Signal, Image and Video Processing (2024) 18:5477–5489
https://doi.org/10.1007/s11760-024-03248-4

ORIGINAL PAPER

Customized convolutional neural network model for IoT botnet attack detection

Balaganesh Bojarajulu[1] · **Sarvesh Tanwar**[1]

Received: 25 April 2023 / Revised: 30 October 2023 / Accepted: 28 April 2024 / Published online: 17 June 2024

Abstract
The Internet of Things is a disruptive technology that has changed the face of many industries. On the contrary, the unpresidential growth of IoT has also introduced many cybersecurity challenges. An adversary can exploit a zero-day vulnerability in an IoT to create a botnet of things. An IoT botnet is a group of compromised Internet of Things weaponized to launch cyber attacks. Machine learning and other artificial intelligence techniques are being used to combat the wide range of cyberattacks on the Internet of Things. However, in order to overcome challenges such as early diagnosis, real-time monitoring, and adaptability to different threats, these Machine Learning approaches still require significant feature engineering. In order to identify IoT botnet assaults early on, this paper suggests using a customized convolutional neural network (CCNN) model. The four phases of the model are feature extraction, attack detection, mitigation, and pre-processing. The class imbalance has been improved and the input data pre-processed using the Enhanced Synthetic minority oversampling approach. Furthermore, flow-based features, raw attributes, mean, median, standard deviation, improved entropy, mutual information, and other statistical features are retrieved and regarded as part of the feature set. The CCNN model provides the detection or classification output during the attack detection phase, which operates depending on the features derived from the input data. Additionally, a mitigation process based on entropy has been suggested to locate the attacker node, aiding in the removal of the susceptible attacker IoT node from the network. The compromised IoT node is removed through the entropy-based mitigation method, which establishes the entropy formulation based on the node's activity. The suggested model's specificity is 97.09%, compared to the minimal specificity reached by conventional techniques, including CNN (83.58%), RNN (86.17%), RF (60.46%), SVM (78.50%), and DNN (84.12%) and SMIE (88.42%), respectively.

Keywords SMOTE-ENC · Mitigation · CNN · CCNN · IoT · Botnet attacks

1 Introduction

The number of IoT botnet-based attacks [1, 2] has steadily increased due to the widespread use of IoT devices. One of the most destructive types of cyber attacks are botnet attacks [3, 4]. IoT Botnets are collections of infected IoT devices linked to the network and controlled by the adversary. Due to the diversification of attack methods and the continuous rise of threats, IoT systems [5] face significant challenges in providing strategies to detect and mitigate cyber-attacks. Once an adversary exploits an IoT's vulnerability, the device can be deployed as a botnet for launching cyberattacks. If it is feasible to identify and remove the bots before they initiate an actual attack, the IoT Botnet detection solution will prove more effective. Therefore, an efficient IoT Botnet detection methodology has become necessary for threat hunting and incident response to mitigate IoT attacks [6–8]. Recently, numerous researchers have proposed many techniques to detect botnet attacks and mitigate bots based on machine learning and deep learning models [9–14]. The feature selection is used as the sub selection of features [15]. But still, these techniques require enhancement to improve the accuracy and precision and reduce the time taken to detect zero-day attacks.

This research paper addresses the class imbalance problem while using real-time datasets to train Machine learning models. When a class imbalance is present in the dataset,

✉ Balaganesh Bojarajulu
balaganeshb@gmail.com

Sarvesh Tanwar
s.tanwar1521@gmail.com

[1] Amity Institute of Information Technology, Amity University, Noida, Uttar Pradesh 201301, India

these models proved inaccurate and ineffective with unseen attacks since the imbalanced data is inclined towards most attacks seen in the training dataset. In Addition to unbalanced data, poor feature selection leads to low accuracy, high computation time, extensive training data requirements, and overfitting challenges. To address the existing research gaps, we developed a customized CNN, a deep-learning strategy for IoT Botnet attack detection, with the following contributions,

1. Proposed and implemented an improved class imbalance processing using SMOTE-ENC for input data pre-processing.
2. Proposed and implemented feature extraction phase to determine improved entropy-based feature calculation and other feature sets.
3. A Customized Convolutional Neural Network Model has been devised to classify the captured network traffic as an attack or legitimate traffic.
4. In the final stage, we propose an entropy-based mitigation procedure to identify the attacker and non-attacker nodes in the network by calculating the entropy value of each node based on the behavior.

This research paper is organized as follows: Sect. 1 introduces the research gaps and the motivation behind the research work. Section 2, Reviews the literature on similar work done previously. In Sect. 3, the customized CNN model has been elaborated. Section 4 presents the results, validation, and comparison with of existing methodologies, and finally, the conclusion is derived in Sect. 5.

2 Literature review

In 2020, Huy-Trung Nguyena et al. [16] offered a new high-level, subgraph-based PSI function for detecting IoT botnets. In 2022, Michal Motylinski et al. [17] offered a method for classifying and pre-processing the different attack types included in the IoT-Bot database. In 2021, Chirag Joshi et al. [18] suggested a fuzzy logic-assisted feature engineering technique that finds the dataset's fuzzy elements before creating fuzzy sets. In 2020, Mehdi Asadi et al. [19] used the BD-PSO-V method to overcome the limitations of earlier investigations. In 2021, Jithu et al. [20] created a DL-based intrusion detection system to find IoT DDoS Botnet attacks. In 2021, Amir hossein Rezaei [21] aimed to reduce the number of features required and improve the accuracy of botnet detection on IoT by building an ensemble learning model combining the best ML techniques from supervised, unsupervised, and regression learning. In 2020, Abdulghani Ali Ahmed et al. [22] developed a method for detecting botnets that uses DL to recognize current botnet attacks. In 2021, Mehdi Asadi et al. [23] implemented a cooperative game theory and LSTM, Autoencoder, and SVM techniques to identify IoT botnet attacks. In 2023, Mohammed et al. [24], have suggested a strong technique for identifying botnet assaults on Internet of Things devices. This was accomplished by creatively fusing the long short-term memory (CNN-LSTM) algorithm mechanism with the model of a convolutional neural network to identify two prevalent and dangerous IoT attacks (BASHLITE and Mirai) on four different kinds of security cameras. In 2023, Hazem et al. [25] have introduced the BiLSTM-CNN model, which combines a convolutional neural network (CNN) and a bidirectional long-short-term memory recurrent neural network (RNN), is proposed in this research. This model uses the BiLSTM for classification and CNN for feature optimization and data processing.

2.1 Research gap

Many studies have been conducted on the topic of botnet detection. Due to their complex behavior, repeated nature, ambiguity, and nearly imperceptible existence in the compromised system, very little study has been done on botnet prediction. Furthermore, only a small amount of research has adequately addressed the issues of rigorous parameter optimization, overfitting, appropriate data preparation, and—above all—the computational cost of the model in order to use these models in real time. In order to investigate botnet identification, the current study employed.

3 Customized convolutional neural network model for IoT botnet attack detection

As IoT botnet remains the most alarming cybersecurity threat, it requires continuous detection and mitigation strategies improvement due to the growing cybersecurity threat landscape. Also, from the literature, we discovered the research gaps in the early-stage detection of IoT botnet attacks. To overcome the shortcomings, we have used an enhanced Synthetic minority over-sampling technique to preprocess and address the class imbalance in the dataset. In Addition, to yield better results, we have used statistical methods for extracting features and feeding the customized convolutional neural network, also called a feed-forward network, which is generally used for classification. Finally, an entropy method widely used to calculate the impurity in a system is used to detect the attacker node. In this context, this paper presents the detection and mitigation of IoT botnets in the early stage with the following phases:

- Pre-processing
- Feature extraction

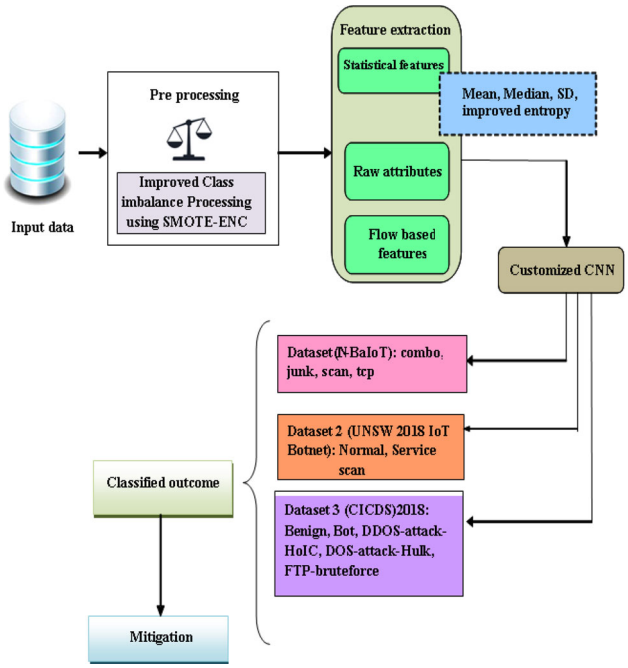

Fig. 1 Proposed model for attack detection

- Attack detection
- Mitigation

The overall design of the suggested Model for attack detection is shown in Fig. 1.

3.1 Pre-processing via SMOTE-ENC

This is the initial step, where the improved class imbalance problem of the input dataset $D = \{d_1, d_2, \ldots d_n\}$ where n the number of data will be solved via SMOTE-ENC processing. Class imbalance is among the main obstacles to attaining high accuracy of minority class data points in classification issues, which is resolved by the SMOTE-ENC approach. The SMOTE method depends on the KNN algorithm. Calculating the distance between the two instances is crucial in creating synthetic samples. Calculating the distance between continuous variables within the feature space is easy, but it can be challenging for categorical variables to calculate the distance between two labels. SMOTE-ENC gives a fixed value to the distance calculation whenever the categorical attribute label varies among an instance and its closest neighbors. As a result, using this method, the distance among any two labels inside a multi-label categorized feature is the same. According to SMOTE-ENC [26], a categorical variable's labels are characterized by their likelihood to associate with the minority class objective.

3.2 Feature extraction: extracting the MI-based feature, flow-based feature, and statistical features

The process of transforming unprocessed data into manageable numerical traits while preserving the specifics of the underlying data set is referred to as feature extraction. Feature extraction is the second phase of our work. Here, several features [27, 28] were extracted based on the pre-processed data set, including statistical features (Mean, Median, SD, improved entropy), MI, and flow-based features. The set of features that were extracted $F = [\overline{D_p}\ Med(D_p)\ \sigma\ IE\ MI\ RAFF]$.

Algorithm 1: Improved class imbalance processing using SMOTE-ENC

Input: $n\%$ (oversampling count), s (instances count inside training set), t (minority class count), k (nearest neighbor count), c (continuous variable count), m (median of continual features SD if $c \succ 0$)

$ir = t/s$

For every classified feature do

For every l inside the distinct label do

$e = l$ labeled instance count

$e^{'} = e * ir$

$o = l$ labeled minority class instance count

$\chi = \frac{(o - e^{'})}{e^{'}}$

if $c \succ 0$

$l = \chi * m * wf$; i.e., the label l is calculated as per the proposed Model with PWLCM randomization, which is a simple chaotic system popularly used for generating PRN.

Here,
$$wf_{k+1} = \begin{cases} wf_k/p & if\ 0 \le wf_k \le p \\ \frac{wf_k - p}{0.5 - p} & if\ p \le wf_k \le 0.5 \\ \frac{1 - p - wf_k}{0.5 - p} & if\ 0.5 \le wf_k \le 1 - p \\ 1 - wf_k/p & if\ 1 - p \le wf_k \le 1 \end{cases},$$

where $0 \le p \le 0.5$

else

$l = \chi * wf$

end

end

Set SMOTE(t,n,k)

The artificial data points' values are calculated using the maximum value of the identified attributes' nearest neighbors.

Categorical values should be inverse-encoded to their actual labels.

3.3 CCNN-based attack detection model

The attack detection process will be done based on the extracted feature set F. First, the extracted feature set F is given as input to the customized CNN model [29] to obtain the final output. Figure 2 represents the layer-wise customization of the CNN model. For the proposed model, the one layer is considered for input layer. InThe one layer is considered for convolutional layer, the one layer is considered for max-pooling 1D layer, the one layer is considered for attention layer and one layer is considered for output layer. In this paper, layer-wise customization of CNN includes the following layers:

- Input layer
- Convolution layer
- Max pooling layer
- Modified attention layer
- Fully connected layer

3.3.1 Input layer

The extracted feature set F will be given as input in the input layer.

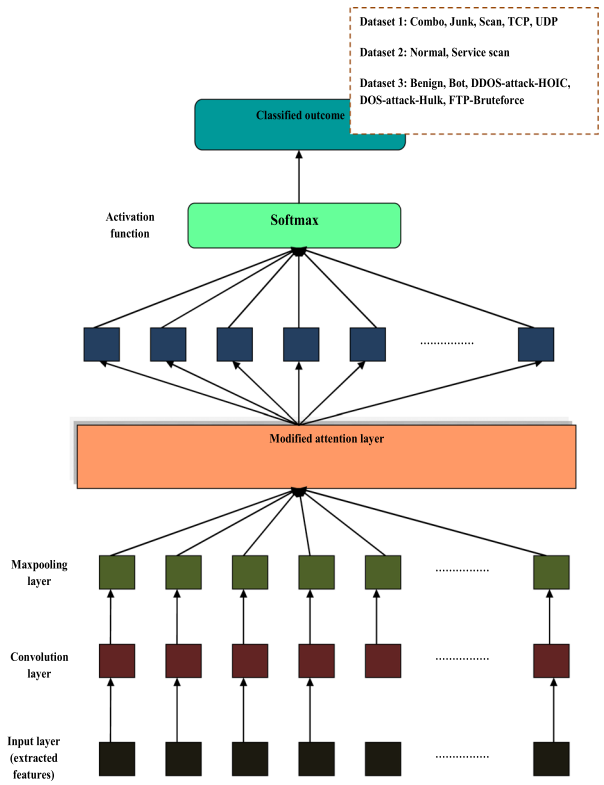

Fig. 2 Customized CNN model for attack detection

3.3.2 Convolution layer

In CNN architectures, convolutional layers are the primary building blocks. By employing a filter or kernel, convolution layers change the input feature. To create the feature map, the layer executes a dot product operation between the filters and the area of the input layer's neurons. Equation (1) describes how the convolutional layer operates, depicting a 2D feature map output and 2D filter having the size of $m \times n$.

$$M(i,\ j) = (F * K)(i,\ j) = \sum\sum F(i+m,\ j+n)K(m,\ n) \quad (1)$$

3.3.3 Max pooling layer

After each additional convolutional layer, pooling layers are added. To lower the spatial dimension of feature maps, CNNs use a pooling layer to combine the outputs of nearby neurons within the same filter or map. Pooling layers help manage to overfit and reduce data representation. The area that each pooling covers is summarised. Max pooling resizes input spatially using the max () technique. With a 2×2 filter, for instance, max () selects the greatest value among the four values. A reduced feature map is the outcome of the max pooling layer.

3.3.4 Modified attention layer

Attention mechanisms are input processing methods for neural networks that allow the network to concentrate on particular properties of a complicated input one at a time till the complete set of data has been classified. We follow a new attention mechanism to enhance the feature's importance, including the computation of Information Gain (IG). This evaluation boosts the impact of input features to be trained well to categorize the detection result. The procedure for improved attention mechanism includes:

Step 1: Find the Information Gain IG calculated as in Eq. (2) for all features, where $H(S)$ depicts the entropy of the dataset S and $H(S|a)$ depicts the conditional entropy of S a given variable a. Then the feature set F which having useful information (IF) will be selected, i.e., $F + IF$

$$IG = H(S) = H(S|a) \quad (2)$$

Step 2: The input variable e is calculated as in Eq. (3), where w depicts weight function and b depicts bias.

$$e = \tanh(IG \times w + b) \quad (3)$$

Here, $\tanh(F) = \frac{1}{1+e^{-2F}} - 1$

Step 3: Compute the context vector C using Eq. (4).

$$C = \sum (IG \times \alpha) \tag{4}$$

Here, $\alpha = soft\max(e)$

3.3.5 Fully connected (FC) layer

Every input from earlier layers is coupled to all FC layer neurons, which function on a flattened input. FC layers may be utilized to improve objectives like class scores and are typically seen towards the finish of CNN systems. This layer calculates the probability of the output classes for the input feature. The FC layer produces the final classified output: For dataset 1, the classified result will be a combo, junk, scan, TCP, and UDP. For dataset 2, the classified outcome will be normal and service scan. For dataset 3, the classified outcome will be Benign, Bot, DDOS-attack-HOIC, DDOS-attack-Hulk, and FTP-Brute force.

Loss function: Conventionally, cross-entropy EL (defined in Eq. (5)) loss function is calculated, which is a metric employed for calculating how well the classification algorithm in ML performs, and as per our proposed work, we used the improved tanh-based cross-entropy loss (CEL) calculation (defined in Eq. (6)) between actual and predicted value.

$$EL = -\sum_{i=1}^{n} t_i \log t_i \tag{5}$$

$$CEL = -\sum_{i=1}^{n} t_i \log(f(s)_i) \tag{6}$$

Here, $f(s) = \frac{1}{m}[(\tanh)+(m-1)(\tanh)]$, $\tanh = \frac{2}{1+e^{-2x}} - 1$

3.4 Mitigation

After the detection, there is a need for the mitigation procedure to remove the attacker node from the network. Hence, this work follows a new mitigation method termed a modified entropy-based mitigation procedure to mitigate the attacker node. Let us consider the network scenario: The network consists of N several nodes where $N = 1$ with an attacker node and a non-attacker node.

For each node, calculate the Entropy value Etp as defined in Eq. (7). The formulation evaluates the behavior of the attacker and non-attacker nodes. This will be finalized based on the threshold value fixed, the threshold value th (defined in Eq. (8), which Cr depicts the correlation coefficient. If the threshold value exceeds the entropy value, the node is determined as an attacker node. If the threshold value is less than the entropy, the node is termed the non-attacker node.

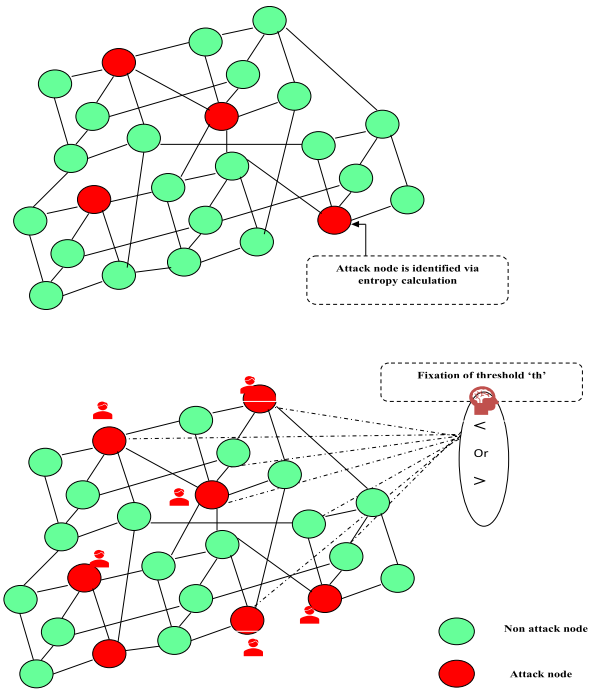

Fig. 3 A network model of the mitigation process

Figure 3 represents the network model of the mitigation process

$$Etp = -\sum_{i=1}^{N} p_i \log p_i + Cr \tag{7}$$

Here, $Cr = \frac{\sum (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum (x_i - \overline{x})^2 (y_i - \overline{y})}}$

$$th = Mean(Etp) \tag{8}$$

4 Result from analysis on CCNN-based IoT-botnet attack detection

4.1 Experimentation

The CCNN approach was implemented in Python. The dataset was accumulated in [30, 31], and [32]. The CCNN was compared with the traditional algorithms, such as CNN, SVM, DNN [33], DBN, RF [34], NN, NB, and RNN, respectively. Similarly, the assessment was conducted regarding Specificity, FPR, MCC, Accuracy, and other performance measures while adjusting the learning percentage. Figures 4, 5, 6 illustrates the mitigation process.

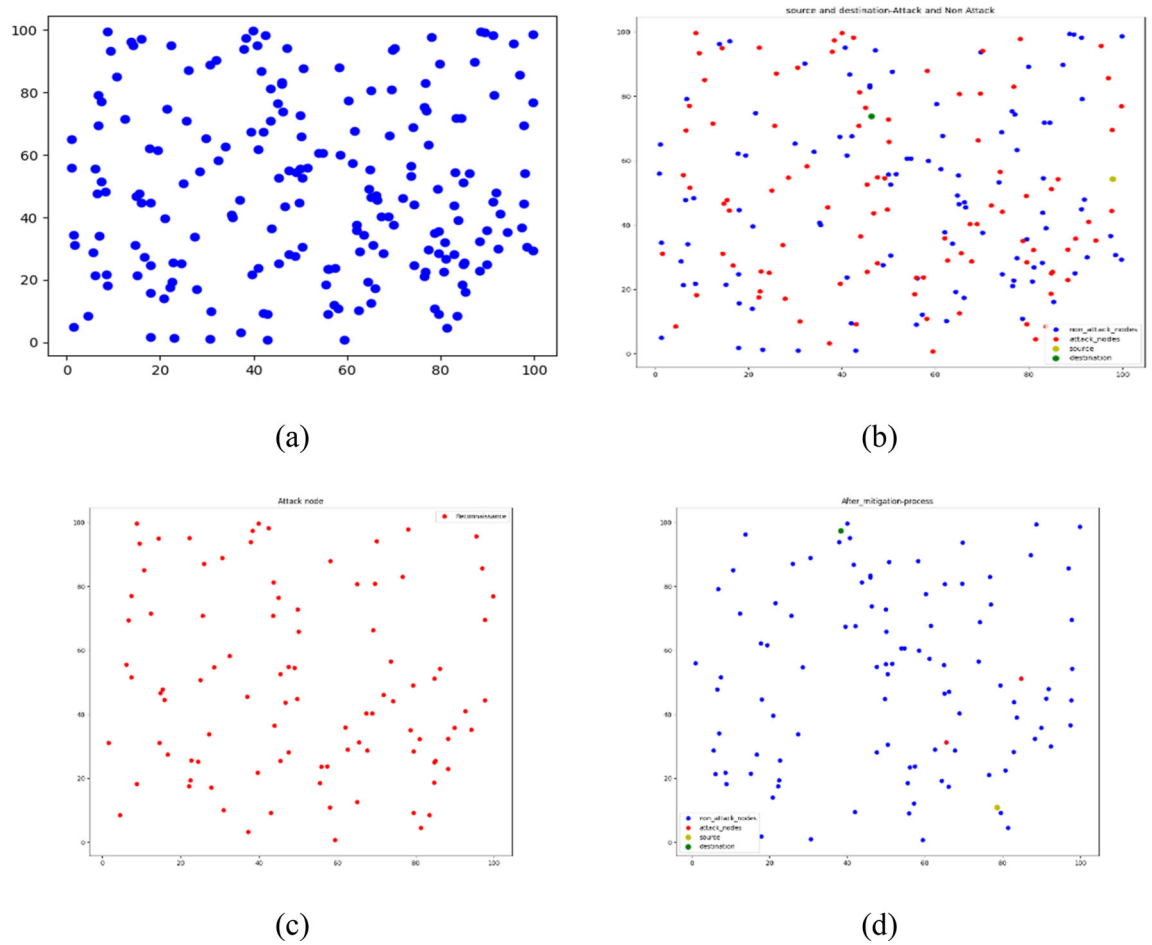

Fig. 4 Mitigation analysis of the proposed Model for dataset 1 **a** total count of nodes in a network **b** attack and normal nodes in a network **c** attack node detection and **d** after mitigation

4.2 Dataset 1 description

The dataset has captured pcap files that are 69.3 GB, with more than 72,000,000 records. The extracted flow traffic, in CSV format, is 16.7 GB. The dataset includes DDoS, DoS, OS and Service Scan, Keylogging and Data exfiltration attacks, with the DDoS and DoS attacks further organized, based on the protocol used.

4.3 Dataset 2 description

For each of the 9 IoT devices, we trained and optimized a deep auto-encoder on 2/3 of its benign data (i.e., the training set of each device). This was done to capture standard network traffic patterns.

The test data of each device comprised the remaining 1/3 of benign data plus all the malicious data. We applied the respective trained (deep) auto-encoder on each test set as an anomaly detector. The detection of anomalies (i.e., the cyberattacks launched from each of the above IoT devices) concluded with 100% TPR.

4.4 Dataset3 description

The BoT-IoT dataset was created by designing a realistic network environment in the Cyber Range Lab of UNSW Canberra. The network environment incorporated a combination of normal and botnet traffic. The dataset's source files are provided in different formats, including the original pcap files, the generated argus files, and CSV files. The files were separated, based on attack category and subcategory, to better assist in the labeling process.

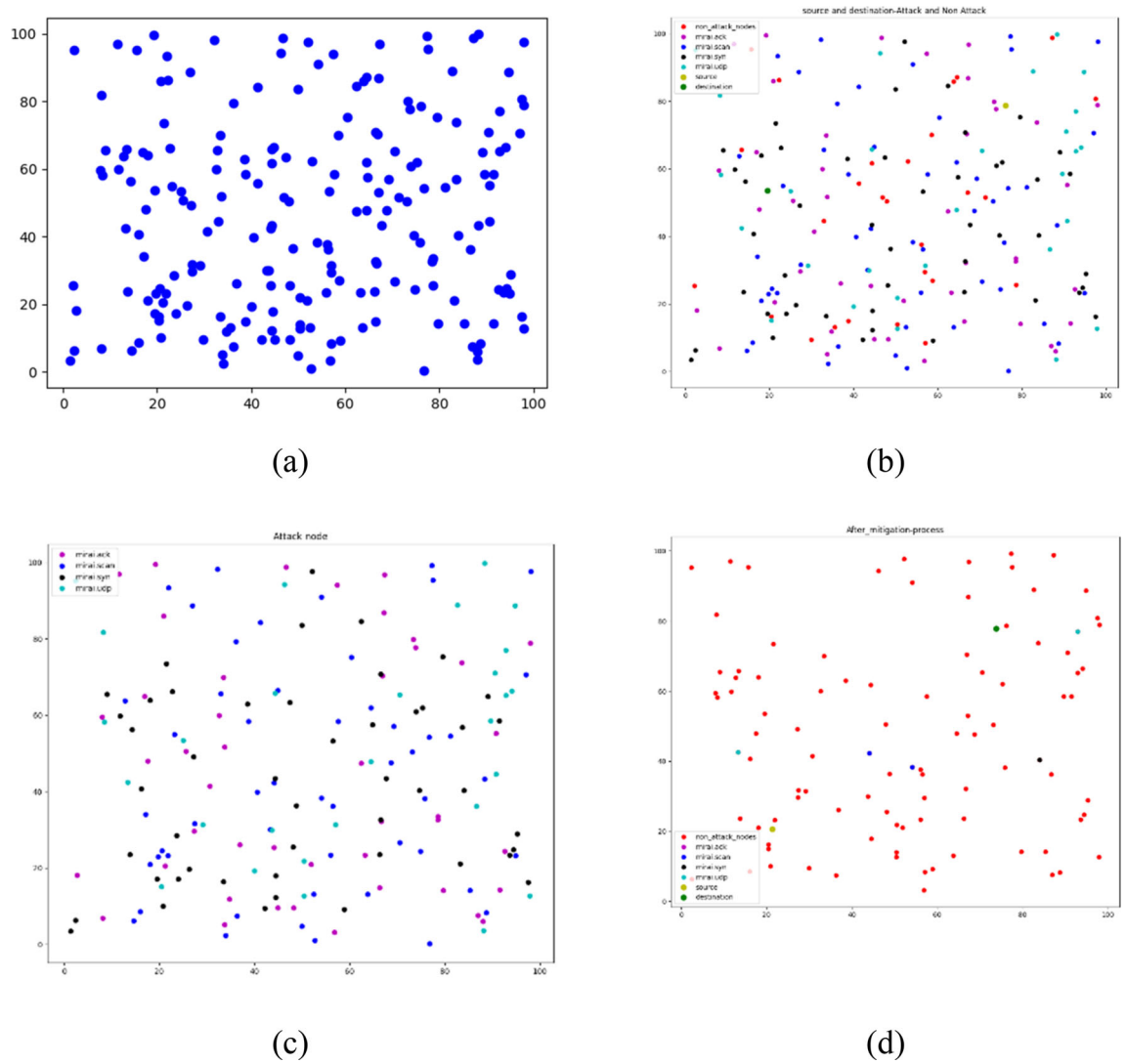

Fig. 5 Mitigation analysis of a proposed model for dataset 2 **a** total count of nodes in the network **b** attack and normal nodes in the network **c** attack node detection and **d** after mitigation

4.5 Evaluation of CCNN and the conventional strategies regarding other measures for IoT-botnet attack detection using dataset1

The performance assessment on CCNN over the traditional schemes for accurately detecting IoT-botnet attacks employs three different datasets. Also, the CCNN is contrasted against the RNN, NB, DNN, SVM, DBN, RF, CNN, and NN as regards positive, negative, and other measures. Moreover, the specificity of the CCNN for the 90th learning IoT-botnet attack detection results are displayed in Figs. 7, 8 and 9. As we examined in Fig. 7a, while fixing the learning percentage to 60, most algorithms achieved accuracy ratings above 90%. However, the CCNN accomplished the maximal accuracy value. In particular, the accuracy of the CCNN is 97.53%, even though the CNN is 87.46%, NB is 94.50%, SVM is 84.17%, RNN is 86.76%, RF is 90.14%, NN is 94.50%, and DNN is 92.32%, SMIE is 87.51% respectively. This emphasizes that the CCNN is substantially more accurate at detecting the IoT-botnet attack due to the Improved class imbalance processing with SMOTE-ENC and Improved entropy calculation-based feature extraction. Also, using the Modified entropy-based Mitigation method, the attacker node is eliminated from the network.

4.6 Impact of CCNN-based detection model, a model with conventional entropy, a model with conventional CNN, and a model with conventional Smote for IoT-botnet attack detection using dataset1, dataset2, and dataset3

The ablation examination on CCNN, a model with conventional entropy, a model with conventional CNN, and a model

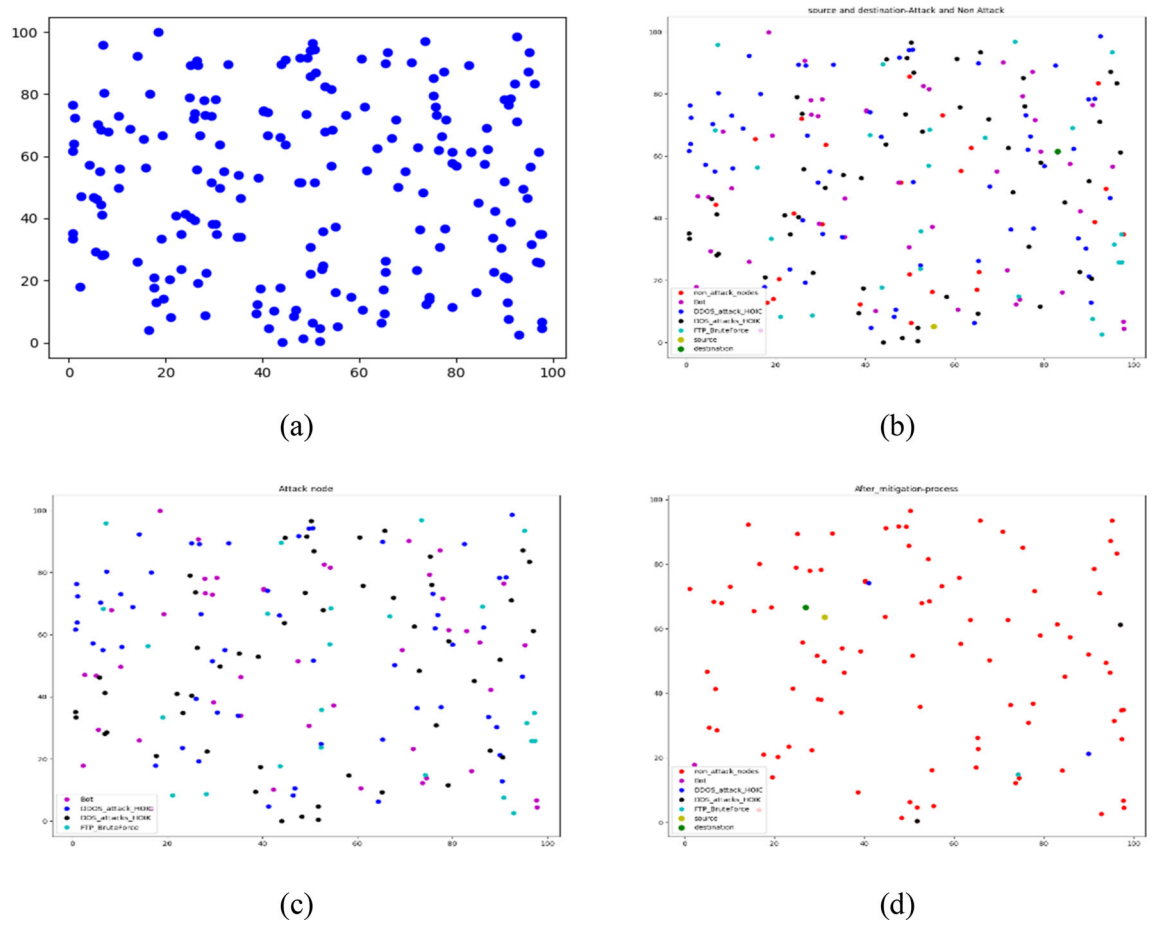

Fig. 6 Mitigation analysis of the proposed Model for dataset 3 **a** total count of nodes in the network **b** attack and normal nodes in the network **c** attack node detection and **d** after mitigation

with conventional smote for IoT-botnet attack detection using dataset1, dataset2, and dataset3, is displayed in Table 1. The Model, with all the improvements, can accurately detect the IoT-botnet attack. Concerning dataset 1, the precision of the CCNN is 96.33%, the Model with conventional entropy is 95.62%, the Model with conventional CNN is 86.69%, and the Model with conventional Smote is 91.15%, respectively. Similarly, the CCNN is 0.036697, the Model with conventional entropy, the Model with conventional CNN, and the Model with conventional Smote have acquired the FNR of 0.0467, 0.1412, 0.0940. Additionally, the CCNN generated an accuracy of 94.24%, the Model with conventional entropy = 92.94%, the Model with conventional CNN = 79.72%, and the Model with conventional Smote = 86.33%. This implies that the CCNN enhances the system's capability in detecting the IoT-botnet attack with Improved entropy calculation-based feature extraction, Customized CNN, and Improved class imbalance processing with SMOTE-ENC-based preprocessing.

4.7 Statistical assessment on CCNN and the traditional approaches with regard to error for IoT-Botnet attack detection using dataset1, dataset2, and dataset3

The statistical assessment of CCNN is contrasted over the RNN, DNN, SVM, NN, DBN, RF, CNN, and NB for IoT-Botnet attack detection under distinct statistical measures is represented in Table 2. This examination assesses the CCNN work's effectiveness using three different datasets. The findings are frequently examined through statistical study due to the unpredictable nature of the optimization procedure. The recorded values for the attack detection illustrate that the CCNN surpassed the traditional methodologies. Further, the findings revealed that the CCNN scored the least error rate over the conventional approaches. Mainly, analyzing the dataset1, the CCNN attained the minimal error value for the maximum statistical measure is 0.909483, meanwhile the CNN = 0.7269, DBN = 0.672, SVM = 0.477, NN = 0.353, RNN = 0.768, NB = 0.7264, DNN = 0.7408, RF = 0.7264, and SMIE = 0.8425 at LP 60 respectively.

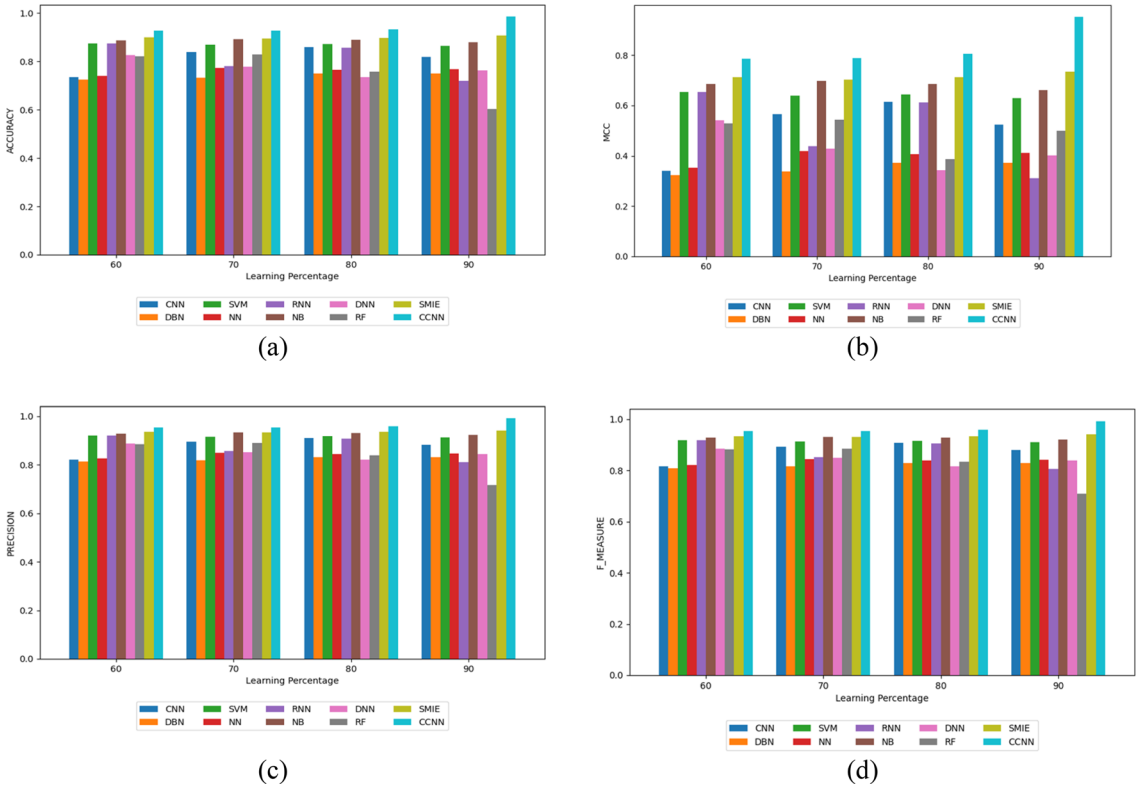

Fig. 7 Comparison of CCNN with other traditional strategies concerning measures for dataset 1

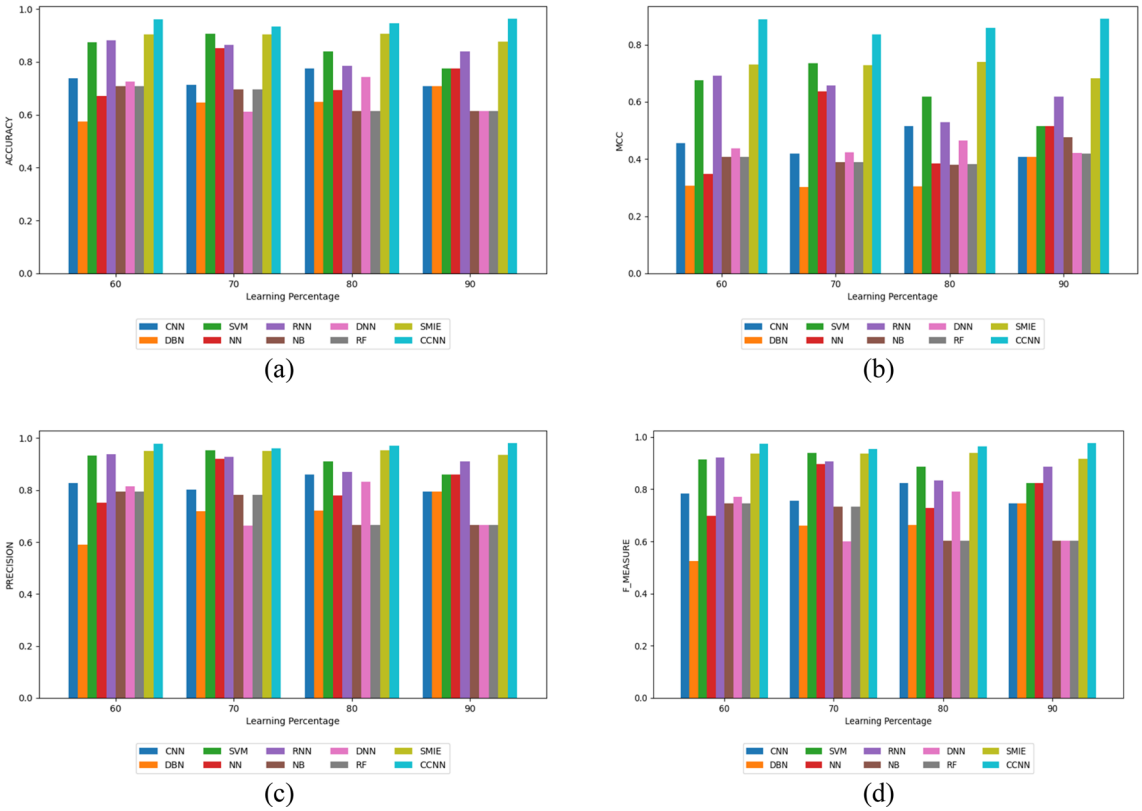

Fig. 8 Comparison of CCNN with other traditional strategies concerning measures for IoT-botnet attack detection using dataset2

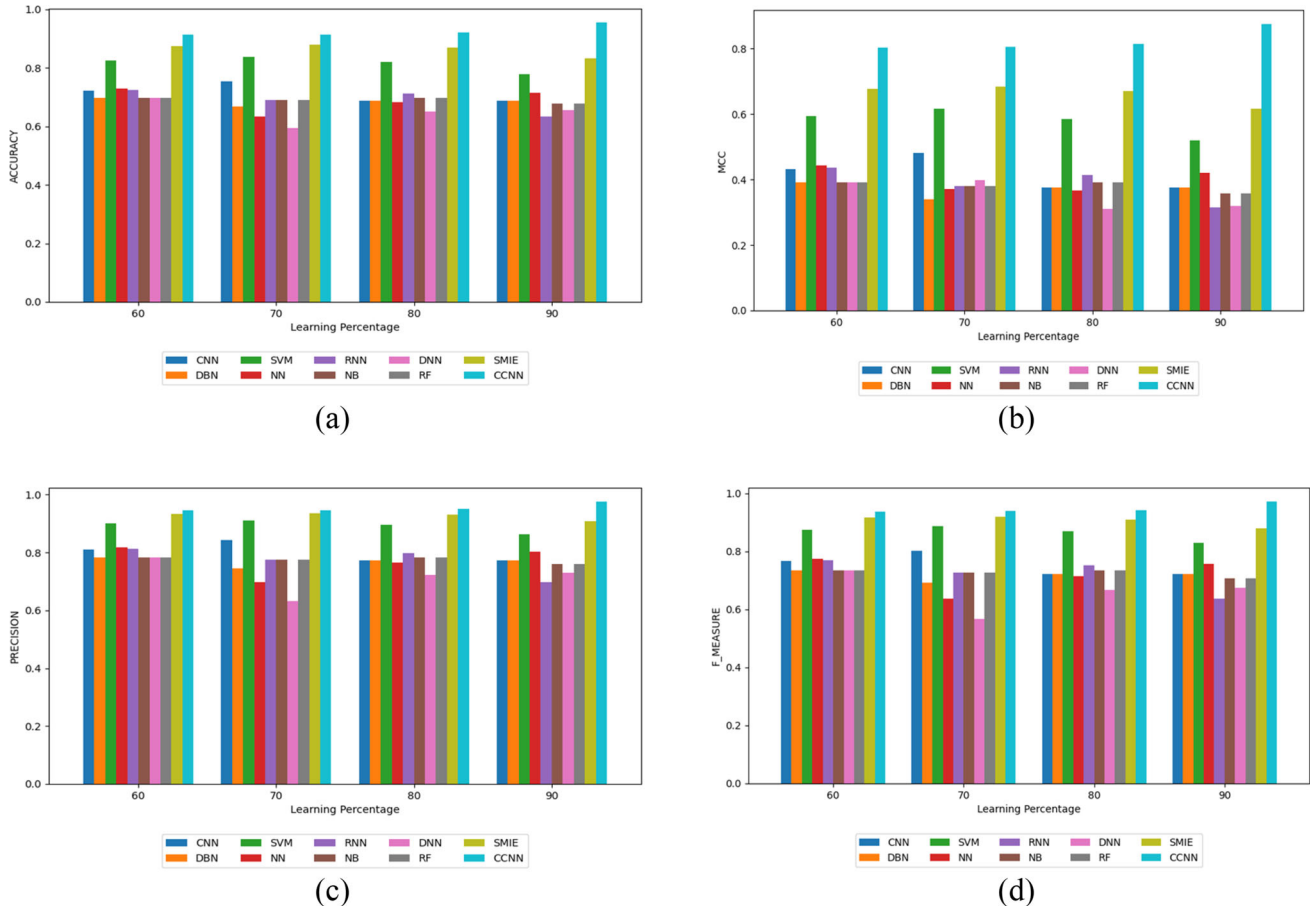

Fig. 9 Comparison of CCNN with other traditional strategies concerning measures for IoT-botnet attack detection using dataset3

Table 1 Ablation study on CCNN, the model with conventional entropy, the model with conventional CNN, and the model with conventional smote for IoT-botnet attack detection employing dataset1, dataset2, and dataset3

	The model with conventional entropy	The model with conventional CNN	The model with conventional Smote	CCNN
Dataset1				
Accuracy	0.929439	0.797266	0.863353	0.942446
Precision	0.956201	0.866985	0.911593	0.963303
f_measure	0.954701	0.862866	0.908783	0.963303
MCC	0.795203	0.474331	0.634767	0.829969
Dataset2				
Accuracy	0.881336	0.908578	0.894957	0.944444
Precision	0.918944	0.941331	0.930138	0.973684
f_measure	0.912487	0.936534	0.92451	0.961039
MCC	0.728438	0.773282	0.75086	0.865207
Dataset3				
Accuracy	0.823288	0.900563	0.861926	0.958333
Precision	0.860096	0.935031	0.897564	0.982143
f_measure	0.849717	0.929757	0.889737	0.973451
MCC	0.635645	0.75976	0.697703	0.877454

Table 2 Statistical evaluation on CCNN over the conventional schemes to error for IoT-Botnet attack detection employing dataset1, dataset2, and dataset3

	CNN	DBN	SVM	NN	RNN	NB	DNN	RF	SMIE	CCNN
Dataset1										
Mean	0.79565	0.72016	0.6616	0.6605	0.82202	0.738	0.7855	0.7383	0.85221	0.93744
Median	0.80655	0.72887	0.7131	0.7587	0.83703	0.737	0.78944	0.73793	0.85472	0.94485
Std	0.04896	0.02928	0.1072	0.1774	0.03133	0.009	0.03601	0.00984	0.00564	0.01640
Min	0.72690	0.672	0.4770	0.3532	0.76859	0.726	0.74074	0.72641	0.84259	0.90948
Max	0.84259	0.75091	0.7431	0.7714	0.84545	0.750	0.82236	0.75091	0.85682	0.95058
Dataset 2										
Mean	0.6710	0.68815	0.71302	0.6850	0.7679	0.66484	0.69903	0.68865	0.8872	0.92650
Median	0.6648	0.69871	0.73084	0.7312	0.78285	0.66825	0.68775	0.68364	0.91400	0.93634
Std	0.0624	0.04884	0.06664	0.1097	0.07975	0.01503	0.05304	0.05910	0.05054	0.02223
Min	0.6046	0.61818	0.61538	0.5	0.64285	0.64285	0.64285	0.61538	0.8	0.88889
Max	0.75	0.73684	0.775	0.7777	0.86315	0.68	0.77777	0.77193	0.92079	0.94444
Dataset 3										
Mean	0.6922	0.64158	0.70922	0.7156	0.68572	0.71003	0.69690	0.70143	0.82680	0.93116
Median	0.6970	0.63965	0.72254	0.7149	0.68535	0.71583	0.69574	0.69672	0.82643	0.92541
Std	0.0372	0.01166	0.03218	0.0171	0.03445	0.01235	0.03300	0.01310	0.00751	0.01686
Min	0.6375	0.62820	0.65625	0.6927	0.6375	0.68888	0.65354	0.68888	0.81666	0.91549
Max	0.7373	0.65882	0.73553	0.74	0.73469	0.71957	0.74257	0.72340	0.83769	0.958333

Simultaneously, assessing the mean statistical measure, the strategies like CNN, DBN, SVM, NN, RNN, NB, DNN, RF, and SMIE have scored the error value of 0.795, 0.720, 0.661, 0.660, 0.822, 0.7383, 0.7855, 0.7383, and 0.8522 whereas the CCNN obtained the error value of 0.9374. Likewise, for dataset 2 and dataset 3, the CCNN accomplished minimized error values in most statistical measures than the conventional schemes. Therefore, the betterment of the CCNN is attained for detecting IoT-botnet attacks with Improved class imbalance with SMOTE-ENC, Improved Entropy calculation, and Customized CNN.

4.8 Computational time analysis

Table 3 shows the computational time analysis. The computational time of the proposed model is better (~ 57.024) when compared to the CNN (~ 91.064), DBN (~ 93.651), SVM (~ 87.926), NN (~ 88.602), RNN (~ 83.762), NB (~ 94.91), DNN (~ 87.926), RF (~ 82.32), and SMIE (~ 81.14). Thus, the superiority of the proposed model is proved over other existing models.

Table 3 Analysis on computational time

Methods	Computation time
CNN	91.06443
DBN	93.65158
SVM	87.92617
NN	88.60289
RNN	83.76241
NB	94.91794
DNN	87.08512
RF	82.32108
SMIE	81.14128
CCNN	57.02474

5 Conclusion

Sophisticated techniques used by threat actors to create and manage IoT botnets for cyber security crimes have become a serious concern, as it is challenging to identify these persistent threats on the network early. With the intent to address this issue, this research paper has analyzed the most recent research on identifying and classifying botnet attacks in the IoT using machine learning and deep learning approaches and has summarized the taxonomies with the aim of resolving this issue. Additionally, employing a four stage method that comprises pre-processing, feature extraction, attack detection, and mitigation, this research has developed a customized CNN model for IoT Botnet attack detection and eradication. The pre-processing stage

employed the Enhanced Synthetic Minority Oversampling Technique (SMOTE-ENC) to enhance the class imbalance in the input data. Statistical features including Mean, Median, Standard Deviation, Improved Entropy, Mutual information (MI), Flow-Based features, and Raw Attributes are extracted during the feature extraction step and used as input to the customized CNN to determine whether the traffic is legitimate or malicious. After the attack was discovered, the attacker and non-attacker nodes were located using an entropy-based mitigation approach. Finally, a number of experiments were run to compare the suggested Model's findings to those of earlier models. Significantly, the CCNN's highest Matthews correlation coefficient (MCC) was 94.25%, with an 80 learning percent. The traditional approaches, on the other hand, had Matthews correlation coefficients (MCC) that were significantly lower, with values like NN = 87.41%, DNN = 79.91%, CNN = 86.77%, RF = 64.45%, DBN = 88.67%, and RNN = 54.77%, respectively. The limitation of the proposed model it is not used for different scenarios. In the future, we would like to investigate this model in an SDN framework, adding Open Flow-enabled switches to the research scenario designed. Through this research, we will be able to measure the mitigation time in an actual equipment setting. Additionally, in order to add even more protection, we intend to apply this approach to virtual computers housed in data centres.

Authors contributions BB conceived the presented idea and designed the analysis. Also, he carried out the experiment and wrote the manuscript with support from ST. All authors discussed the results and contributed to the final manuscript. All authors read and approved the final manuscript.

Funding This research did not receive any specific funding.

Data availability https://www.impactcybertrust.org/dataset_view?idDataset=1296
https://www.kaggle.com/datasets/mkashifn/nbaiot-dataset
https://research.unsw.edu.au/projects/bot-iot-dataset

Declarations

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

Ethical approval Not applicable.

Informed consent Not applicable.

References

1. Nguyen, H.-T., Ngo, Q.-D., Le, V.-H.: A novel graph-based approach for IoT botnet detection. Int. J. Inf. Secur. (2019). https://doi.org/10.1007/s10207-019-00475-6
2. Al Shorman, A., Faris, H., Aljarah, I.: Unsupervised intelligent system based on one class support vector machine and Grey Wolf optimization for IoT botnet detection. J. Ambient. Intell. Human. Comput. (2019). https://doi.org/10.1007/s12652-019-01387-y
3. Lee, S., Abdullah, A., Jhanjhi, N., Kok, S.: Classification of botnet attacks in IoT smart factory using honeypot combined with machine learning. PeerJ Comput. Sci. **7**, e350 (2021)
4. Lee, S., Abdullah, A., Jhanjhi, N.Z.: A review on honeypot-based botnet detection models for smart factory. Int. J. Adv. Comput. Sci. Appl. **11**(6), 418–435 (2020)
5. Zago, M., Gil Pérez, M., Martínez Pérez, G.: Early DGA-based botnet identification: pushing detection to the edges. Clust. Comput. **24**(3), 1695–1710 (2021). https://doi.org/10.1007/s10586-020-03213-z
6. Gelenbe, E., Nakıp, M.: Traffic based sequential learning during botnet attacks to identify compromised IoT devices. IEEE Access **10**, 126536–126549 (2022). https://doi.org/10.1109/ACCESS.2022.3226700
7. Sattari, F., Farooqi, A.H., Qadir, Z., Raza, B., Nazari, H., Almutiry, M.: A hybrid deep learning approach for bottleneck detection in IoT. IEEE Access **10**, 77039–77053 (2022). https://doi.org/10.1109/ACCESS.2022.3188635
8. Hatzivasilis, G., Soultatos, O., Chatziadam, P., Fysarakis, K., Askoxylakis, I., Ioannidis, S., Spanoudakis, G.: WARDOG: awareness detection watchbog for Botnet infection on the host device. IEEE Trans. Sustain. Comput. (2019). https://doi.org/10.1109/tsusc.2019.2914917
9. Hussain, F., et al.: A two-fold machine learning approach to prevent and detect IoT botnet attacks. IEEE Access **9**, 163412–163430 (2021). https://doi.org/10.1109/ACCESS.2021.3131014
10. Kalakoti, R., Nõmm, S., Bahsi, H.: In-depth feature selection for the statistical machine learning-based botnet detection in IoT networks. IEEE Access **10**, 94518–94535 (2022). https://doi.org/10.1109/ACCESS.2022.3204001
11. Panda, M., Mousa, A.A.A., Hassanien, A.E.: Developing an efficient feature engineering and machine learning model for detecting IoT-Botnet cyber attacks. IEEE Access **9**, 91038–91052 (2021). https://doi.org/10.1109/ACCESS.2021.3092054
12. Sajjad, S.M., Yousaf, M., Afzal, H., Mufti, M.R.: eMUD: enhanced manufacturer usage description for IoT botnets prevention on home WiFi routers. IEEE Access **8**, 164200–164213 (2020). https://doi.org/10.1109/ACCESS.2020.3022272
13. Yin, L., Luo, X., Zhu, C., Wang, L., Xu, Z., Lu, H.: ConnSpoiler: disrupting C&C communication of IoT-based botnet through fast detection of anomalous domain queries. IEEE Trans. Ind. Inform. **16**(2), 1373–1384 (2020). https://doi.org/10.1109/TII.2019.2940742
14. Popoola, S.I., Adebisi, B., Hammoudeh, M., Gui, G., Gacanin, H.: Hybrid deep learning for botnet attack detection in the Internet-of-Things networks. IEEE Internet Things J. **8**(6), 4944–4956 (2021). https://doi.org/10.1109/JIOT.2020.3034156
15. Beraha, M., Metelliy, A.M., Papiniy, M., Tirinzoniy, A., Restelli, M.: Feature selection via mutual information: new theoretical insights. arXiv:1907.07384v1 [cs.LG] (2019)
16. Nguyen, H.-T., Ngo, Q.-D., Nguyen, D.-H., Le, V.-H.: PSI-rooted subgraph: a novel feature for IoT botnet detection using classifier algorithms. ICT Express (2020). https://doi.org/10.1016/j.icte.2019.12.001
17. Motylinski, M., Dermott, Á.M., Iqbal, F., Shah, B.: A GPU-based machine learning approach for detection of botnet attacks. Comput. Secur. **123**, 102918 (2022)
18. Joshi, C., Ranjan, R.K., Bharti, V.: A fuzzy logic based feature engineering approach for Botnet detection using ANN. J. King Saud Univ. Comput. Inf. Sci. (2021). https://doi.org/10.1016/j.jksuci.2021.06.018
19. Asadi, M., Jamali, M.A.J., Parsa, S., Majidnezhad, V.: Detecting botnet by using particle swarm optimization algorithm based on

voting system. Future Gener. Comput. Syst. (2020). https://doi.org/10.1016/j.future.2020.01.055
20. Shareena, J., Ramdas, A., AP, H.: Intrusion detection system for IOT botnet attacks using deep learning. SN Comput. Sci. (2021). https://doi.org/10.1007/s42979-021-00516-9
21. Rezaei, A.: Using ensemble learning technique for detecting botnet on IoT. SN Comput. Sci. (2021). https://doi.org/10.1007/s42979-021-00585-w
22. Ahmed, A.A., Jabbar, W.A., Sadiq, A.S., Patel, H.: Deep learning-based classification model for botnet attack detection. J. Ambient. Intell. Human. Comput. (2020). https://doi.org/10.1007/s12652-020-01848-9
23. Asadi, M.: Detecting IoT botnets based on the combination of cooperative game theory with deep and machine learning approaches. J. Ambient. Intell. Human. Comput. (2021). https://doi.org/10.1007/s12652-021-03185-x
24. Alzahrani, M.Y., Bamhdi, A.M.: Hybrid deep-learning model to detect botnet attacks over internet of things environments. Soft. Comput. **26**(16), 7721–7735 (2022)
25. Hezam, A.A., Mostafa, S.A., Baharum, Z., Alanda, A., Salikon, M.Z.: Combining deep learning models for enhancing the detection of botnet attacks in multiple sensors internet of things networks. Int. J. Inform. Visual. **5**(4), 380–387 (2021)
26. Mukherjee, M., Khushi, M.: SMOTE-ENC: a novel SMOTE-based method to generate synthetic data for nominal and continuous features. Appl. Syst. Innov. **4**, 18 (2021). https://doi.org/10.3390/asi4010018
27. https://www.csueastbay.edu/scaa/files/docs/student-handouts/marija-stanojcic-mean-median-mode-variance-standard-deviation.pdf
28. Yong Deng, Y.: Deng entropy. Chaos Solitons Fractals **91**, 549–553 (2016). https://doi.org/10.1016/j.chaos.2016.07.014
29. Sowmya, S., Jose, D.: Contemplate on ECG signals and classification of arrhythmia signals using CNN-LSTM deep learning model. Meas. Sens. **24**, 100558 (2022)
30. https://www.impactcybertrust.org/dataset_view?idDataset=1296
31. https://www.kaggle.com/datasets/mkashifn/nbaiot-dataset
32. https://research.unsw.edu.au/projects/bot-iot-dataset
33. Sriram, S., Vinayakumar, R., Alazab, M., Soman, K.P.: Network flow based IoT botnet attack detection using deep learning. In: IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) (2020)
34. Alissa, K., Alyas, T., Zafar, K., Abbas, Q., Tabassum, N., Sakib, S.: Botnet attack detection in IoT using machine learning. Comput. Intell. Neurosci.**2022**(1), 4515642 (2022)