SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V11I4P108

Volume 11 Issue 4, 68-79, April 2024

Original Article

A Network Intrusion Detection System Based on Enhanced CNN2D for IoT Architecture

Manasa Koppula[1], L.M.I. Leo Joseph[2]

[1,2]*Department of Electronics and Communication Engineering, SR University, Telangana, India.*

[1]*Corresponding Author : manasa436@gmail.com*

Received: 10 February 2024 Revised: 07 March 2024 Accepted: 07 April 2024 Published: 30 April 2024

Abstract - *The Internet of Things has experienced explosive evolution as a ground-breaking phenomenon since its conception. The security sector has witnessed enormous growth in cyberattacks as a consequence of the increasing growth of IoT devices, which expanded the attack vector for hackers to carry out significantly more damaging vulnerabilities. A key component of assuring the cybersecurity of IoT is the identification of anomalies in network activity using an intrusion detection system. Conventional machine learning methods appear vain in the face of inconsistent network expertise and several attack tactics. Deep learning methods have proved their capability to recognize irregularities in a wide range of research fields accurately. An excellent substitute for conventional methods of anomaly detection and classification is Convolutional Neural Networks (CNN). In this research, a novel IDS-based improved CNN model for IoT networks has been developed. To solve the issue of overfitting and improve the sophistication of the classifier, various regularization techniques, including L1, L2, Dropout, and multi-regularization, have been deployed. The experimental findings demonstrate that, when contrasted to the other CNN2D models, the proposed method outperforms with an above 98% accuracy. The Detection Rate and False Discovery Rate of individual classes are above 0.9 and below 0.1, respectively.*

Keywords - *Internet of Things, Convolutional Neural Networks, Regularization, Overfitting, Intrusion Detection System.*

1. Introduction

The Internet of Things (IoT), a revolutionary technology that allows gadgets equipped with sensors and network support to connect to the Internet, has recently developed. IoT increases the effectiveness of every device and makes it easier and more rational to use limited resources. Users have access to the capacity to combine internet-enabled devices, information, and applications [1]. Statista projects that by 2030, there will be about 29 billion IoT devices in use worldwide, up from 9.7 billion in 2020 [2].

The IoT network design has grown proportionately as a result of the exponential growth in the number of IoT devices, which already stand in the billions. The drawback of this, though, is that it raises issues of security and privacy concerning the gathering and transfer of information. When data is moved from one device to another over a wide area of the network, it is frequently endangered in terms of both individual and organizational security.

An Intrusion Detection System (IDS) is frequently utilized to track network activity and provides a prompt alert signal as a vital countermeasure for IoT/cyber security threats. An IDS leverages hybrid, anomalous, and signature-based approaches for detection. By comparing known patterns or a predetermined set of criteria, signature-based techniques identify intrusions, whereas anomaly-based strategies concentrate on the actions of the current user to identify disruptions [3]. Attacks can be detected using anomaly monitoring frameworks, but developing complicated rules for larger collections of data can be costly, time-consuming, and prone to errors [4].

In cybersecurity applications, including intrusion detection, authentication mechanisms, and privacy protection, ML (Machine Learning) and DL (Deep Learning) algorithms are being employed progressively more often. With the help of these cutting-edge learning techniques, it may be possible to examine and draw conclusions from the underpinning IoT information to enhance threat analysis and detection systems and, ultimately, uncover security holes in the IoT platform. ML and DL for cybersecurity are extensible and independent because they enable a system to adapt from its expertise as it expands and self-tune to grow more effectively and efficiently.

A framework was developed by Sattari et al. [5] that utilizes DL techniques to identify IoT security threats. A hybrid IDS utilizing Random Forest (RF) and Autoencoder (AE) was proposed by Chao Wang et al. [6]. The probability

output of the RF classifier is used in the first step to identify attack samples, which is particularly helpful for unidentified attacks. To lower false positives, an extra AE is included in the second phase. Their method showed a high detection rate and significantly reduced false positives in studies emulating unknown attacks when compared to different baselines.

Jingyi et al. [7] used the IoTID20 [8] dataset for attack classification. The authors utilized Decision-Tree, Gradient-boosting, and Random-Forest machine algorithms for attack classification [7]. The IoTID20 dataset has an imbalanced class distribution, but the authors did not consider data imbalances. Another drawback of this research is that the authors designed a classification model for binary classes, and each class of attack uses different feature sets which will complicate the IDS deployment process in the IoT architecture.

Peng et al. [9] suggested an information-based virtual MAC spoof detector using deep CNN (Convolutional Neural Network). The deep CNN model was compared with the SVM model for binary classification [9]. R. Aiyshwariya et al. [10] suggested a Deep LSTM-based attack detection method that has given 95% accuracy. The problem with the model is class imbalance. CNN is a DL structure that has attracted investigators' interest primarily for its outstanding capability to handle image data in image identification, classification, and computer vision. As a result, it has been cast in a range of fields, together with networking and diagnostic image processing [11]. The success of CNN in resolving several challenging classification problems served as the inspiration for the proposed signature-based IDS for efficient and prompt detection of IoT security vulnerabilities.

The proposed work employs image data generation and regularization techniques to minimize overfitting and offer a comprehensive paradigm that can suit properly on unfamiliar data. Five types of algorithms are used for the evaluation and analysis of results, namely CNN2D, L1 regularized CNN2D, L2 regularized CNN2D, CNN2D with dropout, and multi-regularized CNN2D.

The work contribution uses two separate datasets, UNSW-NB15 [12] and Bot-IoT [13], to thoroughly evaluate botnet vulnerabilities for various IoT devices and growing cyber-attacks in IoT. The proposed method incorporates the CNN2D algorithm with the L1, L2, and dropout regularization methods. Analysis of the proposed system using reliable evaluation metrics like Recall, Accuracy, F1-Score, and Precision. The proposed method resulted in a higher Detection Rate (DR). The following are the contributions made by this research work:

- The proposed use of a CNN2D-based deep learning scheme for identifying Botnet threats in an IoT scenario is an innovative approach. This choice leverages the proven success of CNNs in image-related tasks and extends their application to the complex domain of IoT security.
- Proposes a powerful CNN2D deep-learning scheme designed to reduce overfitting issues, providing adaptability to unknown attacks.
- Utilized five algorithms, including CNN2D, L1 regularized CNN2D, L2 regularized CNN2D, CNN2D with dropout, and multi-regularized CNN2D, ensuring a thorough analysis of results.
- Employed two distinct datasets, UNSW-NB15 and Bot-IoT, to comprehensively assess botnet vulnerabilities in diverse IoT devices amid growing cyber threats.
- Acknowledged and mitigated class distribution imbalances in datasets using SMOTE (Synthetic Minority Oversampling Technique), enhancing the reliability of the experimental results.
- Implemented image data generation from the datasets using OpenCV, contributing to enhanced model interpretability.
- Innovatively combined images from the datasets Bot-IoT and UNSW-NB15 to create a new dataset, further validating the proposed method's effectiveness.

In comparison to existing research efforts, this work not only addresses the limitations of previous IDS solutions but also introduces several key innovations. The acknowledgement and mitigation of class distribution imbalances in datasets, enhancement of model interpretability through image data generation, and utilization of multiple datasets to comprehensively assess botnet vulnerabilities are significant contributions. Through rigorous experimentation and analysis, the efficacy of the proposed technique in achieving higher detection rates and reducing false positives is demonstrated.

The rest of the article is structured as follows: Section 2 is brief about the proposed methodology and data preprocessing techniques. A comparison of results with previous research and results analysis is described in section 3. At last, section 4 concludes the paper.

2. Proposed Methodology

The proposed framework will be trained using image-based data that includes both benign and diverse types of threats. The structure of the proposed IDS technique is represented in Figure 1. To assess the functioning of the trained model, three cases are considered:

1. In the first instance, the evaluation of each model is done by using the image data produced from the UNSW-NB15 dataset, which contains 10 classes, including the normal type of data.
2. Bot-IoT dataset with 5 different classes including Normal data, is used for the evaluation of models in the second case.

3. In the last instance of the evaluation phase, a newly created composite dataset with 12 different classes has been used.

Result analysis of all the instances was done after evaluating the outcomes from different models like CNN2D, CNN2D with L1 regularization, CNN2D with L2 regularization, CNN2D with dropout, and CNN2D with multi regularization (Combination of L1, L2, and dropout).

For training and testing, two well-known datasets of network packets, the Bot-IoT and UNSW-NB15, are employed. More than 2.5 million network packets are used to replicate the dataset UNSW-NB15, which includes nine different attacks: exploitation, reconnaissance, denial-of-service, generic, shellcode, backdoors, fuzzers, worms, and analysis, along with normal packets.

The dataset is severely imbalanced because more than 87% of the packets are of the Normal type. With a mix of virtual and real-time IoT setups, the Bot-IoT dataset has more than 72 million records. Although there are four different sorts of attacks, DoS and DDoS packet types make up the majority of the dataset. Similar to the dataset UNSW-NB15, Bot-IoT is also unbalanced.

Table 1. Class distribution of the UNSW-NB15 dataset before and after the SMOTE algorithm

Class Type	Percentage of Data	
	Before SMOTE	After SMOTE
Normal	87.35	10.00
Backdoor	0.1	10.00
Analysis	0.11	10.00
Fuzzers	0.95	10.00
Shellcode	0.06	10.00
Reconnaissance	0.55	10.00
Exploits	1.75	10.00
DoS	0.64	10.00
Worms	0.01	10.00
Generic	8.48	10.00

It is essential to preprocess the information prior to applying it to the CNN2D model. The function "pd.to_numeric()" to transform category data into numeric data is used. The features 'id' and 'pkSeqID' are omitted in UNSW-NB15 and Bot-IoT datasets, respectively, because they are merely representational numbers from records; they have no bearing on network traffic or the nature of attacks. After omitting labels and IDs from both datasets, 42 features are employed for the attack detection. Every classification model must be given enough data, at least during the training phase, that includes a respectable count of each class, to function successfully.

On examining the data distribution of the two datasets in Tables 1 and 2, it is evident that the distribution of the data classes is not balanced before applying it to the SMOTE algorithm. The prediction performance for the minor classes is substantially impacted by an imbalanced dataset since the model is biased towards the category that has a greater record count in the training data. The classifier had a better chance of learning about the majority class throughout the process of learning, however, it was unable to acquire sufficiently about the minority class. As a result, the classifier has a tendency to classify a given data as belonging to the majority class even while it does not.

In order to increase the minority class records, the feature vectors for each member of the minor class are identified and plotted as dots in a 2D space in SMOTE [14]. Then, for each point, the nearest neighbor is determined, creating a new-fangled point that depends on the concerning line amid the starting point and its adjacent neighbor. To obtain balanced data in the proposed work, the SMOTE mechanism is employed. Tables 1 and 2 show the data sample distribution for the two datasets before and after applying the SMOTE algorithm.

Table 2. Class distribution of the BoT-IoT dataset before and after the SMOTE algorithm

Class Type	Percentage of Data	
	Before SMOTE	After SMOTE
Normal	0.013	20.00
DDoS	52.50	20.00
DoS	45.00	20.00
Reconnaissance	2.48	20.00
Theft	0.002	20.00

In order to make the traffic flow suitable for CNN's input, the traffic flow needs to be initially converted into an image. The characteristic distribution is transformed into a normal distribution using the quantile transformation normalizing technique. As a result, most of the variable values are within a few standard deviations of the median, effectively handling outliers [15]. Quantile transformation distributed each feature according to the same desired pattern depending on the following:

$$Y_{\mathrm{i}} = Q^{-1}(F(X_{\mathrm{i}})). \quad (1)$$

Where Q^(-1) is the quantile function of the intended output distribution, and F is the feature's cumulative distribution function Q.

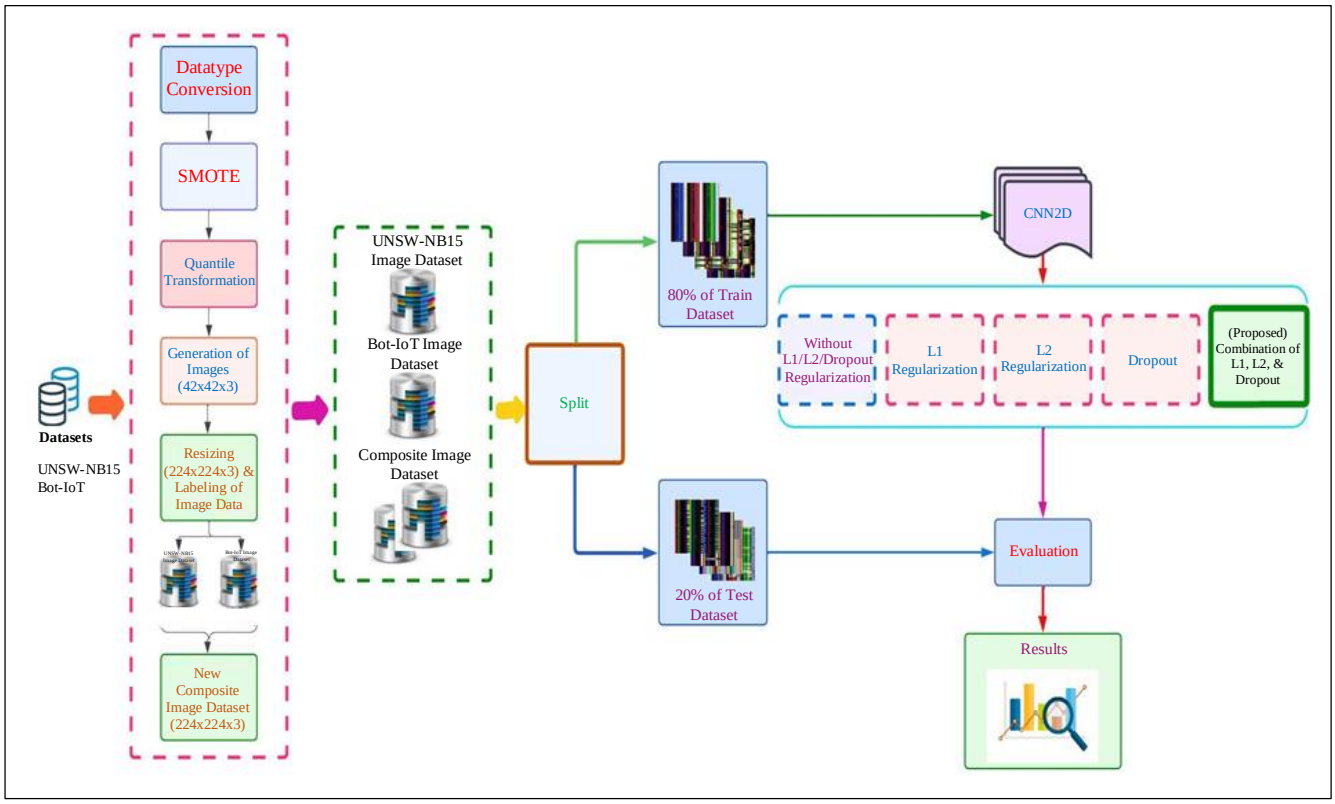


Fig. 1 Framework for proposed Intrusion Detection System

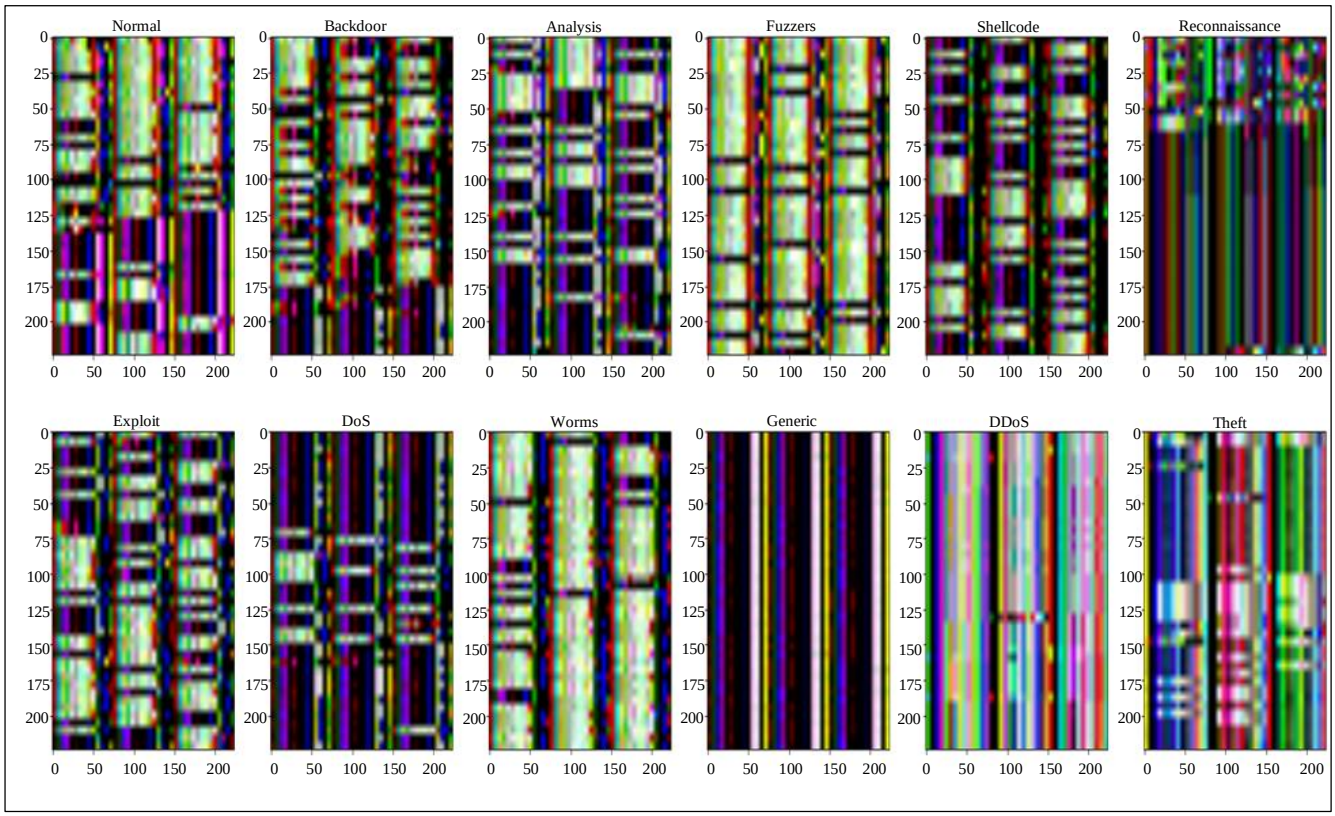


Fig. 2 Sample class images from the composite dataset

The data records are divided into parts of varying sizes throughout the image creation phase, dependent on the timestamp and attribute dimensions of the traffic flow datasets. Each one of the datasets contains 42 features, which are turned into a 3D image with three channels to reflect the image's color. As a result, each color image that is produced is converted to a maximum of 42 × 42 x 3 feature values.

Inferring from the previous sentence, the 42 records of every portion were transformed into image matrices of each channel, and then all of them were typically translated into the RGB channels. The OpenCV package is used to translate the information into the image matrix. Each portion of a dataset has 42 X 3 = 126 successive data samples. Repeat this process until all of the labels in both datasets have undergone the necessary transformation.

The images produced have relied on the timestamps of the dataset, therefore the outcomes are guaranteed to be accurate. It is worth noting that the time-series association of the initial traffic information can be preserved in this particular instance. Also, the ability to easily distinguish the various outlines in the images via color makes it crucial in this situation to transform the information into an image trajectory with three color channels.

The image is then resized into a shape that the CNN2D models can use as input. We, therefore, increased the images' size to 224 X 224 X 3. As a result, CNN2D will be able to quickly pick up on all of the image's features and increase learning speed. Figure 2 exhibits illustrative examples of the various attack samples from the composite datasets. It is evident from both datasets that the feature patterns of various classes differ significantly from one another. The variations in the visual patterns determine the occurrence and tactics of every attack plan used by the adversary. The CNN2D algorithm can acquire extra features in addition to these differences, resulting in greater accuracy for detecting attacks.

For the first two cases, the image dataset generated by datasets [UNSW-NB15 and Bot-IoT] has been used correspondingly. The combination of both image datasets is used to assess case 3. Normal, Backdoor, Reconnaissance, Analysis, Shellcode, Fuzzers, Exploits, DoS, Worm, and Generic are among the ten classes in the UNSW-NB-15 dataset.

In contrast, there are 5 categories for Bot-IoT data: DDoS, DoS, Reconnaissance, Normal, and Theft. The analysis of the two datasets was done, and discovered that the common kinds of classes were DoS, Reconnaissance, and Normal. Then, leaving the other files intact, the images from these three files were combined. In this way, a composite image dataset with 12 classes has been generated at this stage. It can be clinched that the intended method is successful at identifying vulnerabilities if it performs as well for a combined dataset as it does for other individual datasets.

As a result of their ability to retain the peculiar trends in the training sample rather than generalizing to new information, complex models like DL models are vulnerable to overfitting. Regularization is the term for any change that is performed to a learning model that aims to lower its generalization error but not its training error. By utilizing regularization methods to maintain the model simple enough, the network is able to generalize successfully to data points it has never encountered before (zero-day attacks).

Early stopping, dropout, L1 regularization, and L2 regularization are employed in the proposed work. The early-stopping technique is utilized to obstruct learning the model whenever its performance on the validating dataset worsens-when it experiences increased loss, declining accuracy, or worsening outcomes of the scoring metric. When errors from the training and validation datasets are shown simultaneously, it can be seen that both errors increase smaller over time till the model becomes overfit.

The training error continues to decrease after such a point while the validation error rises. The model will, therefore, be able to have minimum variance and higher generalization. The Dropout layer proceeds as a veil by eradicating some neurons' influences to the successive layer while upholding the integrity of the other neurons. The connections to the neurons' incoming and outgoing signals are likewise cut off when the neurons are turned off. It is always prudent to switch off the neurons up to 50%. There is a likelihood that the algorithm leaning and the estimates would be poor if further after 50% of neurons switched off. The main benefit of dropout is that it enables a single network to represent a variety of distinct sub-networks in a straightforward way for both training and testing [16]. In the research, a 20% dropout probability is used.

In CNN, one can define the cost function or loss function "F" as the squared error, where the error represents the variation between the y_{T_n} (actual value) and the y_{P_n} (predicted value). The cost function can be written as:

$$F = \frac{1}{N}\sum_{n=1}^{N}(y_{P_n} - y_{T_n})^2. \quad (2)$$

Ridge Regression is an approach that employs L2 regularization, whereas Lasso Regression utilizes L1 regularization. The loss function is modified by the L1 regularization, which is illustrated below, by adding the absolute value of the coefficient weights as a penalty term.

$$F_{L1} = \frac{1}{N}\sum_{n=1}^{N}(y_{P_n} - y_{T_n})^2 + \lambda\sum_{n=1}^{N}|W_n| \quad (3)$$

Where λ is the regularization parameter, the squared weights of the coefficients are added as a penalty term to the loss function for L2 regularization.

$$F_{L2} = \frac{1}{N}\sum_{n=1}^{N}(y_{P_n} - y_{T_n})^2 + \lambda\sum_{n=1}^{N}|W_n|^2 \quad (4)$$

If λ=0, the loss function will revert to the initial stage in both the L1 and L2 regularization scenarios. λ, conversely, will add too much weight and give rise to underfitting when it's tremendously huge.

After that, it is important to consider the procedure used to choose λ. This approach works well to avoid the over-fitting issue. Combining L1, L2, and dropout offers the best of all possible worlds, which is a useful advantage. L2 is typically more accurate than L1, and it is also simpler to modify. Dropout can streamline the network by eliminating the presence of particular neurons. However, L1 can function with sparse feature spaces and aids in the feature selection process. In order to reach the best accuracy, combine L2 and L1 with Dropout. To improve model efficiency and identify unforeseen attacks, early stopping together with dropout, L1 regularization, L2 regularization, and a combination of the aforementioned techniques have been employed.

The CNN2D model was fed with a 224 × 224 x 3 RGB input image. The proposed IDS has an input layer, seven convolution, three pooling, two dense layers, and a dropout layer. To ensure the smooth distribution consistent throughout the forward propagation and the backpropagation, Glorot uniform initialization is utilized. Relu activation is utilized for all the CNN2D convolutional layers, along with a (3, 3) kernel and the "same" padding parameter. 64 filters are applied at the first two convolution layers, and the output is (224, 224, 64).

The max pooling layer provides a way to create sample feature maps by adding up the positions of attributes in segments. The result from the initial max pooling layer (112, 112, 64). The third and fourth convolution layers used the same parameters as the first one, except the number of kernels is 128. So, the fourth convolution layer output will be (112, 112, 128). Then max-pooling layer with size (2, 2) is used to get the output of shape (56, 56, 128).

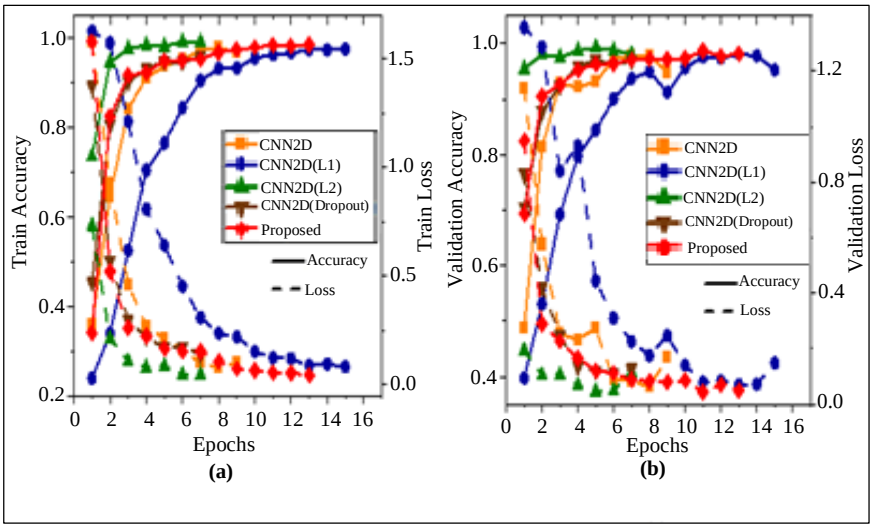


Fig. 3 Case 1 graph for accuracy-loss (a) Train, and (b) Validation.

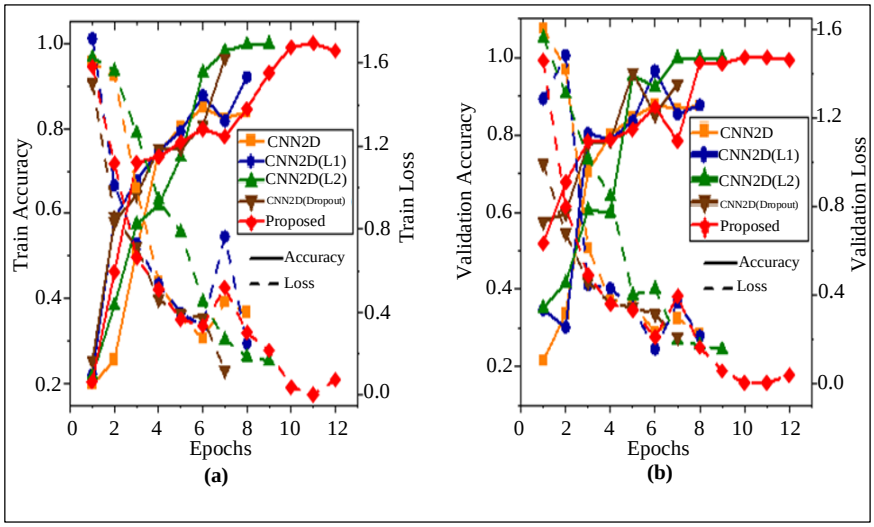


Fig. 4 Case 2 graph for accuracy-loss (a) Train, and (b) Validation.

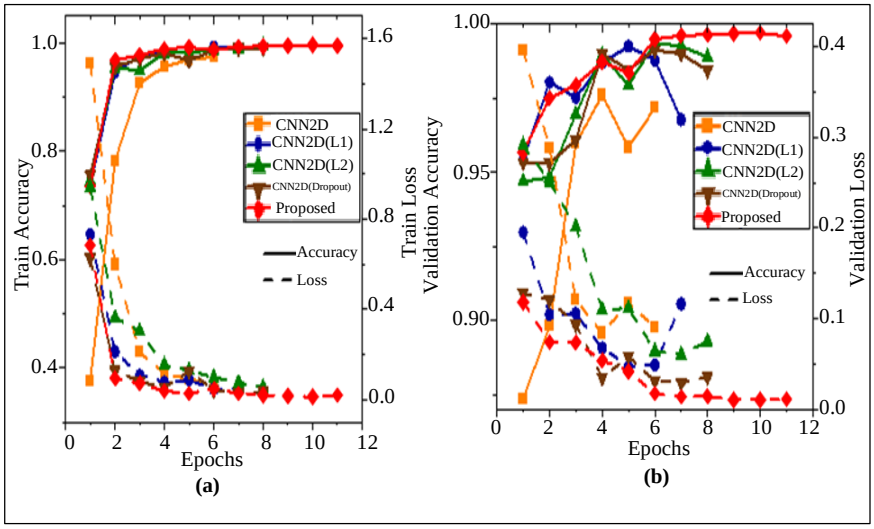
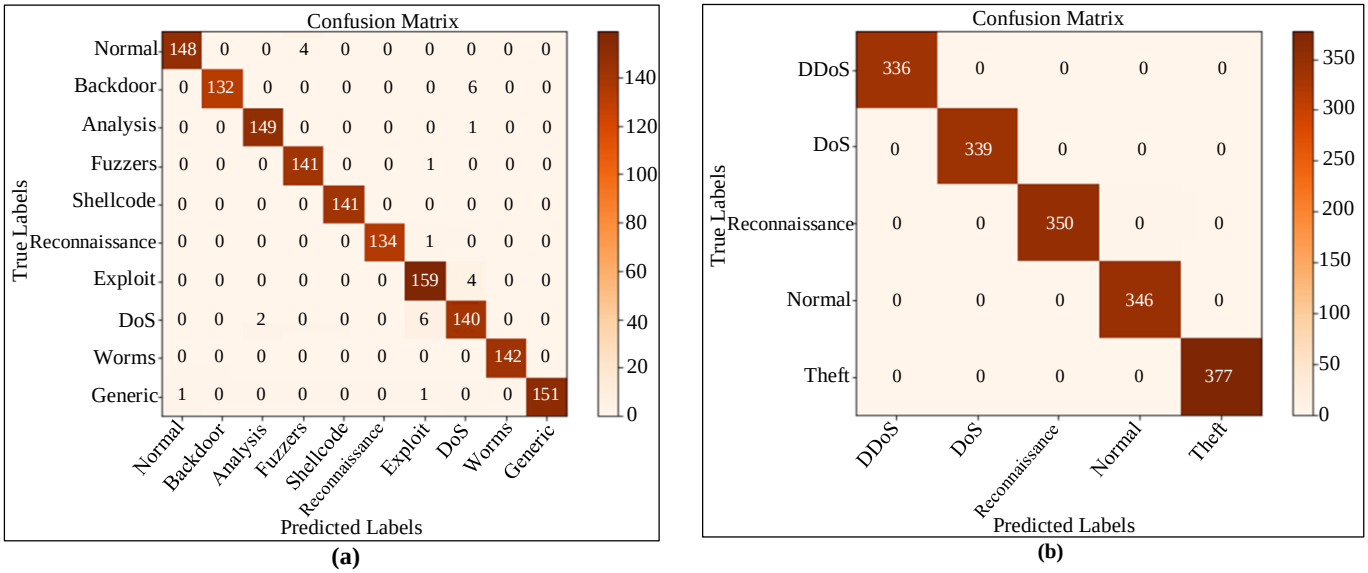


Fig. 5 Case 3 graph for accuracy-loss (a) Train, and (b) Validation.



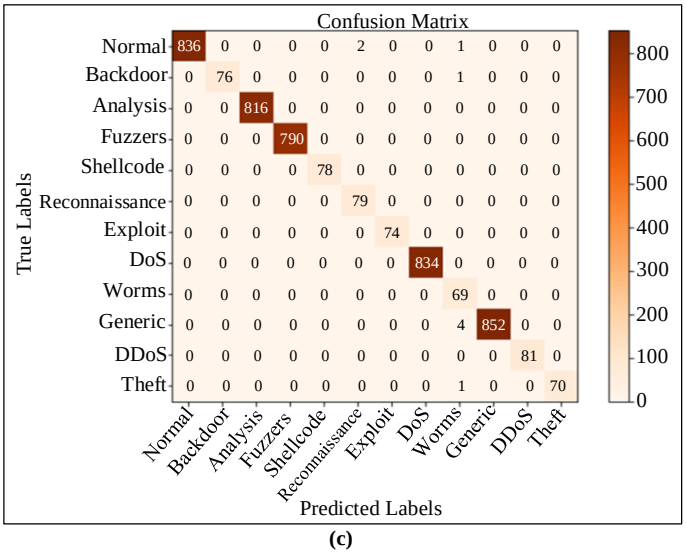


Fig. 6 Confusion matrix of proposed method of (a) Case 1, (b) Case 2, and (c) Case 3.

The remaining convolution layers have used 256 kernels of size (3,3) with (1,1) strides, which results in (56, 56, 256) output shape. The Global average pooling is used instead of the FC layer in the proposed algorithm. The benefit of global average pooling beyond FC layers is that it enforces correlations among feature maps and classes, enabling it to be more naturally adapted to the convolution framework.

As a result, the feature maps are simply classifications of confidence maps. A dense layer with both L1 (0.001) and L2 (0.001) bias regularizations is used after the global average pooling. A dropout regularization is inserted here to get much simpler and more effective results with a 0.2 dropout rate. Again, a dense-layer is used as the output layer of the proposed model with a SoftMax activation layer.

3. Performance Evaluation

In this section, the analysis of multi-regularized CNN2D IDS's performance is done by using the UNSW-NB15 and Bot-IoT datasets. For comparison, CNN2D with L1 regularization, CNN2D with L2 regularization, and CNN2D with dropout have been used. Each model must be evaluated for three separate datasets, including UNSW-NB15, Bot-IoT, and composite dataset, to carry out an efficient and precise experimental analysis.

3.1. Analysis Tools

In this research, the experiment was done by using a workstation with the following features: Windows 11 Pro 64-bit OS, Intel(R) Core (TM) i7-10700 CPU at 2.90GHz speed, 2904 MHz, 8 Cores, and 16 Logical Processors with 64.0 GB of RAM. The libraries utilized in the research work are shown in Table 3. Python version 3.9.12 from Anaconda version 23.1.0 is used. The Python interpreter, several libraries, and the Spyder IDE are all included in Anaconda.

Table 3. Python libraries used for research

Name of Library	Version
Pandas	1.4.2
SciKit-learn	1.0.2
Numpy	1.21.5
OpenCV	4.7.0
Tensorflow	2.10.0
Keras	2.10.0

3.2. Result and Analysis

The IDS connected to the defined cases was run to show and contrast each one's performance after the effective measurement metrics were defined. For all the models the early stopping technique is used to prevent the model from overfitting and to reduce the time for training the model. 20 epochs are used to evaluate each model. Due to the early stopping effect, all the models stopped training the model before reaching 20 epochs.

In the first case, the UNSW-NB15 is used for the evaluation. In this case, a total of 10 classes have been categorized for classification. Figure 3 shows loss and accuracy graphs for training and validation sets of the UNSW-NB15. From Figure 3, It can be noticed that the anticipated method stopped at the 13th epoch. From the 4th epoch, all models almost reached approximately stable accuracy levels except CNN2D with L1 regularization. But CNN2D with L1 regularization has trained up to 15 epochs which are highest in Case 1. Table 4 depicts the evaluation metrics of all Cases, which include Accuracy, Precision, Recall, and F1-Score. From the metrics, it can be observed that the multi-regularized CNN2D model achieved the best scores. At the same time, CNN2D with dropout has the lowest scores. The best accuracy of 98.16% is achieved by employing the proposed algorithm, whereas the lowest accuracy of 94.81% is achieved using CNN2D for UNSW-NB15. The confusion matrix of the proposed method for Case 1 is represented in Figure 6(a).

Bot-IoT dataset employed in Case 2 for evaluation. The loss and accuracy graphs of the Bot-IoT are represented in Figure 4. The proposed model stopped training at the 12th epoch because the training accuracy reached 100% accuracy at the 10th and 11th epochs, and it reduced to 99.37% at the 12th epoch. The validation accuracy is 100% at the 11th epoch and it reduced to 98.21% at the 12th epoch.

Upon considering Table 4, the proposed model has an effective performance (99.83% accuracy) compared to others. The CNN2D with L2 regularization also got good scores (99.37%) but was unable to reach the proposed method. Figure 6(b) depicts the confusion matrix of the proposed technique. From Figure 6(b), it can be observed that all classes are predicted absolutely except for the 'Reconnaissance' class. Out of 353 samples, 350 samples were predicted correctly as Reconnaissance, and the remaining 3 samples were predicted wrong as a 'Normal' class in this case.

The images generated from the aforementioned datasets are combined and utilized as composite image datasets in Case 3. The accuracy and loss scores of train and validation sets for the composite dataset are represented in Figure 5. Here, the proposed model is trained up to the 11th epoch. Finally, for the composite dataset, the proposed model outperformed the competition, which shows the effectiveness of the proposed technique. In Case 3, the proposed method reached an accuracy level of 99.68%. The confusion matrix of the proposed model of Case 3 is mounted in Figure 6(c). According to the confusion matrix, 8 out of 12 classes predicted exactly as true labels, and in the remaining 4 classes, slight misprediction occurred.

Table 4. Evaluation metrics of CNN2D with different regularizations

	Case-1				Case-2				Case-3			
	A	**P**	**R**	**F1**	**A**	**P**	**R**	**F1**	**A**	**P**	**R**	**F1**
CNN2D	94.81	95.37	94.81	94.72	87.44	92.01	87.44	86.09	96.77	97.8	96.77	96.94
CNN2D (L1)	95.22	95.64	95.22	95.2	87.61	92.31	87.61	86.18	97.17	95.73	97.17	96.38
CNN2D (L2)	96.99	97.16	96.99	97	99.37	99.38	99.37	9937	98.89	98.9	98.89	98.89
CNN2D (Dropout)	96.86	97.13	96.86	96.9	93.03	94.24	93.03	92.95	98.44	98.43	98.44	98.27
Proposed*	**98.16**	**98.2**	**98.16**	**98.17**	**99.83**	**99.83**	**99.83**	**99.83**	**99.68**	**99.7**	**99.69**	**99.68**

Table 5. Comparison of related work on the Bot-IoT and UNSW-NB15 datasets

Year	**Technique**	**Dataset**	**No. of Classes**	**Accuracy**	**Ref.**
2019	(Improved Conditional Variational AutoEncoder) ICVAE-DNN	UNSW-NB15	10	89.08	[17]
2019	AutoEncoder-Support Vector Machine- Artificial Bee Colony (AE-SVM-ABC)	UNSW-NB15	2	97.00	[18]
2020	RNN-LSTM	UNSW-NB15	2	87	[19]
2020	ANN-RFE (Artificial Neural Network - Recursive Feature Elimination)	UNSW-NB15	-	90.21	[20]
2019	Feed Forward Neural Networks (FNN)	Bot-IoT	5	95.1	[21]
2019	RNN using BPTT (Back Propagation Through Time)	Bot-IoT	11	98.20	[22]
2020	CNN	Bot-IoT	11	97.01	[23]
2021	LSTM	Combination of UNSW-NB15 & Bot-IoT	3	96.3	[24]
2022	Lightweight Deep Neural Network (LNN)	UNSW-NB15 Bot-IoT	10 5	86.11 96.15	[25]
This Work	**Multi-Regularized CNN2D**	**UNSW-NB15 Bot-IoT Composite Dataset**	**10 5 12**	**98.16 99.83 99.68**	-

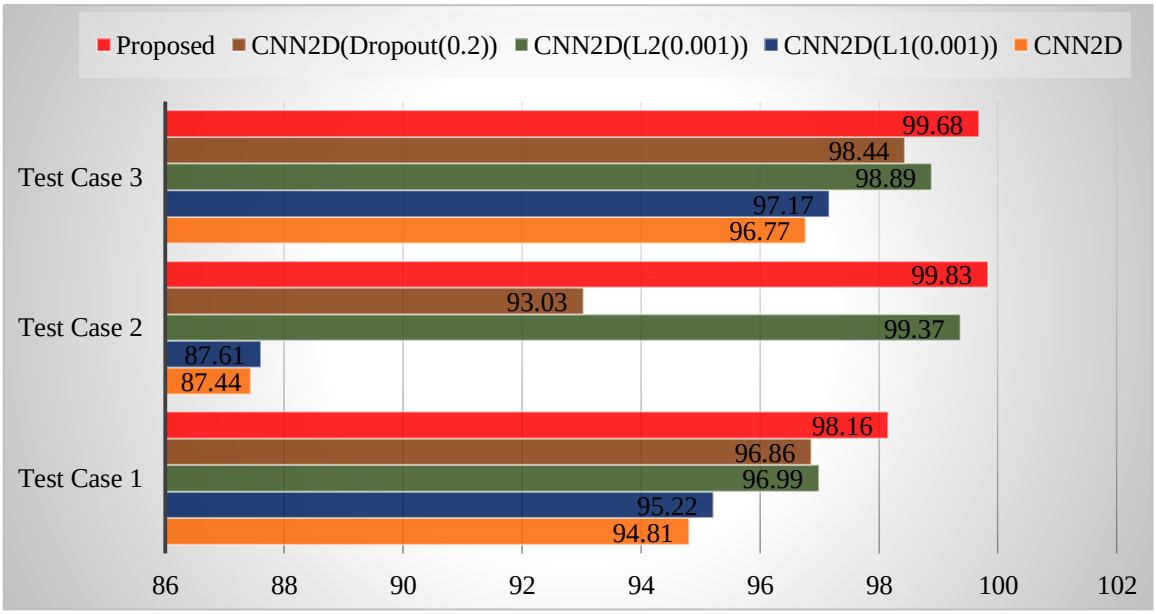


Fig. 7 Accuracy comparison of CNN2D models with different regularizations

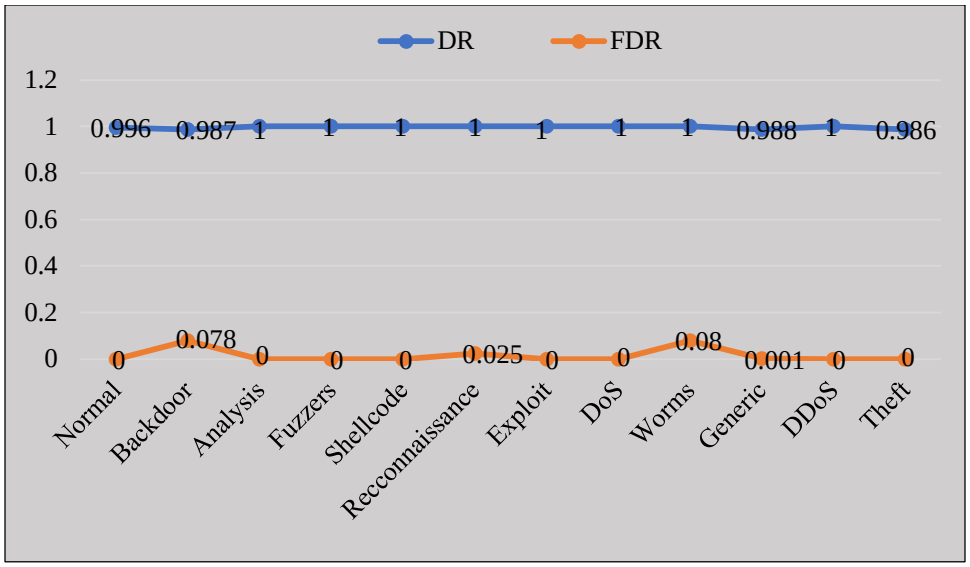


Fig. 8 Detection rate and false discovery rate of proposed model for individual classes

Whereas the lowest accuracies achieved with the CNN2D model are 94.81%, 87.44%, and 96.77% for Cases 1, 2, and 3, respectively. Figure 8 depicts the False discovery rate and detection rate of individual classes. From the graph (Figure 8), it is observed that the DR for every class is above 0.9, and FDR is below 0.1, which shows the success of the proposed model.

3.3. Comparison with Related Research

The proposed research findings were contrasted with those of other publications that employed the UNSW-NB15 and/or Bot-IoT datasets. Most research that has already been done has only used one of the two datasets for evaluation, namely Bot-IoT or UNSW-NB15. In this paper, image data for both tabular datasets produced a new composite image dataset and proposed an IDS architecture based on a multi-regularized CNN2D. Table 5 provides an overview of the contrast among the existing practices and the proposed one.

Yang et al. [17] implemented DNN in three datasets individually and used the ICAVE method to balance the dataset. One of the three datasets, UNSW-NB15, has a maximum accuracy score of 89.08% when employing ICAVE-DNN. The research proposal made by Q. Tian et al. [18] adopts the UNSWNB15 dataset as an object of research. An autoencoder from deep learning is utilized to decrease features during the data reprocessing stage.

The ABC algorithm is utilized to discover the optimal parameter, while SVM is employed as a decision engine. Using the AE-based SVM-ABC approach, an accuracy of 97% was attained. N. Guizani et al. [19] employed the RNN-LSTM technique and were successful in classifying binary data with an accuracy of 87%. R. A. Khamis et al. [20] used CNN, RNN, and ANN for evaluating the UNSW-NB15. For feature selection, RFE is employed in the dataset. The ANN with the RFE method outperformed here with an accuracy of 90.21%.

Ibitoye et al. [21] assessment of the Bot-IoT with 10 leading features using FNN and Self-normalized Neural Network (SNN) yielded an accuracy of 95% with FNN, which is superior to 91% with SNN. Ferrag et al. [22] presented a method called DeepCoin by using RNN with BPTT and yielded an accuracy of 98.2% for the evaluation of Bot-IoT.

Ferrag et al. [23] used Random Forest, Naïve Bayes, SVM, ANN, and CNN for the evaluation of the Bot-IoT dataset and scored the highest accuracy of 97.01% with the CNN algorithm. Bot-IoT and UNSW-NB15 datasets were taken into consideration for the study by Zeeshan et al. [24]. The authors analyzed the characteristics of the two datasets and found comparable characteristics. A new dataset was created using the common features that were found. With the new dataset, the LSTM approach was used to classify non-anomalous, DoS, and DDoS traffic with an accuracy of 96.3%.

Zhao et al. [25] developed LNN, SNN, and CNN for the multi-class classification of both datasets separately. The LNN technique yielded the highest accuracy compared to others, with 86.11% for UNSW-NB15 and 96.15% for Bot-IoT. The proposed multi-regularized CNN2D-based IDS used the composite image dataset generated from both the aforementioned datasets and performed effectively compared to existing research.

4. Conclusion

In the proposed work, a multi-regularized CNN2D is proposed in which two benchmark datasets, Bot-IoT and UNSW-NB15, are used. These two datasets are imbalanced class records and the issue of imbalance is solved by the SMOTE algorithm. Bot-IoT and UNSW-NB15 datasets are

used in Cases 1 and 2, respectively. For Case 3, A composite dataset generated from both datasets is used.

In each case, 5 models that include CNN2D, CNN2D with L1 regularization, CNN2D with L2 regularization, CNN2D with dropout method, and CNN2D with muti regularization have been used for evaluation. The multi-regularized CNN2D incorporated with L1, L2, and dropout regularization combined. Apart from these regularization techniques, all five models used an early stopping technique for minimizing model training time and increasing efficiency in detecting vulnerabilities. The proposed multi-regularized CNN2D model outperformed all other models in all the Cases.

The maximum accuracies yielded by the proposed algorithms are 98.16%, 99.83%, and 99.68% for Cases 1, 2, and 3, respectively. The effectiveness of this research can be seen in the manner in which the proposed approach was performed with both the benchmark datasets and the newly formed composite dataset.

References

[1] Phillip Williams et al., "A Survey on Security in Internet of Things with a Focus on the Impact of Emerging Technologies," *Internet of Things*, vol. 19, pp. 1-24, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[2] Lionel Sujay Vailshery, Number of Internet of Things (IoT) Connected Devices Worldwide from 2019 to 2021, with Forecasts from 2022 to 2030, 2022. [Online]. Available: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/

[3] Swathy Akshaya M., and Padmavathi G., "A Survey on Various Intrusion Detection System Tools and Methods in Cloud Computing," *2019 6th International Conference on Computing for Sustainable Global Development (INDIACom)*, New Delhi, India, pp. 439-445, 2019. [Google Scholar] [Publisher Link]

[4] Hongxin Hu, Gail-Joon Ahn, and Ketan Kulkarni, "Detecting and Resolving Firewall Policy Anomalies," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, pp. 318-331, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[5] Fraidoon Sattari et al., "A Hybrid Deep Learning Approach for Bottleneck Detection in IoT," *IEEE Access*, vol. 10, pp. 77039-77053, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[6] Chao Wang et al., "Hybrid Intrusion Detection System Based on Combination of Random Forest and Autoencoder," *Symmetry*, vol. 15, no. 3, pp. 1-16, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[7] Jingyi Su, Shan He, and Yan Wu, "Features Selection and Prediction for IoT Attacks," *High-Confidence Computing*, vol. 2, no. 2, pp. 1-6, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[8] Imtiaz Ullah, and Qusay H. Mahmoud, "A Scheme for Generating a Dataset for Anomalous Activity Detection in IoT Networks," *Advances in Artificial Intelligence*, pp. 508-520, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[9] Peng Jiang, Hongyi Wu, and Chunsheng Xin, "A Channel State Information Based Virtual MAC Spoofing Detector," *High-Confidence Computing*, vol. 2, no. 3, pp. 1-6, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[10] R. Aiyshwariya Devi, and A.R. Arunachalam, "Enhancement of IoT Device Security Using an Improved Elliptic Curve Cryptography Algorithm and Malware Detection Utilizing Deep LSTM," *High-Confidence Computing*, vol. 3, no. 2, pp. 1-14, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[11] Arthur A.M. Teodoro et al., "An Analysis of Image Features Extracted by CNNs to Design Classification Models for COVID-19 and Non-COVID-19," *Journal of Signal Processing Systems*, vol. 95, no. 2-3, pp. 101-113, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[12] Nour Moustafa, and Jill Slay, "UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems (UNSW-NB15 Network Data Set," *2015 Military Communications and Information Systems Conference (MilCIS)*, Canberra, Australia, pp. 1-6, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[13] Nickolaos Koroniotis et al., "Towards the Development of Realistic Botnet Dataset in the Internet of Things for Network Forensic Analytics: Bot-IoT Dataset," *Future Generation Computer Systems*, vol. 100, pp. 779-796, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[14] N.V. Chawla et al., "SMOTE: Synthetic Minority Over-Sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321-357, 2002. [CrossRef] [Google Scholar] [Publisher Link]

[15] Ogobuchi Daniel Okey et al., "Transfer Learning Approach to IDS on Cloud IoT Devices Using Optimized CNN," *IEEE Access*, vol. 11, pp. 1023-1038, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[16] Nickolaos Koroniotis, "*Designing an Effective Network Forensic Framework for the Investigation of Botnets in the Internet of Things*," Ph.D. Dissertation and Thesis, University of New South Wales, Australia, pp. 1-245, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[17] Yanqing Yang et al., "Improving the Classification Effectiveness of Intrusion Detection by Using Improved Conditional Variational AutoEncoder and Deep Neural Network," *Sensors*, vol. 19, no. 11, pp. 1-20, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[18] Qiao Tian, Jingmei Li, and Haibo Liu, "A Method for Guaranteeing Wireless Communication Based on a Combination of Deep and Shallow Learning," *IEEE Access*, vol. 7, pp. 38688-38695, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[19] Nadra Guizani, and Arif Ghafoor, "A Network Function Virtualization System for Detecting Malware in Large IoT Based Networks," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 6, pp. 1218-1228, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[20] Rana Abou Khamis, "*Evaluating Adversarial Learning on Different Types of Deep Learning-Based Intrusion Detection Systems Using Min-Max Optimization*," Master's Thesis, Carleton University, Canada, pp. 1-139, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[21] Olakunle Ibitoye, Omair Shafiq, and Ashraf Matrawy, "Analyzing Adversarial Attacks against Deep Learning for Intrusion Detection in IoT Networks," *2019 IEEE Global Communications Conference (GLOBECOM)*, Waikoloa, USA, pp. 1-6, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[22] Mohamed Amine Ferrag, and Leandros Maglaras, "DeepCoin: A Novel Deep Learning and Blockchain-Based Energy Exchange Framework for Smart Grids," *IEEE Transactions on Engineering Management*, vol. 67, no. 4, pp. 1285-1297, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[23] Mohamed Amine Ferrag et al., "Deep Learning for Cyber Security Intrusion Detection: Approaches, Datasets, and Comparative Study," *Journal of Information Security and Applications*, vol. 50, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[24] Muhammad Zeeshan et al., "Protocol-Based Deep Intrusion Detection for DoS and DDoS Attacks Using UNSW-NB15 and Bot-IoT Data-Sets," *IEEE Access*, vol. 10, pp. 2269-2283, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[25] Ruijie Zhao et al., "A Novel Intrusion Detection Method Based on Lightweight Neural Network for Internet of Things," *IEEE Internet of Things Journal*, vol. 9, no. 12, pp. 9960-9972, 2022. [CrossRef] [Google Scholar] [Publisher Link]