MS-ADS: multistage spectrogram image-based anomaly detection system for IoT security.

AHMAD, Z., KHAN, A.S., ZEN, K. and AHMAD, F.

2023

Received: 8 February 2023 | Revised: 2 May 2023 | Accepted: 25 May 2023

DOI: 10.1002/ett.4810

RESEARCH ARTICLE

WILEY

MS-ADS: Multistage Spectrogram image-based Anomaly Detection System for IoT security

Zeeshan Ahmad[1,2] | **Adnan Shahid Khan**[1] | **Kartinah Zen**[1] | **Farhan Ahmad**[3]

[1] Faculty of Computer Science and Information Technology, Universiti Malaysia Sarawak, Kota Samarahan, Malaysia

[2] Department of Electrical Engineering, College of Engineering, King Khalid University, Abha, Kingdom of Saudi Arabia

[3] Explеo Group UK, Derby, United Kingdom

Correspondence
Zeeshan Ahmad and Adnan Shahid Khan, Faculty of Computer Science and Information Technology, Universiti Malaysia Sarawak, Kota Samarahan 94300, Malaysia.
Email: zayshan@kku.edu.sa and skadnan@unimas.my

Funding information
Universiti Malaysia Sarawak

Abstract

The innovative computing idea of Internet-of-Things (IoT) architecture has gained tremendous popularity over the last decade, resulting in an exponential increase in the connected devices and the data processed in the IoT networks. Since IoT devices collect a massive amount of sensitive information exchanged over the traditional internet, security has become a prime concern due to the more frequent generation of network anomalies. A network-based anomaly detection system can provide the much-needed efficient security solution to the IoT network by detecting anomalies at the network entry points through constant traffic monitoring. Despite enormous efforts by researchers, these detection systems still suffer from lower detection accuracy in detecting anomalies and generate a high false alarm rate and false-negative rate in classifying network traffic. To this end, this paper proposes an efficient Multistage Spectrogram image-based network Anomaly Detection System (MS-ADS) using a deep convolution neural network that utilizes a short-time Fourier Transform to transform flow features into spectrogram images. The results demonstrate that the proposed method achieves high detection accuracy of 99.98% with a reduction in the false alarm rate to 0.006% in classifying network traffic. Also, the proposed scheme improves predicting the anomaly instances by 0.75% to 4.82%, comparing the benchmark methodologies to exhibit its efficiency for the IoT network. To minimize the computational and training cost for the model re-training phase, the proposed solution demonstrates that only 40500 network flows from the dataset suffice to achieve a detection accuracy of 99.5%.

1 | INTRODUCTION

The Internet-of-Things (IoT) has emerged as a novel, revolutionary, and ground-breaking computing idea over the previous few years that has been widely welcomed by the technological sectors, such as smart cities having smart homes, smart healthcare, smart industries, smart grids, smart transportation, etc. to name a few.[1] It contains many IoT devices (called Things) that have limited storage, computation, and communication capabilities and are embedded with various types of sensors and actuators to gather and share sensitive information over the conventional Internet.[2] The IoT market is predicted to generate massive revenue over the next decade, starting with 2 billion U.S. dollars in 2020 to grow to 8.131 trillion U.S. dollars by 2030.[3] This prediction attracts stakeholders, such as suppliers, vendors, corporations, manufacturers, etc., to invest in such ground-breaking technology.[1,4]

In an IoT network, a massive amount of critical and sensitive data is captured by the IoT devices and then exchanged over the internet with other IoT devices or cloud data centers for storage and processing purposes. For instance, in the smart healthcare sector, the patient's health information, such as heart rate, blood pressure, etc., is gathered by IoT devices and then exchanged over the internet with healthcare centers.[5] Similarly, smart cars are equipped with many sensors that continuously collect sensitive information about the car and its surroundings in the smart transportation industry. This critical information is then shared with the neighboring vehicles to help ensure road traffic safety and fuel consumption improvement by optimizing the journey times.[6] Hence, the technological paradigm of connecting smart devices to the internet for communication is quite significant and has improved the quality of living efficiently and cost-effectively.[7]

However, security and privacy from malicious threats are needed due to the sensitive and critical nature of the data collected and processed within the IoT network. For instance, any compromise to the patient's health records by an attacker may cause a risk to their lives.[1] Similarly, compromises to the smart car Wi-Fi system may risk the in-car data, devices, and road safety messages.[7] To provide the required security to IoT networks against security threats, various security procedures, such as firewalls, authentication methods, different encryption schemes, and antiviruses are currently adopted as the first protection shield.[8] However, due to the integration of many connected IoT devices and the massive volume of data production, new anomalies that can be either novel or the mutation of an old anomaly are frequently generated. For strengthening the IoT network security, a second protection shield provided by an intrusion detection system (IDS) can be deployed.[2,4] An IDS is a system that can detect intrusions by constantly monitoring the network traffic for any malicious behavior.[9] It can be classified into different types based on its deployment or detection strategy. Regarding the deployment strategies, the IDS can be host- or network-based. While in terms of detection strategy, the IDS can be signature-based, anomaly-based, specification-based, or hybrid detection-based.[10]

Although the idea of IDS was first coined in 1980 by Jim Anderson, many IDS products have evolved to satisfy network security needs.[11] However, immense technological growth has resulted in a significant expansion of network size, interconnected devices, and the applications and the data handled, thereby has demanded an improvement in the current IDS systems, which have shown inefficiency in detecting new malicious security threats by monitoring the vast network traffic behavior. Hence resulting in a decline in the detection accuracy in detecting security attacks and a rise in the false alarm rate (FAR) and false negative rate (FNR).[12]

Researchers have recently explored integrating IDS with artificial intelligence (AI) methods such as machine learning (ML) and deep learning (DL) to address these issues. ML and DL techniques are extremely powerful tools that have gained significant popularity over the last decade due to the invention of very powerful Graphics Processing Units (GPUs).[13,14] Recent studies have highlighted the importance of ML, and DL approaches for network-based IDS (NIDS) in effectively processing the network traffic data and learning the meaningful patterns to help predict them as benign or anomaly flows. The ML approaches heavily rely on feature engineering for learning valuable features from the network flow.[15] In contrast, DL approaches are promising in automatically learning the needful features due to the deep architecture without requiring feature engineering and human involvement, making DL an ideal tool that can be integrated with the NIDS for improving anomaly prediction in an IoT network.

This research mainly focuses on the network-based IDS (NIDS) deployment strategy to secure the entry points of the IoT network from all types of intrusions by adopting the anomaly detection-based detection scheme. A multistage spectrogram image-based anomaly detection system is proposed using a deep convolution neural network (CNN) that utilizes a short-time Fourier Transform (STFT) to transform flow features into spectrogram images. Deep CNN then processes these images to perform the efficient anomaly prediction task. This research extends our previous works[16,17] based on the spectrogram approach for anomaly detection systems (ADS). The proposed works used a single-stage DL approach for the traditional network[16] and IoT network,[17] respectively.

The main contributions of this study are four-fold. (1) To extensively discuss the state-of-the-art DL-based network anomaly detection methodologies. (2) To propose a multistage effective anomaly detection method for an IoT network using spectrograms and deep CNN. (3) To generate a Spectrogram images dataset from the BoT-IoT dataset (4) To evaluate the effectiveness of MS-ADS using the BoT-IoT dataset against different ADS models based on supervised DL algorithms. We also extensively compared MS-ADS with some of the recent state-of-the-art ADS works.

The rest of the paper is organized as follows: Section 2 provides the recent works on DL-based ADS solutions for IoT. Section 3 details the preliminary concepts, followed by the system model explanation in Section 4. Section 5 extensively discusses the dataset, experimental setup, and simulation results. Finally, Section 6 concludes this research article.

2 | RELATED WORK

Researchers have widely utilized AI methods over the last decade to propose efficient ADS. Most of the proposed ADS solutions for an IoT network are ML-based. These solutions have improved the accuracy and minimized the FAR and FNR. However, their performance will be saturated at a certain point due to the gigantic volume of information collected by IoT devices. So DL-based is the preferable choice in such IoT scenarios due to its ability to learn optimal features automatically from the raw data to predict efficiently.[18]

Aversano et al.[19] proposed a hybrid Autoencoder (AE) and DNN-based IDS to protect IoT networks from security threats such as DoS/DDoS, Scanning, Mirai, and malware attacks. The AE performs the feature space reduction task followed by traffic prediction and classification using DNN. The methodology exhibited a lower detection accuracy for Benign and Mirai traffic. Another DL-based ADS is proposed by Wei Ma using the improved method of activation function in Recurrent Neural Network (RNN) for the cloud computing system.[20] The proposed optimized RNN-based detection system improved the detection rate, and effectively reduce the detection time and cost.

Similarly, another hybrid method is proposed by Popaala et al.[21] using AE arranged in Long Short-Term Memory (LSTM) arrangement (LAE) and bidirectional LSTM (BLSTM). The LAE task is to perform feature extraction for dimensionality reduction. At the same time, the BLSTM is used to perform the classification task by predicting the anomalies. Again, the accuracy for detecting Benign flows declined by 6.6% compared to the anomaly flows.

Another DL-based IDS is proposed using the deep recursive recurrent neural by Almiani et al.[22] for IoT network protection from different intrusions, particularly DoS, Privilege Escalation (R2L and U2R), and probe anomalies. It comprises two main engines: traffic analysis engine and classification engine. The system can be deployed at the fog computation layer close to IoT devices and the end-users. Using two stages of network packet filtering helps to detect the anomalies undetected by the first detection level, which improves detection accuracy. However, their proposed methodology is evaluated using a very old NSL-KDD dataset collected from the traditional network flows. The detection accuracy for the minority class anomalies, such as Privilege Escalation anomalies, was on the lower side, exhibiting only 77% and 65% detection accuracy for U2R and R2L anomalies.

Similarly, Diro et al.[23] proposed a distributed DNN-based attack detection system for securing the IoT network. They proposed deploying the attack detection system at the fog nodes as these are closer to the smart infrastructure of the IoT in a distributed fashion. They also used a coordinating master node deployed at the edge of the distributed fog network for efficient parameter sharing.

Thamilarasu et al.[24] proposed a three-stage IDS framework using Deep Belief Network to fabricate the feed-forward DNN as the perceptual learning model for IoT network prevention. The model is evaluated using real network traces and simulation to show the superiority of the proposed solution in detecting blackhole, opportunistic service, DDoS, sinkhole, and wormhole attacks.

Sriram et al.[25] proposed a DL-based IoT botnet attack detection framework based on network traffic flows. The proposed solution captures the network flows, transforms them into connection records, and uses a DL-based solution to detect anomalies from compromised IoT devices. A real-time hybrid anomaly and specification-based IDS is proposed by Bostani et al.[26] to detect routing attacks such as sinkhole and selective-forwarding attacks in IoT networks. Reddy et al.[27] proposed a novel DL-based framework for classifying anomalies from normal behaviors based on the type of attack in the Distributed Smart Space Orchestration System traffic traces data set. The proposed solution exhibits a noticeable improvement for most of the attacks within the dataset.

The literature suggests that many proposed solutions were effective in detecting most of the considered anomalies. However, the detection accuracy was not very promising for a few anomalies. Also, some DL-based methodologies exhibited lower detection accuracy for Benign traffic, comparing the anomalies causing extra overhead to the detection model. Literature shows a tradeoff between the model's complexity and detection accuracy. A few studies also exhibited that the detection accuracy is improved at the cost of increased model complexity, considering training time and the resource consumed. Also, DL research is still in its early phase for the ADS for IoT scenarios, with excessive research room available that needs to be explored by researchers to improve IDS efficiency in detecting both new and old attacks efficiently and correctly. To this end, we explore using a CNN to propose our efficient MS-ADS solution utilizing the spectrogram images generated from the IoT network flows.

3 | PRELIMINARY CONCEPTS

This section furnishes the details about the convolutional neural network and the spectrogram and IoT hierarchical architecture concepts that are an integral part of MS-ADS.

3.1 | Convolutional Neural Network

CNN is one of the promising feed-forward DL mechanisms that has proven its effectiveness in processing the specific arrangement of data in an array-like topology (e.g., time series data) or grid-like topology (e.g., an image with a grid-like pixel arrangement).[28,29] CNN employs a specialized linear mathematical operation called convolution in at least one of its layers. A typical CNN comprises an input layer, a stack of a convolutional layer (CL) with a specific activation function and a pooling layer (PL), the fully connected layer (FCL), and a final classification output layer, as depicted in Figure 1. The multiple layers of the CL and PL stack arrangement perform the feature extraction and feature map reduction tasks. At the same time, the FCLs and classification output layer handle the classification and prediction tasks.[16] The different layers of CNN are discussed in the following text.

3.1.1 | Convolutional layer

A CL constitutes the core of the CNN that adopts the convolution operations for extracting the useful features from the input image. This layer incorporates the convolution operation by utilizing the dot product between the input image and the convolution kernel (filter) to learn features and generate the feature map. The operations performed in this layer can be shown mathematically,[30,31]

$$F_m = b_m + \sum_n (X_n * K_{nm}) \tag{1}$$

where F_m is the mth feature map, b_m is the mth bias function, and X_n is the nth input. K_{nm} is a convolutional kernel connecting nth input with mth output. The symbol $*$ represents the convolution operation. The feature maps are then passed through an activation function σ to generate layer output Y_m.

$$Y_m = \sigma(F_m) \tag{2}$$

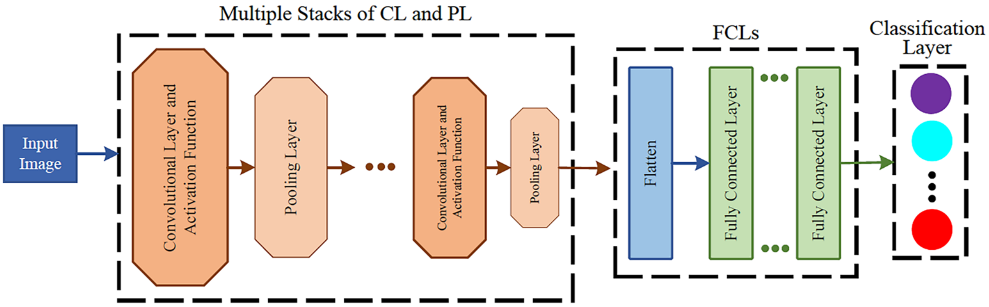
Multiple Stacks of CL and PL
Input Image
Convolutional Layer and Activation Function
Pooling Layer
Convolutional Layer and Activation Function
Pooling Layer
FCLs
Flatten
Fully Connected Layer
Fully Connected Layer
Classification Layer

FIGURE 1 Deep CNN block diagram.

After the back-propagation step during CNN training, the biases and weights adjustment constitute competent feature detection filters.

Activation functions

In a neural network, the activation functions control the layer's output. These functions can be linear or nonlinear depending upon the type of application used. This study employs ReLU as an activation function for the CL, while sigmoid and softmax as activation functions for the final classification layer for binary and multiclass classification and prediction, respectively. Mathematically, these activation functions for any given input x are given as,

$$\text{ReLU}(x) = \max(0, x) \tag{3}$$

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}} \tag{4}$$

$$\text{softmax}(x)_k = \frac{e^{x_k}}{\sum_{i=1}^{N} e^{x_i}} \text{ for } k = 1, \ldots, N \tag{5}$$

3.1.2 | Pooling layer

The PL follows the CL layer to reduce the feature maps using the nonlinear down-sampling process (such as maximum over a nonoverlapping subset of the feature map employed in this research). This process effectively reduces the image size to improve memory usage, eventually reducing the number of parameters to help avoid model overfitting. Also, dropout is employed as the regularization technique to enhance the model's accuracy by preventing its overfitting.

3.1.3 | Fully connected layers

FCLs typically follow the multiple layers of the CL and PL stack. It first converts the stack output image, the reduced 2-D image matrix, to a 1-D vector and then passes it through the network of dense layers to prepare it for classification.

3.1.4 | Classification layer

The last layer of the CNN is generally a classification layer that employs a specific activation function to classify network traffic by performing prediction tasks. Sigmoid is the activation function when network traffic is classified as benign or an anomaly. At the same time, softmax is the activation function when the network flows are classified as benign and one of the specific anomaly classes.

3.2 | Spectrogram

A spectrogram is a pictorial representation of the signal's visual details. It depicts the signal's frequency with time in the form of an image by representing the magnitude of the frequency at a specific time contained in the signal by varying the color heatmaps against the vertical frequency axis to express the energy.[32] Spectrograms are very efficiently utilized in different fields, for instance, speech analysis[33] and the medical field for ECG analysis.[34] In this study, we adopted the STFT technique for performing the time-frequency analysis to obtain the spectrogram images from the available discrete data. The STFT will first divide the discrete-time signal into small chunks using windowing techniques such as the Hanning window, etc., and then compute each chunk's Fourier transform individually. The STFT of a discrete-time signal $x[n]$ is mathematically given as,[34,35]

$$\text{STFT}\{x[n]\} = X(m, \omega) = \sum_{n=-\infty}^{\infty} x[n]w[n-m]e^{-j\omega n} \tag{6}$$

where $X(m, \omega)$ is the STFT of the discrete-time data samples $x[n]$ represented as STFT$\{x[n]\}$. Also, m is the reference time indicating the location in the time domain signal, while ω is the angular frequency. The $w[n]$ is the analysis window function that is only non-zero during the interval $[0, N-1]$ and is used to divide $x[n]$ into small chunks, The spectrogram is then calculated as,

$$\text{Spectrogram}\ (m, \omega) = |STFT\{x[n]\}|^2 = |X(m, \omega)|^2 \tag{7}$$

where the energy is distributed and visualized in the form of heatmaps in the two-dimensional time-frequency plane.[36]

3.3 | IoT hierarchical architecture

The IoT has revolutionized networking in recent years with the potential to enhance the overall quality of life. It contains a vast network of interconnected internet-enabled devices equipped with sensors, storage, computational, and communication capabilities. It generates a vast amount of sensitive data that is shared over the unpredictable internet and needs to be secured. This study proposes the ADS for the typical three-layered IoT architecture,[4,37] as depicted in Figure 2. It consists of a perception layer, network layer, and application layer. The perception layer is the lowest, also called the physical layer. It involves the hardware equipment and devices, such as sensors, actuators, etc., that regularly collect data and then exchange it using various communication standards and protocols like Bluetooth, Wi-Fi, ZigBee, 6LowPAN, etc. The perception layer performs the task of device-to-device communication, anomaly detection, and sending/receiving the data to and from the network layer.[38]

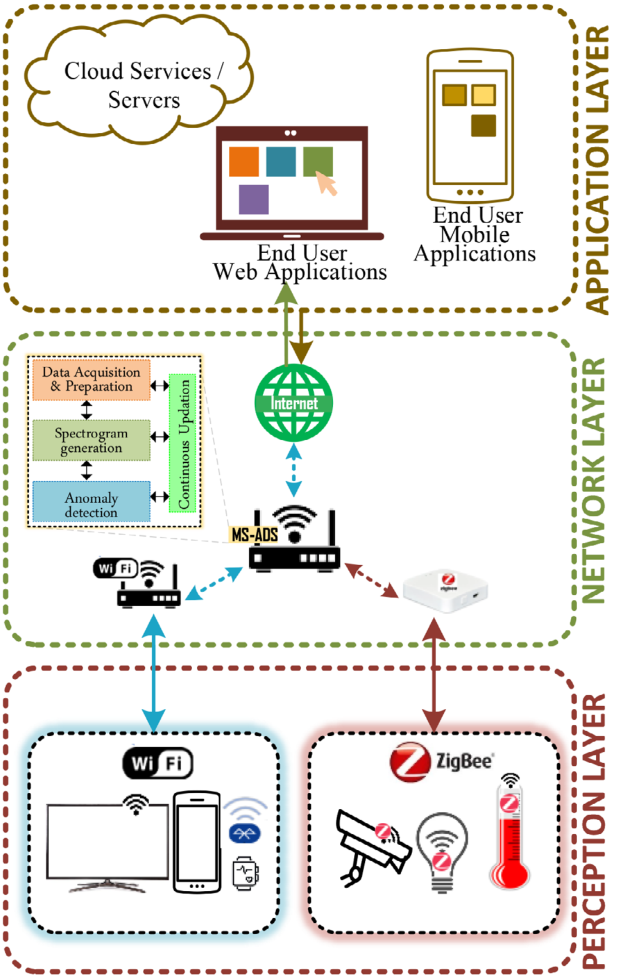

FIGURE 2 A typical three-layered hierarchical IoT architecture. Universiti Malaysia Sarawak, Kota Samarahan, Malaysia

The network layer is the gateway layer to ensure the routing of data packets using different communication standards such as 4G, 5G, Wi-Fi, ZigBee, IPv6, etc. It controls the communication between the perception layer and the application layer. The perception layer also performs the task of device-to-device transmission, anomaly detection, and sending and receiving data from the application and perception layers.[38]

The final application layer, also called the software layer, processes the data for visualization for end users' applications, for example, smart health monitoring. This layer is also responsible for handling the cloud-based activities of the IoT network. Some protocols used in this layer are the Constrained Application Protocol (CoAP) and Data Distribution Service (DDS). The application layer provides services to the end-user, application-to-application communication, anomaly detection, and sending and receiving data to the network layer.[38]

4 | PROPOSED SOLUTION

We adopted a NIDS-based anomaly detection strategy to secure the IoT network from all possible anomalies. The proposed multistage spectrogram image-based anomaly detection system using a deep convolutional neural network (MS-ADS) is deployed at the entry points of an IoT network, for example, at the edge router considering the network layer of a three-layered IoT architecture. Also, MS-ADS will use the cloud or network fogs to maintain the network flow definitions and perform the training in the cloud servers or network fogs to ease the computational and storage requirements. The MS-ADS, as depicted in Figure 3, is a multistage security solution comprised of four main stages: the Data acquisition and preparation stage, the Spectrogram dataset generation stage, the Deep CNN-based Anomaly Detection stage, and the continuous learning stage.

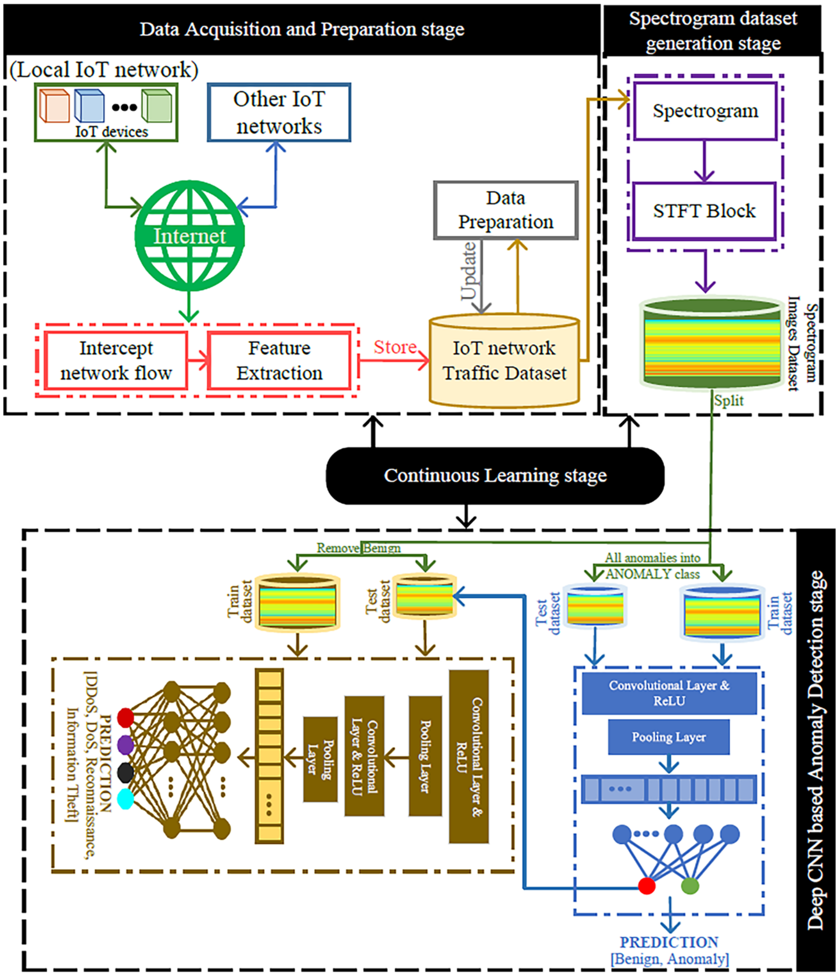
Data Acquisition and Preparation stage
Spectrogram dataset generation stage
(Local IoT network)
IoT devices
Other IoT networks
Internet
Data Preparation
Update
Spectrogram
STFT Block
Spectrogram Images Dataset
Intercept network flow
Feature Extraction
Store
IoT network Traffic Dataset
Split
Continuous Learning stage
Remove Benign
All anomalies into ANOMALY class
Train dataset
Test dataset
Test dataset
Train dataset
Convolutional Layer & ReLU
Pooling Layer
Convolutional Layer & ReLU
Pooling Layer
Convolutional Layer & ReLU
Pooling Layer
PREDICTION [DDoS, DoS, Reconnaissance, Information Theft]
PREDICTION [Benign, Anomaly]
Deep CNN based Anomaly Detection stage

FIGURE 3 Proposed MS-ADS model.

4.1 | Data acquisition and preparation stage

This stage will acquire the network data packets, extract the useful features, and then process them to remove any redundant information to prepare for the next stage. The network traffic will be intercepted either within the network, such as traffic from the IoT devices, or from outside the IoT network through the internet. For this study, we assume that all the communication by the IoT devices within the IoT network is routed through the edge router where our proposed MS-ADS is deployed.

The network traffic data can be intercepted using network sniffing tools such as TCPdump, Wireshark, Ettercap, Argus, EtherApe, etc. The main tasks of the sniffing tools are to acquire, examine, analyze, and visualize the network packets.[39] The sniffer thoroughly analyzes the captured network flows to generate the raw packet features. These features are then stored to form a dataset which is the first step in the generation of the feature set, which will be used to train the DL model in the last stage of MS-ADS. Assume that D_{nm} represents the data stored as a dataset with n network packet records of m features each such as,

$$D_{nm} = \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1m} \\ f_{21} & f_{22} & \cdots & f_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ f_{n1} & f_{n2} & \cdots & f_{nm} \end{bmatrix} \tag{8}$$

The dataset is then passed to the Data preparation stage to pre-process by first removing redundant records with null or infinite features. Then encoding of the categorical features is performed using a one-hot encoding mechanism followed by normalizing the numerical features based on each feature value between 0 and 1.

4.2 | Spectrogram dataset generation stage

The main task of this stage is to generate the spectrogram images using the D_{nm} dataset prepared in stage 1. To formulate the spectrogram generation process, we consider a single record k within the dataset as a discrete-time vector $D_{km} = x_k[n] = \{f_{k1}, f_{k2}, \cdots, f_{km}\}$ with features values considered as sample values for each discrete time vector. These samples are then passed to the STFT block to find the frequency representation of the signal for time-frequency analysis, such as,

$$X_k(m, \omega) = \sum_{n=-\infty}^{\infty} x_k[n] w_{hn}[m-n] e^{-j\omega n} \tag{9}$$

where X_k is the STFT of the signal $x_k[n]$ using the Hanning window function $w_{hn}[n]$, that is mathematically represented as,

$$w_{hn}[n] = \frac{1}{2}\left(1 - \cos\left(2\pi \frac{n}{N}\right)\right), 0 \le n \le N-1 \tag{10}$$

where N indicates the time observation length. Now to generate the spectrogram, the square of the magnitude of the X_k is calculated as,

$$\text{Spectrogram } (m, \omega) = |X_k\ (m, \omega)|^2 \tag{11}$$

Similarly, spectrogram images of all the network traffic records in the dataset are generated. All the spectrograms generated are then re-processed to remove both x-axis (time) and y-axis (frequency) values to treat them as a specialized image. These images are then stored in another dataset called the Spectrogram Images Dataset, which will be used in be next stage to train the CNN model for efficient prediction.

4.3 | Deep CNN-based anomaly detection stage

This stage is the heart of MS-ADS to detect anomalies in an IoT network. It is further subdivided into two substages. Substage-1 will perform the initial screening of the data packets by performing the binary classification to predict network packets as either benign or anomaly. The benign packet can then proceed to the network without further action. In contrast, the predicted anomaly packet is forwarded to substage-2, and an initial alarm signal is generated to notify the administrator. Substage-2, once it receives the anomaly packet, will predict the exact type of the anomaly to help the administrator take timely action.

For substage-1, CNN is chosen because of its ability to process, learn and predict efficiently from the images. The image from the spectrogram images dataset will be input into the deep CNN network. The spectrogram image is input as $(28 \times 28 \times 3)$ images. The first layer of CNN is the CL which will perform the convolution task on the given spectrogram image using the convolutional kernel of dimension (3×3), with the stride of 1, and the padding is kept as same to keep the same spatial dimensions as the input. Stride describes the number of pixels that shift over the image matrix. Also, the padding process means adding empty pixels around the image to help preserve the original image dimensions. For the CL, ReLU is used as an activation function. The CL will transform the spectrogram into the image into $(28 \times 28 \times 32)$ due to 32 learning filters.

The PL will follow the CL layer to reduce the size of the CL feature maps. The PL strategy followed in this study is the Max pooling that will choose the highest value in each patch of the feature map. The pooling layer filter is (2×2) with the stride of 1 and keeps the padding the same to match the input spatial dimension. As a result of PL, the feature map is transformed and reduced to $(14 \times 14 \times 32)$, which is then forwarded to the classification block for the binary classification. The classification block is constructed from the flatten layer followed by a dense neural network to prepare it for the prediction task as shown in Figure 1. The flatten layer will convert the 2-D matrix into a 1-D array of 3136, which is then passed through a fully connected dense network. This study uses a single FCL with 128 neurons for substage-1. The activation operation considered in the FCL is ReLU. The last layer of substage-1 is the classification layer, with only two neurons representing either the flow as benign or anomaly. The sigmoid is used as an activation function for the binary classification layer.

The substage-2 will be engaged only once an anomaly packet is detected by MS-ADS. In that case, an alarm will be sent to the network administrator to notify about the anomaly flow. Also, the detected anomaly flow will be processed in substage-2 to find out the exact nature of the anomaly. The main motive behind allowing benign traffic to pass through directly without activating substage-2 is to reduce the complexity of the MS-ADS. For this study, we focus on detecting four different types of anomalies as DoS, DDoS, Reconnaissance, and information theft. The main architecture of the CNN used for substage-2 is almost similar but has more feature extraction and classification block layers. The spectrogram image of the detected anomaly flow is then forwarded to the substage-2 as $(28 \times 28 \times 3)$ image, which will be transformed to $(28 \times 28 \times 32)$ and $(14 \times 14 \times 32)$ feature map by the first set of CL and PL layer. Again, ReLU is the activation function used for the CL layer. Also, the used filters for CL and PL are (3×3) and (2×2), respectively, with stride one and the same pooling to keep the same spatial dimension as the input. The second set of CL and PL layers transform the feature maps into $(14 \times 14 \times 64)$ and $(7 \times 7 \times 64)$ feature maps using 64 filters of (3×3) and (2×2) each for CL and PL, respectively.

The $(7 \times 7 \times 64)$ feature map is first flattened into a 1-D vector of 3136 in the classification block. Then it is passed through a 2-layer dense FCL with 128 neurons each. Finally, it is passed through the classification layer with four neurons, one each for DDoS, DoS, Reconnaissance, and information theft anomaly, to perform the multiclass prediction task. The softmax activation function is used in this study for the multiclass classification task to find the exact type of anomaly, which can help the network administrator to take necessary actions based on the nature and intensity of the threat.

4.4 | Continuous learning stage

This stage is an important block for the MS-ADS and will help keep the detection model up to date with the new anomaly patterns. In the context of IoT, network threats are increasing daily with the more frequent generation of anomalies that are either the mutation of an old anomaly or new ones. So, to make MS-ADS more efficient in detecting all anomalies and flexible for deployment, a continuous learning stage is included,[40] as depicted in Figure 4. It will update the model in the offline mode in the cloud servers/fog nodes by the network administrator.

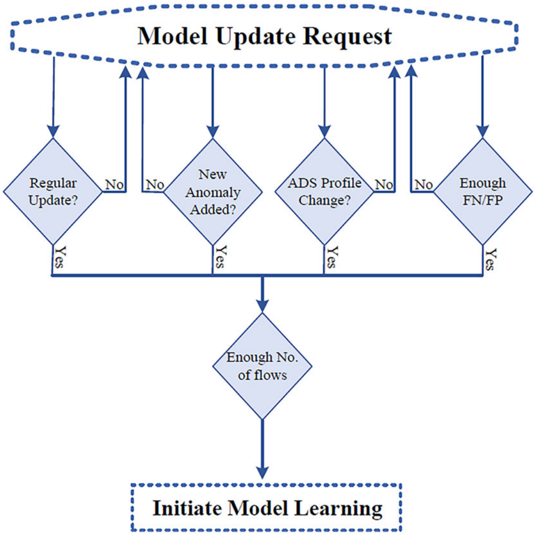

FIGURE 4 Flow diagram for continuous learning stage.

This study considers the different conditions to initiate the model learning process. These conditions include checking whether it is a regular learning process, a new anomaly detected, or a pattern added by the network administrator, the ADS profile updated by an administrator, or if the model generates enough false positives and false negatives.

5 | RESULTS AND ANALYSIS

5.1 | Dataset

This study utilizes the BoT-IoT dataset[41] to evaluate MS-ADS. The dataset is by the Cyber Range Lab, UNSW Canberra, Australia, using the realistic network environment for the IoT network. The Bot-IoT dataset is generated using a testbed consisting of Virtual machines, simulated IoT scenarios, and feature extracting and analytic tools. The dataset contains network traces from five different IoT scenarios from a smart home system: a weather station, smart fridge, motion-activated light, smart garage door, and a smart thermostat. The dataset files are publicly available in the PCAP and CSV formats. The CSV file contains 46 features, but it lacks flow-based features. The feature set is improved by Ullah et al.[42] by extracting more network and flow-based features from the PCAP file provided in the Bot-IoT dataset and is made publicly available in the CSV format. For this study, we adopted 82 features of different data types, such as integer, float, and categorial.

The Bot-IoT dataset contained the network traces for the five different classes: the Benign class and the Benign, DDoS anomaly, DoS anomaly, Reconnaissance (Information gathering) anomaly, and Information theft anomaly classes. Each anomaly class also contains a subcategory under which the network flows for other types of anomalies are gathered. The dataset's DDoS and DoS anomaly traces are collected considering TCP, UDP, and HTTP protocols. While the Information theft anomaly flows are collected considering Data theft and Keylogging instances. Also, the Reconnaissance anomaly instances include the traces of Service Scanning and OS Fingerprinting. Our focus in this study is to detect and predict only the main category of anomalies. Since, for IoT networks, all four considered anomalies can cause severe threats to the network, detecting and reporting the main anomaly category will be sufficient to prevent IoT networks. Adding another block to detect the subcategories will add more overhead for the IoT network. However, to extract each anomaly category, an equal number of flows from different subcategories is included for fair prediction tasks. The detailed number of flows included in this study for the binary and multiclass classification tasks considering the benign and the anomaly (DDoS, DoS, Reconnaissance, and Information Theft) classes are detailed in Table 1. All extracted features are cleaned to remove the infinite, empty, and duplicate entries, followed by the encoding and normalization process.

TABLE 1 BoT-IoT dataset distribution.

Binary classification		Multiclass classification	
Category	**No. of instances**	**Category**	**No of instances**
Benign	30 000	Benign	30 000
Anomaly	105 000	DDoS	30 000
		DoS	30 000
		Reconnaissance	30 000
		Information theft	15 000
Total flows	*135 000*	*Total flows*	*135 000*

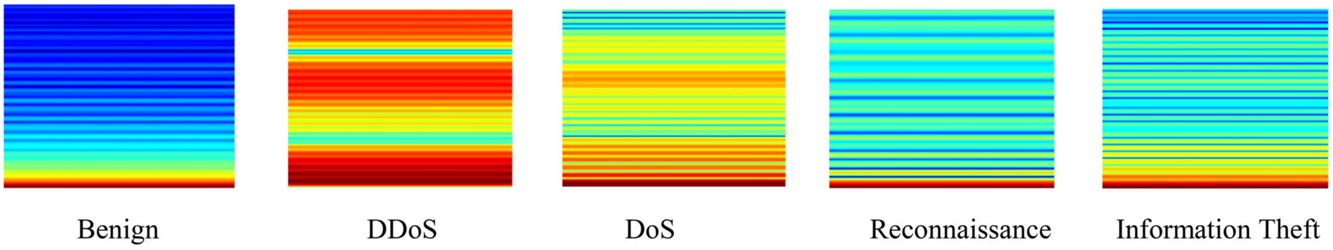

FIGURE 5 Samples from spectrogram images dataset.

TABLE 2 A confusion matrix.

		Predicted class	
		Anomaly	**Benign**
Actual class	Anomaly	True positive (*TP*)	False negative (*FN*)
	Benign	False positive (*FP*)	True negative (*TN*)

5.1.1 | Spectrogram images dataset

This study focuses on training the DL algorithms in the deep CNN Anomaly Detection stage using spectrogram images.[16] For that, spectrogram images of the pre-processed dataset are generated using Equations (9)–(11) and are stored as another dataset called the spectrogram images dataset. Figure 5 depicts each class's random spectrogram image sample from the spectrogram images dataset.

5.2 | Evaluation metrics

This study's performance evaluation metrics include accuracy, Precision, Recall, F1-Score, True Negative Rate, FAR, and FNR. These metrics are calculated from the different fields of the standard confusion matrix given in its binary version in Table 2. A confusion matrix is a two-dimensional matrix that details the actual ground classes and the predicted classes based on the experiment's results. *TP* and *TN* are the correctly predicted anomaly and benign instances, whereas *FN* and *FP* are the incorrect prediction of the classifier as benign and anomaly, respectively. The formal definition and the mathematical formula of different evaluation metrics considered in this study are[43] as follows:

5.2.1 | Accuracy

Accuracy represents the ratio of the correctly predicted instances (both benign and anomaly instances) to the total number of test instances. It measures the overall effectiveness of the model. Mathematically, the Accuracy is calculated as,

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

5.2.2 | Precision

Precision denotes the ratio of correctly predicted anomaly instances to all the samples predicted as anomalies. Mathematically, the Precision is calculated as,

$$\text{Precision} = \frac{TP}{TP + FP} \tag{13}$$

5.2.3 | Recall

The recall is also called the detection rate and is equivalent to the true positive rate (*TPR*). It is denoted as the ratio of all the correctly classified anomaly samples to all the anomaly samples. Mathematically, Recall is calculated as,

$$\text{Recall} = \frac{TP}{TP + FN} \tag{14}$$

5.2.4 | F1 score

F1 score is the harmonic mean of the Precision and Recall and provides a statistical technique for examining the accuracy of a system. Mathematically $F1$ score is calculated as,

$$F1\ \text{score} = 2 \times \left(\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \right) \tag{15}$$

5.2.5 | True negative rate

True negative rate (TNR) is defined as the ratio of the number of correctly classified benign samples to all the samples labeled as benign. Mathematically TNR is given as,

$$\text{TNR} = \frac{TN}{FP + TN} \tag{16}$$

5.2.6 | False alarm rate

The false alarm rate (FAR), also called the false positive rate (*FPR*), is defined as the ratio of wrongly predicted anomaly instances to all the instances labeled as benign. It reflects the probability that a false alarm will be raised. It is mathematically given as,

$$\text{FAR} = FPR = \frac{FP}{FP + TN} \tag{17}$$

5.2.7 | False negative rate

False-negative rate (FNR) denotes the miss rate and shows the possibility of the classifier missing the anomaly instances. It is denoted as the ratio of wrongly predicted benign instances to all the actual anomaly instances.

$$\text{FNR} = \frac{FN}{TP + FN} \tag{18}$$

In terms of Accuracy, Precision, Recall, F1 score, and TNR, the higher the evaluation metric score is, the better the DL algorithms in terms of that evaluation metric. While in terms of FAR and FNR, the lower the evaluation metric score is, the better the DL algorithm.

5.3 | Experimental setup

For this research, we performed all the experiments on an H.P. Laptop with 8 G.B. of RAM, Intel Core I7-8550U, and NVIDIA GeForce MX150 with a 64-bit Windows 10 operating system. MATLAB 2019a and Python (version 3.6.9) are the tools used to implement the proposed solution. MATLAB 2019a generates a spectrogram of the considered dataset while the Python Deep Learning library Keras library is used as the main programming tool to implement and evaluate the proposed and other DL methodologies in the Google Colab environment with GPU selected as hardware accelerator.[44]

5.4 | Results and discussion

For performing experiments, we updated the dataset by combing all the anomalies into a single anomaly class for binary classification since the first stage of the detection block performs only the initial screening of packets. Also, the dataset is divided into the train and test datasets by a random split ratio of 75% and 25%, respectively. The performance of SCNN is compared with five different supervised DL methods as Deep Neural Network (DNN), one-dimensional Convolutional Neural Network (CNN-1D), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU). The optimum hyperparameters used in this study for implementing all supervised DL-based ADS methodologies are detailed in Table 3.

To find the optimum number of hidden layers for all the DL-based ADS, we perform different experiments to calculate the detection accuracy by considering different batch sizes and the number of hidden layers. All the DL models are trained for 100 epochs to find the appropriate number of hidden layers and batch sizes. We observe the initial trend that our proposed spectrogram-based CNN detection model (SCNN) model achieves a high detection accuracy comparing the other DL models for binary and multiclass classification. Figure 6 shows the average accuracy percentage scores for the SCNN by varying batch sizes by considering different layers for binary and multiclass classification scenarios. The SCNN model exhibited a higher detection accuracy for two layers (one pair of CL and PL and a single FCL), considering the batch size of 256 for binary classification. Similarly, for multiclass classification, four layers of SCNN (Two pairs of CL and PL and two FCL) achieved higher accuracy for the batch size of 256.

Table 4 summarizes the percentage evaluation metrics scores achieved by different DL-based ADS methodologies for binary and multiclass classifications. It is observed that SCNN exhibits superior performance in terms of all the considered evaluation metrics for binary and multiclass classification scenarios. In particular, SCNN exhibited a high detection accuracy of almost 99.90%, with the lowest FAR of 0.09% and 0.03%, respectively, for the binary and multiclass scenarios. Also, SCNN recorded the lowest FNR of 0.10% and 0.09% for both scenarios. We observe that both DNN and CNN-1D performed almost identically for binary/multiclass classification. CNN-1D-based ADS approach performed slightly better in terms of Accuracy, Recall, F1 score, and FNR, while DNN performs slightly better in precision, TNR, and FAR considering binary classification.

TABLE 3 Optimum hyperparameters used for DL algorithms.

Parameter	Value
Learning Rate	0.001
Optimizer	Adam
Loss Function	binary cross-entropy, categorical cross-entropy
Activation	ReLU, Sigmoid, Softmax
Batch size	$\{2^6, 2^7, 2^8, 2^9\}$

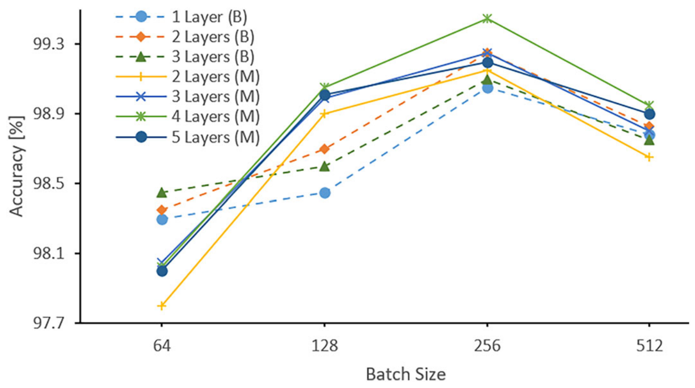

FIGURE 6 SCNN: Accuracy percentage per batch size using different layers for Binary (B) and Multiclass (M) classification scenarios.

TABLE 4 Binary/multiclass classification: Performance evaluation metric score (%).

DL Algorithm	Binary Classification							Multiclass Classification						
	Accuracy	Precision	Recall	F1 score	TNR	FAR	FNR	Accuracy	Precision	Recall	F1 score	TNR	FAR	FNR
DNN	98.16	98.71	97.45	98.07	98.82	1.18	2.56	99.18	99.22	99.26	99.24	99.79	0.21	0.74
CNN-1D	98.34	98.16	98.39	98.27	98.30	1.70	1.61	98.53	98.63	98.66	98.64	99.63	0.38	1.34
RNN	96.17	96.51	95.47	95.99	96.81	3.19	4.53	97.67	97.66	97.80	97.72	99.41	0.59	2.20
LSTM	95.63	93.27	97.97	95.56	93.48	6.52	2.03	95.92	95.95	95.87	95.91	98.97	1.03	4.13
GRU	96.86	96.58	96.88	96.73	96.84	3.16	3.12	97.92	97.95	98.02	97.97	99.47	0.53	1.99
SCNN	99.90	99.91	99.90	99.91	99.91	0.09	0.10	99.90	99.90	99.91	99.91	99.98	0.03	0.09

Similarly, for multiclass classification, CNN-1D performs slightly better in the Accuracy, Recall, precision, and F1 score, while DNN performs slightly better in TNR, FAR, and FNR. It is also noted that RNN, LSTM, and GRU performed almost similarly by exhibiting the lowest evaluation scores. LSTM scored high FAR for binary/multiclass scenarios and FNR for multiclass classification. RNN performed worst in terms of FNR for binary scenarios.

Figure 7 depicts the percentage improvement in the performance of the SCNN comparing other considered DL-based ADS solutions. It is noticed that SCNN performed well by exhibiting an improvement of 1.56%–4.27% in terms of the model's detection accuracy while at the same time reducing the FAR by 1.08%–6.43%. It is also observed that the highest improvement is observed comparing the LSTM-based ADS in terms of detection accuracy and FAR. It is also observed that RNN based approach has a higher miss rate for predicting benign traffic, and SCNN achieved a 4.4% improvement in reducing the miss rate. So, based on the metric evaluation performance, SCNN is chosen as the suitable DL approach to be adopted in substage-1 of the MS-ADS detection stage for the initial screening of the network traffic.

Figure 8 shows the percentage improvement in the performance exhibited by the SCNN comparing the other DL-based detection methodologies. It is observed that SCNN improved by 0.71%–3.98% in the model's detection accuracy and reduced the FAR by 0.18%–1%. It is also observed that the highest improvement is observed by the SCNN comparing the LSTM-based ADS in terms of all the considered performance evaluation metrics. So, based on the evaluation metric performance scores for the multiclass classification scenario, SCNN is chosen as the suitable DL approach to be adopted in substage-2 to classify the anomaly flows into one of the DoS, DDoS, Reconnaissance, or information theft anomalies.

5.5 | Evaluation of MS-ADS

The performance of the MS-ADS is compared with the three benchmark methods LAE-BLSTM,[21] D-RNN,[22] and AE-DNN.[19] For a fair comparison among the MS-ADS with other benchmark methods, we evaluated all the methods

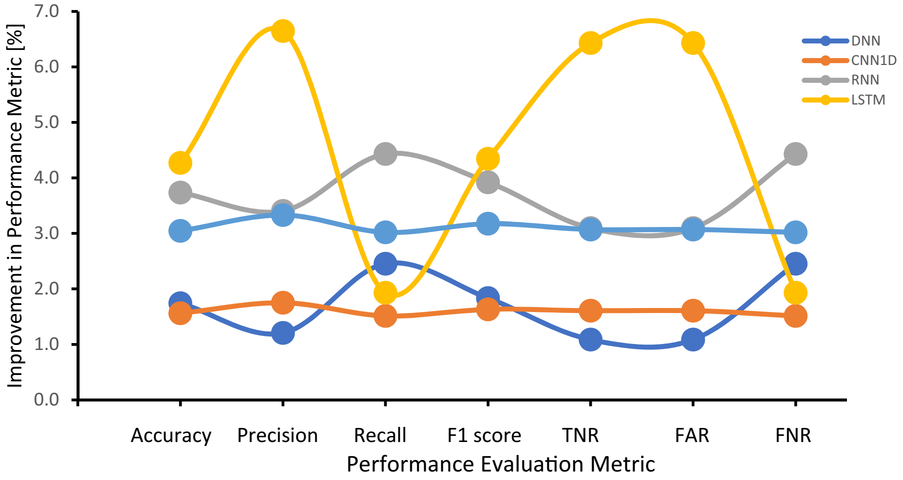

FIGURE 7 Binary classification: SCNN performance improvement comparing other DL-based detection methodologies.

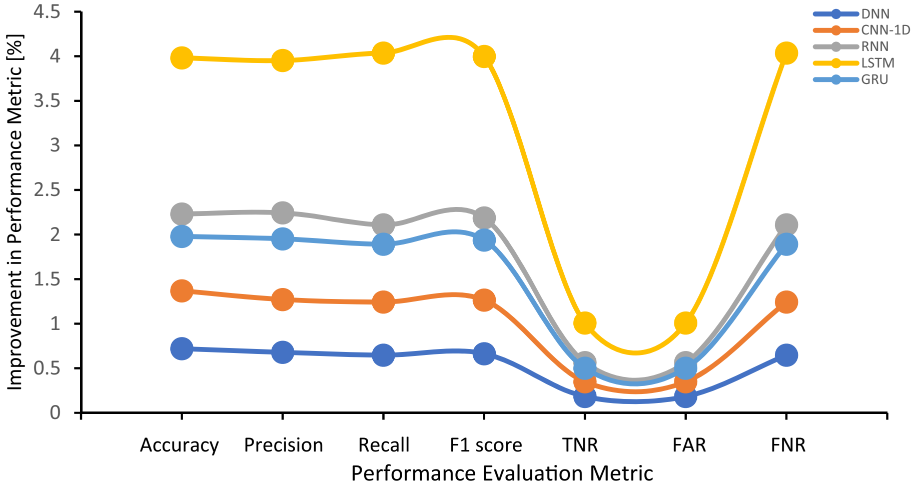

FIGURE 8 Multiclass classification: Improvement in SCNN performance comparing other DL schemes.

under the same common condition, for example, using the same dataset and simulation settings, as it is rather misleading to claim that one technique is better than the other by merely comparing the performance metrics listed in the related original publications. We performed different experiments to find the best possible combination of the hyperparameters, as in Table 5, resulting in higher detection accuracy. It is observed that LAE-BLSTM and D-RNN used eight hidden layers to achieve higher accuracy, while AE-DNN needed 11 hidden layers to achieve higher detection accuracy.

Table 6 summarizes the evaluation metric scores of LAE-BLSTM, D-RNN, AE-DNN, and MS-ADS. It is noted that for the considered scenario, MS-ADS exhibited its superiority in the performance comparing the benchmark methods by achieving a higher evaluation score in terms of all the evaluation metrics considered. It is also noted that among the benchmark methods, AE-DNN exhibited better performance comparing LAE-BLSTM and D-RNN. Also, the performance evaluation scores for the D-RNN are worst comparing all the considered methodologies. It is observed that considering the accuracy metric, 99.98% of the time, MS-ADS correctly predicted the network flows to be either benign or one of the anomalies flows. Also, considering the precision, it is observed that when MS-ADS predicts that the flow is an anomaly flow, the model is correct 99.98% time. MS-ADS also correctly identified the anomaly flows by achieving a Recall score of 99.98%, exhibiting the higher detection rate among all the methods. We also observed that the MS-ADS model achieved a high F1 score of 99.98% to statistically exhibit the model's accuracy. Results also show that almost all the models correctly

TABLE 5 Hyperparameters for benchmarks.

	Methodology			
Hyperparameters	LAE-BLSTM	D-RNN	AE-DNN	MS-ADS
Hidden Layers	8 (LAE = 3, BLSTM = 5)	8 (Level-1 = 3, Level-2 = 5)	11 (AE = 6, DNN = 5)	6 (Substage-1 = 2, Substage-2 = 4)
Learning Rate	0.001	0.001	0.01	0.001
Loss Function	Adam	Adam	Nadam	Adam
Activation	ReLU, Softmax	ReLU, Softmax	ReLU, Softmax	ReLU, Sigmoid, Softmax
Batch size	256	256	512	256

TABLE 6 Performance comparison (%) of MS-ADS w.r.t. benchmark methods.

	Performance evaluation metric						
Methodology	Accuracy	Precision	Recall	F1 score	TNR	FAR	FNR
LAE-BLSTM	98.726	98.874	98.814	98.827	99.673	0.327	1.186
D-RNN	96.208	96.689	95.161	95.788	99.024	0.976	4.839
AE-DNN	99.230	99.188	99.229	99.208	99.806	0.194	0.771
MS-ADS	99.981	99.980	99.980	99.980	99.994	0.006	0.020

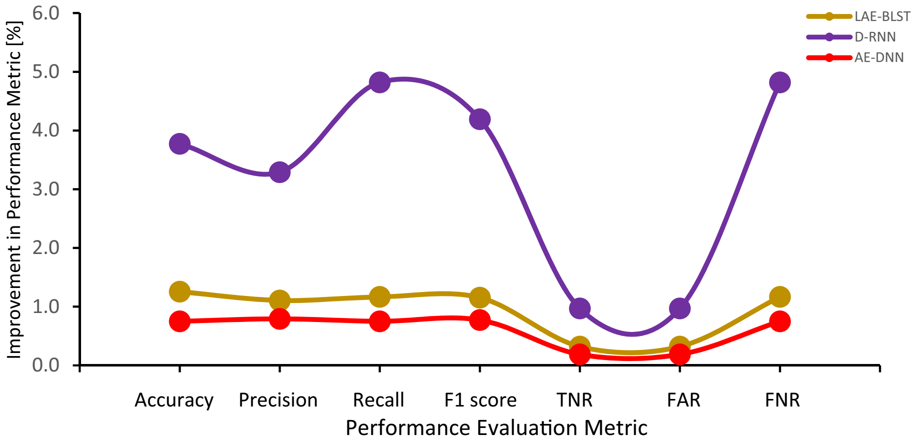

FIGURE 9 Improvement in MS-ADS performance comparing benchmarks.

detected the Benign flows, with MS-ADS achieving the higher score of 99.99%. The MS-ADS also reduced FAR to 0.006% and FNR to 0.02% exhibiting its effectiveness for IoT networks.

The percentage improvement in the metric evaluation performance of MS-ADS against benchmarks is depicted in Figure 9. It is observed that MS-ADS improved the detection accuracy by 0.75%–3.77% while at the same time reducing the FAR by 0.18%–0.96%. The figure also depicts that the MS-ADS improves the detection rate by 0.75%–4.82% exhibiting its efficiency for the IoT network. Also, MS-ADS exhibited a very high improvement in the miss rate by minimizing it by 0.75%–4.82%, which exhibited the proposed solution superiority in reducing the miss rate to classify the anomalies to their specific types correctly.

Figure 10 depicts the confusion matrix for the MS-ADS. Since our proposed solution is a 2-stage DL solution where the first DL substage performs the binary classification task to predict the network flows into benign and anomaly traffic. The anomaly flows are then transferred to the second DL substage to find out the exact type of anomaly. Figure 10A depicts the confusion matrix for the first substage. It is observed that the model predicted both anomaly and benign classes with 99.9% efficiency. The number of benign flows wrongly predicted as an anomaly is more than the anomaly flows wrongly predicted as benign flows. It is observed that the number of FN flows is less than the FP flows, so it causes little threat to

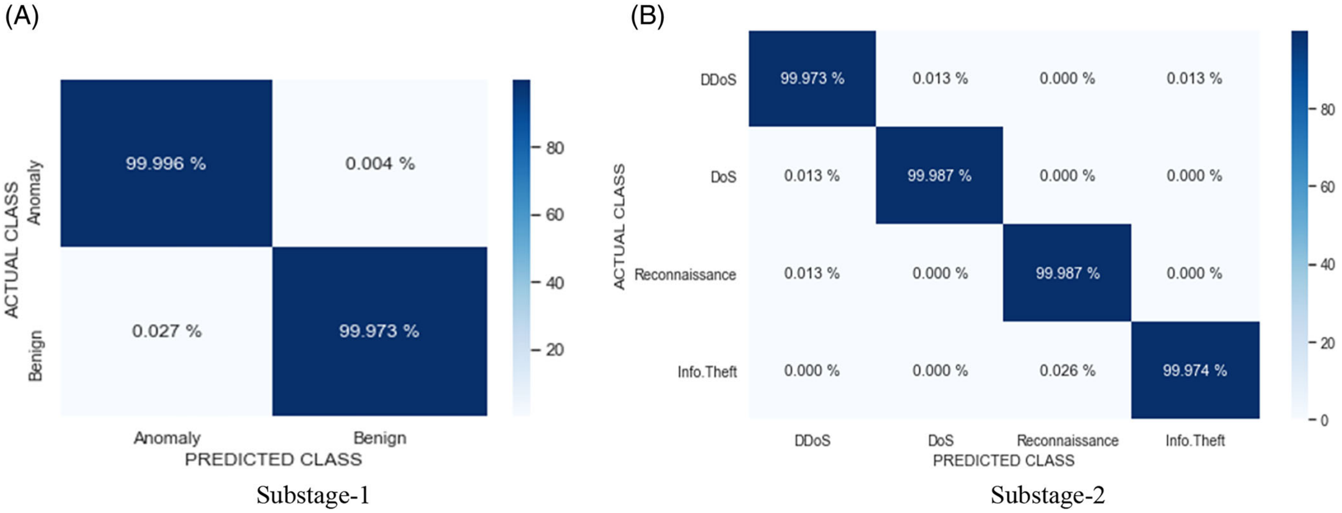

FIGURE 10 Confusion Matrix (MS-ADS).

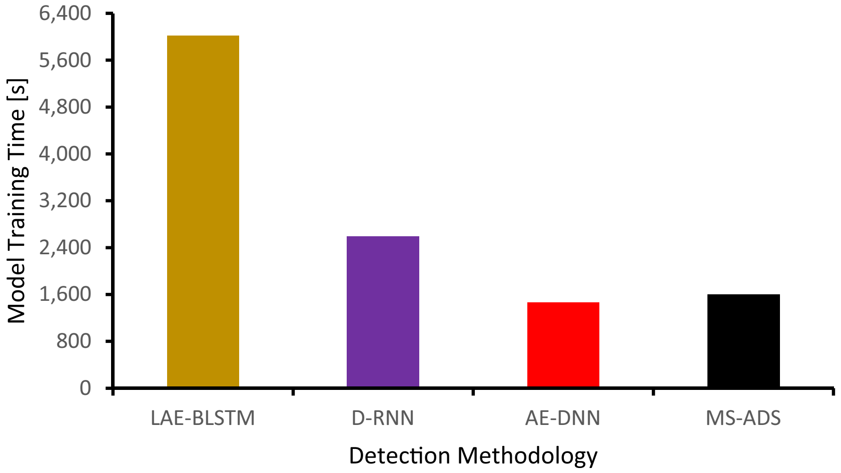

FIGURE 11 Training time (s) for DL-based detection methodologies.

the IoT network. Figure 10B shows the confusion matrix of substage-2, which only classifies the transferred anomalies as DDoS, DoS, Reconnaissance, or Information theft anomalies. Again, the confusion matrix shows that the model correctly predicted the specific type of anomaly with 99.9% efficiency. Also, the model's wrong prediction of any individual class spans over the remaining classes.

Training time is the time needed to build the DL model by training it using the training samples. Training time for the model depends on many factors, for example, type of the dataset, size of the dataset, type of the DL algorithm, no. of hidden layers, no. of epochs, batch sizes, etc. For this study, we measured the model's training time for each methodology considering 100 epochs. Figure 11 depicts the training time in seconds for MS-ADS and the benchmarks. It is observed that AE-DNN is the more efficient model in terms of training time to generate the trained model. It is observed that the proposed MS-ADS performed well in training comparing D-RNN and LAE-BLSTM models by finishing the training early by 993 and 4423 s, respectively. Comparing the efficient model AE-DNN, the MS-ADS took 130 s (2.16 min) more for training, which is a reasonable training time due to the presence of two similar types of the DL methodology in a multistage setting for MS-ADS. Also, this slight increase in training time results in better detection accuracy than AE-DNN exhibiting an improvement of 0.75%.

Table 7 details the training time in seconds considering 100 epochs to achieve 95%–99.5% accuracy. It is observed that the proposed MS-ADS methodology outperforms the benchmark detection methodologies in training time to achieve

TABLE 7 Training time (s) to achieve different accuracies.

	Training time (s) to achieve accuracy (95%–99.5%)					
Methodology	**95%**	**96%**	**97%**	**98%**	**99%**	**99.5%**
LAE-BLSTM	301.17	481.76	903.30	2107.70	–	–
D-RNN	259.21	1555.20	–	–	–	–
AE-DNN	29.38	44.07	73.45	367.25	1028.30	–
MS-ADS	15.9	31.98	47.97	63.96	111.93	367.99

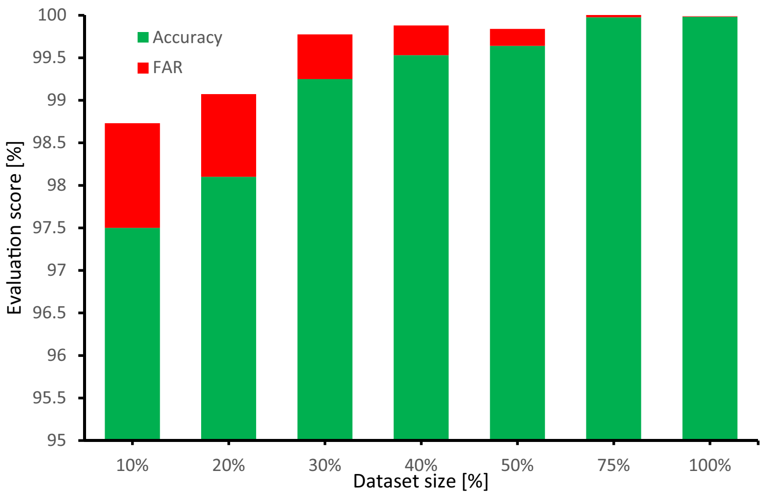

FIGURE 12 MS-ADS: Dataset size effect on the performance evaluation.

TABLE 8 Average training time (s) w.r.t. dataset size.

Dataset size (%)	**10%**	**20%**	**30%**	**40%**	**50%**	**75%**	**100%**
Training Time (s)	231.6	459.7	423.0	529.7	729.7	981.3	1598.7

accuracy in the range of 95%–99.5%. Only MS-ADS is observed to achieve more than 99.5% accuracy among all the detection. Also, D-RNN detection accuracy was the lowest, with 96% accuracy achieved in 1555.2 s. D-RNN can achieve 96.21% accuracy in 2592 s considering 100 epochs. It is also observed that LAE-BLSTM is the more time taking methodology during the training process by consuming more time in training to get the desired accuracy comparing the other detection methodologies.

The dataset is an important factor considering the DL model training, which requires a regular update to keep the model up to date with the latest anomaly patterns. We want to find a more feasible dataset size for re-training MS-ADS in this context. Figure 12 illustrates the feasible size of the dataset by comparing it with the two important evaluation metrics such as detection accuracy and FAR. It is obvious from the figure that detection accuracy improved with the increase in the dataset size. Also, the FAR has shown a reasonable reduction with increased dataset size. We observe that by only using roughly 40% of the dataset, we still can achieve the detection accuracy of 99.53% with the FAR of 0.35%, which makes it a reasonable choice of dataset size for the re-training to update the model with the new patterns to make it more effective and efficient for anomaly detection.

We observe from Table 8 that the training time increases as the size of the complete dataset increases. If we consider using only 40% of the dataset size (40 500 flows), then we only need to train the model for 529.7 s (8.8 min), which is an improvement of 17.8 min comparing training the whole dataset. Training of the model can be done in the off-time mode, and then the trained model can be replaced with the old model to use an updated model for detection purposes.

6 | CONCLUSIONS

This research has developed a Multistage Spectrogram image-based Anomaly Detection System (MS-ADS) to ensure the security of the IoT network from DoS, DDoS, Reconnaissance, and Information theft anomalies. The proposed solution utilizes CNN as the DL algorithm in the detection module and uses two variants of CNN arranged in sequence to perform the anomaly prediction tasks on multiple levels. The CNN models are trained using spectrogram images, generated from the IoT network flows using STFT to improve detection accuracy and reduce FAR and FNR. Also, the usage of the second-level CNN algorithm only if an anomaly is detected minimizes the overall overhead on the system in terms of computation. Moreover, the continuous learning module of the MS-ADS improves the model's effectiveness in accurately detecting new anomalies to minimize the FAR and FNR by periodically updating the model with the latest anomaly patterns. The results demonstrate that the proposed method achieves high detection accuracy of 99.98% with a reduction in the FAR to 0.006% and FNR to 0.020% in classifying network traffic showing the model efficiency for the IoT network. MS-ADS outperforms the benchmark methodologies by improving the detection accuracy of AE-DNN by 0.75%, LAE-BLSTM by 1.25%, and D-RNN by 3.77%. Also, MS-ADS reduced the FAR and FNR for AE-DNN by 0.18% and 0.75%, LAE-BLSTM by 0.32% and 1.16%, and D-RNN by 0.96% and 4.82%. Also, MS-ADS improves the detection rate of AE-DNN by 0.75% and D-RNN by 4.82%, exhibiting its efficiency for the IoT network. To minimize the computational and training cost for the model re-training phase, the proposed solution demonstrates that only 40 500 network flows from the dataset suffice to achieve a detection accuracy of 99.5%.

For future work, we will extend this research by testing its performance and effectiveness by implementing it in the real-time IoT environment. We will also extend our proposed idea for the unsupervised DL-based methods to make the proposed solution more effective in automatically processing the huge amount of IoT data.

ACKNOWLEDGMENTS

This research was fully funded by Universiti Malaysia Sarawak, Kota Samarahan, Malaysia under grant number F08/PGRG/2058/2021.

CONFLICT OF INTEREST STATEMENT

All the authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Zeeshan Ahmad https://orcid.org/0000-0002-8530-864X

REFERENCES

1. Ahmad F, Ahmad Z, Kerrache CA, Kurugollu F, Adnane A, Barka E. Blockchain in internet-of-things: architecture, applications and research directions. *2019 Int Conf Comput Inf Sci ICCIS 2019*. IEEE. Published online May 15; 2019. doi:10.1109/ICCISCI.2019.8716450
2. Ahmad Z, Khan AS, Nisar K, et al. Anomaly detection using deep neural network for IoT architecture. *Appl Sci*. 2021;11(15):7050. doi:10.3390/APP11157050
3. Asim J, Khan AS, Saqib RM, et al. Blockchain-based multifactor authentication for future 6G cellular networks: A systematic review. *Appl Sci*. 2022;12:3551. doi:10.3390/app12073551
4. Chaabouni N, Mosbah M, Zemmari A, Sauvignac C, Faruki P. Network intrusion detection for IoT security based on learning techniques. *IEEE Commun Surv Tutorials*. 2019;21(3):2671-2701. doi:10.1109/COMST.2019.2896380
5. Luo E, Bhuiyan MZA, Wang G, Rahman MA, Wu J, Atiquzzaman M. PrivacyProtector: privacy-protected patient data collection in IoT-based healthcare systems. *IEEE Commun Mag*. 2018;56(2):163-168. doi:10.1109/MCOM.2018.1700364
6. Lu N, Cheng N, Zhang N, Shen X, Mark JW. Connected vehicles: solutions and challenges. *IEEE Internet Things J*. 2014;1(4):289-299. doi:10.1109/JIOT.2014.2327587
7. Arshad J, Azad MA, Amad R, Salah K, Alazab M, Iqbal R. A review of performance, energy and privacy of intrusion detection systems for IoT. *Electron*. 2020;9(4):629. doi:10.3390/ELECTRONICS9040629
8. Panigrahi R, Borah S. Dual-stage intrusion detection for class imbalance scenarios. *Comput Fraud Secur*. 2019;2019(12):12-19. doi:10.1016/S1361-3723(19)30128-9
9. Habeeb MS, Babu TR. Network intrusion detection system: a survey on artificial intelligence-based techniques. *Expert Syst* Published online. 2022;39:e13066. doi:10.1111/EXSY.13066

10. Verwoerd T, Hunt R. Intrusion detection techniques and approaches. *Comput Commun*. 2002;25(15):1356-1365. doi:10.1016/S0140-3664(02)00037-3
11. Debar H, Dacier M, Wespi A. Towards a taxonomy of intrusion-detection systems. *Comput Networks*. 1999;31(8):805-822. doi:10.1016/S1389-1286(98)00017-6
12. Khan AS, Javed Y, Saqib RM, et al. Lightweight multifactor authentication scheme for nextgen cellular networks. *IEEE Access*. 2022;10:31273-31288.
13. Maikol SO, Khan AS, Javed Y, et al. A novel authentication and key agreement scheme for countering MITM and impersonation attack in medical facilities. *Int J Integr Eng*. 2020;13(2):127-135. doi:10.30880/ijie.2021.13.02.015
14. Aqeel S, Khan AS, Ahmad Z, Abdullah J. A comprehensive study on DNA based Security scheme Using Deep Learning in Healthcare. 2021. doi:10.1080/07366981.2021.1958742
15. Najafabadi MM, Villanustre F, Khoshgoftaar TM, Seliya N, Wald R, Muharemagic E. Deep learning applications and challenges in big data analytics. *J Big Data*. 2015;2(1):1-21. doi:10.1186/S40537-014-0007-7
16. Khan AS, Ahmad Z, Abdullah J, Ahmad F. A spectrogram image-based network anomaly detection system using deep convolutional neural network. *IEEE Access*. 2021;9:87079-87093. doi:10.1109/ACCESS.2021.3088149
17. Ahmad Z, Khan AS, Aqeel S, et al. S-ADS: spectrogram image-based anomaly detection system for IoT networks. In *Proc – Applied Informatics International Conference (AiIC 2022)*, Serdang, Malaysia; 2022, pp. 105-110.
18. Khan AS, Javed Y, Abdullah J, Zen K. Trust-based lightweight security protocol for device to device multihop cellular communication (TLwS). *J Ambient Intell Humaniz Comput*. 2021;1:1-18. doi:10.1007/S12652-021-02968-6/TABLES/5
19. Aversano L, Bernardi ML, Cimitile M, Pecori R, Veltri L. Effective anomaly detection using deep learning in IoT systems. *Wirel Commun Mob Comput*. 2021;2021:1-14. doi:10.1155/2021/9054336
20. Ma W. Analysis of anomaly detection method for internet of things based on deep learning. *Trans Emerg Telecommun Technol*. 2020;31(12):e3893. doi:10.1002/ETT.3893
21. Popoola SI, Adebisi B, Hammoudeh M, Gui G, Gacanin H. Hybrid deep learning for botnet attack detection in the internet-of-things networks. *IEEE Internet Things J*. 2021;8(6):4944-4956. doi:10.1109/JIOT.2020.3034156
22. Almiani M, AbuGhazleh A, Al-Rahayfeh A, Atiewi S, Razaque A. Deep recurrent neural network for IoT intrusion detection system. *Simul Model Pract Theory*. 2020;101:102031. doi:10.1016/J.SIMPAT.2019.102031
23. Diro AA, Chilamkurti N. Distributed attack detection scheme using deep learning approach for internet of things. *Futur Gener Comput Syst*. 2018;82:761-768. doi:10.1016/J.FUTURE.2017.08.043
24. Thamilarasu G, Chawla S. Towards deep-learning-driven intrusion detection for the internet of things. *Sensors*. 2019;19(9):1977. doi:10.3390/S19091977
25. Khan AS, Sattar MA, Nisar K, et al. A survey on 6G enabled light weight authentication protocol for UAVs, security, open research issues and future directions. *Appl Sci*. 2023;13:277. doi:10.3390/app13010277
26. Bostani H, Sheikhan M. Hybrid of anomaly-based and specification-based IDS for internet of things using unsupervised OPF based on MapReduce approach. *Comput Commun*. 2017;98:52-71. doi:10.1016/J.COMCOM.2016.12.001
27. Reddy DKK, Behera HS, Nayak J, Vijayakumar P, Naik B, Singh PK. Deep neural network-based anomaly detection in internet of things network traffic tracking for the applications of future smart cities. *Trans Emerg Telecommun Technol*. 2021;32(7):e4121. doi:10.1002/ETT.4121
28. Goodfellow I, Bengio Y, Courville A. *Deep Learning*. MIT Press; 2016 https://www.deeplearningbook.org/
29. Khan AS, Balan K, Javed Y, Abdullah J, Tarmizi S. Secure trust-based blockchain architecture to prevent attacks in VANET. *Sensors*. 2019;19(22):4954. doi:10.3390/S19224954
30. Zhu Y, Yin X, Hu J. Robust fingerprint matching based on convolutional neural networks. *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*. Vol 235. Springer Verlag; 2018:56-65. doi:10.1007/978-3-319-90775-8_5
31. Kowsari K, Sali R, Ehsan L, et al. HMIC: hierarchical medical image classification, a deep learning approach. *Information*. 2020;11(6):318. doi:10.3390/INFO11060318
32. Azab A, Khasawneh M. MSIC: malware spectrogram image classification. *IEEE Access*. 2020;8:102007-102021. doi:10.1109/ACCESS.2020.2999320
33. Satt A, Rozenberg S, Hoory R. Efficient Emotion Recognition from Speech Using Deep Learning on Spectrograms. Published online 2017. doi:10.21437/Interspeech.2017-200
34. Huang J, Chen B, Yao B, He W. ECG arrhythmia classification using STFT-based spectrogram and convolutional neural network. *IEEE Access*. 2019;7:92871-92880. doi:10.1109/ACCESS.2019.2928017
35. Yuan L, Cao J. Patients' EEG data analysis via spectrogram image with a convolution neural network. *Smart Innovation, Systems and Technologies*. Vol 72. Springer Science and Business Media Deutschland GmbH; 2018:13-21. doi:10.1007/978-3-319-59421-7_2
36. Saqib RM, Khan AS, Javed Y, et al. Analysis and intellectual structure of the multi-factor authentication in information security. *Intell. Autom. Soft Comput*. 2022;32:1633-1647.
37. Khraisat A, Gondal I, Vamplew P, Kamruzzaman J, Alazab A. A novel Ensemble of Hybrid Intrusion Detection System for detecting internet of things attacks. *Electron*. 2019;8(11):1210. doi:10.3390/ELECTRONICS8111210
38. Ullah I, Mahmoud QH. A two-level flow-based anomalous activity detection system for IoT networks. *Electron*. 2020;9(3):530. doi:10.3390/ELECTRONICS9030530

39. Hoque N, Bhuyan MH, Baishya RC, Bhattacharyya DK, Kalita JK. Network attacks: taxonomy, tools and systems. *J Netw Comput Appl*. 2014;40(1):307-324. doi:10.1016/j.jnca.2013.08.001
40. Chaabouni N. Intrusion detection and prevention for IoT systems using Machine Learning. Published online 2020.
41. Koroniotis N, Moustafa N, Sitnikova E, Turnbull B. Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: bot-IoT dataset. *Futur Gener Comput Syst*. 2019;100:779-796. doi:10.1016/J.FUTURE.2019.05.041
42. Ullah I, Mahmoud QH. A technique for generating a botnet dataset for anomalous activity detection in IoT networks. Paper presented at: Conf Proc – IEEE Int Conf Syst Man Cybern. 2020; 2020 October, pp. 134–140. doi:10.1109/SMC42975.2020.9283220
43. Ahmad Z, Khan AS, Shiang CW, Abdullah J, Ahmad F. Network intrusion detection system: a systematic study of machine learning and deep learning approaches. *Trans Emerg Telecommun Technol*. 2021;32(1):e4150. doi:10.1002/ett.4150
44. Bisong E, Bisong E. Google Colaboratory. *Building Machine Learning and Deep Learning Models on Google Cloud Platform*. Apress; 2019:59-64. doi:10.1007/978-1-4842-4470-8_7

How to cite this article: Ahmad Z, Khan AS, Zen K, Ahmad F. MS-ADS: Multistage Spectrogram image-based Anomaly Detection System for IoT security. *Trans Emerging Tel Tech*. 2023;34(8):e4810. doi: 10.1002/ett.4810