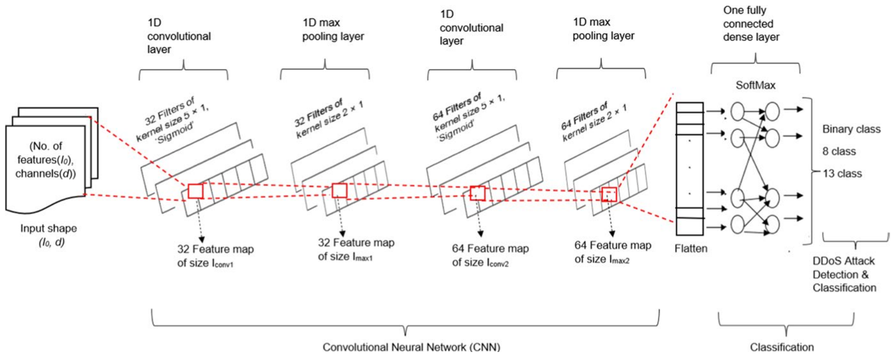
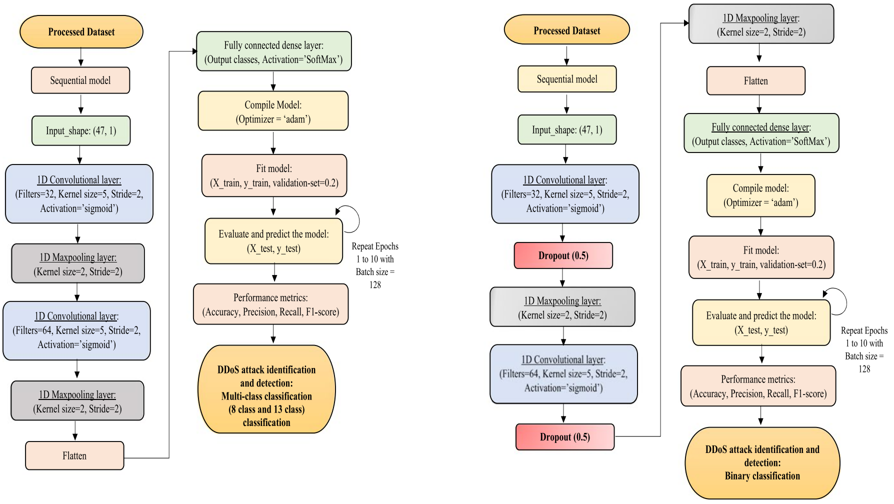

1D convolutional layer
1D max pooling layer
1D convolutional layer
1D max pooling layer
One fully connected dense layer
SoftMax
(No. of features(l_0), channels(d))
Input shape (l_0, d)
32 Filters of kernel size 5 × 1, 'Sigmoid'
32 Filters of kernel size 2 × 1
64 Filters of kernel size 5 × 1, 'Sigmoid'
64 Filters of kernel size 2 × 1
32 Feature map of size l_{conv1}
32 Feature map of size l_{max1}
64 Feature map of size l_{conv2}
64 Feature map of size l_{max2}
Flatten
Binary class
8 class
13 class
DDoS Attack Detection & Classification
Convolutional Neural Network (CNN)
Classification

Fig. 2 Architecture of the proposed HetIoT-CNN IDS


Processed Dataset
Sequential model
Input_shape: (47, 1)
1D Convolutional layer: (Filters=32, Kernel size=5, Stride=2, Activation='sigmoid')
1D Maxpooling layer: (Kernel size=2, Stride=2)
1D Convolutional layer: (Filters=64, Kernel size=5, Stride=2, Activation='sigmoid')
1D Maxpooling layer: (Kernel size=2, Stride=2)
Flatten
Fully connected dense layer: (Output classes, Activation='SoftMax')
Compile Model: (Optimizer = 'adam')
Fit model: (X_train, y_train, validation-set=0.2)
Evaluate and predict the model: (X_test, y_test)
Repeat Epochs 1 to 10 with Batch size = 128
Performance metrics: (Accuracy, Precision, Recall, F1-score)
DDoS attack identification and detection: Multi-class classification (8 class and 13 class) classification
Processed Dataset
Sequential model
Input_shape: (47, 1)
1D Convolutional layer: (Filters=32, Kernel size=5, Stride=2, Activation='sigmoid')
Dropout (0.5)
1D Maxpooling layer: (Kernel size=2, Stride=2)
1D Convolutional layer: (Filters=64, Kernel size=5, Stride=2, Activation='sigmoid')
Dropout (0.5)
1D Maxpooling layer: (Kernel size=2, Stride=2)
Flatten
Fully connected dense layer: (Output classes, Activation='SoftMax')
Compile model: (Optimizer = 'adam')
Fit model: (X_train, y_train, validation-set=0.2)
Evaluate and predict the model: (X_test, y_test)
Repeat Epochs 1 to 10 with Batch size = 128
Performance metrics: (Accuracy, Precision, Recall, F1-score)
DDoS attack identification and detection: Binary classification

(a) Multi-class classification **(b)** Binary classification

Fig. 3 **a** Multi-class classification. **b** Binary classification. Proposed HetIoT-CNN IDS

parameter settings such as a number of layers, filters, kernel size, activation function used, etc. are outlined in this section. The input for multi-class classification and binary classification is processed datasets (refer to Sect. 3.4 and Fig. 5) with an 80–20 train-test split. The section also highlighted the pseudocode used for the proposed HetIoT-CNN IDS. Algorithm 1 is used for multi-class (8- and 13-class) classifications, whereas Algorithm 2 is used for binary classifications.

Algorithm 1 Pseudocode for multi-class classification

```
Require: Batch size=128, Epoch = 10, Input_shape($I_{0,d}$) = (47,1)
Ensure: Accuracy, Precision, Recall, F1_score (for 8 class or 13 class)
1: Define sequential model
2: for Epoch = 1 to 10 do
3:     Compute 1D-convolution ($CL1$) operation with filters $f$ and kernel_size $k$ using equation, $CL1^{f}_{i,j} = \sigma(\sum_{p=0}^{k}\sum_{q=0}^{k}((\mathrm{I}_{0,\mathrm{d}})_{i+m,j+n} * \mathrm{W}^{f}_{p,q} + \mathrm{B}^{f})$, where $(p, q)$ indices of $f^{th}$ filters, $(i, j)$ are indices of output, weight W and bias B
4:     Extract maximum activated features with 1D-maxpooling ($MPL1$) layer using equation, $MPL1^{f}_{i,j} = MaxPool(CL1^{f}_{i,j})$
5:     Compute 1D-convolution ($CL2$) operation with filters $f$ and kernel_size $k$ using equation, $CL2^{f}_{i,j} = \sigma(\sum_{p=0}^{k}\sum_{q=0}^{k}((\mathrm{MPL1}^{\mathrm{f}})_{i+m,j+n} * \mathrm{W}^{f}_{p,q} + \mathrm{B}^{f})$
6:     Extract maximum activated features with 1D-maxpooling ($MPL2$) layer using equation, $MPL2^{f}_{i,j} = MaxPool(CL2^{f}_{i,j})$
7:     Flatten the resultant output of $MPL2^{f}$
8:     Detect and classify DDoS attacks using fully connected Dense layer with neurons = 8 or 13 and activation = 'softmax'i.e., each $MPL2^{f}$ neuron is fed to activation function to classify the output classes.
9:     Compile the model using optimizer = 'adam', loss function = 'sparse_categorical_ crossentropy'
10:    Fit the model on training_set, validation_ set = 0.2, and batch size
11:    Repeat steps 3 to 10 for each Epochs
12: end for
13: Evaluate and predict model for testing_set
14: Print confusion matrix and classification report
15: Plot overall training and testing loss, accuracy
```

Algorithm 2 Pseudocode for binary class classification

```
Require: Batch size=128, Epoch = 10, Input_shape($I_{0,d}$) = (47,1)
Ensure: Accuracy, Precision, Recall, F1_score (for 2 class)
1: Define sequential model
2: for Epoch = 1 to 10 do
3:     Compute 1D-convolution ($CL1$) operation with filters $f$ and kernel_size $k$ using equation, $CL1^{f}_{i,j} = \sigma(\sum_{p=0}^{k}\sum_{q=0}^{k}((\mathrm{I_{0,d}})_{i+m,j+n} * \mathrm{W}^{f}_{p,q} + \mathrm{B}^{f})$, where $(p, q)$ indices of $f^{th}$ filters, $(i, j)$ are indices of output, weight W and bias B
4:     Dropped-out 0.5 neurons and enable the model to learn more relevant features.
5:     Extract maximum activated features with 1D-maxpooling ($MPL1$) layer using equation, $MPL1^{f}_{i,j} = MaxPool(CL1^{f}_{i,j})$
6:     Compute 1D-convolution ($CL2$) operation with filters $f$ and kernel_size $k$ using equation, $CL2^{f}_{i,j} = \sigma(\sum_{p=0}^{k}\sum_{q=0}^{k}((\mathrm{MPL1^{f}})_{i+m,j+n} * \mathrm{W}^{f}_{p,q} + \mathrm{B}^{f})$
7:     Dropped-out 0.5 neurons and enable the model to learn more relevant features.
8:     Extract maximum activated features with 1D-maxpooling ($MPL2$) layer using equation, $MPL2^{f}_{i,j} = MaxPool(CL2^{f}_{i,j})$
9:     Flatten the resultant output of $MPL2^{f}$
10:    Detect and classify DDoS attacks using fully connected Dense layer with neurons = 2 and activation = 'softmax'i.e., each $MPL2^{f}$ neuron is fed to activation function to classify the output classes.
11:    Compile the model using optimizer = 'adam', loss function = 'sparse_categorical_ crossentropy'
12:    Fit the model on training_set, validation_ set = 0.2, and batch size
13:    Repeat steps 3 to 12 for each Epochs
14: end for
15: Evaluate and predict model for testing_set
16: Print confusion matrix and classification report
17: Plot overall training and testing loss, accuracy
```

As shown in the Fig. 2, the first convolution layer takes the input as, I_0, with the number of channels, d, where d is initially set to 1. The input_shape, (I_0, d), is then convolved with thirty-two filters of kernel size 5, the stride of 2, padding as 'same' and results in thirty-two feature maps of size I_{conv1}. The feature map size, I_{conv1}, is calculated using the following formula [46],

$$I_{conv1} = \frac{I_0 - K + 2P}{S} + 1 \tag{1}$$

$$P = \frac{K - 1}{2}, \tag{2}$$

where K = kernel size, P = padding, S = stride.

The resultant feature map, I_{conv1}, with thirty-two filters, then becomes the new input_shape, $(I_{conv1}, 32)$, for the first 1D-max-pooling layer. The 1D-max-pooling layer downsample it with the kernel size and stride of 2 yielding thirty-two feature map of size, I_{max1}, calculated as,

$$I_{max1} = \frac{I_{conv1} - K + 2P}{S} + 1 \tag{3}$$

Similarly, the second 1D-convolution layer convolves, I_{max1}, with the sixty-four filters of kernel size 5 and outputs a sixty-four feature map of size, I_{conv2}. The algorithm is repeated for the second 1D-max-pooling layer, with a new input shape, $(I_{conv2}, 64)$, yielding sixty-four feature maps of size I_{max2}.

After two 1D-convolution and two 1D-max-pooling layers, the output with the input_shape, $(I_{max2}, 64)$, is flattened and becomes the input for the final layer, which is a fully connected dense layer that accurately classifies various DDoS attacks, with SoftMax classification function. The SoftMax classification function is given by the following,

$$SoftMax, \sigma(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}} \; for \; i = 1, 2, \dots, K \tag{4}$$

The output of each 1D-convolution layer is fed to a Sigmoid activation function that is given by [47],

$$Sigmoid, \sigma(z) = \frac{1}{1 + e^{-z}}, \tag{5}$$

where z, is the resultant output of each convolution layer, i.e.

$$z = \sum I^w * k^w + b^k, \tag{6}$$

b = bias, I^w = weight of each input I, k^w = weight of each element in the kernel, k.

In the last step, adaptive moment estimation (Adam) optimizer is used. The Adam optimizer [40, 48] yielded superior results and is the most widely used optimizer; hence model incorporates it as an optimizer. The Adam optimizer keeps an exponentially decaying average of previous gradients, g_t [47, 49],

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{7}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2, \tag{8}$$

where m_t is first moment and v_t is second moment of the gradient , β_1 and β_2 is decay rates.

3.3 Dataset

CICDDoS2019 [29] dataset is provided by the Canadian Institute for Cybersecurity (CIC) and the University of New Brunswick (UNB). Figure 4 shows the composition of HetIoT based DDoS attacks present in the CICDDoS2019 dataset. The dataset consists of samples of normal traffic and attack traffic of 13 different attacks.

This dataset is commonly utilized in HetIoT-based DDoS attack detection and mitigation studies [28, 31]. The dataset is split into two parts: training-day dataset and testing-day dataset. Tables 3 and 4 describe the details of these datasets, including the name of the attacks present in each dataset, the number of samples (i.e., size of the attack), and features.

3.4 Data Pre-processing

The performance of any learning mechanism relies on the data pre-processing performed [39]. The research employed the following strategies to prepare the dataset for building a HetIoT-CNN IDS model.

(a) Feature selection: Feature selection is an important data pre-processing strategy that helps to reduce the number of features and improve the performance [51, 52]. The work uses an implementation of Random-forest regressor from the python sci-kit learn library [50, 53]. All the features were examined except Flow Id, Source IP, Destination IP, SimillarHTTP, and Timestamp. These exclusions were made because the features were either intrinsically uninformative or made

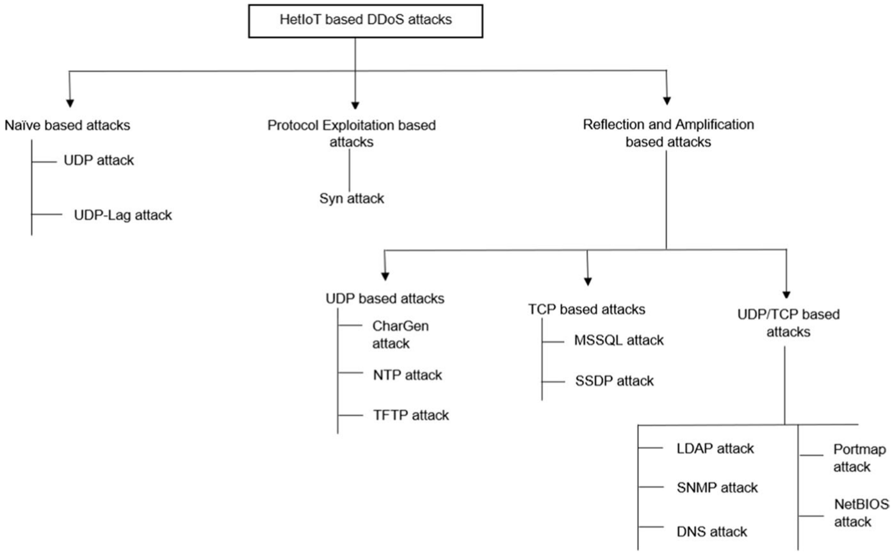


Fig. 4 Composition of the CICDDoS2019 DDoS dataset [28, 50]

Table 3 Details of CICDDoS2019 training-day dataset

Sr.no.	Name of the files	No. of samples	Features	Name of the attacks present in each file
1	DrDoS_DNS	5,074,413	86	DrDoS_DNS
2	DrDoS_LDAP	1,048,575	86	DrDoS_LDAP
3	DrDoS_MSSQL	4,524,498	86	DrDoS_MSSQL
4	DrDoS_NetBIOS	4,094,986	86	DrDoS_NetBIOS
5	DrDoS_NTP	1,217,007	86	DrDoS_NTP
6	DrDoS_SNMP	5,161,377	86	DrDoS_SNMP
7	DrDoS_SSDP	2,611,374	86	DrDoS_SSDP
8	DrDoS_UDP	3,136,802	86	DrDoS_UDP
9	Syn	1,582,681	86	Syn
10	TFTP	20,107,827	86	TFTP
11	UDP-Lag	370,605	86	UDP-Lag,WebDDoS
	Total	48,930,145		

Table 4 Details of CICDDoS2019 testing-day dataset

Sr.no.	Name of the files	No. of samples	Features	Name of the attacks present in each file
1	DrDoS_LDAP	2,113,234	86	LDAP, NetBIOS
2	DrDoS_MSSQL	5,775,786	86	LDAP, MSSQL
3	DrDoS_NetBIOS	3,455,899	86	NetBIOS
4	DrDoS_UDP	3,782,206	86	UDP, MSSQL
5	Syn	4,320,541	86	Syn
6	UDP-Lag	725,165	86	UDP-Lag, Syn, UDP
7	Portmap	191,694	86	Portmap
	Total	20,364,525		

up within a simulated context [33]. Table 5 below exhibits a list of the selected features after feature selection.

(b) Memory optimization: Memory optimization is a method of freeing or cleaning memory during a program in execution. An out-of-memory is a common error during the building model using huge datasets. The memory optimization is performed to save memory and fit the model with the entire dataset within the available system resources. The research used gc.collect(), del statement, downcast() feature from python libraries to save the memory for effective training and testing of the proposed model.

(c) Data cleaning: The data cleaning is performed to remove null, NAN, and infinity values from data. The work used isnull(), isinf(), and isNAN() methods from the python sci-kit libraries to remove it. Such values are replaced with constant integers.

Table 5 List of the selected features

Sr.no.	Feature name	Sr.no.	Feature name	Sr.no.	Feature name
1	Source Port	17	Idle Min	33	Active Mean
2	Destination Port	18	Idle Mean	34	Active Max
3	Protocol	19	act_data_pkt_fwd	35	Bwd IAT Total
4	Flow Duration	20	URG Flag Count	36	Bwd IAT Mean
5	Inbound	21	Subflow Fwd Packets	37	Bwd IAT Max
6	Min Packet Length	22	Subflow Bwd Packets	38	Bwd IAT Min
7	Flow IAT Std	23	Fwd Packet Length Min	39	Bwd IAT Std
8	Flow IAT Max	24	Init_Win_bytes_forward	40	Flow Packets/s
9	Flow IAT Min	25	Init_Win_bytes_backward	41	Flow Bytes/s
10	Fwd IAT Min	26	Total Fwd Packets	42	Fwd Packets/s
11	Fwd IAT Max	27	Total Backward Packets	43	Bwd Packets/s
12	min_seg_size_forward	28	Bwd Header Length	44	Fwd Packet Length Std
13	Active Min	29	Packet Length Std	45	Total Length of Bwd Packets
14	Avg Bwd Segment Size	30	Packet Length Variance	46	ACK Flag Count
15	Fwd Header Length	31	Bwd Packet Length Mean	47	Bwd Packet Length Min
16	Fwd Header Length.1	32	Bwd Packet Length Max	48	Label

(d) Feature scaling: Feature scaling is another key part of data pre-processing. The most prevalent feature scaling technique is the Standard Scaler. The Standard Scaler from sci-kit learn is adopted for the proposed HetIoT-CNN IDS.

Since the CICDDoS2019 dataset is huge (refer Tables 3, 4), 70% data from the training-day dataset is taken at random while data from the testing-day dataset is used entirely. These datasets are used to create three distinct datasets, Dataset 1, Dataset 2, and Dataset 3. Dataset 1 is used for binary classification, Dataset 2 for 8-class classification, and Dataset 3 for 13-class classification. The details of the three datasets used throughout the research are listed in Table 6.

Figure 5 represents the deployment model for the proposed HetIoT-CNN IDS, which starts with data pre-processing steps and creates three unique datasets. The proposed HetIoT-CNN IDS is trained and tested on each of these datasets to check its performance and identify DDoS attacks in the HetIoT environment. The proposed HetIoT-CNN IDS is deployed at the edge of the network layer to block the traffic from the intruder devices. The gateways will serve as edge nodes, processing data locally. To provide security against DDoS attacks, the proposed HetIoT-CNN IDS benefits from edge computing over cloud computing [54, 55].

Table 6 Details of dataset used for the proposed HetIoT-CNN IDS

Name of the dataset	Total no. of samples	No. of training samples	No. of testing samples	No. of features	No. of classes
Dataset 1: Binary classification	54,615,626	34,251,101	20,364,525	47	2
Dataset 2: 8-class classification	20,364,525	16,291,620	4,072,905	47	8
Dataset 3: 13-class classification	34,251,101	27,400,881	6,850,220	47	13

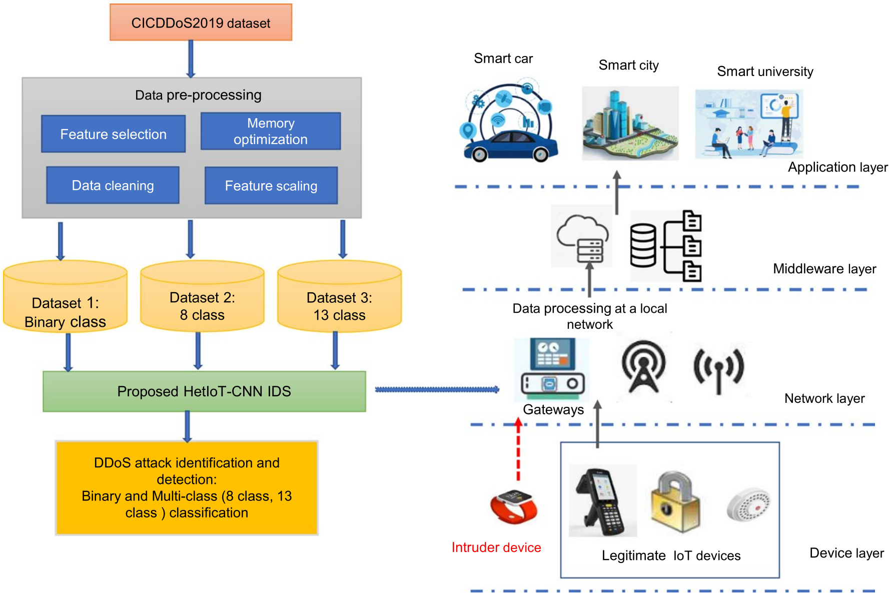


Fig. 5 Deployment model of the proposed HetIoT-CNN IDS

4 HetIoT-CNN IDS Analysis

This section analyses the asymptotic time complexity of the proposed HetIoT-CNN IDS. The 1D convolution is the sum of the row-wise dot product of two matrices, where one matrix is the kernel of size, k, and another matrix is the input of size, I_c. Therefore, the total computational complexity for a single convolution layer is, $O(I_c kd)$, where d is number of channels [56]. The convolution with number of filters, f, is $O(I_c kdf)$. So, the time complexity with two 1D-convolution layers is, $2 * O(I_c kdf)$.

The max-pooling layer doesn't contain any trainable parameters. It will help to reduce the computational overhead. It will search for the maximum weight from the given input data, I_m, with the help of kernel size, k. Fundamentally it is a one dimensional unsorted array. So, the time complexity with two 1D-max-pooling layers is, $2 * O(I_m)$.

The fully connected dense layer, nothing but the output layer, takes input after flattening the data from the max-pooling layer and processes it to detect the various DDoS attacks. This layer, connect each neuron, I_{fc}, to every SoftMax activation neuron, I_s. Hence, the time complexity with one fully connected dense layer is , $O(I_{fc} I_s)$. The asymptotic time complexity for the proposed HetIoT-CNN IDS is, $\{2 * O(I_c kdf) + 2 * O(I_m) + O(I_{fc} I_s) \}$

The research analyses the proposed HetIoT-CNN IDS time complexity with state-of-the-art [27] model that comprises three 1D-convolution layers, one 1D-GlobalAveragePooling layer, and three fully connected dense layers. The global average pooling layer requires more computation time than the max-pooling layer. The

Table 7 Comparative asymptotic time complexity

	Asymptotic time complexity
[27]	$\{\ 3 * O(I_c kdf) + 3 * O(I_{fc} I_s)\ \}$
HetIoT-CNN IDS	$\{\ 2 * O(I_c kdf) + O(I_{fc} I_s)\ \}$

Table 8 Comparative experimental training, testing, and total-time

Time in second	[27]		HetIoT-CNN IDS	
	8-class	13-class	8-class	13-class
Training time	774	1173	659	1148
Testing time	297	451	152	241
Total time	1071	1624	811	1389

global average pooling will compute the average of first k, element, where k is kernel size and then slide the window to the next k element to find the average and so on until it reaches the end of the list, I_g. Hence, the time complexity for the one 1D-GlobalAveragePooling layer is, $O(kI_g)$. Therefore, the asymptotic time complexity for [27], is, $\{\ 3 * O(I_c kdf) + O(kI_g) + 3 * O(I_{fc} I_s)\ \}$

The computational cost of the pooling layer is low [57]; so, by ignoring it, the overall comparative asymptotic time complexity for the proposed HetIoT-CNN IDS and state-of-the-art model in [27] is given in Table 7 below. The proposed HetIoT-CNN IDS used less number of layers. Hence computational cost (i.e., addition and multiplication operations) is less. As a result, the proposed HetIoT-CNN IDS is lightweight, simple, and less complex.

The asymptotic time complexity analysis of both the IDS shows that the proposed HetIoT-CNN IDS is efficient in computation time. Further, to quantify the time required for training and testing, the proposed HetIoT-CNN IDS and model in [27] have experimented on the CICDDoS2019 dataset. Table 8 shows the comparison of training and testing time required for both the IDS in the case of 8-class and 13-class. The results in Table 8 reveal that the proposed HetIoT-CNN IDS outperforms as compared with state-of-the-art work in [27].

5 Result and Discussion

The section examines the performance of the proposed HetIoT-CNN IDS to appropriately detect DDoS attacks by considering binary class, 8-class, and 13-class classification. The section also provides a comparative performance analysis of the proposed technique with two state-of-the-art works, DL-based model [27], and Flowguard model [28]. The simulation set-up, as well as performance metrics used, are mentioned below.

(i) Simulation set-up: The specification of the software and hardware set-up are provided in Table 9.

Table 9 Simulation set-up

Software specification	
Software tool	Spyder 5.0.1 IDE
Programming language	Python 3.8, Sklearn library, Matplotlib 3.5
API tool	Keras on TensorFlow
Hardware specification	
Processor: Intel(R) Core(TM) i7-10750H CPU @ 2.60 GHz, RAM: 16 GB	
OS: Windows 10, 64-bit, Dedicated 4 GB NVIDIA GEFORCE GTX 1650 Ti	

(ii) Performance metrics used: The measurement of the proposed IDS is performed using the standard performance metrics, accuracy, precision, recall, and f1-score [42, 58, 59].

(a) Accuracy: Accuracy is the ratio of correctly predicted DDoS attacks and Benign Flow out of all predicted data.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

(b) Precision: Precision is a metric that quantifies a system's ability to provide only relevant outcomes.

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

(c) Recall: Recall is also considered as True-Positive-Rate(TPR). It is a metric that quantifies a systems' ability to provide all relevant outcomes.

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

(d) F1-score: F1-score calculates the harmonic mean of precision and recall.

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{12}$$

Here in the above Eqs. (9), (10), (11), (12), true positive (TP) indicates attack data are correctly classified as DDoS attacks. False positive (FP) indicates attack data are incorrectly classified as benign. True negative (TN) indicates benign flow is correctly classified as benign. False negative (FN) indicates benign flow is incorrectly classified as a DDoS attack.

(iii) Performance analysis: To perform the comparative performance analysis of the proposed HetIoT-CNN IDS with state-of-the-art IDS mentioned in [27, 28], the research considered experimentation under binary and multi-class (8- and 13-class) classification. Three datasets are prepared as per the details given

Table 10 Hyperparameter settings

Hyperparameter settings	Value
Learning rate	0.001
Activation function	Sigmoid
Classification function	SoftMax
Batch size	128
Epoch	10
Dropout	0.5
Optimizer	Adam
Loss function	Sparse_categorical_Crossentropy

Table 11 Dataset 1: binary classification

Sr.no.	Dataset name	No. of samples		Total no. of samples
		Benign	DDoS attack	
1	Training_dataset1	51,953	34,199,148	34,251,101
2	Testing_dataset1	56,965	20,307,560	20,364,525

in Sect. 3.4. Following are the steps that are applied on the three datasets and are common throughout the research, data pre-processing, as mentioned in the preceding section 3.4, is the first step towards training the proposed HetIoT-CNN IDS (refer Fig. 5). The considered datasets are split as 80% training data and 20% testing data. The number of training samples is further split into training and validation sets for 8- and 13-class classification to ensure that all results are consistent. The hyperparameter settings are depicted in Table 10.

5.1 Binary Classification

Dataset 1 is used for binary classification. In Dataset 1, there are two independent datasets: training_dataset1 and testing_dataset1. The training_dataset1 is used to train the proposed HetIoT-CNN IDS. Then the model is validated and tested on the testing_dataset1 with the ratio of 20:80 (i.e., the validation set is 20% and the testing set is 80%). The proposed HetIoT-CNN IDS performed admirably and achieved a good accuracy rate. Table 11 shows the details of two unique training and testing datasets.

The proposed HetIoT-CNN IDS hyper-parameter settings are depicted in Algorithm 2. In addition to the convolution and max-pooling layers, two dropout layers are added to identify the binary class classification and accurately achieve better outcomes. As mentioned in [29, 50], the training and testing sets are two independent datasets collected on different days, times, and in various circumstances. As a result, the distributions of the two datasets diverge. As a result, the model is overfitted, and binary classification utilizes two dropout layers to avoid overfitting. The Fig. 6 depicts the comparative performance accuracy to detect benign and DDoS attacks for the proposed HetIoT-CNN IDS. The HetIoT-CNN IDS show a 99.75% accuracy. The accuracy of the state-of-the-art model in [27], is 99.95%, and [28], is 98.9%. As indicated in Table 9, DL-based IDS model [27], performs somewhat better than the proposed HetIoT-CNN IDS. The performance of the HetIoT-CNN IDS, on the other hand, is superior to that of the Flowguard model [28].

5.2 8-Class Classification

The Dataset 2 contains eight classes: LDAP, MSSQL, NetBIOS, UDP, Syn, UDP-Lag, Portmap, and Benign. As previously mentioned, this research also focused on the Portmap attack, which receives comparatively little attention according to the literature review. The statistics for each class in Dataset 2 are reported in Table 12.

The proposed HetIoT-CNN IDS shows 100% accuracy for detecting six different classes of attacks such as LDAP, MSSQL, NetBIOS, UDP, Syn, and Portmap DDoS attacks. The outcomes of these specific attack identifications are compared with the model in [27]. The Fig. 7 depicts the performance outcomes of the same. Figure 7 indicates that the proposed HetIoT-CNN IDS outperforms as compared to the state-of-the-art model in [27]. As reported in the [27], it identifies UDP-Lag attacks with an accuracy of 0%, whereas a Portmap attack is not considered. However, the proposed HetIoT-CNN IDS identifies UDP-Lag attack with an accuracy rate of 47% and Portmap with 100%.

Furthermore, the proposed HetIoT-CNN IDS identifies 8-classes of attacks with an accuracy of 99.95% which is higher as compared with the state-of-the-art model accuracy of 95.90% [27].

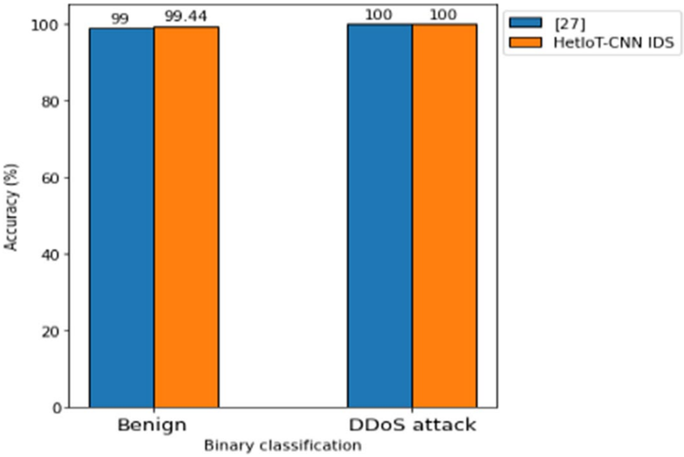


Fig. 6 Performance result for binary classification

Table 12 Dataset 2: 8-class classification

Sr.no.	Attacks name	Training sample	Testing sample
1	Benign	45,572	11,393
2	DrDoS_LDAP	1,532,098	383,024
3	DrDoS_MSSQL	4,629,962	1,157,491
4	DrDoS_NetBIOS	2,925,998	731,499
5	DrDoS_UDP	3,093,724	773,431
6	Syn	3,913,200	978,300
7	UDP-Lag	1498	375
8	Portmap	149,568	37,392
	Total no. of samples	16,291,620	4,072,905

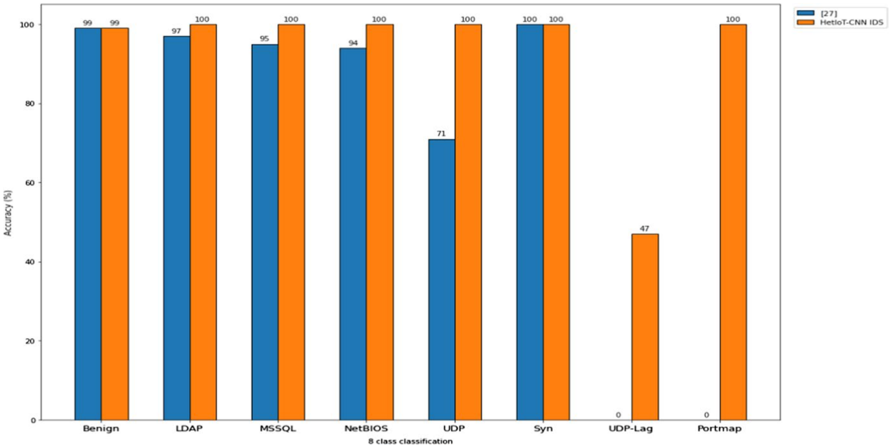


Fig. 7 Performance result for 8-class classification

5.3 13-Class Classification

The 13 classes considered in the experimentation are DNS, LDAP, MSSQL, NetBIOS, NTP, SNMP, SSDP, UDP, Syn, TFTP, UDP-Lag, WebDDoS, and Benign. Further, Dataset 3 with 13 classes classifications is exposed to the proposed HetIoT-CNN IDS. The statistics for each DDoS attack in Dataset 3 are reported in Table 13.

The HetIoT-CNN IDS accurately predicts DNS, LDAP, MSSQL, NetBIOS, NTP, SNMP, SSDP, UDP, Syn, TFTP, and UDP-Lag DDoS attacks with an accuracy rate of 100%; however, WebDDoS attack is identified with 44% accuracy. The reason behind this is that while training the proposed HetIoT-CNN IDS, the volume of WebDDos attacks is relatively low (refer to Table 13). Hence, the model performs poorly compared to other DDoS attacks mentioned above. Despite this, WebDDoS attacks have a precision of 1, and recall is 28%. The performance results for the same are shown in Fig. 8. Figure 8 indicates that the proposed HetIoT-CNN IDS explicitly outperforms the state-of-the-art model in [27]. Furthermore, the proposed

Table 13 Dataset 3: 13-class classification

Sr.no.	Attacks name	Training sample	Testing sample
1	Benign	31,455	7864
2	DrDoS_DNS	2,839,784	709,946
3	DrDoS_LDAP	586,753	146,688
4	DrDoS_MSSQL	2,532,573	633,143
5	DrDoS_NetBIOS	2,292,221	573,055
6	DrDoS_NTP	673,491	168,373
7	UDP-Lag	205,205	51,301
8	DrDoS_SNMP	2,889,524	722,381
9	DrDoS_SSDP	1,461,928	365,482
10	DrDoS_UDP	1,755,430	438,858
11	Syn	886,086	221,522
12	TFTP	11,246,188	2,811,547
13	WebDDoS	242	61
	Total no. of samples	27,400,881	6,850,220

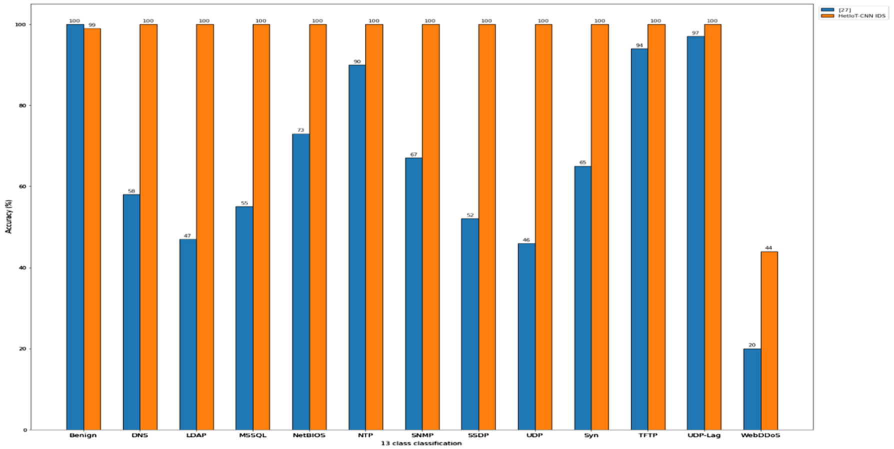


Fig. 8 Performance result for 13-class classification

model identiifes 13 classes of attacks with accuracy of 99.99% as compared with 95.12% in [27] and 99.9% in [28].

5.4 Summary of Results

The summary of results is shown in Fig. 9. It shows that the proposed model outperforms as compared with state-of-the-art models. The proposed model shows 99.75% for binary classification, 99.95% for 8-class classification and 99.99% for 13-class classification, which is higher as compared with state-of-the-art model discussed in [27, 28]. The rationale for the proposed HetIoT-CNN IDS's

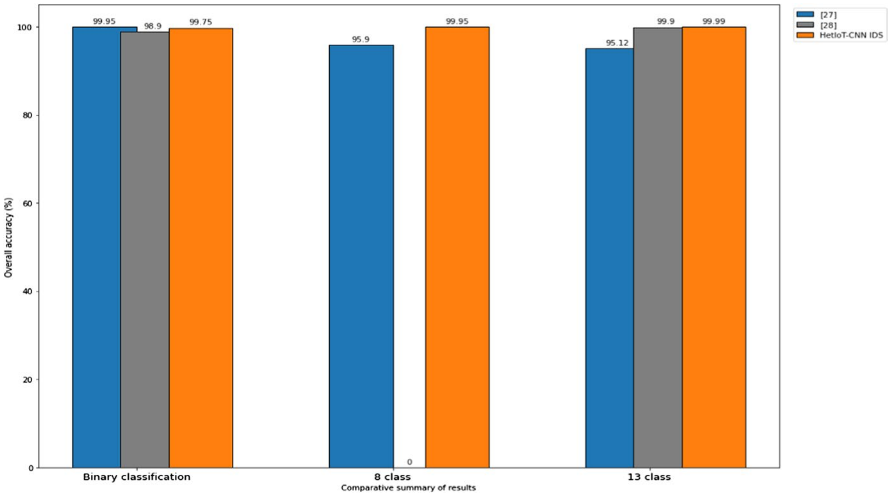


Fig. 9 Comparative summary of results

high accuracy includes data pre-processing steps after exploratory data analysis, the number of selected features, dataset training, and many other aspects, which are represented in Table 14 below. It highlights that the proposed model is lightweight, simple, and less complex for various DDoS attack detection and classification compared to the state-of-the-art model [27]. Compared to the state-of-the-art model [27], the proposed model uses two convolution layers, two max-pooling, and one fully connected dense layer. In contrast, the state-of-the-art model [27] uses three convolution layers, one global average pooling, and three fully connected dense layers. As mentioned in Sect. 4, the asymptotic time complexity of the proposed HetIoT-CNN IDS is less than the state-of-the-art mode [27].

Table 14 Comparative summary of HetIoT-CNN IDS with state-of-the-art [27]

Model parameters	[27]	HetIoT-CNN IDS
No of features	67	47
Batch size	10,000	128
Epoch	35	10
Convolution layer	3	2
Filter	64, 32, 16	32, 64
Kernel size	3, 3, 2	5, 5
Pooling layer	One GlobalAverage-pooling	Two Max-pooling
Fully connected dense layer	3	1
Activation function	Relu	Sigmoid
Classification function	SoftMax	SoftMax

6 Conclusions

This research proposed the HetIoT-CNN IDS, an innovative deep learning-based CNN for the HetIoT environment. The proposed IDS shows 99.75% accuracy for detecting benign and DDoS attacks (binary classification), 99.95% for detecting 8 different classes of DDoS attack (8-class classification), and 99.99% accuracy for detecting 13 different classes of DDoS attack (13-class classification). The asymptotic time complexity analysis also revealed that the proposed IDS is efficient in terms of time, lightweight and less complex. The accuracy performance and time complexity of the proposed IDS is compared with state-of-the-art intelligent IDS. The work also concentrated on the individual detection accuracy of each attack in case 8-class and 13-class. The future work of the paper is to develop a RNN model for the detection and prediction of DDoS attacks. The work will also be extended further by considering reinforcement learning models for real-time detection of attacks in HetIoT systems.

Funding Not applicable.

Data Availability The dataset generated during and/or analyse during the current study are available from the corresponding author.

Code availability Not applicable.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Kamble, A., Bhutad, S.: Survey on Internet of Things (IoT) security issues & solutions. In: 2nd International Conference on Inventive Systems and Control (ICISC), pp. 307–312. IEEE (2018)
2. Kumar, R.P., Smys, S.: A novel report on architecture, protocols and applications in Internet of Things (IoT). In: 2nd International Conference on Inventive Systems and Control (ICISC), pp. 1156–1161. IEEE (2018)
3. Gupta, B.B., Quamara, M.: An overview of Internet of Things (IoT): architectural aspects, challenges, and protocols. Concurr. Comput.: Pract. Exp. **32**(21), 1–24 (2020)
4. IoT Examples Of 2021 Real World Apps. https://www.softwaretestinghelp.com/best-iot-examples/ (2021). Accessed 27 Sept 2021
5. Elkobaisi, M.R., Al Machot, F.: Human emotion modeling (HEM): an interface for IoT systems. J. Ambient Intell. Humaniz. Comput. **13**(8), 4009–4017 (2021)
6. Shu, L., Mukherjee, M., Pecht, M., Crespi, N., Han, S.N.: Challenges and research issues of data management in IoT for large-scale petrochemical plants. IEEE Syst. J. **12**(3), 2509–2523 (2017)
7. Qiu, T., Chen, N., Li, K., Atiquzzaman, M., Zhao, W.: How can heterogeneous Internet of Things build our future: a survey. IEEE Commun. Surv. Tutor. **20**(3), 2011–2027 (2018)
8. Sun, X., Ansari, N.: EdgeIot: mobile edge computing for the Internet of Things. IEEE Commun. Mag. **54**(12), 22–29 (2016)
9. Alam, T.: A reliable communication framework and its use in Internet of Things (IoT). Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol. (CSEIT1835111) **3**(5), 450–456 (2018)

10. Lueth, K.L.: IoT analytics: state of the IoT 2018: number of IoT devices now at 7B-market accelerating. https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/ (2018). Accessed 8 Aug 2018
11. Hassija, V., Chamola, V., Saxena, V., Jain, D., Goyal, P., Sikdar, B.: A survey on IoT security: application areas, security threats, and solution architectures. IEEE Access **7**, 82721–82743 (2019)
12. De Schepper, T., Latré, S., Famaey, J.: Scalable load balancing and flow management in dynamic heterogeneous wireless networks. J. Netw. Syst. Manag. **28**(1), 133–159 (2020)
13. Ko, E., Kim, T., Kim, H.: Management platform of threats information in IoT environment. J. Ambient Intell. Humaniz. Comput. **9**(4), 1167–1176 (2018)
14. Kim, S., Lee, I.: IoT device security based on proxy re-encryption. J. Ambient Intell. Humaniz. Comput. **9**(4), 1267–1273 (2018)
15. Hassan, W.H.: Current research on Internet of Things (IoT) security: a survey. Comput. Netw. **148**, 283–294 (2019)
16. Wani, A., Revathi, S.: DDoS detection and alleviation in IoT using SDN (SDIoT-DDoS-DA). J. Inst. Eng. (India): Ser. B **101**(2), 117–128 (2020)
17. Rodrigues, B., Scheid, E., Killer, C., Franco, M., Stiller, B.: Blockchain signaling system (bloss): cooperative signaling of distributed denial-of-service attacks. J. Netw. Syst. Manag. **28**(4), 953–989 (2020)
18. Moayad, Aloqaily, Otoum, Safa, Al Ridhawi, Ismaeel, Jararweh, Yaser: An intrusion detection system for connected vehicles in smart cities. Ad Hoc Netw. **90**, 1–14 (2020)
19. Corero Network security. https://internetofbusiness.com/ddos-attacks-double-iot-target-corero/ (2020). Accessed 10 July 2020
20. Hameed, S., Khan, F.I., Hameed, B.: Understanding security requirements and challenges in Internet of Things (IoT): a review. Hindawi J. Comput. Netw. Commun. **2019**, 1–14 (2019)
21. Golubov, R.: Winter breaks on the darknet: our top 10 IoT cyber stories of Q1 2020. https://firedome.io/blog/top-10-iot-cyber-stories-of-q1-2020/ (2020). Accessed 20 Mar 2020
22. Dyn. DNS and DDoS. https://www.kaspersky.com/blog/attack-on-dyn-explained/13325/ (2020). Accessed 30 May 2016
23. Acohido, B.: IoT attacks intensified by Covid-19 Avast Blog. https://securityboulevard.com/2020/11/iot-attacks-intensified-by-covid-19-avast/ (2020). Accessed 6 Nov 2020
24. Vivek Ganti, O.Y.: Network-layer DDoS attack trends for Q3 2020. https://blog.cloudflare.com/network-layer-ddos-attack-trends-for-q3-2020/ (2020). Accessed 18 Nov 2020
25. Balasubramanian, V., Otoum, S., Reisslein, M.: VeNet: hybrid stacked autoencoder learning for cooperative edge intelligence in IoV. IEEE Trans. Intell. Transp. Syst. (2022) . https://doi.org/10.1109/TITS.2022.3170372
26. Tsimenidis, S., Lagkas, T., Rantos, K.: Deep learning in IoT intrusion detection. J. Netw. Syst. Manag. **30**(1), 1–40 (2022)
27. 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 **10**(11), 1–26 (2021)
28. Jia, Y., Zhong, F., Alrawais, A., Gong, B., Cheng, X.: Flowguard: an intelligent edge defense mechanism against IoT DDoS attacks. IEEE Internet Things J. **7**(10), 9552–9562 (2020)
29. DDoS Evaluation Dataset CICDDoS2019. https://www.unb.ca/cic/datasets/ddos-2019.html (2019). Accessed 10 Jun 2020
30. Elsayed, M.S., Le-Khac, N.-A., Dev, S., Jurcut, A.D.: DDoSnet: a deep-learning model for detecting network attacks. In: IEEE 21st International Symposium on “A World of Wireless, Mobile and Multimedia Networks” (WoWMoM), pp. 391–396. IEEE (2020)
31. de Assis, M.V., Carvalho, L.F., Rodrigues, J.J., Lloret, J., Proença, M.L., Jr.: Near real-time security system applied to SDN environments in IoT networks using convolutional neural network. Comput. Electr. Eng. **86**, 106738 (2020)
32. Assis, M.V., Carvalho, L.F., Lloret, J., Proença, M.L., Jr.: A GRU deep learning system against attacks in software defined networks. J. Netw. Comput. Appl. **177**, 1–13 (2021)
33. Pontes, C., Souza, M., Gondim, J., Bishop, M., Marotta, M.: A new method for flow-based network intrusion detection using the inverse Potts model. IEEE Trans. Netw. Serv. Manag. **18**(2), 1125–1136 (2021)
34. Javeed, D., Gao, T., Khan, M.T.: SDN-enabled hybrid DL-driven framework for the detection of emerging cyber threats in IoT. Electronics **10**(8), 918–934 (2021)

35. Nie, L., Wu, Y., Wang, X., Guo, L., Wang, G., Gao, X., Li, S.: Intrusion detection for secure social Internet of Things based on collaborative edge computing: a generative adversarial network-based approach. IEEE Trans. Comput. Soc. Syst. **9**(1), 134–145 (2021)
36. Amaizu, G.C., Nwakanma, C.I., Bhardwaj, S., Lee, J., Kim, D.-S.: Composite and efficient DDoS attack detection framework for B5G networks. Comput. Netw. **188**, 107871 (2021)
37. ur Rehman, S., Khaliq, M., Imtiaz, S.I., Rasool, A., Shafiq, M., Javed, A.R., Jalil, Z., Bashir, A.K.: DiDDoS: an approach for detection and identification of distributed denial of service (DDoS) cyberattacks using gated recurrent units (GRU). Future Gener. Comput. Syst. **118**, 453–466 (2021)
38. Badamasi, U.M., Khaliq, S., Babalola, O., Musa, S., Iqbal, T.: A deep learning based approach for DDoS attack detection in IoT-enabled smart environments. Int. J. Comput. Netw. Commun. Secur. **8**(10), 93–99 (2020)
39. Pal, K.K., Sudeep, K.: Preprocessing for image classification by convolutional neural networks. In: International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), pp. 1778–1781. IEEE (2016)
40. Indolia, S., Goswami, A.K., Mishra, S.P., Asopa, P.: Conceptual understanding of convolutional neural network—a deep learning approach. Procedia Comput. Sci. **132**, 679–688 (2018)
41. Izadi, S., Ahmadi, M., Rajabzadeh, A.: Network traffic classification using deep learning networks and Bayesian data fusion. J. Netw. Syst. Manag. **30**(2), 1–21 (2022)
42. Gaur, V., Kumar, R.: Analysis of machine learning classifiers for early detection of DDoS attacks on IoT devices. Arab. J. Sci. Eng. **47**(2), 1353–1374 (2021)
43. Drewek-Ossowicka, A., Pietrołaj, M., Rumiński, J.: A survey of neural networks usage for intrusion detection systems. J. Ambient Intell. Humaniz. Comput. **12**(1), 497–514 (2021)
44. Yamashita, R., Nishio, M., Do, R.K.G., Togashi, K.: Convolutional neural networks: an overview and application in radiology. Insights Imaging **9**(4), 611–629 (2018)
45. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. J. Mach. Learn. Res. **13**(2), 281–305 (2012)
46. Kim, J., Kim, J., Kim, H., Shim, M., Choi, E.: CNN-based network intrusion detection against denial-of-service attacks. Electronics **9**(6), 916–937 (2020)
47. Wani, M.A., Bhat, F.A., Afzal, S., Khan, A.I.: Training supervised deep learning networks. In: Broy, M., Denert, E. (eds.) Advances in Deep Learning. Studies in Big Data, vol. 57, pp. 31–52. Springer, Singapore (2020)
48. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv:1412.6980v9 [cs.LG] (2017). https://doi.org/10.48550/arXiv.1412.6980
49. Taqi, A.M., Awad, A., Al-Azzo, F., Milanova, M.: The impact of multi-optimizers and data augmentation on tensorflow convolutional neural network performance. In: Conference on Multimedia Information Processing and Retrieval (MIPR), pp. 140–145. IEEE (2018)
50. Sharafaldin, I., Lashkari, A.H., Hakak, S., Ghorbani, A.A.: Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. In: International Carnahan Conference on Security Technology (ICCST), pp. 1–8. IEEE (2019)
51. Huang, Y., Jin, W., Yu, Z., Li, B.: Supervised feature selection through deep neural networks with pairwise connected structure. Knowl. Based Syst. **204**, 1–13 (2020)
52. Ghori, K.M., Imran, M., Nawaz, A., Abbasi, R.A., Ullah, A., Szathmary, L.: Performance analysis of machine learning classifiers for non-technical loss detection. J. Ambient Intell. Humaniz. Comput. (2020). https://doi.org/10.1007/s12652-019-01649-9
53. RFR. Selecting good features: random forest Regressor. https://blog.datadive.net/selecting-good-features-part-iii-random-forests/ (2014). Accessed 1 Dec 2014
54. Aliyu, F., Sheltami, T., Deriche, M.: Human immune-based intrusion detection and prevention system for fog computing. J. Netw. Syst. Manag. **30**(11), 1–27 (2022). https://doi.org/10.1007/s10922-021-09616-6
55. Al Ridhawi, I., Aloqaily, M., Kotb, Y., Al Ridhawi, Y., Jararweh, Y.: A collaborative mobile edge computing and user solution for service composition in 5G systems. Trans. Emerg. Telecommun. Technol. **29**(11), e3446 (2018)
56. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Adv. Neural Inf. Process. Syst. **30**, 5998–6008 (2017)
57. Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., Inman, D.J.: Inman 1D convolutional neural networks and applications: a survey. Mech. Syst. Signal Process. **151**, 1–20 (2021)

58. Shahhosseini, M., Mashayekhi, H., Rezvani, M.: A deep learning approach for botnet detection using raw network traffic data. J. Netw. Syst. Manag. **30**(3), 1–23 (2022). https://doi.org/10.1007/s10922-022-09655-7
59. Otoum, S., Kantarci, B., Mouftah, H.T.: A novel ensemble method for advanced intrusion detection in wireless sensor networks. In: ICC 2020–2020 IEEE International Conference on Communications (ICC), pp. 1–6. (2020). https://doi.org/10.1109/ICC40277.2020.9149413

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Shalaka Mahadik is a Doctoral student in Birla Institute of Technology and Science Pilani, Dubai, UAE. She received her B.E. and M.E. (Computer Engineering) from Mumbai University (India) in 2005 and 2011, respectively. She worked as a lecturer from 2005 to 2011 and then as an Assistant Professor from 2011 to 2014. She authored and presented her master's thesis at international conferences. In 2022, she won the "Best Poster Presentation" award at BITS Pilani's Dubai campus on Research Day. Her research focuses on Security and Privacy in heterogeneous IoT (HetIoT).

Pranav M. Pawar currently working as an Assistant Professor in Birla Institute of Technology and Science Pilani, Dubai, UAE. He was a postdoctoral fellow at Bar-Ilan University, Israel from March 2019 to October 2020 in the area of Wireless Communication and Deep Leaning. He is the recipient of an out-standing postdoctoral fellowship from the Israel Planning and Budgeting Committee. His research interests are Energy efficient MAC for WSN, QoS in WSN, wireless security, green technology, computer architecture, database management system, and bioinformatics.

Raja Muthalagu is currently an Assistant Professor with Birla Institute of Technology and Science, Pilani, Dubai Campus, Dubai, UAE. He was a Postdoctoral Research Fellow with Air Traffic Management Research Institute, Nanyang Technological University, Singapore, from 2014 to 2015. Dr. Muthalagu was the recipient of the Canadian Commonwealth Scholarship Award 2010 for the Graduate Student Exchange Program in the Department of Electrical and Computer Engineering, University of Saskatchewan, Saskatoon, SK, Canada. His research interest includes wireless communication, signal processing, aeronautical communication, and cyber security.

Authors and Affiliations

Shalaka Mahadik[1] · Pranav M. Pawar[1] · Raja Muthalagu[1]

Shalaka Mahadik
p20200002@dubai.bits-pilani.ac.in

Raja Muthalagu
raja.m@dubai.bits-pilani.ac.in

[1] Department of Computer Science, Birla Institute of Technology and Science Pilani,
Dubai Campus, Dubai, UAE