IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of

Digital

Design and Development of a Deep Learning-Based Model for Anomaly Detection in IoT Networks

Imtiaz Ullah, (Member, IEEE), Qusay H. Mahmoud, (Senior Member, IEEE)
Department of Electrical, Computer and Software Engineering, Ontario Tech University, Oshawa, ON, L1G 0C5 Canada

Corresponding author: Imtiaz Ullah (e-mail: imtiaz.ullah@ontariotechu.net)

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

ABSTRACT The growing development of IoT (Internet of Things) devices creates a large attack surface for cybercriminals to conduct potentially more destructive cyberattacks; as a result, the security industry has seen an exponential increase in cyber-attacks. Many of these attacks have effectively accomplished their malicious goals because intruders conduct cyber-attacks using novel and innovative techniques. An anomaly-based IDS (Intrusion Detection System) uses machine learning techniques to detect and classify attacks in IoT networks. In the presence of unpredictable network technologies and various intrusion methods, traditional machine learning techniques appear inefficient. In many research areas, deep learning methods have shown their ability to identify anomalies accurately. Convolutional neural networks are an excellent alternative for anomaly detection and classification due to their ability to automatically categorize main characteristics in input data and their effectiveness in performing faster computations. In this paper, we design and develop a novel anomaly-based intrusion detection model for IoT networks. First, a convolutional neural network model is used to create a multiclass classification model. The proposed model is then implemented using convolutional neural networks in 1D, 2D, and 3D. The proposed convolutional neural network model is validated using the BoT-IoT, IoT Network Intrusion, MQTT-IoT-IDS2020, and IoT-23 intrusion detection datasets. Transfer learning is used to implement binary and multiclass classification using a convolutional neural network multiclass pre-trained model. Our proposed binary and multiclass classification models have achieved high accuracy, precision, recall, and F1 score compared to existing deep learning implementations.

INDEX TERMS Internet of Things, Anomaly detection, IoT intrusion detection, machine learning, deep learning, transfer learning, network security, convolutional neural network.

I. INTRODUCTION

Cybersecurity is a crucial part of the information management framework of today's IoT environment. The following factors contributed to the widespread exposure of IoT vulnerabilities to cyber-attacks: the large-scale distribution of IoT devices from every household to every home, smart power grids, and smart cars, as well as the complexity of the communication protocols used by IoT users, will create significant security threats. Although the IoT increases efficiency and productivity through smart and remote control, it also increases cyberattack risk. The IoT information protection architecture is essential in today's technological innovations. The number of IoT devices in use has risen significantly from 16 billion in 2015 to over 30 billion in 2020, increasing since homes and companies are steadily relying on web technology. By 2024, the IoT is projected to reach 83 billion devices [1]. The increased variety of IoT systems being produced demonstrates that the IoT manufacturing industry is progressing toward revolutionizing IoT architecture. Therefore, the requirements that govern IoT devices connectivity are complex, needing a shared forum to promote communication between devices. Industry and manufacturing use 40.2 % of IoT devices; 30.3 % of IoT equipment is used in the medical sector; retail uses 8.3 %, security uses 7.7% of IoT equipment; and transport uses 4.1 %, IoT equipment [1].

The growing variety of IoT devices developed for various applications ensures that the IoT manufacturer is increasingly evolving IoT technologies and reducing the time to sell their produce. End users have benefited from IoT

devices, and critical facilities have also used them successfully in carrying out their daily tasks. Besides taking significant measures to have improved protection features, the IoT makes consumers potentially susceptible to cyber-attacks on their personal information. A large number of critical vulnerabilities on IoT networks are also a threat. Common cyber-attacks involve DDoS (Distributed Denial of Service), ransomware, and botnet attacks, which seek to exploit IoT networks and destroy their computational capabilities. The volume of data produced by these devices increases exponentially and can contain confidential information. The IoT-generated data is expected to reach 73.1 ZB by 2025 [2].

Although IoT applications are favored over traditional devices and frameworks, such implementations remain susceptible to a range of attack approaches that take advantage of both well-known and novel attack routes. Since IoT and web-based framework infiltration became more accessible in recent years, attacks such as DoS (Denial of Service), DDoS, and other remote hacking techniques are more commonly used to breach their confidentiality. The attacker intended to overwhelm the target IoT networks with malicious behavior. Hackers often exploit unpatched, non-patchable, or unencrypted IoT networks to access valuable data held on insecure IoT devices. While protection systems are better now than in the past, some people always try to deceive devices by breaking into smart locks and garage doors [3].

IoT systems have benefited people in several ways; however, there are several weaknesses in IoT systems. Security is the most challenging aspect of IoT networks. It is hard to prevent IoT threats since there are no agreed-upon guidelines for developing IoT devices. Different communication protocols can introduce additional complexity when implementing an IoT framework [4]. The challenges associated with the wide variety of IoT protocols complicate delivering a reliable and uniform cybersecurity approach for IoT networks. Adversaries can attack and compromise the IoT networks due to many vulnerabilities available in the IoT protocols. Device disruption, data theft, interruption, and MiTM (Man-in-The-Middle) are all threats that can be applied to any of these scenarios [5]. The rise of malicious threats on critical infrastructure has required proactive protection technology to enhance the protection of critical systems. The IDS has gained popularity as reactive network security. Network intrusion detection aims to evaluate different network data through various behavioral analyses of the network to ensure its security is maintained. Commercial security products are typically mainly focused on thresholds, signatures, heuristics-driven approaches, or statistics. These techniques work well for known threats but fail when attempting to identify new or unknown threats. In addition, these approaches often necessarily require domain training and knowledge and continuing updates. The inability to identify new developing cyber threats and the need for manual signature database updates restrict the effectiveness of signature-based detection systems.

The increased use of the Internet has prompted IoT protection firms to build more sophisticated technologies and standard security approaches to protect IoT devices from intruders. There are a wide variety of methods that are available for network anomaly detection. Machine learning has been both necessary and effective in the timely identification of cyber-attacks. While several anomaly-detection methods are used, fewer attempts were made to implement CNN (Convolutional Neural Network) for anomaly detection. Malicious actions in IoT networks must be detected and stopped immediately; therefore, the function of IDS has become critical in securing IoT networks by identifying anomalies. Our proposed IDS use a deep learning-based model for multiclass and binary class traffic classification. The proposed system uses a convolutional neural network architecture in the multiclass classifier to categorize 15 types of attacks and effectively distinguishing them from regular network traffic. A model also built using a CNN and focuses on transfer learning for binary and multiclass class traffic classification. This article makes the following significant contributions:

- A novel anomaly detection model for IoT networks based on a convolutional neural network.
- A transfer learning model for binary and multiclass classification based on a convolutional neural network.
- A strategy for creating a new dataset from a current one to detect anomalous behavior in IoT networks. The processing and generation of features focused on the flow of raw network traffic. We created four datasets using this strategy and then combined them with increasing the number of attack categories. The proposed datasets may be accessed [6].
- Performance improvements of binary and multiclass classification models.

The rest of the paper proceeds as follows: Section II discussed the related work. The proposed model is presented in Section III. In Section IV, data collection strategies and datasets used are discussed. The analysis of the evaluation results are presented in Section V, with discussion and comparison results in Section VI. Finally, Section VII concludes the paper and offers ideas for future work.

II. RELATED WORK

The massive rise in data transmission via various IoT devices and communication protocols has increased security concerns, emphasizing the need for effective IDS. Deep learning is a form of artificial intelligence that uses several neurons to model the learning process. Researchers have focused their attention on more comprehensive artificial intelligence methods of anomaly detection, and that is why deep learning has gained more importance in the industry. The review of deep learning approaches carried out by Aldweesh et al. [7] provides a comprehensive evaluation of intrusions. In the past, several researchers used KDD99 or NSL-KDD datasets to identify malicious activities; the survey paper main findings emphasized the need for a

modern and legitimate dataset to get accurate output results. Kaur et al. [8] use a CNN model to identify and describe several attacks. They analyzed their model via CICIDS2017 and CICIDS2018 datasets. Their model provides multiclass attack classification, but the detection rate for many attacks was not satisfactory.

Ferrag et al. [9] conduct a deep learning survey for data security intrusion detection. The authors compare seven deep learning models using 35 well-known datasets and classify them into seven separate categories. They conducted binary and multiclass classification and checked their strategies on BoT-IoT and CIC-IDS2018 datasets. The authors investigated several attack methods to evaluate the effectiveness across different deep learning models. These models were evaluated using their false alarm rates, accuracy, and detection rates. The CNN model is more effective than the FFN (FeedForward Neural Network) and RNN (Recurrent Neural Network) models. The convolutional neural networks have proven successful in several applications, including target tracking, image processing, and surveillance systems. A convolutional neural network extracts features from labeled files to perform classification. Due to the computational and memory requirements of these multilabel convolutional neural network models, deployment on edge devices is complicated. Odetola et al. [10] develop a multilabel classification method using a convolution neural network on edge IoT devices. The framework uses a single convolutional neural network with many predefined layers and configurable loss functions. Their model achieved less latency and MACC (multiply and accumulate) operations. They suggest a multilabel identification technique that enhances the capabilities of a model prepared for traditional classification to perform multilabel classification. Their approach is perfect for extracting different features from a single image. Their technique enables multilabel classification at a low cost and with a substantial degree of precision.

Ge et al. [11] propose a novel scheme for connected IoT networks based on deep learning principles. They used the FFN model for binary and multiclass classification. Their model produced accurate binary classification results but failed to produce precise multiclass classification results. Pecori et al. [12] developed IoT benign and malicious network traces by combining separate datasets. The integrated dataset consists of four attack categories and a normal category. They used seven hidden layers and 50 epochs to achieve the best performance; however, accuracy, recall, and F1 score are not satisfactory for binary and multiclass classification. Their model is inapplicable due to the complex structure of neural networks and the inadequate detection rate.

Idrissi et al. [13] conducted a study to identify IoT vulnerabilities and security threats. Their paper uses real-world threats and vulnerabilities to identify several types of IoT threats and vulnerabilities. They also discuss recent research in IoT security, focusing on intrusion detection techniques based on neural network strategies. Tian et al. [14] proposed a distributed approach for identifying network threats through URLs using deep learning algorithms. Their system was designed to protect multiple web applications in the EoT (Edge of Things) distributed environment. Their framework can be practically effective because of its automated function collection, ease of upgrading, and reliability in defending against attacks on distributed deep models. Hassan et al. [15] suggest a hybrid deep learning algorithm that uses a CNN and a WDLSTM (Weight-Dropped Long Short-Term Memory) model to identify intrusions in a large data context. CNN is used to discover the best features, and the WDLSTM technique is used to help a neural network resist overfitting. Using the UNSW-NB15 dataset, the model had a binary classification accuracy of 97.1% and a multiclass classification accuracy of 98.4%. The entire computational environment has evolved as a result of significant advances in information and communication technology.

Swarna et al. [16] suggested a DNN (Deep Neural Network) to recognize and forecast unexpected cyberattacks in IoMT (Internet of Medical Things) networks to establish reliable and productive IDS. The proposed DNN framework achieved improved accuracy and a 32% reduction in computation time, allowing quicker detection to prevent post-intrusion effects in critical cloud computing. The development of networks has always been associated with advancing information technology, but now, the Internet economy is growing rapidly with IoT. Li et al. [17] propose a deep migration learning model architecture for IoT feature selection and anomaly detection in smart cities that incorporate deep learning and intrusion detection technologies. Their paper provides a migration learning model analysis scheme as well as system feature selection. The proposed algorithm has a fast detection time, but the detection rate is insufficient for some attacks.

Governments worldwide are encouraging smart city applications to increase the standard of everyday living in metropolitan environments. The growing rise of Internet-enabled devices is triggering an increase in botnet attacks centered on the IoT. Sriram et al. [18] propose a network traffic flow-based deep learning botnet identification technique. In certain situations, the efficiency of IoT applications relies on the consistency and credibility of the information. Yin et al. [19] developed an integrated deep learning model for anomaly detection in IoT networks. They used LSTM autoencoder and CNN to identify the anomalies. They use a two-stage window-based data preprocessing to achieve improved learning predictions. Their suggested approach was restricted to binary classification and achieved better accuracy, precision, recall, and F1-score.

Privacy and confidentiality are seen as critical concerns when it comes to the IoT. Intruders may carry out various attacks, resulting in issues with the privacy and security of IoT devices. Manimurugan et al. [20] suggested a DBN (Deep Belief Network) model for anomaly identification in smart medical environments. Their model provided better outcomes for the normal class, but the anomaly class

outcome was not satisfactory. Wang et al. [21] build a deep hierarchical network for packet-level analysis of malicious traffic capable of understanding traffic characteristics from raw packet data. They extracted spatial features from the raw packet using a CNN and temporal features using GRU (Gated Recurrent Units). Their model had a high detection rate for specific attack categories but a low detection rate for others. Table I presents a summary of the intrusion detection models using deep learning techniques. The table shows that many research papers only focused on accuracy performance measures and binary classification to evaluate the deep learning intrusion detection model. The detection rate for some articles is not satisfactory. Several academic papers evaluated intrusion detection techniques using an old KDD intrusion detection dataset. Many latest cyber-attacks are not considered in the KDD99 dataset, and the KDD99 dataset was not developed considering the IoT network. In Table I, DR represents detection rate, Acc means accuracy.

TABLE I.
OVERVIEW OF RELATED WORK

Article	Year	Model	Dataset	Classification	Performance
[22]	2019	Autoencoder	KDD99	Binary	Acc= 84-100
[23]	2019	CNN	KDD99	----------	Acc=97.34
[24]	2020	DNN	KDD99	----------	Acc=92.70
[25]	2021	SRDLM	KDD99	Multiclass	Acc=94.03
[26]	2018	DNN	NSL-KDD	Binary	Acc =99.29
[27]	2018	CNN	NSL-KDD	----------	Acc=82.83
[28]	2018	CNN	NSL-KDD	Multiclass	Acc=85.07
[29]	2019	GA Optimized DBN	NSL-KDD	Multiclass	Acc= 99.45
[30]	2019	SMO	NSL-KDD	Binary	Acc =99.02
[31]	2019	MLP	NSL-KDD	Binary	Acc =79.70
[32]	2019	DNN	NSL-KDD	Binary	Acc=98.00
[33]	2020	CNN	NSL-KDD	Multiclass	Acc = 86.95
[34]	2020	RNN	NSL-KDD	Multiclass	Acc=92.18
[35]	2021	DNN	NSL-KDD	Multiclass	Acc= 83.33
[36]	2017	D-FES	AWID	Binary	Acc=99.90
[37]	2018	LSTM	AWID	Binary	Acc =98.22
[38]	2019	Stacked Autoencoder	AWID	Multiclass	Acc=99.90
[39]	2019	Autoencoder	AWID	Binary	Acc=98.00
[40]	2019	LSTM	ISCX2012	Binary	Acc =99.99
[41]	2018	Autoencoder	UNSW-NB15	Multiclass	DR=68.91
[42]	2018	Bidirectional LSTM	UNSW-NB15	Binary	Acc=95.71
[15]	2019	CNN	UNSW-NB15	Multiclass	Acc =98.43
[43]	2019	Autoencoder, SVM	UNSW-NB15	Binary	Acc=97.00
[44]	2019	MLP, CNN, DNN	UNSW-NB15	Binary	Acc=99.24
[45]	2020	CNN	UNSW-NB15	----------	Acc=89.02
[46]	2020	CNN	UNSW-NB15	----------	Acc=96
[47]	2019	Stacked Autoencoders	Personal Dataset	----------	Acc=95.00
[48]	2019	LSTM, GRU	Personal Dataset	Multiclass	Acc =96.08
[49]	2019	Feed Forward DNN	Personal Dataset	Binary	Recall =97
[50]	2019	CNN	Personal Dataset	----------	Acc=98.80
[51]	2019	MLP	Personal Dataset	----------	Acc=87.10
[52]	2019	GRU	Personal Dataset	Multiclass	Acc=95.60
[53]	2019	GRU	Personal Dataset	Multiclass	F1 score =80.30
[54]	2020	CNN	----------	Multiclass	Acc=92.53
[55]	2019	CNN	VAST 2013	----------	Acc=95.86
[56]	2020	C-CMU	P1 & DMD2018	----------	Acc=99.20
[16]	2020	DNN	Kaggle	Multiclass	Acc=99.90
[21]	2020	CNN, GRU	Multiple	Binary	Acc=99.42
[57]	2020	Fast GRNN	MedBIoT	Multiclass	F1 score =99.99
[58]	2020	CNN	SWaT	Multiclass	Acc=98.02
[19]	2020	C-LSTM_AE	Webscopre S5	----------	Acc=99.62
[59]	2021	CNN	Gas pipeline	Multiclass	Acc=95.97
[11]	2019	FFN	BoT- IoT	Multiclass	Acc= 98.09
[60]	2019	RNN	BoT-IoT	Multiclass	Acc =98.20
[9]	2020	DNN, RNN, CNN	BoT-IoT	Multiclass	Acc =98.37
[61]	2020	FFN	BoT-IoT	Multiclass	Acc=99.79
[62]	2020	SNN	BoT-IoT	Multiclass	Acc=98.73

III. PROPOSED MODEL

A. MODEL DESIGN

The convolutional neural networks have recently produced excellent responses in voice recognition and image recognition. This paper proposes a model focused on convolutional neural networks due to its superior image analysis capability. The results produced by convolutional neural networks are attractive in these fields. Furthermore, by transforming intrusion detection issues into image recognition problems, convolutional neural network benefits may be more fully utilized. A convolutional neural network allocates weights and biases to different image elements and distinguishes one from the other.

In this paper, we design and develop 1D, 2D, 3D convolutional neural networks for anomaly detection in IoT networks. A general structure of the proposed model is presented in Fig. 1. The model consists of an input layer, four blocks of convolution layers, flatten layer, a fully connected layer, and an output layer. Each block consists of a convolutional layer, normalization layer, pooling layer, and dropout layer. The input layer receives inputs from the reshaping system. The reshaping system transforms network data into a format compatible with CNN1D, CNN2D, and CNN3D models. The blocks of convolution layers are the primary components of a convolutional neural network. The pooling layer was omitted from the CNN2D model fourth block and excluded from the CNN3D model third and fourth blocks. The reason for eliminating the pooling layer from these models is because additional downsampling of the input data is not possible. The convolution layer extract features from the input image and learns image attributes from tiny squares of input data, preserving the association between pixels. The layer normalization aims to normalize all the inputs to a neural network layer. The layer normalization layer standardizes the output of the convolution layer for the average pooling layer. The pooling layer lets to improve features by summarizing features in sub-maps with robust features. The average pooling layer determines the overall number of features in each patch by computing the total number of updates throughout the whole function map.

A neural network that uses convolution has a possibly overfitting issue and will have to undergo extensive fine-tuning of the test dataset parameters. A dropout layer reduces the chance of overfitting by ignoring some neurons during the training phase. When adjacent frames are interrelated strongly with feature maps, a regular dropdown does not regularize the activations. We used a spatial dropout layer that drops the entire feature maps instead of individual units. The tensor is reshaped to create a flat operation on a tensor with the number of elements in the tensor, excluding the batch size, equal to the number of elements in the tensor. The flattening layer is fully connected to a dense layer. A total of 512 neurons were used in the dense layer. The last layer of the model is the output layer, and the number of neurons in the output layer equals the number of classes. To further demonstrate the effectiveness of the CNN model in detecting an anomaly in IoT networks, we implement the same structure across the CNN1D, CNN2D, and CNN3D models. Six IoT intrusion detection datasets were used to train, validate, and test CNN1D, CNN2D, and CNN3D models.

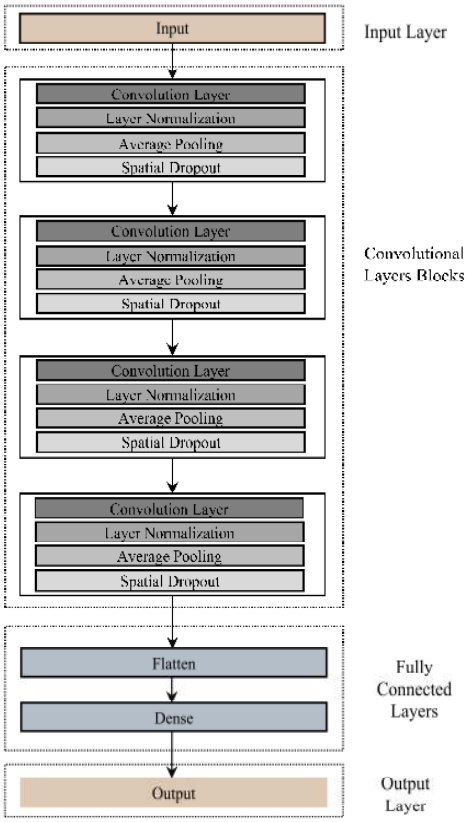


FIGURE 1. Layers View of Proposed Model

B. CONVOLUTION1D MODEL DESIGN

Convolution is calculated in 1D using time access and the kernel movies in one direction. The input and output data for CNN1D are two-dimensional, and it is most often used for time series data. First, an input vector 64×1 is generated to fit the 64 best features chosen by the feature selection algorithm. After the input layer, four convolution layer blocks were added to the model. The four-convolution layer blocks are considered a method of feature learning. The convolution layers extract features from the input image and find image properties from small data samples within the input, retaining the vector relationship. The proposed CNN1D model layer's view is presented in Fig. 2.

Convolution first layer use relu activation, 32 filters, kernel size 5, and the same padding parameter. The layer normalization adjusts the preceding layer activation separately for each given sample in a batch. The output of the first convolution layer and layer normalization is (64, 32). The average pooling layer offers a solution to downsample feature maps by summarizing features in a feature map segment. The average pooling layer output (32, 32). We used a spatial dropout layer to regularize the training

data model and reduce overfitting with a drop value of 0.05. Each of the four convolution layers uses identical parameters except the filters, which are doubled in each subsequent layer. The classification part consists of fully connected flatten and dense layers. The flatten layer is applied to the model transforming the tensor into a shape equivalent to tensor elements. There is no batch size parameter for the flatten layer. The flatten layer is connected to a fully connected dense layer, and the dense layer is connected to the output layer. The dense layer has 512 neurons, whereas the output layer has a class number dependent number of neurons.

C. CONVOLUTION2D MODEL DESIGN

The CNN2D is a three-dimensional neural network that is most often used to process image data. Kernel moves in two directions in a CNN2D model. First, an 8 × 8 input image is created to match the 64 features of the CNN2D model. The CNN2D model consists of an input layer, four convolution layer blocks, fully connected layers, and an output layer. The CNN2D first convolutional layer uses a relu activation, with a (5, 5) kernel, 32 filters, and same padding parameter. The proposed CNN2D model layer view is presented in Fig. 3. The output of the first convolution layer and normalization layer is (8, 8, 32). By summing the locations of features in segments, the average pooling layer offers a method for generating sample characteristic maps. The average pooling layer output (4, 4, 32). A spatial dropout layer is added at the end of the first convolution layer to minimize the model overfitting and regularize the training data output. A dropout value of 0.05 was used for the dropout layer. The same parameter values are used throughout all four convolution layers, with the exception of the filters, which are doubled in each successive layer. The average pooling layer was not used in the fourth convolution layer block because the input vector shapes were reduced to (1, 1, 128) in the third convolution layer block. To construct a feature vector, we flatten the output of the convolutional layer. The flatten layer is linked to the fully connected dense layer, and the dense layer is attached to the output layer. The number of neurons in the output layer is calculated by the number of classes in the dataset.

D. CONVOLUTION3D MODEL DESIGN

The CNN3D is a four-dimensional neural network that is commonly used to process three-dimensional image data. The kernel moves in three directions in CNN3D. First, an input image with the dimensions 4 × 4 × 4 is generated to match the 64 features of the CNN3D model. The CNN3D layer view of the proposed model is shown in Fig. 4. In addition to the input layer, the CNN3D model has four convolutions layers blocks, flatten layer, a dense layer, and an output layer. Relu activation, 32 filters, kernel size (5, 5, 5), and the same padding were used in the CNN3D first layer. Layer normalization and average pooling layer are used next to the convolution layer. The output of the average pooling layer (2, 2, 2, 32). A spatial dropout layer is created at the end of the first convolution layer block to prevent overfitting the model and regularize the performance. The average pooling layer is used only in the first two convolution layer blocks because, in the second convolution layer, the pooling layer reduces the vector dimension to (1, 1, 1, 64). The layer normalization and spital dropout were used along with all convolution layer blocks. The flatten layer was added to reshape the number of elements of the tensor. The flatten layer is connected to the dense layer, and the dense layer connects to the output layer. The dense layer has 512 neurons, while the output layer contains an undetermined number of neurons based on the class number.

E. TRANSFER LEARNING

Transfer learning is a kind of machine learning technique in which a model produced for one activity is utilized as the starting point for a model on a different task. Transfer learning principle is used to deploy a pre-train multiclass CNN model for the multiclass and binary models. In the first phase, we used IoT-DS-2 pre-trained multiclass classification CNN1D, CNN2D, CNN3D models for the binary classification of the IoT-DS-2 dataset via the transfer learning principle. In the next phase, we used the same pre-trained learning model for multiclass classification of BoT-IoT, IoT Network Intrusion, MQTT-IoT-IDS2020, IoT-23, and IoT-DS-1 datasets. Because the BoT-IoT, IoT Network Intrusion, MQTT-IoT-IDS2020, IoT-23, and IoT-DS-1 datasets are subsets of the IoT-DS-2 dataset, the transfer learning concept was used in the multiclass classification model for these datasets. Transfer learning from a multiclass CNN model to a binary class CNN model is effective because the binary CNN model is trained using a subset of data used by the multiclass classification model. Using the multiclass CNN model for binary CNN model, the input, convolution layers, and fully connected layers are adopted from the already IoT-DS-2 dataset trained model.

The output layer was removed from the pre-trained multiclass CNN model. A new output layer was added to the model with two neurons and a softmax activation function for binary classification. The new model is trained using a binary class dataset. All current model layers were disabled during training except the dense and output layers used in the binary dataset training process. A binary classification model based on transfer learning is presented in Fig. 5. The binary classification model uses a pre-train multiclass classification model. The input layer has the same number of features. The convolution layers, normalization layers, pooling layers, dropout layers, and flatten layer were frozen while the binary classification model was being trained. During the training phase, only the dense and output layers were allowed to learn. The binary classification model uses an IoT-DS-2 dataset pre-trained model consisting of all attack classes from BoT-IoT, IoT Network Intrusion, MQTT-IoT-IDS2020, IoT-23, and IoT-DS-1 datasets. The binary CNN model has a relu activation function and 256 neurons at the dense layer. The output layer uses two neurons and a softmax

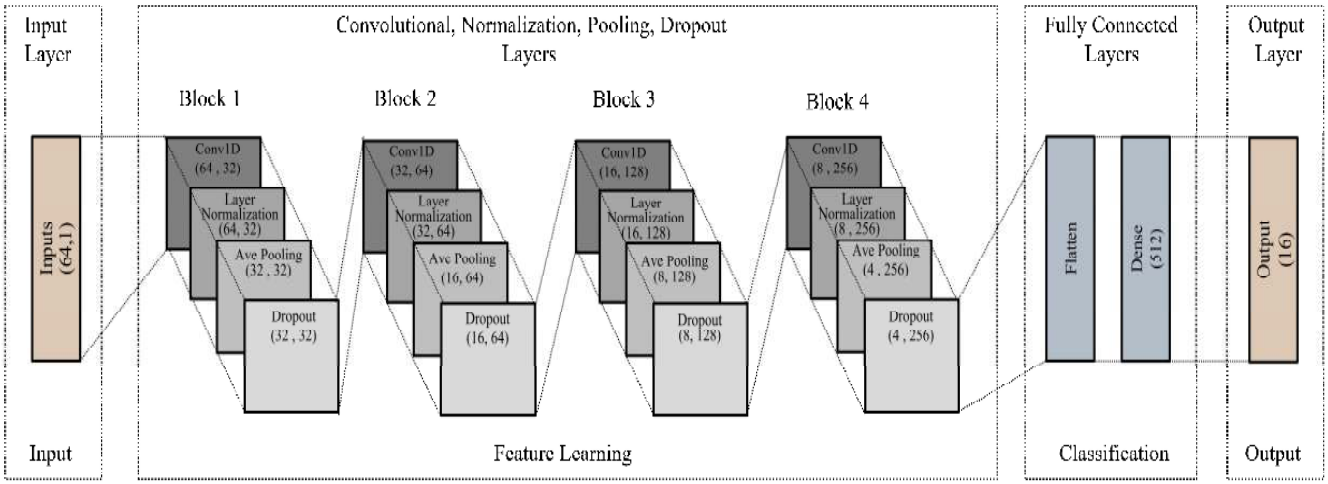


FIGURE 2. Convolution1D Layers View of Proposed Model

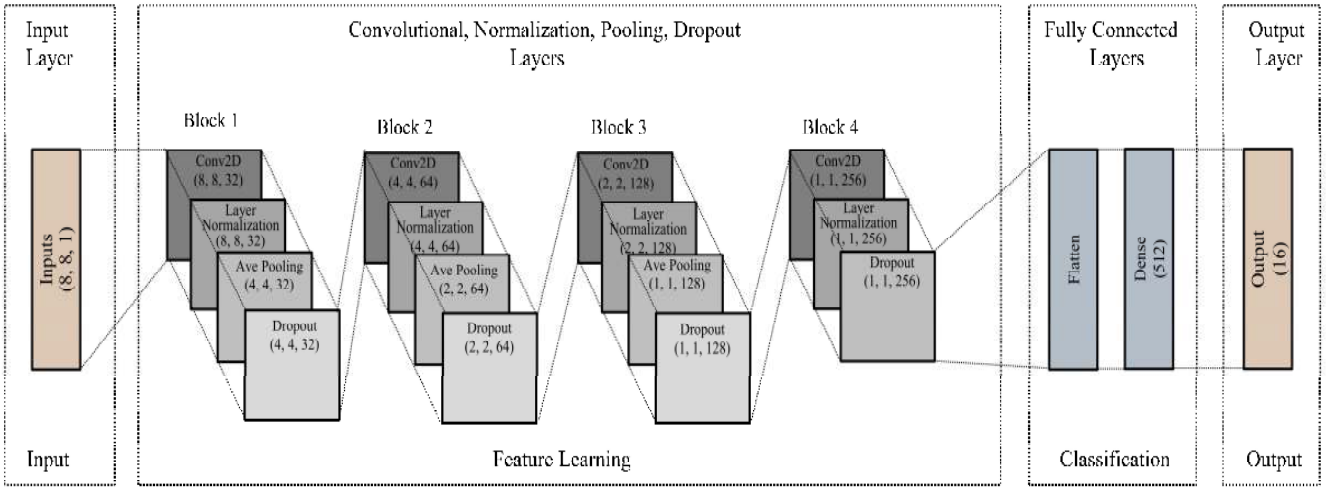


FIGURE 3. Convolution2D Layers View of Proposed Model

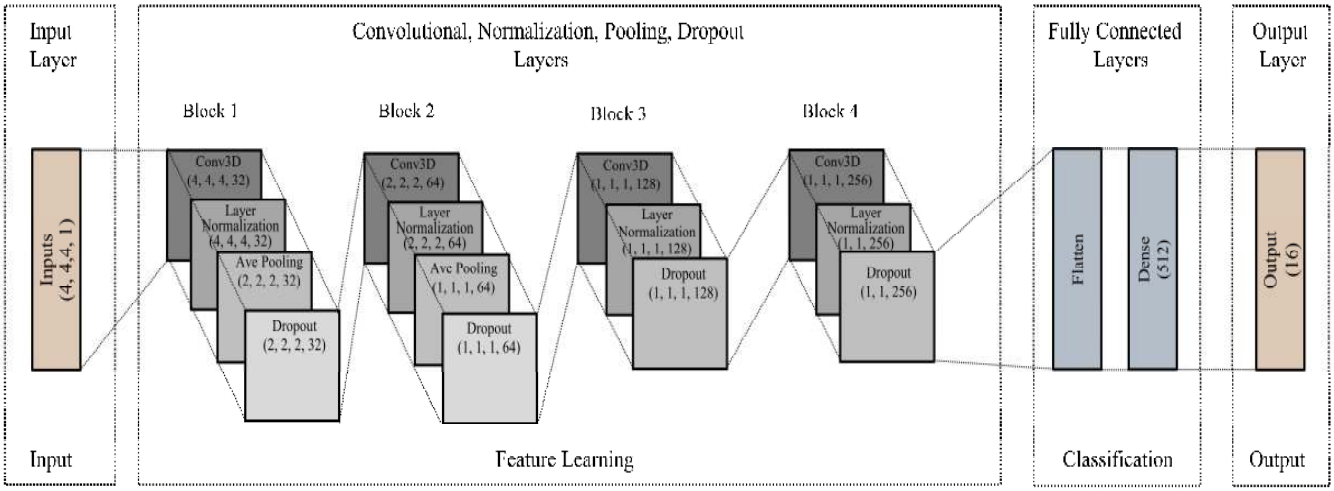


FIGURE 4. Convolution3D Layers View of Proposed Model

activation function. The same hyperparameters are used for CNN1D, CNN2D, and CNN3D for binary classification. Next, the transfer learning methodology was applied to multiclass classification using the BoT-IoT, IoT Network Intrusion, MQTT-IoT-IDS2020, IoT-23, and IoT-DS-1 datasets. The multiclass model uses the IoT-DS-2 dataset pre-trained model to detect and classify anomalies in these datasets.

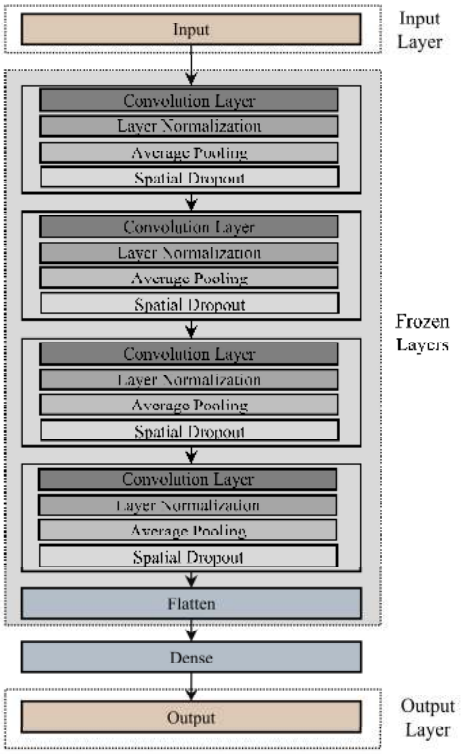


FIGURE 5. Layers View of Proposed Binary Model using Transfer Learning

F. MODEL TUNING

Most deep learning networks require many training iterations to reach the convergence stage, but iterations may be reduced by choosing a precise parameters configuration that enables more convolution in the training process, creating and guiding the network structure. In addition, regularization is also beneficial in avoiding overfitting. We used three regularization methods and various hyperparameters to tune the multiclass and binary class models. We used the same hyperparameters and monitoring optimization for the multiclass and binary CNN models to implement the model generalization for different classification problems. As the baseline construct, we used a multiclass CNN model consisting of four blocks of convolution layers, a flatten layer, a dense layer with 512 neurons, and the number of classes in the dataset represented by the neurons in the output layer. We initialize the CNN model layers with random values to help it learn the features over time.

We used three different methods for regularization: L1, L2, and dropout. The kernel, bias, and activity regularizers are used on the L1 and L2 data preparation levels. L1 is additionally randomized, and L2 is integrated with L1. Dropout, L1, and L2 produce a more generalized model. Convolutional neural networks with three architectures were investigated. We specifically increased/decreased the number of convolutional layer blocks, increasing/reducing filters and kernel size. We also used different dropout rates at the convolutional layer blocks and dense layers. The findings indicate that the reference convolutional neural network model performs better. We choose adam optimizer and apply a sparse categorically cross-entropy loss function to adjust the optimizer weights. In deep learning algorithms, the learning rate is essential since it specifies the size of the steps taken by a model during each iteration. We performed a series of tests, varying the learning rate for adam optimizer (0.01, 0.001, 0.0001, 0.00001), and 0.0001 was chosen as the best learning rate with maximum detection rate. As the network learns, the loss function becomes inversely proportional to its output, and the error trend decreases as accuracy increases. Finally, to prevent overfitting, we implemented an early stopping strategy. When the validation loss does not reduce over a set number of iterations, the training process will stop. The number of epochs must be adjusted to guarantee the highest potential network output during the testing period, to the point that the network accuracy vs. epochs no longer increases. We used 50, 100, 200, and 500 epochs in each CNN model. Since all CNN models converge within 100 epochs, we consider 100 epochs to be the optimum number of epochs.

The activation function of a deep learning algorithm is important. The relu activation function is used in convolution layers, as well as in the dense layer. Softmax activation is used in the output layer. The batch size is also a key hyperparameter to adjust in deep learning systems. By increasing the batch size, we can improve the degree to which computations are parallelized, and we can distribute the training examples across several worker nodes. As a result, model training may be significantly accelerated. However, larger batch sizes have seemed to generalize poorly for testing results despite producing comparable training losses to smaller batch sizes [63]. The generalization gap refers to the difference between train and test error. We ran a set of experiments of different batch sizes to see what would work best (16, 32, 64, 128, 256, 512). A batch size of 64 to 128 was considered the optimal choice for training and testing the CNN model.

IV. DATA COLLECTION

A. DATASETS

The initial phase involves the processing of raw network traffic. This process extracts network functionality from pcap files from datasets. In this study, we used four publicly available IoT datasets. We used CICFlowmeter [64] to extract the features from pcap files and export them in a CSV format. The CICFlowmeter is an open-source flow generation platform that generates network features from pcap data. TCP flows are typically terminated by link

teardown, while UDP flows are terminated by a flow timeout. Our suggested IoT datasets provide general network features and flow-based network features. Each dataset contains 80 distinct network features.

1). BoT-IoT DATASET

Koroniotis et al. [65] created the BoT-IoT dataset. The BoT-IoT dataset testbed includes VMs linked to the network both over a LAN and the Internet. The PFSense system connects the VMs to the Internet. The Ubuntu server provides IoT resources to emulate a real IoT network, while Kali Linux is used as an attack system. The ostinato tool is used to generate normal network traffic. A realistic smart home framework was generated using five IoT devices that were run locally and linked to the cloud services through the node-red system for developing network traffic. The MQTT protocol is used to transmit IoT messages into the Cloud. The taxonomy of attacks in the BoT-IoT dataset is shown in Fig. 6. There are four attack categories which are further divided into ten subcategories. A comprehensive explanation of the testbed configuration and attacks is available in the referenced article [65]. Our adapted BoT-IoT dataset may be accessed [6].

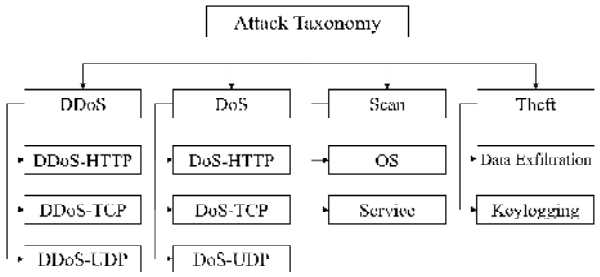


FIGURE 6. BoT-IoT Dataset Attack Taxonomy

2). IoT INTRUSION DETECTION DATASET

Kang et al. [66] developed the IoT Network Intrusion detection dataset. A standard smart home system consisting of a smart home SKT NGU and EZVIZ Wi-Fi camera was used to produce an IoT Network Intrusion d12ataset. These two IoT devices are used victim devices and are wired to a smart home Wi-Fi router. Laptops, tablets, and smartphones are also linked to the smart home router. These devices were used as attacking devices in the testbed. Fig. 7 shows the taxonomy of attacks in the IoT Network Intrusion dataset.

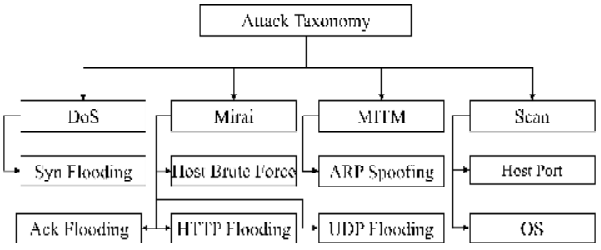


FIGURE 7. IoT Network Intrusion Dataset Attack Taxonomy

There are four attack categories which are further divided into eight subcategories. Binary, category, or subcategory can be used for as label features. Our adapted IoT intrusion detection dataset is available for download on the website [6].

3). MQTT-IoT-IDS2020 DATASET

Hindy et al. [67] develop the MQTT-IoT-IDS2020 dataset. This dataset comprises both common attacks and brute force attacks from the MQTT networking framework. Twelve MQTT sensors, a broker, a system to replicate a camera feed, and an intruder make up the network. The twelve sensors automatically publish random messages during regular service. The dataset includes the most common MQTT attacks and scenarios for testing real-world devices. There are four attack categories in the MQTT-IoT-IDS2020 dataset. Fig. 8 shows the attack categorization in the MQTT-IoT-IDS2020 dataset. Our adapted MQTT-IoT-IDS2020 dataset is available [6].

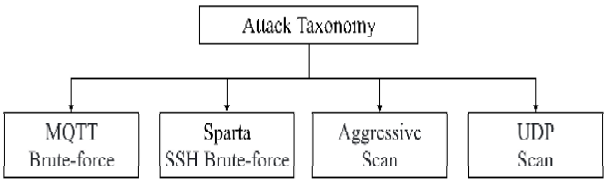


FIGURE 8. MQTT-IoT-IDS2020 Dataset Attack Taxonomy

4). IoT-23 DATASET

The IoT-23 dataset was developed by Stratosphere Laboratory CTU University, Czech Republic [68]. There are 20 malicious-related events and three non-malicious-related events for IoT devices. The objective of the IoT-23 dataset is to give researchers a massive, labeled dataset of real IoT and IoT malware infections to build machine learning models. Attacks in the IoT-23 dataset are classified into nine types, as shown in Fig. 9.

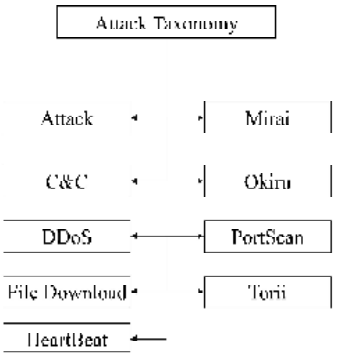


FIGURE 9. IoT-23 Dataset Attack Taxonomy

The IoT-23 dataset includes twenty different network activities to simulate multiple use cases for IoT devices. The benign network traffic was collected by gathering the network traffic of three separate IoT devices. These three devices are real hardware devices, not simulated. Malicious and normal situations operate with unrestricted Internet connectivity in a managed network setting, like every other actual IoT network system. This dataset aims to provide the community with different datasets: the first category contains normal and malicious networks, while the other includes only benign IoT network capture. The primary advantage of the IoT intrusion detection dataset is that it accurately mimics a recent trend in IoT network traffic; it is

also one of the few publicly accessible IoT intrusion detection datasets. Our adapted IoT-23 dataset may be accessed [6].

B. FEATURE PROCESSING

A flow has the same source IP, destination IP, source port, destination port, and protocol. After extracting the features, the next step is to label each dataset instance according to a predefined condition. Each dataset we used in this paper has its own set of rules for labeling dataset instances as normal or malicious. CICFlowmeter extracts 80 network features from pcap files. First, the network features flow ID, source IP, destination IP, and timestamp were removed from all datasets. These network features describe communication in a specific IoT network; however, our proposed model applies to all IoT networks. Second, the dataset non-numeric category features are coveted to a numeric field. We used 0 to fill NaN values in all datasets. After converting the pcap files to CSV files, duplicate instances were introduced. Finally, redundant instances were removed from all datasets.

The BoT-IoT dataset instances presented in Table II, the IoT Network Intrusion Detection dataset instances are shown in Table III, MQTT-IoT dataset instances presented in Table IV, and IoT-23 dataset instances presented in Table V. After removing redundant instances, we can assess the model output during the testing process using previously unseen data. We normalized input feature columns within a defined range $(-1, 1)$ to remove extreme high values and effectively speed up calculations. The binary label column is encoded as 0 for normal and 1 for attack network instances. BoT-IoT dataset multiclass were labeled from 0 to 3 for normal, DoS/DDoS, Scan, and Data theft. The IoT Network Intrusion detection dataset multiclass were labeled from 0 to 4 for normal, DoS, MITM ARP Spoofing, Mirai, and Scan. MQTT-IoT dataset multiclass were labeled from 0 to 4 for normal, MQTT Bruteforce, Scane_A, Scan UDP, and Sparta. The IoT-23 dataset multiclass was labeled from 0 to 9 for normal, attack, Mirai, file download, heartbeat, C&C, Torii, port scan, DDoS, Okiru.

C. PREPROCESSING DATASET

We present four adapted datasets in Table II to V. We developed these datasets using the same software to ensure precise regularity in all datasets. First, we combined BoT-IoT, IoT Network Intrusion, and MQTT-IoT-IDS2020 datasets to increase the number of attacks in the dataset. The new dataset consists of 9 attack classes and a normal class. The new dataset, named IoT-DS-1, is described in Table VI. The IoT-DS-1 dataset multiclass was labeled 0 for normal and 1 to 9 for attack categories. The first reason for the generation of IoT-DS-1 is to increase the number of attacks. The second reason is to evaluate our model with two different datasets having the same number of attack categories. Next, we combined BoT-IoT, IoT Network Intrusion, MQTT-IoT-IDS2020, and IoT-23 datasets further to increase the number of attacks in the dataset. Table VII shows the new dataset named IoT-DS-2, which contains 15 attack classes and a normal class. The IoT-DS-2 dataset multiclass was labeled 0 for normal and 1-15 for attack categories. Due to the imbalance in the training set, we adjusted the class weights to give the classifiers distinct sensitivity to each class. To simplify the class weights calculation, we divided the number of instances in each class using all class quotients to determine the weight. As a result, the under-represented class with fewer samples would have a higher weight score.

The preprocessed data is divided into three sets for classification purposes: training, validation, and testing. The training phase input selected features from the training set and fed them into a neural network model. The testing procedure is used to evaluate the classifier performance against a given test set. We investigated binary and multiclass classification methods. A binary classification model generates either a normal or an attack category for each instance, while a multiclass classification model produces either a normal or an attack category. We used the TensorFlow library and Keras implementations. All our experiments were conducted with Google Colab Pro on a Tesla V100-SXM2 with 27.4 GB RAM. The Pareto Theory, also known as the 80/20 rule, is used to partition the dataset. The dataset is first divided into 80 % for training and 20 % for testing in a stratified way. The stratified methodology ensures an equivalent number of samples from each division of training, validation, and testing sets. The training set is then divided into 80 % for training and 20 % for validation in a stratified way. The total number of instances and class numbers present in each dataset as shown in Table VIII. All three convolution neural network models were evaluated using these six datasets described in Table VIII.

D. FEATURE SELECTION

The selection of features is an important step in the development of a deep learning model. Model improvement techniques known as feature selection include identifying and then only choosing certain features that are needed to enhance prediction. The feature selection strategy minimizes overfitting, speeds up model training, and allows the model less prone to test errors. We used a feature selection technique called RFE (Recursive Feature Elimination) in this paper to extract relevant features from our proposed datasets. Accuracy, precision, recall, and F1 score are used to rank features. A random forest algorithm was used to estimate the overall importance of features. To validate the subset of selected features and the RFE model overfitting, we used a 5- and 10-fold cross-validation test. We use the RFE algorithm to extract 64 features from the IoT-DS-2 dataset. Our suggested feature selection technique utilizes the feature significance, and coefficient attributes to determine the relevance of each feature and then eliminates the least important item from the current collection of features. IoT-DS-2 was chosen as the feature selection dataset because it contains malicious data from all the other datasets. We used the same set of 64 features in CNN1D, CNN2D, and CNN3D for BoT-IoT, IoT Network Intrusion, MQTT-IoT-IDS2020, IoT-23, IoT-DS-1, and IoT-DS-2 datasets.

TABLE II
BOT-IOT DATASET INSTANCES

No	Category	Subcategory	With Redundancy	Without Redundancy
0	Normal	Normal	105202	77511
1	DoS	HTTP	34057	33392
		TCP	19111830	8264448
		UDP	37881485	9122245
2	DDoS	HTTP	51934	50709
		TCP	15975894	8410058
		UDP	21049846	9738949
3	Scan	OS Fingerprinting	350093	35675
		Service Scanning	1481465	221276
4	Data Theft	Data Exfiltration	5003	4944
		Keylogging	1387	1313
Total			96048196	35960520

TABLE III
IOT NETWORK INTRUSION DATASET INSTANCES

No	Category	Subcategory	With Redundancy	Without Redundancy
0	Normal	Normal	40073	39851
1	DoS	DoS-Synflooding	59391	59391
2	MITM	MITM ARP Spoofing	35377	32909
3	Mirai	Mirai-Ackflooding	55124	41998
		Mirai-HTTP Flooding	55818	43008
		Mirai-Host Brute force	121181	112990
		Mirai-UDP Flooding	183554	168975
4	Scan	Scan Host Port	22192	21240
		Scan Port OS	53073	50882
Total			625783	571244

TABLE IV
MQTT-IOT-IDS2020 DATASET INSTANCES

No	Category	With Redundancy	Without Redundancy
0	Normal	334318	167159
1	MQTT_Bruteforce	2002780	2001972
2	Scan-A	31245	29276
3	Scan-U	33404	27843
5	Sparta	1252259	1217198
Total		3654006	3443448

TABLE V
IOT-23 DATASET INSTANCES

No	Category	With Redundancy	Without Redundancy
0	Normal	4313776	4253672
1	Attack	1716778	1699608
2	Mirai	756	756
3	File Download	8035	7707
4	HeartBeat	12895	12648
5	C&C	23981	20612
6	Torii	33858	24492
7	Port Scan	65944863	2999999
8	DDoS	20768988	4619869
9	Okiru	13718252	12908506
Total		106542182	26547869

TABLE VI
IOT-DS-1 DATASET CLASSES

No	Category	BoT-IoT	IoT Net-ID	MQTT	Total
0	Normal	77511	----------	167159	244670
1	DDoS	17420085	----------	----------	17420085
2	DoS	----------	59391	----------	59391
3	MITM ARP Spoofing	----------	32909	----------	32909
4	MQTT Bruteforce	----------	----------	2001972	2001972
5	Mirai	----------	366971	----------	366971
6	OS Scan	35675	----------	----------	35675
7	Port Scan	----------	----------	57119	57119
8	Sparta	----------	----------	1217198	1217198
9	Theft	6257	----------	----------	6257
Total					21442247

TABLE VII
IOT-DS-2 DATASET CLASSES

No	Category	BoT-IoT	IoT Net-ID	MQTT	IoT-23	Total
0	Normal	----------	----------	----------	4253672	4253672
1	DDoS	17420085	----------	----------	----------	17420085
2	DoS	----------	59391	----------	----------	59391
3	MITM ARP Spoofing	----------	32909	----------	----------	32909
4	Mirai	----------	366971	----------	----------	366971
5	MQTT Bruteforce	----------	----------	2001972	----------	2001972
6	Sparta	----------	----------	1217198	----------	1217198
7	Theft	6257	----------	----------	----------	6257
8	Attack	----------	----------	----------	1699608	1699608
9	C&C	----------	----------	----------	20612	20612
10	File Download	----------	----------	----------	7707	7707
11	HeartBeat	----------	----------	----------	12648	12648
12	Okiru	----------	----------	----------	12908506	12908506
13	OS Scan	35675	----------	----------	----------	35675
14	Port Scan	----------	----------	----------	2999999	2999999
15	Torii	----------	----------	----------	24492	24492
Total						43067702

TABLE VIII.
DATASET INSTANCES AND NUMBER OF CLASSES

Dataset	Dataset Name	Instances	Classes
BoT-IoT	BoT-IoT	35960520	4
IoT NI	IoT Network Intrusion Dataset	571244	5
MQTT	MQTT-IoT-IDS2020 Dataset	3443448	5
IoT-23	IoT-23 Dataset	26547869	10
IoT-DS-1	BoT-IoT, IoT Network Intrusion, MQTT-IoT-IDS2020	21442247	10
IoT-DS-2	BoT-IoT, IoT Network Intrusion, MQTT-IoT-IDS2020, IoT-23	43067702	17

V. EVALUATION RESULTS

A. ANALYSIS OF RESULTS

The multiclass and binary CNN models are validated using the accuracy, precision, recall, and F1 score. Accuracy is expressed as the proportion of accurately identified samples to the total number of samples. Precision is measured by the ratio of appropriately classified items to the total TP (True Positive) and FP (False Positive). The recall value is determined by calculating the overall amount of TP measurements by the total number of TP and FN (False Negative). Finally, the F1 score is computed as the weighted average of precision and recall. Additionally, we also calculate TPR, TNR, FPR, and FNR. Where TPR (True Positive Rate) refers to the number of abnormal items that test positive, the TNR (True Negative Rate) is the number of normal samples that are found to be negative, the number of

normal samples that test positive is known as the FPR (False Positive Rate), and FNR (False Negative Rate) is the number of abnormal samples that test negative.

$$Accuracy = \frac{(TP + TN)}{(TP + FP + TN + FN)} \quad (1)$$

$$Precision = \frac{TP}{(TP + FP)} \quad (2)$$

$$Recall = \frac{TP}{(TP + FN)} \quad (3)$$

$$F1\ score = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} \quad (4)$$

The CNN model accuracy and loss were measured for both the training and validation sets at each epoch value. It allows us to assess if the model has been sufficiently learned to differentiate between various anomalies and how many data points in the validation set have been correctly identified. The loss function is perhaps the most important aspect of neural networks. The gradients are calculated using the loss function, and the gradient is used to update the neural network biases, increasing or decreasing the neural network weights. TensorFlow has a variety of loss functions that can be used to accomplish a variety of tasks. In this paper, we used adam optimizer and applied a sparse categorically cross-entropy loss function. Fig. 10 shows the loss of the CNN model during training and validation. The logarithmic loss function measures the total deviation for each test within the training set. The average loss for training 0.05, while the testing loss measured 0.0007. When the validation loss does not reduce for a certain number of iterations, the early stopping technique will end the training process to reduce the over-fitting problem. We trained the CNN model using a 100 epoch, a batch size of 128, and patience of 5 iterations. The loss function and accuracy plot are inversely related, as seen in Fig. 10. The average accuracy was 99.20 for training, 99.30 for validation, and 99.90 for testing using the BoT-IoT dataset. The accuracy did not improve with 200 and 500 epochs and 10 iterations of patience. Consequently, operating a model over a large number of epochs results in overfitting the training data.

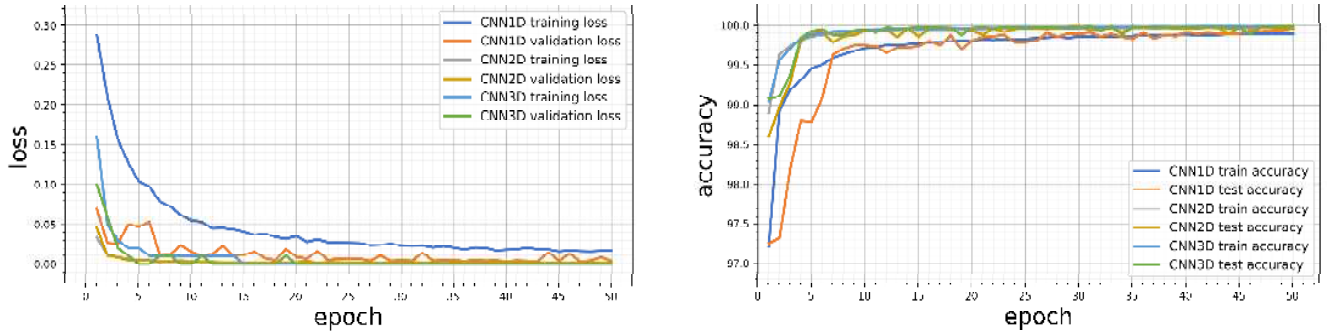


FIGURE 10. Performance of Multiclass CNN1D Model in Training and Validation

C. MULTICLASS CLASSIFICATION

The multiclass classification was used to categorize the dataset as normal network traffic or any attack described in Tables II to VII. Fig. 10 presents the multiclass classification model efficiency during training and validation in terms of loss and accuracy. It is found on the function curves; the accuracy and loss values have inverse functions. Overfitting is reduced due to the early stopping strategy with patience of 5 iterations. The training and validation processes take less than 100 epochs to complete. The loss of training and validation dropped slowly up to 100 epochs. This evidence confirms that these models would correctly categorize various cyber-attacks present in the datasets or real IoT networks. The effectiveness of the multiclass CNN model is then accomplished utilizing BoT-IoT, IoT Network Intrusion, MQTT-IoT-IDS2020, IoT-23, IoT-DS-1, and IoT-DS-2 datasets. The BoT-IoT dataset consists of three attack categories and a normal category. The performance of CNN models using the BoT-IoT dataset is presented in Table IX (a). All CNN models produce high accuracy over four classes. The detection rate for Normal, DoS, Scan, and Theft was 99.90%, 99.96%, 99.91%, and 98.10%. Overall, FNR was 0.67 %, and FPR was 0.05 %.

The IoT Network Intrusion dataset consists of five classes. The CNN model performance for the IoT Network Intrusion dataset is presented in Table IX (b). The detection rate for the Normal class is 99.63%, while the detection rate for the DoS class is 99.94%. The detection rates for Mirai, MITM, and Scan were 97.85%, 88.23%, and 93.30%. The detection rate for Mirai, MITM, and Scan classes was not high as Normal and DoS classes. There are five classes in the MQTT-IoT-IDS2020 dataset. Table IX (c) shows that the MQTT-IoT-IDS2020 dataset achieved a high detection rate for the Normal class and all malicious classes. Normal, Scan, and Sparta classes correctly detected. The only misclassification occurred in the MQTT brute force attack class, resulting in an FNR of 1.48 %. The CNN model capacity to classify larger multiclass datasets was evaluated by merging the BoT-IoT, IoT Network Intrusion, and MQTT-IoT-IDS2020 datasets.

The new IoT-DS-1 consists of 9 attack classes and a normal class. The findings of CNN models on the IoT-DS-1 dataset are summarized in Table X. All CNN models obtained a high detection score for Normal, DDoS, DoS, MQTT_BF, OS Scan, Sparta, and Theft classes. On the other hand, malicious classes MITM, Mirai, and Port scan belong to the IoT Network Intrusion dataset with a high misclassification ratio. Next, we illustrate the CNN model efficiency using the IoT-23 dataset, which comprises a normal class and nine malicious classes. Normal, Attack, Mirai, C&C, Torii, Portscan, DDoS, and Okiru had a detection rate greater than 99.70%. While the detection rate for FileDownload 98.29% and the detection rate for Heartbeat 97.60%. In comparison to IoT-DS-1, the proposed model performs better on the IoT-23 dataset. The findings of CNN models on the IoT-23 dataset are shown in Table XI.

Finally, the CNN model capability for larger multiclass classification was evaluated. In Table XII, we present the results of CNN models using the IoT-DS-2 dataset. The proposed IoT-DS-2 dataset consists of a normal class and 15 attack classes. The Normal class has a detection score of 99.98%. DDoS, Okiru, Portscan, Torii achieved a detection rate of ~100%. The detection rate for DoS, Mirai, MQTT-BF, Sparta, Theft, Attack, and OS scan measured greater or equal to 99.60%. The lowest detection rate was measured for MITM. A summary of the multiclass classification TPR, TNR, FPR, and FNR is presented in Table XIII.

Next, we used transfer learning for multiclass classification for BoT-IoT, IoT Network Intrusion, MQTT-IoT-IDS2020, IoT-23, and IoT-DS-1. This was performed using a pre-train model using the IoT-DS-2 dataset for the CNN1D, CNN2D, and CNN3D models. We chose a batch size of 128 across all datasets for regular multiclass classification; however, this batch size does not perform well for transfer learning multiclass classification models. Several experiments were carried out to determine the most appropriate batch size to use. Batch sizes of 32 and 64 work well for transfer learning models. Table XIV summarizes the average accuracy, precision, recall, and F1 score for regular multiclass classification, whereas Table XV summarizes the average accuracy, precision, recall, and F1 score for transfer learning multiclass classification.

The CNN1D and CNN2D models perform better than CNN3D. These models achieved approximately the same detection rate as normal multiclass classification models. The CNN3D model has a relatively high error rate throughout training, validation, and testing. The CNN3D model detection rate is insufficient since it has very high FPR and FNR rates. In addition, it has a lower detection rate for the normal class compared to other malicious classes. We conclude from our study that CNN1D and CNN2D are more effective at detecting anomalies in multiclass classification. These models are also better at detecting anomalies in transfer learning multiclass classification. This research shows that the proposed model will help create an effective network intrusion detection system with a high detection rate for IoT networks.

TABLE IX
MULTICLASS CLASSIFICATION BOT-IOT, IOT NETWORK INTRUSION, AND MQTT-IOT-IDS2020 DATASETS

Model	CNN1D				CNN2D				CNN3D			
Class	Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score
(a) BOT-IOT DATASET												
Normal	99.96	99.90	99.53	99.71	99.96	99.88	99.62	99.75	99.95	99.86	99.46	99.66
DoS	99.97	99.96	99.99	99.98	99.97	99.97	99.99	99.98	99.97	99.96	99.99	99.98
Scan	99.98	99.91	99.92	99.92	99.97	99.90	99.91	99.91	99.95	99.76	99.88	99.82
Theft	99.99	98.10	96.26	97.17	99.99	100.00	91.82	95.74	99.99	96.83	76.73	85.61
(b) IOT NETWORK INTRUSION DATASET												
Normal	99.86	99.47	98.48	98.97	99.86	99.47	98.48	98.97	99.82	99.63	97.76	98.69
DoS	99.99	99.97	99.90	99.94	99.99	99.97	99.90	99.94	99.98	99.94	99.90	99.92
Mirai	97.77	98.44	98.09	98.27	97.77	98.44	98.09	98.27	97.27	97.85	97.92	97.89
MITM	98.80	90.61	87.90	89.24	98.80	90.61	87.90	89.24	98.43	88.23	83.53	85.82
Scan	98.69	93.18	96.71	94.91	98.69	93.18	96.71	94.91	98.65	93.30	96.19	94.72
(c) MQTT-IOT-IDS2020 DATASET												
Normal	99.93	98.87	99.72	99.29	99.93	98.85	99.73	99.29	99.93	98.52	100.00	99.26
MQTT-BF	99.93	99.98	99.91	99.94	99.93	99.98	99.90	99.94	99.93	100.00	99.88	99.94
Scan-A	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Scan-U	100.00	100.00	99.97	99.99	100.00	100.00	99.97	99.99	100.00	100.00	99.98	99.99
Sparta	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00

TABLE X
MULTI CLASS CLASSIFICATION IoT-DS-1 (BoT-IoT, IoT NETWORK INTRUSION AND MQTT-IoT-IDS2020 DATASETS)

Model	CNN1D				CNN2D				CNN3D			
Class N	Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score
Normal	99.92	99.57	99.43	99.50	99.93	99.65	99.54	99.60	99.94	99.72	99.52	99.62
DDoS	99.96	99.95	99.97	99.96	99.98	99.97	99.99	99.98	99.98	99.96	99.99	99.98
DoS	99.97	99.07	99.50	99.29	99.98	99.80	99.15	99.47	99.97	99.50	99.25	99.38
MITM	99.23	53.37	94.55	68.23	99.58	76.40	75.18	75.78	99.56	74.06	76.38	75.21
MQTT-BF	99.99	99.95	99.87	99.91	99.99	99.99	99.79	99.89	99.99	99.99	99.80	99.89
Mirai	99.01	99.37	91.12	95.07	99.40	96.72	97.62	97.17	99.39	96.91	97.30	97.10
OS Scan	99.95	99.85	99.81	99.83	99.95	99.80	99.87	99.83	99.94	99.80	99.85	99.82
Port Scan	99.60	76.09	89.84	82.40	99.74	90.66	83.82	87.10	99.74	90.17	84.18	87.07
Sparta	99.99	99.91	99.96	99.93	99.98	99.82	99.99	99.90	99.99	99.85	99.99	99.92
Theft	100.00	99.89	99.89	99.89	99.99	99.95	99.31	99.63	100.00	99.89	99.68	99.79

TABLE XI
MULTI CLASS CLASSIFICATION IoT-23

Model	CNN1D				CNN2D				CNN3D			
Class N	Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score
Normal	99.97	99.99	99.91	99.95	99.98	99.96	99.96	99.96	99.92	99.83	99.89	99.86
DDoS	99.99	99.98	99.98	99.98	99.99	99.96	99.98	99.97	99.99	99.95	99.98	99.97
Attack	100.00	100.00	100.00	100.00	100.00	99.55	100.00	99.77	100.00	99.55	100.00	99.77
Mirai	99.99	98.29	99.57	98.93	99.99	99.34	99.09	99.21	99.99	99.34	99.05	99.19
File Download	99.98	97.60	99.81	98.69	99.92	90.38	99.15	94.56	99.92	89.77	99.15	94.23
HeartBeat	99.99	99.69	99.85	99.77	99.93	99.63	94.56	97.03	99.87	98.18	91.36	94.65
C&C	99.99	99.98	99.98	99.98	99.99	99.96	99.96	99.96	99.99	99.97	99.97	99.97
Torii	100.00	99.99	100.00	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99
Port Scan	99.99	99.999	99.99	99.99	100.00	100.00	100.00	100.00	99.99	99.99	99.99	99.99
Okiru	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.000	99.99	100.00	99.99

TABLE XII
MULTI CLASS CLASSIFICATION IoT-DS-2 (BoT-IoT, IoT NETWORK INTRUSION AND MQTT-IoT -IDS2020, IoT-23 DATASETS)

Model	CNN1D				CNN2D				CNN3D			
Class N	Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score
Normal	99.97	99.98	99.82	99.90	99.81	99.21	99.57	99.39	99.79	99.10	99.57	99.33
DDoS	99.99	99.99	99.99	99.99	99.96	99.94	99.91	99.93	99.95	99.89	99.92	99.90
DoS	99.99	99.63	99.53	99.58	99.96	99.80	97.57	98.67	99.96	99.68	97.72	98.69
MITM	99.76	76.46	97.06	85.54	99.55	67.57	72.39	69.89	99.50	69.20	55.06	61.33
Mirai	99.76	99.60	96.67	98.12	99.51	96.33	96.08	96.20	99.47	95.12	96.81	95.96
MQTT-BF	99.99	99.85	99.78	99.82	99.93	99.99	98.12	99.05	99.99	99.95	99.67	99.81
Sparta	99.98	99.77	99.86	99.82	99.88	98.46	99.44	98.95	99.94	99.32	99.63	99.48
Theft	99.99	99.68	99.60	99.64	99.98	99.42	96.21	97.79	99.98	99.45	96.68	98.05
Attack	99.99	99.84	100.00	99.92	99.99	99.27	100.00	99.64	99.99	99.91	99.83	99.87
C&C	99.99	99.54	99.90	99.71	99.80	96.74	90.59	93.57	99.80	97.57	90.43	93.87
File Download	99.99	98.82	99.75	99.28	99.99	99.12	98.49	98.80	99.99	98.13	99.26	98.69
HeartBeat	99.99	98.95	99.89	99.41	99.89	91.16	97.61	94.28	99.90	91.08	99.17	94.95
Okiru	100.00	100.00	100.00	100.00	100.00	100.00	99.99	99.99	100.00	99.99	100.00	99.99
OS Scan	99.99	99.98	99.96	99.97	99.92	99.55	99.61	99.58	99.88	99.30	99.45	99.38
Port Scan	99.99	100.00	99.99	99.99	99.99	100.00	99.99	99.99	99.99	99.99	99.99	99.99
Torii	100.00	100.00	99.98	99.99	99.99	99.98	99.98	99.98	99.99	99.97	99.92	99.94

TABLE XIII
AVERAGE MULTICLASS CLASSIFICATION TPR, TNR, FPR, FNR

Model / Dataset	CNN1D				CNN2D				CNN3D			
	TPR	TNR	FPR	FNR	TPR	TNR	FPR	FNR	TPR	TNR	FPR	FNR
BoT-IoT	99.33	99.94	0.06	0.67	99.35	99.95	0.05	0.65	99.00	99.92	0.08	1.00
IoT Network Intrusion	96.94	99.24	0.76	3.06	96.22	99.11	0.89	3.78	95.06	98.88	1.12	4.94
MQTT-IoT-IDS2020	98.60	99.93	0.07	1.40	98.60	99.93	0.07	1.40	98.52	99.90	0.10	1.48
IoT-23	99.88	99.95	0.05	0.12	99.68	99.86	0.14	0.32	99.47	99.77	0.23	0.53
IoT-DS-1	97.39	99.88	0.12	2.61	95.42	99.92	0.08	4.58	95.60	99.92	0.08	4.60
IoT-DS-2	99.49	99.98	0.02	0.51	96.60	99.94	0.06	3.40	95.84	99.94	0.06	4.16

TABLE XIV
AVERAGE MULTICLASS CLASSIFICATION ACCURACY, PRECISION, RECALL, AND F1SCORE

Model / Dataset	CNN1D				CNN2D				CNN3D			
	Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score
BoT-IoT	99.97	99.95	99.95	99.95	99.95	99.95	99.95	99.95	99.94	99.92	99.92	99.92
IoT Network Intrusion	97.76	97.80	97.76	97.78	97.55	97.56	97.55	97.55	97.08	97.07	97.08	97.07
MQTT-IoT-IDS2020	99.93	99.92	99.93	99.92	99.93	99.93	99.93	99.93	99.92	99.90	99.91	99.90
IoT-23	99.96	99.97	99.96	99.96	99.90	99.91	99.90	99.90	99.84	99.85	99.84	99.84
IoT-DS-1	98.80	99.16	98.80	98.98	99.26	99.20	99.18	99.19	99.25	99.16	99.15	99.15
IoT-DS-2	99.70	99.74	99.70	99.72	99.43	99.42	99.43	99.42	99.07	99.03	99.07	99.05

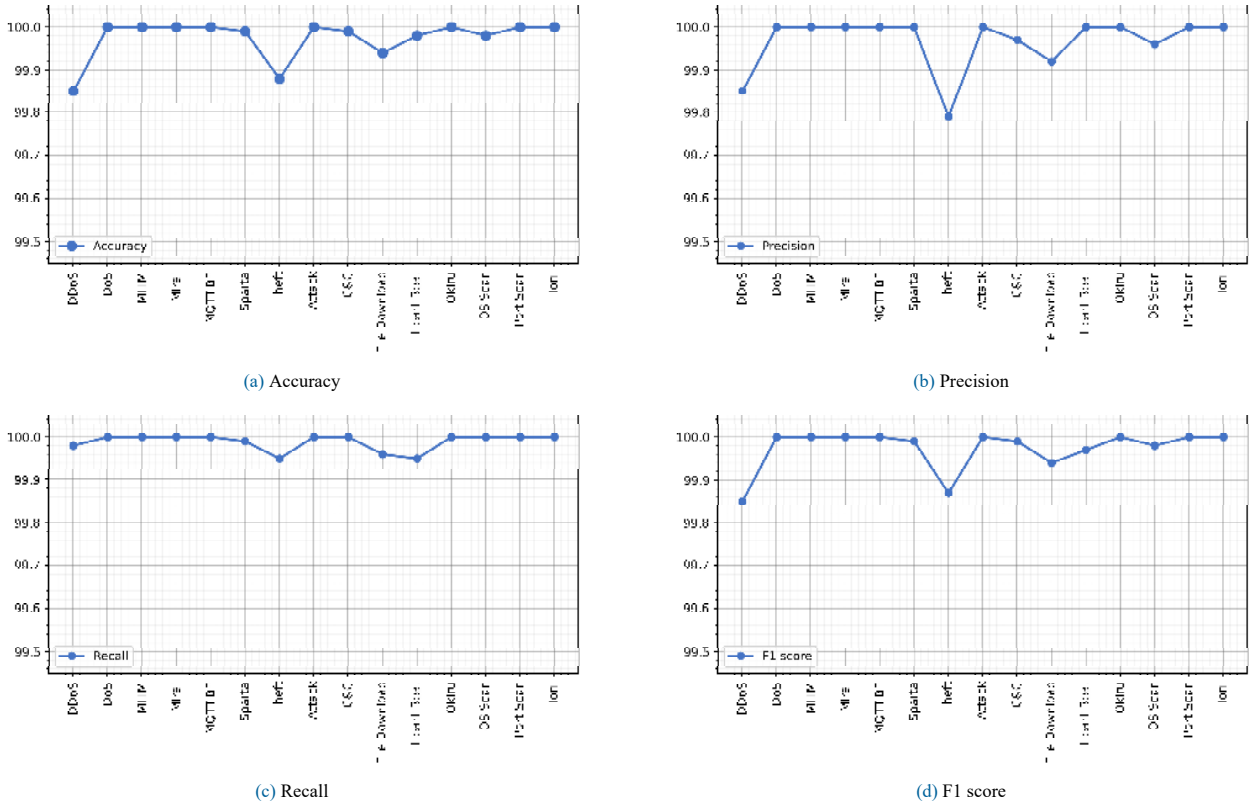
TABLE XV
AVERAGE MULTICLASS CLASSIFICATION ACCURACY, PRECISION, RECALL, AND F1SCORE USING TRANSFER LEARNING

Model / Dataset	CNN1D				CNN2D				CNN3D			
	Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score
BoT-IoT	99.30	99.26	99.20	99.23	98.60	98.56	98.57	98.56	98.13	98.10	98.09	98.10
IoT Network Intrusion	86.28	88.84	86.28	87.54	86.79	87.51	86.79	87.14	82.20	81.80	82.00	81.90
MQTT-IoT-IDS2020	99.92	99.90	99.92	99.91	99.93	99.93	99.93	99.93	96.10	96.21	96.08	95.19
IoT-23	99.62	99.61	99.62	99.61	99.60	99.60	99.58	99.59	86.00	87.45	86.00	82.60
IoT-DS-1	97.72	97.43	97.72	97.57	98.90	98.80	98.75	98.77	86.41	88.00	87.00	87.50

B. BINARY CLASS CLASSIFICATION

The proposed binary CNN model classification accuracy, precision, recall, and F1 score were assessed using the BoT-IoT, IoT Network Intrusion, MQTT-IoT-IDS2020, IoT-23, IoT-DS-1, and IoT-DS-2 datasets. The same set of features were used for binary CNN1D, CNN2D, and CNN3D models. The binary classification assigns the dataset instance to one category: normal network traffic or malicious network traffic. We evaluate and conduct experiments on each dataset mentioned in Table VIII. Fig. 11 shows the accuracy, precision, recall, and F1 score for binary classification using CNN1D and IoT-DS-2 dataset. The minimum detection rate for binary classification was 99.79% for the Theft class. The evaluation metrics for binary classification (Min, Max, Std Dev, Avg) for BoT-IoT, IoT Network Intrusion, MQTT-IoT-IDS2020, IoT-23, IoT-DS-1, and IoT-DS-2 datasets using CNN1D are presented in Table XVI.

The binary classification was performed using a transfer learning approach. The binary CNN model took less time to train and validate compared to multiclass classification. The use of a pre-train model further reduces the training time for binary classification. Early stopping and dynamic learning rates monitor the number of training epochs and increase adam optimization process efficiency during training. The IoT-DS-2 pre-trained model was used for binary classification of BoT-IoT, IoT Network Intrusion, MQTT-IoT-IDS2020, IoT-23, and IoT-DS-1 datasets. The IoT-DS-2 pre-train model was chosen because it contains all the attacks and normal network traffic from the BoT-IoT, IoT Network Intrusion, MQTT-IoT-IDS2020, IoT-23, and IoT-DS-1 datasets. The Binary CNN model has high accuracy, precision, recall, and F1 score with a small number of incorrectly classified instances. Attackers may use various options of the same or other resources to configure the attack. Additionally, several sophisticated attackers can employ evasion mechanisms. However, identical header fields inside the packets may be used to accomplish the same attack objective. Consequently, the dataset network traffic indicates authentic hacker activities, illustrating the proposed model generalizability [61]. In terms of runtime, the CNN model took between 10 and 20 minutes to complete each binary classification's training, validation, and testing process.



(a) Accuracy (b) Precision

(c) Recall (d) F1 score

FIGURE 11. Binary Classification Performance Metrics CNN1D Model for IoT-DS-2 Using Transfer Learning

TABLE XVI
BINARY CLASSIFICATION (MIN, MAX, STD DEV, AVG) USING TRANSFER LEARNING

Dataset	Metrics	Min	Max	Std Dev	Avg
BoT-IoT	Accuracy	99.81	99.95	0.0600	99.90
	Precision	99.60	99.85	0.1315	99.75
	Recall	99.60	100.00	0.1780	99.85
	F1Score	99.60	99.90	0.1328	99.80
IoT Network Intrusion	Accuracy	99.95	100	0.0263	99.98
	Precision	99.95	100	0.0222	99.98
	Recall	99.95	99.99	0.0200	99.98
	F1Score	99.95	99.95	0.0210	99.98
MQTT-IoT-IDS2020	Accuracy	99.99	100	0	99.99
	Precision	99.98	100	0	99.99
	Recall	99.99	100	0	99.99
	F1Score	99.99	100	0	99.99
IoT-23	Accuracy	99.82	100	0.0560	99.98
	Precision	99.60	100	0.1580	99.90
	Recall	99.90	100	0.4456	99.98
	F1Score	99.80	100	0.0870	99.93
IoT-DS-1	Accuracy	99.75	100	0.1085	99.90
	Precision	99.75	100	0.1085	99.90
	Recall	99.75	100	0.1052	99.89
	F1Score	99.75	100	0.1079	99.90
IoT-DS-2	Accuracy	99.85	100	0.0505	99.96
	Precision	99.79	100	0.0756	99.95
	Recall	99.95	100	0.0309	99.98
	F1Score	99.85	100	0.0512	99.97

VI DISCUSSION AND COMPARISON RESULTS

In this section, the results of CNN models are compared to previous results from other research studies. Our proposed models were significantly more effective at identifying anomalies in IoT networks. The research mentioned in this article investigated the possibility of utilizing a convolutional neural network to solve anomaly detection in IoT networks. We investigated various convolutional neural network's abilities to detect and classify anomalies in IoT networks. Furthermore, we evaluated different convolutional neural network's capacity to detect and classify anomalies in IoT networks via binary and multiclass classification through transfer learning. An input layer, four blocks of convolutional layers, a fully connected dense layer, and an output layer make up the model we used in this article. Our proposed architecture is implemented using CNN1D, CNN2D, and CNN3D models. The BoT-IoT, IoT Network Intrusion, MQTT-IoT-IDS2020, IoT-23, IoT-DS-1, and IoT-DS-2 datasets are used to evaluate the CNN model's performance. Several experiments were conducted with the primary objective of classifying attack categories using binary and multiclass classification.

The concept of transfer learning is being used to implement a pre-train multiclass CNN paradigm for binary and multiclass classification. Ge et al. [61] utilized transfer learning to create a representation for high-dimensional categorical features. To the best of our knowledge, a transfer learning approach was never used for anomaly detection where a pre-train multiclass model can be reused binary and multiclass anomaly detection. Initially, we used the transfer learning technique for binary classification. We choose a pre-train multiclass classification model that was trained on the IoT-DS-2 dataset. One reason to select transfer learning for binary is to keep consistency for the binary classification among different datasets. The second reason is to reduce the complexity and run time of binary classification. Convolution layers were excluded from training during the binary classification training process, but the dense and output layers are used. The transfer learning technique considerably reduces the time required for training, validation, and testing in binary classification. Mohammad et al. [15] propose a CNN-based anomaly detection model to demonstrate binary and multiclass classification. The detection rate of their proposed was not satisfactory for binary and multiclass classification. The binary classification model achieved an accuracy of 97.17% for normal and abnormal classes. The binary classification of our proposed model is compared to the previously proposed models in Table XVII. Numerous recent advancements in deep learning technologies demonstrate their ability to identify patterns through various research fields. We consider four, five, ten, and sixteen categories of IoT network traffic in multiclass classification. Several publications in the literature focused exclusively on binary classification while developing a deep learning model for anomaly detection. Many of these publications use accuracy as a metric of performance for assessing their model. However, we used many performance metrics and a large number of malicious categories. The multiclass classification is divided into small category datasets, consisting of up to five classes. The medium dataset consists of up to ten classes, and large datasets consist of up to sixteen classes. The outcome of the proposed model multiclass classification TPR, TNR, FPR, and FNR for six datasets is presented in Table XIII. The proposed model multiclass classification average accuracy, precision, recall, and F1 score are shown in Table XIV.

TABLE XVII
BINARY CLASSIFICATION

MODEL	MODEL	ACCURACY	PRECISION	RECALL	F1SCORE
[15]	CNN	97.17	97.00	97.00	97.00
[33]	CNN	86.95	86.95	86.95	86.95
[44]	CNN	99.24	-------	-------	-------
Proposed Model	CNN1D	99.96	99.90	99.95	99.93
	CNN2D	99.98	99.95	99.96	99.96
	CNN3D	99.98	99.96	99.95	99.95

TABLE XVIII
MULTICLASS CLASSIFICATION

Article	MODEL	ACCURACY	PRECISION	RECALL	F1 SCORE
[15]	CNN	98.43	98.00	98.00	98.00
[19]	C-LSTM-AE	99.62	98.78	97.20	97.98
[33]	CNN	86.95	89.56	87.25	88.41
[58]	CNN	98.02	97.71	98.39	98.05
[9]	CNN	98.37	-------	-------	-------
[55]	CNN	95.86	-------	-------	-------
[23]	CNN	97.34	-------	-------	-------
[54]	CNN	92.53	-------	-------	-------
[56]	C-CMU	99.20	85.40	99.92	91.80
[11]	FFN	98.09	98.88	98.88	98.88
[61]	FFN	99.79	99.79	99.79	99.79
[62]	SNN	98.73	99.17	98.36	98.77
Proposed Model	CNN1D	99.97	99.95	99.95	99.95
	CNN2D	99.95	99.95	99.95	99.95
	CNN3D	99.94	99.92	99.92	99.92

The multiclass classification results of the CNN models are compared to those mentioned previously in other research articles. Many of these articles are only concerned with the suggested model's accuracy. Table XVIII presents our proposed model's accuracy, precision, recall, F1 score, and other models for multiclass classification analysis. As seen in Table XVIII, our proposed CNN models are more accurate than other deep learning models. Ge et al. [61] build a multiclass classification model using feed-forward neural networks with embedding layers. They used the BoT-IoT dataset for their model validation. In certain attack classes, their model performed well, but in others, the model performed poorly. Our proposed model accuracy, precision, recall, and F1 score exceed other deep learning models. However, a few research papers developed an intrusion detection deep learning algorithm using multiclass

classification. We use multiclass classification to classify IoT network traffic into four, five, ten, and sixteen categories. Our proposed CNN models outperform all other implementations in all datasets included in this study: BoT-IoT, IoT Network Intrusion, MQTT-IoT-IDS2020, IoT-23, IoT-DS-1, and IoT-DS-2. In comparison to the CNN3D model, the CNN1D and CNN2D models perform better. CNN3D requires twice as much time to train as CNN1D or CNN2D models. In addition, the CNN3D model needed a large number of epochs for its convergence. The CNN1D model was the most accurate in terms of detection rate and required the least amount of training time.

VII. CONCLUSION AND FUTURE WORK

Deep learning approaches have demonstrated their capacity to classify anomalies in many fields of research correctly. However, intruders employ novel and innovative techniques to launch cyber-attacks. While significant attempts to track and distinguish these attacks continue to occur in multiple ways in collaboration with other potential attacks such as DDoS and Botnets attacks. This article proposes and develops an anomaly detection model for IoT networks using a convolutional neural network to detect and classify binary and multiclass anomalies. We provide a technique for detecting anomalous activity in IoT networks by generating a new dataset from an existing one. This method was used to create four IoT datasets, which were then combined to increase the number of attack categories. We use the BoT-IoT, IoT Network Intrusion, MQTT-IoT-IDS2020, IoT-23, IoT-DS-1, and IoT-DS-2 intrusion detection datasets to validate our proposed convolutional neural network model. We classify a variety of anomalies using 1D, 2D, 3D convolutional neural network models.

Furthermore, we use the transfer learning principle to build multiclass and binary classification models. Our proposed binary and multiclass classification models showed high accuracy, precision, recall, and F1 score compared to existing classification strategies and recent deep learning implementations. The minimum detection rate of the CNN1D model 99.74%, CNN2D model 99.42%, CNN3D 99.03% for BoT-IoT, MQTT-IoT-IDS2020, IoT-23, and IoT-DS-2 datasets. This study findings indicate that the suggested model will aid in the development of an efficient anomaly-based intrusion detection system for IoT networks that has both a high detection rate and a low false alarm rate.

For future work, we plan to investigate further anomaly detection using various deep learning methods, like FFN and RNN, GAN, and contrast the findings to those obtained using a deep convolutional neural network model.

ACKNOWLEDGMENT

The authors acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC).

REFERENCES

[1] S. Smith, "IoT Connections To Reach 83 Billion By 2024, Driven By Maturing Industrial Use Cases," 2020, [Online]. Available:https://www.juniperresearch.com/press/press-release s/iot-connections -to-reach-83-billion-by-2024-driven (accessed Apr. 10, 2021).

[2] A. Mukherjee, "IoT Growth Demands Rethink of Long-Term Storage Strategies, says IDC," 2020. https://www.idc.com /getdoc .jsp?containerId=prAP46737220 (accessed Apr. 04, 2021).

[3] C. Robberts and J. Toft, 'Finding Vulnerabilities in IoT Devices : Ethical Hacking of Electronic Locks', Dissertation, 2019.

[4] C. Meffert, D. Clark, I. Baggili, and F. Breitinger, "Forensic State Acquisition from Internet of Things (FSAIoT): A general framework and practical approach for IoT forensics through IoT device state acquisition," Proceedings of the 12th International Conference on Availability, Reliability, and Security, pp.1-11, 2017, doi: 10.1145/3098954.3104053.

[5] S. N. Firdous, Z. Baig, C. Valli, and A. Ibrahim, "Modelling and Evaluation of Malicious Attacks against the IoT MQTT Protocol," 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), vol. 2018, pp. 748–755, 2018, doi: 10.1109/iThings-GreenCom-CPSCom-SmartData.2017.115.

[6] I. Ullah and Q. H. Mahmoud, "IoT Intrusion Detection Datasets," 2021. https://sites.google.com/view/iotdataset1. (accessed Apr. 10, 2021).

[7] A. Aldweesh, A. Derhab, and A. Z. Emam, "Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues," Knowledge-Based Systems., vol. 189, pp. 105124, 2020, doi: 10.1016/j.knosys. 2019.105124.

[8] G. Kaur, A. Habibi Lashkari, and A. Rahali, "Intrusion Traffic Detection and Characterization using Deep Image Learning," 2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom /CyberSciTech), pp. 55–62, 2020, doi: 10.1109/DASC-PICom-CBDCom-CyberSciTech 49142.2020.00025.

[9] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," Journal of Information Security and Applications., vol. 50, pp. 102419, 2020, doi: 10.1016/j.jisa.2019. 102419.

[10] T. A. Odetola, O. Oderhohwo, and S. R. Hasan, "A Scalable Multilabel Classification to Deploy Deep Learning Architectures for Edge Devices," arXiv, 2019.

[11] M. Ge, X. Fu, N. Syed, Z. Baig, G. Teo, and A. Robles-Kelly, "Deep Learning-Based Intrusion Detection for IoT Networks," 24th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC), vol. 2019-December, pp. 256–265, 2019, doi: 10.1109/PRDC47002.2019.00056.

[12] R. Pecori, A. Tayebi, A. Vannucci, and L. Veltri, "IoT Attack Detection with Deep Learning Analysis," 2020 IEEE International Joint Conference on Neural Networks (IJCNN), pp. 1-8, 2020, doi: 10.1109/IJCNN48605.2020.9207171.

[13] I. Idrissi, M. Azizi, and O. Moussaoui, "IoT security with Deep Learning-based Intrusion Detection Systems: A systematic literature review," 2020 IEEE Fourth International Conference on Intelligent Computing in Data Sciences (ICDS)., pp. 1-10, 2020, doi: 10.1109/ICDS50568.2020.9268713.

[14] Z. Tian, C. Luo, J. Qiu, X. Du, and M. Guizani, "A Distributed Deep Learning System for Web Attack Detection on Edge Devices," IEEE Transactions on Industrial Informatics, vol. 16, no. 3, pp. 1963–1971, 2020, doi: 10.1109/TII.2019.2938778.

[15] M. M. Hassan, A. Gumaei, A. Alsanad, M. Alrubaian, and G. Fortino, "A hybrid deep learning model for efficient intrusion detection in big data environment," Information Sciences., vol. 513, pp. 386–396, 2020, doi: 10.1016/j.ins.2019.10.069.

[16] R. M. Swarna Priya et al., "An effective feature engineering for DNN using hybrid PCA-GWO for intrusion detection in IoMT architecture," Computer Communications., vol. 160, no. May, pp. 139–149, 2020, doi: 10.1016/j.comcom.2020.05.048.

[17] D. Li, L. Deng, M. Lee, and H. Wang, "IoT data feature extraction and

intrusion detection system for smart cities based on deep migration learning," International journal of information management., vol. 49, no. October 2018, pp. 533–545, 2019, doi: 10.1016/j.ijinfomgt.2019.04.006.

[18] S. Sriram, R. Vinayakumar, M. Alazab, and K. P. Soman, "Network Flow based IoT Botnet Attack Detection using Deep Learning," IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). 2020, pp. 189–194, 2020, doi: 10.1109/INFOCOMWKSHPS 50562.2020.9162668.

[19] C. Yin, S. Zhang, J. Wang, and N. N. Xiong, "Anomaly Detection Based on Convolutional Recurrent Autoencoder for IoT Time Series," IEEE Transactions on Systems, Man, and Cybernetics: Systems., pp. 1–11, 2020, doi: 10.1109/tsmc.2020.2968516.

[20] S. Manimurugan, S. Al-Mutairi, M. M. Aborokbah, N. Chilamkurti, S. Ganesan, and R. Patan, "Effective Attack Detection in Internet of Medical Things Smart Environment Using a Deep Belief Neural Network," IEEE Access, vol. 8, pp. 77396–77404, 2020, doi: 10.1109/ACCESS.2020.2986013.

[21] B. Wang, Y. Su, M. Zhang, and J. Nie, "A Deep Hierarchical Network for Packet-Level Malicious Traffic Detection," IEEE Access, vol. 8, pp. 201728–201740, 2020, doi: 10.1109/access. 2020.3035967.

[22] G. Bae, S. Jang, M. Kim, and I. Joe, "Autoencoder-Based on Anomaly Detection with Intrusion Scoring for Smart Factory Environments," International Conference on Parallel and Distributed Computing: Applications and Technologies, pp. 414-423, 2018, Springer Singapore.

[23] H. Yang and F. Wang, "Wireless network intrusion detection based on improved convolutional neural network," IEEE Access, vol. 7, pp. 64366–64374, 2019, doi: 10.1109/ACCESS.2019. 2917299.

[24] R. Kishore and A. Chauhan, "Evaluation of Deep Neural Networks for Advanced Intrusion Detection Systems," 2020 4th IEEE International Conference on Electronics, Communication and Aerospace Technology (ICECA), pp. 1-8, 2020, doi: 10.1109/ICECA49313.2020.9297515.

[25] Z. Wu, J. Wang, L. Hu, Z. Zhang, and H. Wu, "A network intrusion detection method based on semantic Re-encoding and deep learning," Journal of Network and Computer Applications., vol. 164, no. March, 2020, doi: 10.1016/j.jnca.2020.102688.

[26] Y. Zhou, M. Han, L. Liu, J. S. He, and Y. Wang, "Deep learning approach for cyberattack detection," IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)., pp. 262–267, 2018, doi: 10.1109/ INFCOMW.2018.8407032.

[27] W. W. Nsunza, A. Q. R. Tetteh, and X. Hei, "Accelerating a Secure Programmable Edge Network System for Smart Classroom," 2018 IEEE Smart World, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (Smart World/ SCALCOM/UIC/ATC/CBDCom/IOP/SCI), pp. 1384–1389, 2018, doi:10.1109/SmartWorld.2018.00240.

[28] S. Z. Lin, Y. Shi, and Z. Xue, "Character-Level Intrusion Detection Based on Convolutional Neural Networks," 2018 IEEE International Joint Conference on Neural Networks (IJCNN), vol. 2018-July, 2018, doi:10.1109/IJCNN.2018. 8488987.

[29] Y. Zhang, P. Li, and X. Wang, "Intrusion Detection for IoT Based on Improved Genetic Algorithm and Deep Belief Network," IEEE Access, vol. 7, pp. 31711–31722, 2019, doi: 10.1109/ACCESS.2019. 2903723.

[30] Y. Otoum, D. Liu, and A. Nayak, "DL-IDS: a deep learning–based intrusion detection framework for securing IoT," Transactions on Emerging Telecommunications Technologies., no. September 2020, 2019, doi: 10.1002/ett.3803.

[31] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, and J. Lloret, "Shallow neural network with kernel approximation for prediction problems in highly demanding data networks," Expert Systems with Applications., vol. 124, pp. 196–208, 2019, doi: 10.1016/j.eswa.2019.01.063.

[32] C. Liang, B. Shanmugam, S. Azam, M. Jonkman, F. De Boer, and G. Narayansamy, "Intrusion Detection System for Internet of Things based on a Machine Learning approach," 2019 IEEE International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN), pp. 1-6.,2019, pp. 1–6, doi: 10.1109/ViTECoN.2019.8899448.

[33] Y. Li et al., "Robust detection for network intrusion of industrial IoT based on multi-CNN fusion," Measurement, vol. 154, p. 107450, 2020, doi: 10.1016/j.measurement.2019.107450.

[34] M. Almiani, A. AbuGhazleh, A. Al-Rahayfeh, S. Atiewi, and A. Razaque, "Deep recurrent neural network for IoT intrusion detection system," Simulation Modelling Practice and Theory, pp.102031, vol. 101, no. November 2019, p. 102031, 2020, doi: 10.1016/j.simpat.2019.102031.

[35] Y. N. Kunang, S. Nurmaini, D. Stiawan, and B. Y. Suprapto, "Attack classification of an intrusion detection system using deep learning and hyperparameter optimization," Journal of Information Security and Applications., vol. 58, no. March, p. 102804, 2021, doi: 10.1016/j.jisa.2021.102804.

[36] M. E. Aminanto, R. Choi, H. C. Tanuwidjaja, P. D. Yoo, and K. Kim, "Deep Abstraction and Weighted Feature Selection for Wi-Fi Impersonation Detection," IEEE Transactions on Information Forensics and Security., vol. 13, no. 3, pp. 621–636, 2017, doi: 10.1109/TIFS.2017.2762828.

[37] A. Diro and N. Chilamkurti, "Leveraging LSTM Networks for Attack Detection in Fog-to-Things Communications," IEEE Communications Magazine., vol. 56, no. 9, pp. 124–130, 2018, doi: 10.1109/MCOM.2018.1701270.

[38] S. Rezvy, Y. Luo, M. Petridis, A. Lasebae, and T. Zebin, "An efficient deep learning model for intrusion classification and prediction in 5G and IoT networks," 2019 53rd IEEE Annual Conference on Information Sciences and Systems (CISS), 2019, doi: 10.1109/CISS.2019.8693059.

[39] L. R. Parker, P. D. Yoo, T. A. Asyhari, L. Chermak, Y. Jhi, and K. Taha, "Demise: Interpretable Deep Extraction and Mutual Information Selection Techniques for IoT Intrusion Detection," Proceedings of the 14th International Conference on Availability, Reliability and Security., pp. 1-10, 2019, doi: 10.1145/3339252.3340497.

[40] R. H. Hwang, M. C. Peng, V. L. Nguyen, and Y. L. Chang, "An LSTM-Based Deep Learning Approach for Classifying Malicious Traffic at the Packet Level," Applied Sciences., vol. 9, no. 16, 2019, doi: 10.3390/app9163414.

[41] B. A. Pratomo, P. Burnap, and G. Theodorakopoulos, "Unsupervised Approach for Detecting Low Rate Attacks on Network Traffic with Autoencoder," IEEE International Conference on Cyber Security and Protection of Digital Services (Cyber Security). 2018, pp.p 1-8, doi: 10.1109/CyberSecPODS. 2018.8560678.

[42] B. Roy and H. Cheung, "A Deep Learning Approach for Intrusion Detection in Internet of Things using Bi-Directional Long Short-Term Memory Recurrent Neural Network," 2018 IEEE 28th International Telecommunication Networks and Applications Conference (ITNAC)., 2018, doi: 10.1109/ ATNAC.2018.8615294.

[43] Q. Tian, J. Li, and H. Liu, "A Method for Guaranteeing Wireless Communication Based on a Combination of Deep and Shallow Learning," IEEE Access, vol. 7, pp. 38688–38695, 2019, doi: 10.1109/ACCESS.2019.2905754.

[44] A. Nagisetty and G. P. Gupta, "Framework for Detection of Malicious Activities in IoT Networks using Keras Deep Learning Library," 3rd IEEE International Conference on Computing Methodologies and Communication (ICCMC), no. Iccmc, pp. 633–637, 2019, doi: 10.1109/ICCMC.2019.8819688.

[45] R. A. Khamis, "Evaluating Adversarial Learning on Different Types of Deep Learning-based Intrusion Detection Systems using min-max Optimization," 2020, (Master Dissertation, Carleton University).

[46] R. A. Khamis and A. Matrawy, "Evaluation of adversarial training on different types of neural networks in deep learning-based IDSs," 2020 IEEE International Symposium on Networks, Computers and Communications (ISNCC)., pp. 0–5, 2020, doi: 10.1109/ISNCC49221.2020.9297344.

[47] R. M. A. Ujjan, Z. Pervez, K. Dahal, A. K. Bashir, R. Mumtaz, and J. González, "Towards sFlow and adaptive polling sampling for deep learning based DDoS detection in SDN," Future Generation Computer Systems1., vol. 111, pp. 763–779, 2020, doi: 10.1016/j.future.2019.

10.015.

[48] H. Alaiz-Moreton, J. Aveleira-Mata, J. Ondicol-Garcia, A. L. Muñoz-Castañeda, I. García, and C. Benavides, "Multiclass Classification Procedure for Detecting Attacks on MQTT-IoT Protocol," Complexity, vol. 2019, 2019, doi: 10.1155/2019/ 6516253.

[49] G. Thamilarasu and S. Chawla, "Towards Deep-Learning-Driven Intrusion Detection for the Internet of Things," Sensors (Switzerland), vol. 19, no. 9, 2019, doi: 10.3390/s19091977.

[50] J. Bassey, D. Adesina, X. Li, L. Qian, A. Aved, and T. Kroecker, "Intrusion Detection for IoT Devices based on RF Fingerprinting using Deep Learning," 2019 Fourth International IEEE Conference on Fog and Mobile Edge Computing (FMEC)., pp. 98–104, 2019, doi: 10.1109/FMEC.2019.8795319.

[51] C. Callegari, E. Bucchianeri, S. Giordano, and M. Pagano, "Real Time Attack Detection with Deep Learning," 019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)., vol. 2019-June, 2019, doi: 10.1109/ SAHCN.2019.8824811.

[52] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and A. R. Sadeghi, "DÏoT: A Federated Self-learning Anomaly Detection System for IoT," 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)., vol. 2019-July, pp. 756–767, 2019, doi: 10.1109/ ICDCS.2019.00080.

[53] F. Li, A. Shinde, Y. Shi, J. Ye, X. Y. Li, and W. Song, "System Statistics Learning-Based IoT Security: Feasibility and Suitability," IEEE Internet of Things Journal., vol. 6, no. 4, pp. 6396–6403, 2019, doi: 10.1109/JIOT.2019.2897063.

[54] S. J. Bu and S. B. Cho, "A convolutional neural-based learning classifier system for detecting database intrusion via insider attack," Information Sciences. (Ny)., vol. 512, pp. 123–136, 2020, doi: 10.1016/j.ins.2019.09.055.

[55] X. Liu, Z. Tang, and B. Yang, "Predicting Network Attacks with CNN by Constructing Images from NetFlow Data," 2019 IEEE 5th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS), 2019, pp. 61-66, doi: 10.1109/BigDataSecurity-HPSC-IDS.2019.00022.

[56] R. Vinayakumar, M. Alazab, S. Srinivasan, Q. V. Pham, S. K. Padannayil, and K. Simran, "A Visualized Botnet Detection System Based Deep Learning for the Internet of Things Networks of Smart Cities," IEEE Transactions on Industry Applications., vol. 56, no. 4, pp. 4436–4456, 2020, doi: 10.1109/ TIA.2020.2971952.

[57] H. Alzahrani, M. Abulkhair, and E. Alkayal, "A Multi-Class Neural Network Model for Rapid Detection of IoT Botnet Attacks," International Journal of Advanced Computer Science and Application., vol. 11, no. 7, pp. 688–696, 2020, doi: 10.14569/ IJACSA.2020.0110783.

[58] S. Priyanga, K. Krithivasan, S. Pravinraj, and V. S. Shankar Sriram, "Detection of Cyberattacks in Industrial Control Systems Using Enhanced Principal Component Analysis and Hypergraph-Based Convolution Neural Network (EPCA-HG-CNN)," IEEE Transactions on Industry Applications., vol. 56, no. 4, pp. 4394–4404, 2020, doi: 10.1109/TIA.2020.2977872.

[59] J. Ling, Z. Zhu, Y. Luo, and H. Wang, "An intrusion detection method for industrial control systems based on bidirectional simple recurrent unit," Computers & Electrical Engineering., vol. 91, no. February 2020, p. 107049, 2021, doi: 10.1016/j.compeleceng.2021.107049.

[60] M. A. Ferrag and L. 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, doi: 10.1109/TEM.2019.2922936.

[61] M. Ge, N. F. Syed, X. Fu, Z. Baig, and A. Robles-Kelly, "Towards a deep learning-driven intrusion detection approach for Internet of Things," Computer. Networks, vol. 186, no. August 2020, pp. 107784, 2021, doi:10.1016/j.comnet.2020.107784.

[62] S. Aldhaheri, D. Alghazzawi, L. Cheng, B. Alzahrani, and A. Al-Barakati, "DeepDCA: Novel Network-Based Detection of IoT Attacks Using Artificial Immune System," Applied Sciences., vol. 10, no. 6, 2020, doi: 10.3390/app10061909.

[63] N. S. Keskar, J. Nocedal, P. T. P. Tang, D. Mudigere, and M. Smelyanskiy, "On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima," arXiv preprint arXiv:1609. 04836.

[64] A. H. Lashkari, G. D. Gil, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of Tor Traffic using Time based features," ICISSP 2017 - Proc. 3rd Int. Conf. Inf. Syst. Security. Privacy., vol. 2017-Janua, no. September, pp. 253–262, 2017, doi: 10.5220/00061056025 30262.

[65] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "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, doi: 10.1016/ j.future.2019.05.041.

[66] H. K. K. Hyunjae Kang, Dong Hyun Ahn, Gyung Min Lee, Jeong Do Yoo, Kyung Ho Park, "IoT Network Intrusion Dataset". IEEE Dataport (accessed Apr. 10, 2021).

[67] H. Hindy, E. Bayne, M. Bures, R. Atkinson, C. Tachtatzis, and X. Bellekens, "Machine Learning Based IoT Intrusion Detection System: An MQTT Case Study (MQTT-IoT-IDS2020 Dataset)," 12th International Networking Conference. INC 2020. Lecture Notes in Networks and Systems, vol 180. Springer, Cham. https://doi.org/10.1007/978-3-030-64758-2_6, 2020, doi: 10.1007/978-3-030-64758-2_6.

[68] "Stratosphere Laboratory. A Labeled Dataset with Malicious and Benign IoT Network Traffic." Agustin Parmisano, Sebastian Garcia, Maria Jose Erquiaga., [Online]. Available: https://www.stratosphereips.org/datasets-iot23, (accessed Apr. 10, 2021).

IMTIAZ ULLAH (Member, IEEE) is a Ph.D. candidate in the Department of Electrical, Computer and Software Engineering, Ontario Tech University, Oshawa, Ontario, Canada. He received an MSc in Internet, Computer, and System Security from the University of Bradford, United Kingdom. He received MSc in Computer Science from the University of Peshawar, Pakistan. His current research interests include deep learning, anomaly detection models for the Internet of Things.

QUSAY H. MAHMOUD (Senior Member, IEEE) is a professor of software engineering in the Department of Electrical, Computer, and Software Engineering, Ontario Tech University, Canada. He was the Founding Chair of the Department, and more recently, has served as an Associate Dean of the Faculty of Engineering and Applied Science at the same university. His research interests include intelligent software systems and cybersecurity.