IoT

MDPI

Article

Towards a Hybrid Deep Learning Model for Anomalous Activities Detection in Internet of Things Networks

Imtiaz Ullah [1,*], Ayaz Ullah [2] and Mazhar Sajjad [3]

[1] Department of Electrical, Computer, and Software Engineering, Ontario Tech University, Oshawa, ON L1G 0C5, Canada

[2] Department of Computer Science, University of Swabi, Swabi 23430, Pakistan; ayazsb@uoswabi.edu.pk

[3] Department of Computer Science, Comsats University, Islamabad 45550, Pakistan; mazhar.sajjad@comsats.edu.pk

* Correspondence: imtiaz.ullah@ontariotechu.net

Abstract: The tremendous number of Internet of Things (IoT) applications, with their ubiquity, has provided us with unprecedented productivity and simplified our daily life. At the same time, the insecurity of these technologies ensures that our daily lives are surrounded by vulnerable computers, allowing for the launch of multiple attacks via large-scale botnets through the IoT. These attacks have been successful in achieving their heinous objectives. A strong identification strategy is essential to keep devices secured. This paper proposes and implements a model for anomaly-based intrusion detection in IoT networks that uses a convolutional neural network (CNN) and gated recurrent unit (GRU) to detect and classify binary and multiclass IoT network data. The proposed model is validated using the BoT-IoT, IoT Network Intrusion, MQTT-IoT-IDS2020, and IoT-23 intrusion detection datasets. Our proposed binary and multiclass classification model achieved an exceptionally high level of accuracy, precision, recall, and F1 score.

Keywords: anomaly detection; deep learning; convolutional neural network; recurrent neural network; gated recurrent unit; Internet of Things; machine learning; network security

check for updates

Citation: Ullah, I.; Ullah, A.; Sajjad, M. Towards a Hybrid Deep Learning Model for Anomalous Activities Detection in Internet of Things Networks. *IoT* **2021**, *2*, 428–448. https://doi.org/10.3390/iot2030022

Academic Editors: Carlo Giannelli, Marco Picone and Hyun-Ho Choi

Received: 12 May 2021
Accepted: 23 July 2021
Published: 27 July 2021

1. Introduction

The evolution of the IoT network infrastructure has influenced the increasing number of embedded devices and intelligent applications. The IoT objective is to build intelligent environments capable of improving human life quality, comfort, and competitiveness. Devices in smart architectures communicate with one another to execute different tasks. IoT-enabled systems have been used in manufacturing settings as well as for a variety of commercial uses. These intelligent systems include a broad spectrum of capabilities, from smart houses to smart cities, intelligent buildings, and other intelligent utilities such as factory automation and management, power generation networks, and transportation [1]. The IoT raises several challenges, including privacy and security. The security challenges associated with the IoT will grow as it develops and progresses over the next several years [2], which further raises the possibility of cyberattacks. Cybercriminals attacked IoT devices in 2020, according to recent reports, indicating a significant rise in IoT vulnerabilities on wireless networks [3]. There would be more substantial motivations and desire for attackers to discover innovative and creative ways to hack IoT applications due to the increased rewards for effective IoT breaches.

Traditional approaches and strategies used in the conventional Internet to safeguard against cyberattacks prove ineffective in defending against the specific weaknesses discovered on the Internet. Security issues in a network are managed by three general methods: prevention, detection, and mitigation. All three steps will have to be taken to ensure effective mitigation strategies for IoT networks. Cybersecurity is an essential component of the information technology system of today's IoT world. Although the IoT improves performance and competitiveness by smart control, it also raises cyberattacks. The IoT privacy security paradigm is important in today's new technologies. The increased diversity of IoT systems in the market shows that the industry is making strides toward revolutionizing IoT architecture. As a result, the specifications governing IoT system connectivity are complex, requiring the development of a unified system to facilitate device communication. The growing range of IoT devices designed for different applications means that IoT manufacturers constantly develop IoT technology and reduce their time to market their products. Customers have benefited from these technologies on one side, while critical facilities have successfully incorporated IoT devices to implement their operations. Apart from requiring significant security enhancements, the IoT exposes users' details to cyberattacks. Although IoT technologies have helped humanity in a variety of respects, they still have several flaws. Despite the fact that many security protocols have been implemented to defend IoT devices from cyber threats, security guidelines are not well established [4]. More than 85% of companies worldwide would switch to IoT devices in one way or another, and 90% of these companies do not know about IoT device security [5]. A new HP report has also found that 70% of Internet-connected devices are susceptible to multiple attacks [1]. Additionally, active launches of cyberattacks such as Mirai [6], Shamoon-2 [7], and Ransomlike [8] attacks on critical infrastructure indicate that current IoT security measures have been inefficient.

1.1. Motivation

Anomaly identification techniques have been the primary source of motivation for many researchers due to their capacity to identify new threats. Massive IoT devices are integrated into our daily lives in many ways. They are commonly employed in several sectors, such as healthcare, manufacturing, delivery, road traffic management, city life safety, shopping, sustainability, city protection, smart communities, transportation, waste management, smart street lighting, traffic signs, and vehicle networks [9]. Designing a fully safe device is impossible because there is no such thing as total security; humans can make mistakes, most current networks contain security vulnerabilities, insider misuse is common, and all types of intrusions remain unknown. At present, attackers use sophisticated techniques to execute increasingly serious attacks efficiently with little technical knowledge of the network. Currently, a practical attack has a significant impact on IoT infrastructure, while the time taken to carry out these attacks continues to decrease. IoT systems have evolved into an attractive goal for attackers looking to launch destructive attacks, and the threat surface of IoT networks will continue to increase as a result of their rapid growth. The most difficult challenge, which is embedded in expanding the network's stability, is cybersecurity.

Classification of malicious activity in big data is becoming more complex in IoT networks. An important factor for protecting IoT networks is an intrusion detection system (IDS). Anomaly-based and misused-based approaches in the field of intrusion detection research are generally focused and inspired detection methodologies. IDS is now a crucial part of protecting complex IoT networks. IDS can detect malicious activity or protocol failure in an operating system or network. IDS can be divided into a centralized intrusion detection system (CIDS) and a distributed intrusion detection system (DIDS). Data analysis may be done at a single location in CIDS, while DIDS consists of multiple IDSs at various locations where the data analysis is conducted. Security companies have had to develop IoT defense strategies alongside existing ones in response to the popularity of the Internet. In addition to using standard network-based intrusion detection techniques, several alternative tools are provided for those investigating intrusions or identifying network threats. Machine learning has shown to be both essential and useful in detecting cyberattacks in real-time. Various mechanisms can be used to generate knowledge from collected and analyzed data. Supervised learning uses the current anomalous data to define the inconsistency to a reference point. In contrast, unsupervised learning determines the anomalous activity by inferential learning to draw a decision based on identified evidence. Popular machine learning techniques are support vector machine, Markov-based, grammar-based, neural network, change detection, Bayesian, decision tree, nearest prototype, hierarchical, and outlier-based identification techniques for classification and clustering [10].

1.2. Contribution

Machine learning advances have resulted in new solutions, and they can adapt to changes in the environment by continuous learning. Although machine learning is increasingly being applied to IoT intrusion detection, it has some shortcomings, which should be considered. Firstly, comparatively single types of intrusions are analyzed, and diverse types of attacks are not considered. Second, it is incredibly time-consuming to identify effective features for training the machine learning algorithm during the data processing stage. Extracting a huge number of features consumes many resources. As a result, a lightweight method for extracting the relevant and limited number of features for machine-learning-based detection of various IoT attacks is needed. To address these issues, first, we select the four most recent intrusion detection datasets' pcap files and generate adapted datasets using CICFlowmeter [11]. Next, we combine these datasets into large attack classes. Our proposed, adopted datasets have 80 network and flow features. We select the 48 best features using the recursive features elimination (RFE) approach. Although multiple techniques have been developed to detect anomalies, CNN and GRU networks for classifying attacks have received much fewer attempts. The model proposed in this paper is an extension of a model employing convolutional neural networks that we previously proposed [12], in which we used three different models of convolutional neural networks. In this paper, we propose a deep learning model based on CNN and GRU for binary and multiclass class anomaly detection and classification in IoT networks. The proposed model uses an input layer, two convolutional layers, two GRU layers, a flatten layer, a dense layer, and an output layer. The proposed scheme classifies 15 different attacks using a convolutional and GRU-based neural network model in the multiclass classifier, essentially separating them from normal network traffic. A novel anomaly-based intrusion detection system for IoT networks using convolutional and gated recurrent unit neural networks is presented in this article. Our proposed binary classification model achieved an accuracy of 99.96%, while our proposed multiclass classification model achieved an accuracy of 99.92%.

Artificial intelligence has made tremendous improvements in closing the distance between human and computer capabilities. A CNN is a deep neural network that can take in an image as data, allocate significance to various aspects of the image, and distinguish one from the other. In comparison to other classification algorithms, a CNN needs significantly less preprocessing [13]. The spatial and temporal properties can be successfully captured by the CNN using related filters in an image. A CNN consists of several artificial neuronal layers. Each neuron has its weights to determine its action. CNNs are fed with the image values and find different features in them. A CNN is typically made up of many convolution layers, but it may also include other elements. Convolution is the initial layer in which the characteristics of an image are extracted. Convolution retains a link between pixels via the use of tiny input squares to learn picture attributes. If the images are too large, the pooling layers section reduces the number of parameters. Spatial pooling, also known as subsampling or downsampling, is a technique for reducing the dimension of each map while retaining critical information. Max pooling, average pooling, and sum pooling are three types of spatial pooling. A fully connected layer flattens the matrix into a vector and feeds it into a fully connected layer, similar to a neural network. A classification layer is the final layer of a CNN, and it uses the output of the previous convolution layer as input. The classification layer generates a series of confidence values (score between 0 and 1) based on the activation function of the final convolution layer, which indicates how probable the object is to correspond to a "class" [14]. Short-term memory is a limitation in recurrent neural networks (RNN). If a series is too long, it would have difficulty transporting knowledge from earlier to subsequent time steps. RNNs encounter the vanishing gradient problem during backpropagation [15]. Gradients are the variables that are used to change the weights of a neural network. When a gradient value becomes quite minimal, it does not add significantly to learning. Therefore, layers obtaining a small gradient update struggle to learn in recurrent neural networks. As a result of the lack of learning in these layers, RNNs will overlook what they saw in longer sequences, resulting in short-term memory. GRUs have been developed to address the issue of short-term memory. The GRU is a more recent generation of RNN that is very close to the LSTM. The GRU is equipped with two gates: a reset and an update, while the LSTM has three gates [16,17]. The update gate operates similarly to an LSTM's forget and input gates. The update gate assists the model in determining how much past data should be transmitted to the future. It determines what data to discard and what new data to use. The reset gate determines how much previous knowledge is to be forgotten. Since GRUs have fewer tensor operations, they are slightly faster to train than LSTMs [15].

The rest of the paper proceeds as follows: in Section 2, the related work is presented. The proposed model, data collection, and preprocessing dataset are discussed in Section 3. The analysis of the results is presented in Section 4. Finally, Section 5 concludes the paper and offers ideas for future work.

2. Related Work

Internet infrastructure is increasingly evolving because of advances in computing technology. However, we have seen issues such as vulnerabilities because of these advances. Kim et al. [18] use a convolutional neural network to classify malicious traffic using packet size and arrival time. They achieved an accuracy of 95%, which is considered very low for modern IoT networks. Internet traffic is increasing at an exponential rate, with daily data generation ranging from zettabytes to petabytes. Along with this increase in use, security risks to networks, the web, databases, and organizations are increasing. Hassan et al. [19] suggest a hybrid deep learning model for effectively detecting network intrusions focused on the convolutional neural network and a long-term memory network. They used a deep convolutional neural network to extract important features from IDS big data and LSTM characteristics to keep longer-term correlations between derived features to avoid the overfit on recurring connections. The accuracy was measured at 97.10%, which is insufficient for today's IoT networks. The advancement in information technology and economic progress have also accelerated the IoT industry. IoT networks are susceptible to attacks due to the limited infrastructure available to sensor nodes, the difficulty of networking, and the free wireless broadcast transmission characteristics. Li et al. [20] suggest an algorithm for extracting IoT features and detecting intrusions in a smart city built on a deep migration learning paradigm and incorporate deep learning and intrusion detection technologies. Their proposed model has a faster detection time and a better detection performance than previous models. IoT technologies for smart cities have risen to prominence as a primary focus for threats such as botnets. Vinayakumar et al. [21] suggest a botnet identification algorithm built on a two-tier deep learning architecture for semantically distinguishing botnets from valid activities at the domain name system implementation layer. Due to its potential architecture for analyzing the domain name system, the results are highly portable on heterogeneous computing servers.

Intrusion detection and prevention mechanisms continue to be the primary line of protection against severe threats. Kaur et al. [22] suggest an image-based deep neural network model for classifying various attacks using two large datasets, CICIDS2017 and CSE-CICIDS2018. They also provide a selection of the best network flow features for detecting these attacks. Their convolutional neural network model produced poor performance for some attack categories. Most cybercriminals currently use encrypted communication networks to shield malicious activities and mimic legitimate user activity. These threats over a protected channel raise the vulnerability of interconnected networks to emerging threats and the risk of significant harm to many other end users. Ullah and Mahmoud [23,24] proposed a two-level intrusion detection system for IoT networks. The level-1 model categorizes network traffic as regular or irregular, while the level-2 model categorizes observed malicious behavior by category or subcategory. Their model precision, recall, and F1 score are 99.90% for levels 1 and 2. Yang and Lim [25] present a novel deep-learning-based approach for detecting malicious SSL traffic. The suggested method extracts the unencrypted contents of the reconfigured record and produces a series of unencrypted data from successive SSL records for classification using deep learning. A long short-term memory encoder generates SSL sequences and uses them to build an encoded feature map for each flow.

These feature maps are forwarded to the convolutional network classifier to see if the SSL is abnormal or not. The massive number of IoT devices and their pervasive nature have drawn hackers looking to perform cyberattacks and data breaches. Ran et al. [26] propose a framework for intrusion detection focused on bidirectional long short-term memory stacks (LSTM). They used the KDD99 dataset for model evaluation. Their model achieved 91.6% accuracy. The authors did not use an early stopping strategy, which may cause overfitting of the model. Ahmad and Alsemmeari [27] proposed the extreme learning machine (ELM) approach to enhance intrusion detection. The authors examine, investigate, and apply well-known activation functions such as sine, sigmoid, and radial basis to quantify their success on the GA (genetic algorithm) features subset and the full features set. Their findings indicate that the radial base and sine functions work stronger on the GA feature set than the complete feature set. In contrast, the sigmoid function performs almost identically on both feature sets. GA-based function selection achieved 98% accuracy and improved the overall performance of the intrusion detection extreme learning machine.

Ling et al. [28] developed a bidirectionally simple recurrent unit-based intrusion detection system. Their suggested approach is more precise and needs far less training time than alternative approaches. Kunang et al. [29] used a pretraining strategy with a deep autoencoder and a deep neural network to build a deep learning intrusion detection framework. An automatic hyperparameter optimization method helps determine hyperparameter importance and the best categorical hyperparameter configuration to improve detection accuracy. Additionally, the performance outcomes exceed prior techniques in terms of multiclass evaluation criteria. A convolutional neural network model was used to develop an intrusion detection system that interprets network activity data as character sequences. The input matrices of the convolutional networks are united to form a complex matrix structure to perform image classification. Their model worked well on training data but performed poorly on testing data. To build an attack prevention mechanism and a secure network, Sicato et al. [30] offer a comprehensive summary of emerging intrusion detection systems for IoT environments, address cyber-security risks, and evaluate and analyze transparent issues and concerns. They suggest a distributed cloud infrastructure built on software-defined IDS for securing the Internet of Things. Standard intrusion prevention techniques based on rules are insufficient to handle the highly dynamic network intrusion traffic. However, the potential of an intrusion detection system based on a traditional machine learning approach to generalize is still limited, and the false alarm rate is strong. Wu et al. [31] suggest SRDLM, a modern intrusion detection approach focused on deep learning and semantic reencoding. Their suggested model approach reencodes the syntax of network activity, improves the traffic's distinguishability, and strengthens the system's generalizability through deep learning technology, significantly increasing the system's efficiency and performance. They used the NSL-KDD data set, and the average score was improved by more than 8% compared to the standard machine learning approach. Sheu et al. [32] use a reinforcement learning algorithm to design a system for identifying the opening time and load restrictions of electrical equipment. The suggested systems are based on a wireless communication network. They can monitor the energy consumption of home appliances, control smart appliances, and reduce the rate of fires caused by electrical appliance overload. Wireless sensor networks are vulnerable to hostile activity because of their security limits. Tariq [33] has designed an anticipatory and proactive mechanism to predict host and grid anomalies. The proposed anomaly identification system's architecture has been widely distributed to provide an accessible and adaptive technique for avoiding a single point of failure. Table 1, adapted from [12], provides an overview of the related literature that was evaluated. In Table 1, DR represents detection rate, Acc means accuracy, Pr represents precision and F1 represent F1 score.

Table 1. Overview of related literature.

Article	Dataset	Classification	Model	Year	Performance
[34]	AWID	Binary	D-FES	2017	Acc = 99.90
[35]	NSL-KDD	Multiclass	CNN	2018	Acc = 85.07
[36]	NSL-KDD	Binary	DNN	2018	Acc = 99.29
[37]	AWID	Binary	LSTM	2018	Acc = 98.22
[38]	UNSW-NB15	Multiclass	Autoencoder	2018	DR = 68.91
[39]	UNSW-NB15	Binary	Bidirectional LSTM	2018	Acc = 95.71
[40]	NSL-KDD	Multiclass	DNN	2019	Acc = 78.10
[41]	NSL-KDD	Multiclass	DNN	2019	Acc = 97.00
[42]	NSL-KDD	Binary	MLP, CNN, DNN, AE	2019	Acc = 99.24
[43]	NSL-KDD	Binary	SMO	2019	Acc = 99.02
[44]	NSL-KDD	Multiclass	GA optimized DBN	2019	Acc = 99.45
[45]	NSL-KDD	Multiclass	MLP, CNN	2019	Acc = 82.60
[46]	AWID-CLS	Binary	Autoencoder	2019	Acc = 98.00
[47]	Personal	Multiclass	GRU	2019	Acc = 95.60
[48]	Personal	Multiclass	GRU	2019	F1 = 80.30
[49]	AWID	Multiclass	Autoencoder, NN	2019	Acc = 99.90
[50]	ISCX2012,	Binary	LSTM	2019	Acc = 99.99
[51]	UNSW-NB15	Binary	Autoencoder, SVM	2019	Acc = 97.00
[52]	BoT-IoT	Multiclass	FNN	2019	Acc = 98.09
[31]	NSL-KDD	Binary	SRDLM	2020	DR = 99.50
[53]	NSL-KDD	Multiclass	RNN	2020	Acc = 79.00
[54]	Kitsune	Multiclass	Autoencoder	2020	Acc = 83.30
[55]	NSL-KDD	Multiclass	RNN	2020	Acc = 92.18
[56]	NSL-KDD	Binary	CNN	2020	Acc = 86.95
[57]	NSL-KDD	Multiclass	DNN	2020	Acc = 97.64
[58]	CIC-IDS2018	Multiclass	DNN, RNN, CNN	2020	Acc = 97.38
[59]	Varied Dataset	Binary	CNN	2020	Acc = 99.41
[60]	BoT-IoT	Multiclass	FFN	2020	Acc = 99.80
[61]	Multiple	Binary	CNN and GRU	2020	Acc = 99.42
[62]	MedBIoT	Multiclass	Fast GRNN	2020	F1 = 99.99
[63]	SWaT	Multiclass	CNN	2020	Acc = 98.02
[64]	KDD	Binary	GAN	2020	Pr = 88.80
[65]	UNSW-NB15	Binary	CNN	2020	Acc = 96.00
[65]	UNSW-NB15	Binary	ANN	2020	Acc = 97.00
[65]	UNSW-NB15	Binary	RNN	2020	Acc = 96.00
[66]	Mixed	Multiclass	FNN	2020	Acc = 99.73
[67]	Webscope S5	Binary	CNN	2020	Acc = 98.36
[68]	KDD	Binary	DNN	2020	Acc = 92.90
[69]	NSL-KDD	Multiclass	DBN	2021	Acc = 97.10
[70]	NSL-KDD	Multiclass	CNN + LSTM	2021	Acc = 90.67
[71]	UNSW-NB15	Binary	LNN	2021	Acc = 97.54

We studied deep neural network models from 2017 to 2021. Most of the models used the KDD99 dataset for evaluation. The KDD99 dataset is very old and was not created for use in IoT networks. As a result, the KDD99 dataset cannot be used to assess an intrusion detection framework for IoT networks. Many deep learning models for binary classification were created with accuracy as the only performance metric. Multiclass classification models show a very poor degree of accuracy. To perform intrusion detection, neither of the models merged CNN and GRU. This paper used CICFlowmeter [11] to retrieve network features from four publicly available datasets' pcap files. These datasets were developed using real and simulated IoT networks. We evaluate our proposed model using binary and multiclass classification. The proposed model was evaluated using accuracy, precision, recall, and F1 score as performance metrics. The proposed model achieved a high detection rate and a low false rate.

3. Proposed Model

Recently, convolutional neural networks have shown better performance in voice and picture recognition. Recurrent neural networks are frequently used in speech understanding, language synthesis, language modeling, and language generation. Both convolutional neural networks and recurrent neural networks generate interesting results in these areas, and their use is becoming more popular. Intrusion detection concerns can be more effectively repurposed into convolution neural network problems known as feature mapping. In this paper, we used a convolution neural network and a gated recurrent neural network. Our proposed multiclass CNNGRU model is described in Figure 1a, while our proposed binary class CNNGRU model is shown in Figure 1b. The multiclass model consists of an input layer, two convolutional layers, two GRU layers, a flatten layer, a fully connected dense layer, and an output layer. The binary class model consists of an input layer, one convolutional layer, one GRU layer, a flatten layer, a fully connected dense layer, and an output layer. The reshaping system sends information to the input layer. The convolution layer extracts feature characteristics from the input image and maintains the connection between pixels while also learning new image properties from small squares of input data. The batch normalization process seeks to equalize all of a neural network layer's input. The batch normalization layer normalizes the performance of the convolution layer ahead of the average pooling layer. The pooling layer enables the enhancement of functionality by condensing them into sub-maps of robust features. The average pooling layer determines each patch's total number of features by averaging the total number of features in each upgrade around the entire function map.

Overfitting may occur when neural networks have difficulty distinguishing between valid and invalid results; thus, further performance optimization of the test dataset parameters is often required. A dropout layer prevents overfitting by cutting out some training neurons in the process of model building. The tensor is restructured to provide a flat operation on a tensor with an element count equivalent to the element count of the tensor, except the batch size. A flat layer is entirely linked to a dense layer. The dense layer uses 512 neurons. The number of neurons in the output layer is equal to the number of classes used for classification. The CNNGRU model was trained, validated, and tested using six IoT intrusion detection datasets. Determining which features to use is a critical phase in machine learning. Model improvements known as feature selection require the identification and selection of those features required to increase prediction. The feature selection technique minimizes overfitting, accelerates model training, and strengthens the model's resistance to test inaccuracies. In this article, we extract important features from our proposed datasets using a feature selection method known as recursive feature elimination [72,73]. The feature selection method estimates the overall significance of features using a random forest classifier. Tenfold cross-validation tests were performed to ensure that the feature selection model was not suffering from overfitting. The feature selection algorithm uses the IoT-DS-2 dataset and selects the 48 best features. The IoT-DS-2 dataset was used for feature selection since it contains attack data from all datasets.

We can measure convolution in 1D using temporal access and single-direction kernel movements. Convolution 1D uses two-dimensional input and output data, which is often seen in time series data. The input layer received an input vector (48, 1) that contains 48 best features. Two convolution layer blocks were used after the input layer. Each block consists of a convolution layer, activation layer, and dropout layer. Convolution layers collect input layer features and compute vector properties for small data samples within the input. The convolution first layer uses a relu activation function, 64 filters, and kernel size 8. The second convolution layer uses 128 filters and the relu activation function. Batch normalization ensures that the inputs are continuously normalized. The normalization process substantially reduces the difficulty associated with organizing changes across many levels. The average pooling layer downsamples feature maps by summarizing features. We used a dropout layer with a drop value of 0.1 to regularize the training data model and minimize overfitting. There are two GRU layer blocks. Each GRU block consists of the GRU layer, activation layer, and dropout layer. The two GRU layers use 512 units. The activation layer uses the relu activation function, and the dropout layers use a dropout value of 0.1. The flatten layer converts the tensor to a shape as the tensor components. Five hundred and twelve neurons are used in the dense layer, while the number of neurons in the output layer is equal to the number of classes in the dataset.

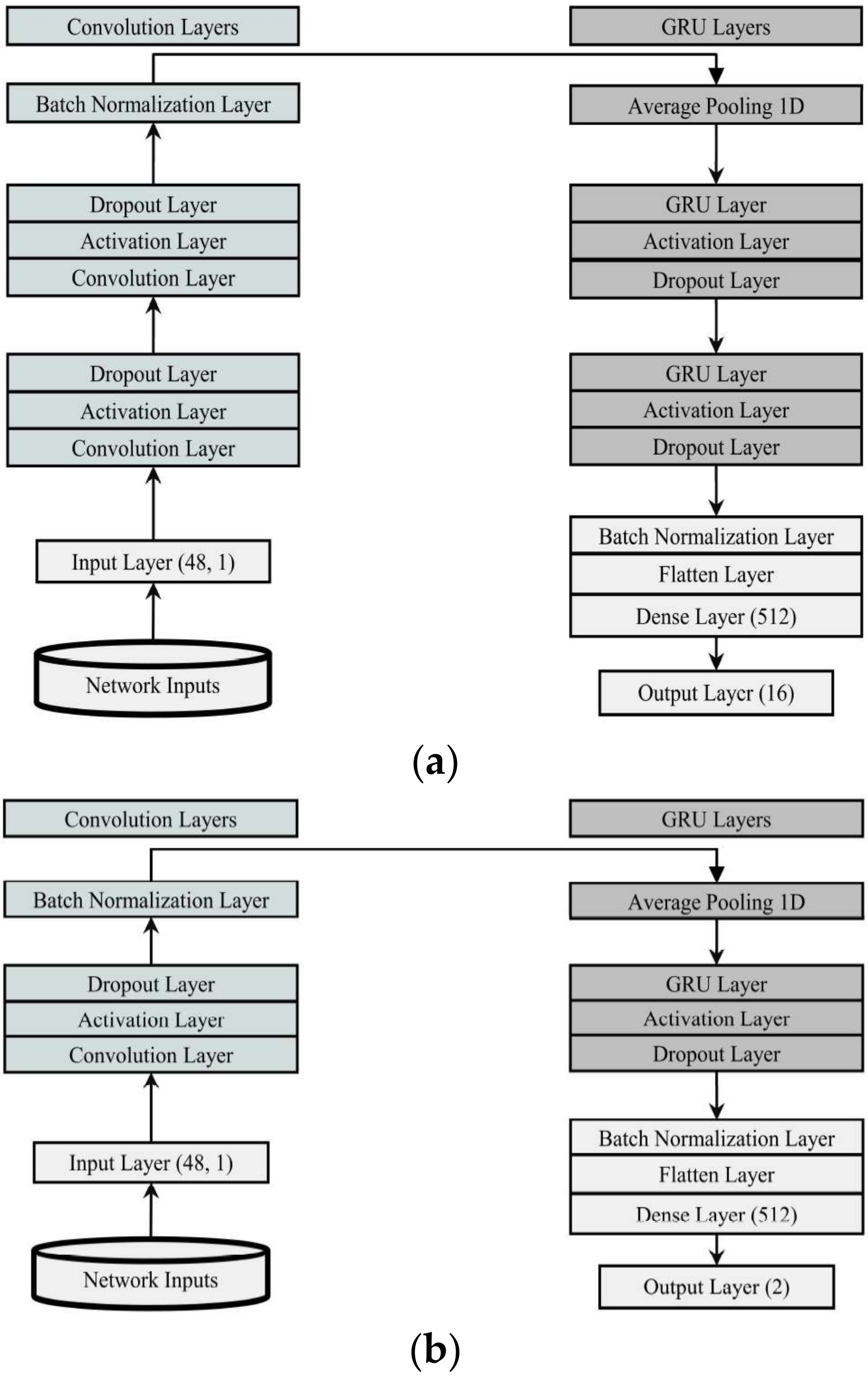


Figure 1. (**a**) Multiclass CNNGRU model. (**b**) Binary CNNGRU model.

Many rounds of training are required to reach the convergence point for deep learning models. However, the number of training rounds can be minimized by selecting a particular parameter configuration that allows for further convolution in the training phase, generating and directing the network structure. Overfitting can be avoided using regularization. We tuned the multiclass model using kernel_regularizer, bias_regularizer, and activity_regularizer regularization approaches. The binary classification model was trained using the same hyperparameters. To help in feature learning over time, we assign random values to the CNN model layers at the beginning. To prevent overfitting, we use L1, L2, and dropout regularization. To modify optimizer weights, we use adam optimizers and sparse categorically cross-entropy loss functions.

A key machine learning algorithm variable is the learning rate, which affects the amount of change each model takes from one iteration to the next. We conducted many tests with various learning rates for the adam optimizer and determined that 0.0001 was the optimum learning rate for the greatest detection rate. Finally, we adopted an early stopping technique to avoid overfitting. The model monitors the validation loss and ends the training phase if it does not reduce after a specified number of cycles. To ensure the maximum possible network performance over the monitoring cycle, the epoch value must be changed before the network accuracy vs. epochs no longer improves. For the proposed model, we used 50, 100, 200, 500, and 1000 epochs. We chose 100 epochs as the optimum number of epochs since all trials using the model converged within this time frame. Activation functions are critical parameters for deep learning algorithms. Convolution, GRU, and dense layers use the relu activation function. Recurrent sigmoid activation in GRU and softmax activation is used in the output layer. The batch size is an essential hyperparameter in deep learning models. A larger batch size can reduce processing time and speed up the training process across multiple nodes. Bigger batch sizes provide similar training losses as smaller batch sizes, but larger batch sizes seem to generalize worse for testing results. For training and testing the proposed model, a batch size ranging from 64 to 128 was determined to be the most optimum option.

3.1. Data Collection

The initial phase includes the management of raw network traffic. This paper used four publicly available datasets' pcap files and extracted network features using CICFlowmeter [11]. The CICFlowmeter is open-source software that generates CSV files from pcap data. The CICFlowmeter generates 80 unique network features. The BoT-IoT dataset was developed by Koroniotis et al. [74]. The testbed environments incorporated five IoT devices and were used to create a practical smart home architecture. These devices were operated locally and connected to cloud networks using the node-red scheme, which allowed for the development of normal network traffic. The new BoT-IoT data collection was shown in Table 2. There are four attack types, which are further subdivided into ten subtypes. A comprehensive description of the testbed settings and attacks can be found in the described article [74]. The newly developed botnet dataset has been made publicly available, and a link is included in [75]. Kang et al. [76] created a dataset for detecting IoT network intrusions. The IoT network intrusion dataset was created using a typical smart home device consisting of a smart home SKT NGU and an EZVIZ Wi-Fi camera. These two IoT devices are connected to a smart home Wi-Fi router and are used as victim devices. There are four attack categories and eight subcategories. The dataset for IoT network intrusions is shown in Table 3. A connection to the newly developed IoT network intrusion dataset is included in [77].

Table 2. BoT-IoT dataset instances.

No	Category	With Redundancy	Without Redundancy
0	Normal	105,202	77,511
1	DoS	57,027,372	17,420,085
2	DDoS	37,077,674	18,199,716
3	Scan	1,831,558	256,951
4	Data Theft	6390	6257

Table 3. IoT network intrusion dataset instances.

No	Category	With Redundancy	Without Redundancy
0	Normal	40,073	39,851
1	DoS	59,391	59,391
2	MITM	35,377	32,909
3	Mirai	415,677	366,971
4	Scan	75,265	72,122

The MQTT-IoT-IDS2020 dataset was created by Hindy et al. [78]. From the MQTT networking platform, this dataset consists of both regular network traffic and brute-force attacks. The network comprises 12 MQTT sensors, a broker, a device for replicating a camera stream, and an attacker. The dataset includes the most popular MQTT attacks and scenarios for analyzing real-world IoT devices. The MQTT-IoT-IDS2020 dataset is presented in Table 4. The MQTT-IoT-IDS2020 dataset contains four attack categories. The new MQTT-IoT-IDS2020 dataset can be accessed at [79]. The Stratosphere Laboratory of the CTU in the Czech Republic created the IoT-23 dataset [80]. There are 20 malicious activities and 3 non-malicious activities. The IoT-23 dataset was created to provide researchers with a large and labeled dataset of real-world IoT devices and IoT malware infections to design a machine learning model. The IoT-23 dataset contains 20 separate network operation models to prototype various IoT device use cases. This dataset aims to provide the world with two distinct datasets: one that comprises benign and malicious network capture and another that only contains benign IoT network capture. The IoT-23 dataset can be seen in Table 5. The IoT-23 dataset contains nine attack categories. The IoT-23 dataset is available at [79]. We merged BoT-IoT, IoT Network Intrusion, and MQTT-IoT-IDS2020 datasets to increase the number of attack classes in the dataset. Nine attack classes and one normal class comprise the new dataset. Table 6 describes the latest dataset known as IoT-DS-1. The latest IoT-DS-1 dataset can be found at [79]. The BoT-IoT, IoT Network Intrusion, MQTT-IoT-IDS2020, and IoT-23 datasets were then merged. The new dataset, named IoT-DS-2, includes 15 attack classes and 1 normal class, as shown in Table 7. The data collection IoT-DS-2 can be accessed at [79].

Table 4. MQTT-IoT-IDS2020 dataset instances.

No	Category	With Redundancy	Without Redundancy
0	Normal	334,318	167,159
1	MQTT_Bruteforce	2,002,780	2,001,972
2	Scan-A	31,245	29,276
3	Scan-U	33,404	27,843
5	Sparta	1,252,259	1,217,198

Table 5. IoT-23 dataset instances.

No	Category	With Redundancy	Without Redundancy
0	Normal	4,313,776	226,451
1	Attack	1,716,778	1,699,608
2	Mirai	756	756
3	File Download	8035	7707
4	HeartBeat	12,895	12,648
5	C&C	23,981	20,612
6	Torii	33,858	24,492
7	Port Scan	65,944,863	2,999,999
8	DDoS	20,768,988	4,619,869
9	Okiru	13,718,252	12,908,506

Table 6. IoT-DS-1 dataset instances.

No	Category	BoT-IoT	IoT Net-ID	MQTT	Total
0	Normal	77,511	-	167,159	244,670
1	DDoS	17,420,085	-	-	17,420,085
2	DoS	-	59,391	-	59,391
3	MITM ARP Spoofing	-	32,909	-	32,909
4	MQTT Bruteforce	-	-	2,001,972	2,001,972
5	Mirai	-	366,971	-	366,971
6	OS Scan	35,675	-	-	35,675
7	Port Scan	-	-	57,119	57,119
8	Sparta	-	-	1,217,198	1,217,198
9	Theft	6257	-	-	6257

Table 7. IoT-DS-2 dataset instances.

No	Category	BoT-IoT	IoT Net-ID	MQTT	IoT-23	Total
0	Normal	-	-	-	4,253,672	4,253,672
1	DDoS	17,420,085	-	-	-	17,420,085
2	DoS	-	59,391	-	-	59,391
3	MITM ARP Spoofing	-	32,909	-	-	32,909
4	Mirai	-	366,971	-	-	366,971
5	MQTT Bruteforce	-	-	2,001,972	-	2,001,972
6	Sparta	-	-	1,217,198	-	1,217,198
7	Theft	6257	-	-	-	6257
8	Attack	-	-	-	1,699,608	1,699,608
9	C&C	-	-	-	20,612	20,612
10	File Download	-	-	-	7707	7707
11	HeartBeat	-	-	-	12,648	12,648
12	Okiru	-	-	-	12,908,506	12,908,506
13	OS Scan	35,675	-	-	-	35,675
14	Port Scan	-	-	-	2,999,999	2,999,999
15	Torii	-	-	-	24,492	24,492

3.2. Preprocessing Dataset

When features from pcap files have been extracted and analyzed, the next step is to label individual dataset instances based on specified criteria. To distinguish between regular and malicious dataset instances, each dataset has its own set of criteria for evaluating whether an instance was normal or malicious. Our proposed model can cover all IoT networks; however, the flow ID, source IP, and destination IP characteristics are unique to a particular IoT network. As a result, these features were removed from all datasets.

We filled NaN values with 0 in all datasets. Redundant instances were generated by CICFlowmeter when the pcap file was converted to CSV files. These duplicate instances were removed from all datasets. After removing redundant instances, we may use previously unseen data to evaluate the model performance throughout the testing phase. We normalize the input feature columns within a specified range $(-1, 1)$ to remove extreme values and significantly accelerate the computations. A non-numeric column is converted to a numeric column. An anomaly is represented by a value of 1, whereas a normal value is represented by 0 in binary classification. The multiclass label encoded 0 to 3 for the BoT-IoT dataset, 0 to 4 IoT for the network intrusion detection dataset, 0 to 4 for the MQTT-IoT dataset, and 0 to 9 for the IoT-23 dataset. The multiclass-encoded label in these datasets represents normal network traffic, and the rest-encoded label represents the desire attack type.

We merged BoT-IoT, IoT network intrusion, and MQTT-IoT-IDS2020 datasets to increase the number of attack classes in the dataset. The updated dataset contains nine attack classes and one normal class. Table 6 describes the latest dataset known as IoT-DS-1. The IoT-DS-1 dataset was classified into normal and attack categories using a multiclass labeling system ranging from 0 to 9. The BoT-IoT, IoT network intrusion, MQTT-IoT-IDS2020, and IoT-23 datasets were then combined. The new dataset, named IoT-DS-2, includes 15 attack classes and one normal class, as shown in Table 7. The IoT-DS-2 dataset was classified into normal and attack categories using a multiclass labeling scheme ranging from 0 to 15. We made class weights more distinctive to better expose the classifiers to each class since there is a clear imbalance in the training set. Google Colab Pro was used to develop the models, which included the TensorFlow framework and Keras implementations. In order to perform the classification, the data are first run through the preprocessing process and are then split into three sets: training, validation, and testing. The dataset was initially split into 80% for training and 20% for testing. The training set is then subdivided into two groups: 80% for training and 20% for validation, with each group being split in a stratified way.

4. Evaluation Results

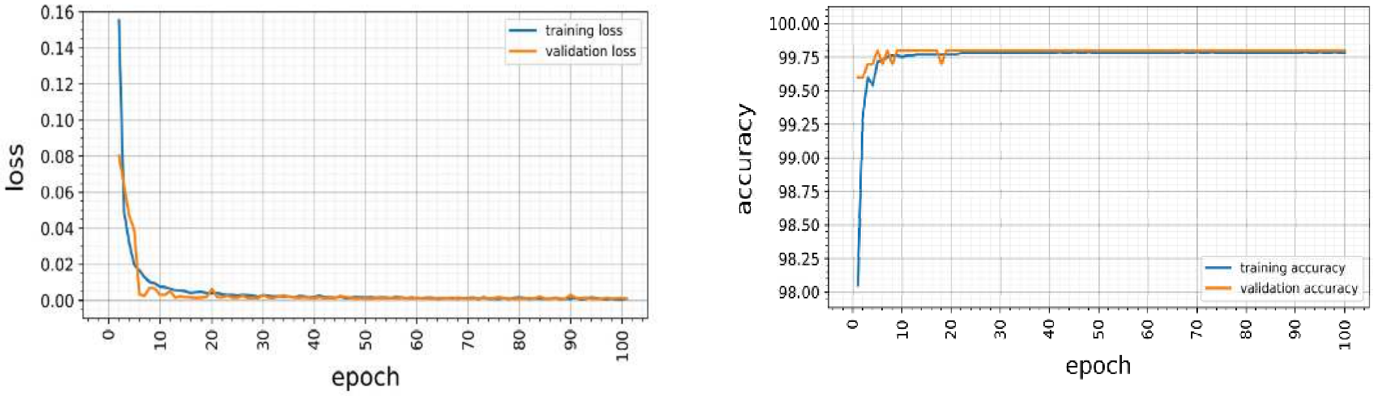
The proposed model accuracy and loss were measured for training and validation sets at each epoch value. This allows us to evaluate whether the model has been sufficiently trained to differentiate between different kinds of anomalies. The gradient is measured from the loss and is used to change the neural network weights. TensorFlow has a variety of loss functions that may be used to tackle a wide variety of issues. We used the adam optimizer and a sparse categorically cross-entropy loss function. The proposed model loss during training and validation using the IoT-DS-2 dataset is represented in Figure 2a, and model accuracy during training and validation using the IoT-DS-2 dataset is represented in Figure 2b. The early stopping technique will stop the training phase if the validation loss does not decrease after a certain number of iterations, reducing the overfitting issue. We used 100 epochs, a batch size of 64, and a patience of 5 iterations to train the proposed model. The average accuracy was 99.20 for training, 99.30 for validation, and 99.38 for testing using the IoT-DS2 dataset. The accuracy did not improve by increasing the epochs beyond 100. Consequently, running a model over many epochs results in the model overfitting to the training data. The effectiveness of the CNNGRU model is evaluated using accuracy, precision, recall, and F1 score evaluation metrics. The accuracy of a system is expressed as the proportion of correctly identified instances to the total number of identified instances. Precision is defined as the ratio of correctly categorized items to the total quantity of TP (true positive) and FP (false positive). The recall value is calculated by dividing the total number of TP measurements by the total TP and FN (false negative) measurements. Precision and recall are combined to form the F1 score, which is the harmonic mean of precision and recall. High F1 scores indicate excellent precision and recall.

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

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

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

$$\text{F1 Score} = 2 \times \frac{(\text{Precision} \times \ \text{Recall})}{(\text{Precision} + \text{Recall})} \tag{4}$$



(**a**) Loss (**b**) Accuracy

Figure 2. Multiclass CNNGRU model loss and accuracy in training and validation.

4.1. Multiclass Classification

The dataset was classified using a multiclass classification as either normal network traffic or the attacks listed in Tables 2–7. It takes fewer than 100 epochs to complete the training and validation processes. The early stopping technique with a patience of five iterations reduces overfitting. Training and validation loss decreased gradually up to 100 epochs. The average training loss level is 0.0049, while the average validation loss is 0.0039. This evidence demonstrates that our proposed CNNGRU model will accurately classify the different types of cyberattacks found in datasets or real-world IoT networks. The BoT-IoT, IoT network intrusion, MQTT-IoT-IDS2020, IoT-23, IoT-DS-1, and IoT-DS-2 datasets are used to demonstrate the multiclass CNNGRU model performance. The BoT-IoT dataset consists of three attack categories and a normal category. The model performed well for normal and malicious categories. The theft class has a limited number of instances, which indicates why the model received a high FNR for the theft class. Next, we used the IoT network intrusion dataset to evaluate the model. The model performance for the IoT network intrusion dataset is at the moderate level for attack categories. A high false negative rate was accomplished for Mirai and scan categories. The proposed model achieved a high detection rate and low false positive and false negative rates for normal and all attack categories in the MQTT-IoT-IDS2020 dataset. The CNNGRU model's capacity to classify large numbers of classes was evaluated using the IoT-23 dataset. The proposed model achieved a high detection rate for normal and all attack classes except the C&C attack class. By combining the BoT-IoT, IoT Network Intrusion, and MQTT IoT-IDS2020 datasets, the CNNGRU model capacity to identify large numbers of classes was also evaluated.

The multiclass classification for BoT-IoT, IoT network intrusion, MQTT-IoT-IDS2020, IoT-23, IoT-DS-1, and IoT-DS-2 datasets is presented in Table 8. This table shows the average performance of the CNNGRU model for each dataset. The proposed model achieved excellent performance for all datasets except the IoT intrusion detection dataset. The highest accuracy, 99.92%, was measured for the BoT-IoT dataset. The performance of the CNNGRU model when applied to the IoT-23 dataset is summarized in Table 9. The average accuracy for the IoT-23 dataset was 99.88%. The instances of Mirai and Okiru were correctly detected and classified. The categories of File Download and HeartBeat have the lowest detection rate of all the categories. Table 10 summarizes the CNNGRU model performance using the IoT-DS-1 dataset. The false positive rate was low for all classes, but the false negative rate for MITM, Mirai, and Scan attacks was high compared to other attacks. To check the model capability for several attack classes, we combined BoT-IoT, IoT network intrusion, MQTT IoT-IDS2020, and IoT-23 datasets. The new dataset was named IoT-DS-2, which contains 15 attack classes and a normal class. Table 11 shows the CNNGRU model performance utilizing the IoT-DS-2 dataset. The detection rate for the normal class was measured at 99.56%, and all malicious categories also had high detection rates except MITM, Mirai, Theft, C&C, and HeartBeat. MITM had the lowest detection rate of 64.25%.

Table 8. Average multiclass classification accuracy, precision, recall, and F1 score.

Dataset	Accuracy%	Precision%	Recall%	F1 Score%
BoT-IoT	99.92	99.91	99.91	99.91
IoT NID	96.77	97.34	96.77	96.91
MQTT	99.91	99.90	99.90	99.90
IoT-23	99.88	99.89	99.88	99.88
IoT-DS-1	99.10	99.32	99.10	99.15
IoT-DS-2	99.38	99.49	99.38	99.41

Table 9. IoT-23 multiclass classification.

Class	Precision%	Recall%	F1 Score%
Normal	99.82	99.26	99.54
Attack	99.92	99.93	99.93
Mirai	100.00	100.00	100.00
File Download	95.44	99.75	97.55
HeartBeat	81.24	99.96	89.64
C&C	91.52	90.70	91.11
Torii	99.96	99.98	99.97
PortScan	99.99	99.99	99.99
DDoS	99.99	99.99	99.99
Okiru	100.00	100.00	100.00

Table 10. IoT-DS-1 multiclass classification (BoT-IoT, IoT network intrusion, and MQTT-IoT-IDS2020 datasets).

Class	Precision%	Recall%	F1 Score%
Normal	99.50	99.78	99.64
DDoS	99.99	99.96	99.97
DoS	99.74	99.38	99.56
MITM ARP Spoofing	60.61	95.93	74.30
MQTT Bruteforce	99.85	99.88	99.87
Mirai	99.54	93.20	96.27
OS Scan	99.94	99.87	99.91
Port Scan	80.35	93.55	86.45
Sparta	99.85	99.84	99.84
Theft	100.00	99.76	99.88

Table 11. IoT-DS-2 multiclass classification (BoT-IoT, IoT network intrusion, MQTT-IoT-IDS2020, and IoT-23 datasets).

Class	Precision%	Recall%	F1 Score%
Normal	99.56	99.36	99.46
DDoS	99.99	99.99	99.99
DoS	99.70	99.40	99.55
MITM ARP Spoofing	64.25	97.78	77.55
Mirai	99.58	93.89	96.65
MQTT Bruteforce	99.85	99.74	99.80
Sparta	99.53	99.50	99.52
Theft	98.78	99.28	99.04
Attack	99.66	99.90	99.78
C&C	96.15	91.55	93.79
File Download	96.12	99.50	97.78
HeartBeat	90.71	99.81	95.04
Okiru	99.99	100.00	99.99
OS Scan	99.94	99.90	99.92
Port Scan	99.99	99.99	99.99
Torii	99.98	99.96	99.97

4.2. Binary Classification

The dataset was classified using binary classification, either normal network traffic or an anomaly. The classification of normal and anomaly traffic inside each subcategory achieves a high accuracy, precision, recall, and F1 score of greater than 99.50%, which signifies low FP and FN forecasts. We used the IoT-DS-2 dataset for binary classification. The reason for using the IoT-DS-2 dataset for binary classification is that the IoT-DS-2 dataset contains all malicious network traffic from the BoT-IoT, IoT network intrusion, MQTT-IoT-IDS2020, and IoT-23 datasets. In comparison to multiclass classification, the binary CNNGRU model required less time to train and validate. Early stopping and dynamic learning rates track the number of training epochs during the training phase, improving the performance of the adam optimization method. The binary CNNGRU model used the same tuning parameter as multiclass classification. Table 12 shows confusion matrices of the binary classification of the IoT-DS-2 dataset. A binary CNNGRU model with a limited amount of wrongly identified instances has a high value of precision, recall, and F1 score. The attack category HeartBeat achieved low accuracy, precision, recall, and F1 score. Eighteen malicious network flows were classified as normal network flows, while 43 normal network flows were classified as malicious network flows. The evaluation matrices of the binary classification of the IoT-DS-2 dataset are presented in Figure 3. The accuracy of binary classification was measured at 99.50% or higher for all malicious categories, except File Download and HeartBeat.

4.3. Discussion and Comparison of Results

Several recent advances in deep learning technology have shown their capacity to recognize trends in various research fields. The CNNGRU model results are related to previous study results in this segment. Our proposed model performed substantially better at detecting anomalies in various IoT networks. We have developed a neural network model that utilizes convolutional and gated recurrent units to identify anomalies in IoT networks. Our effective architecture is based on CNN and GRU models. The CNNGRU model is evaluated using the BoT-IoT, IoT network intrusion, MQTT-IoT-IDS2020, IoT-23, IoT-DS-1, and IoT-DS-2 datasets. Numerous experiments have been performed with the primary goal of binary and multiclass classification of attack categories. For binary classification, we used the IoT-DS-2 dataset. The IoT-DS-2 dataset is used since it includes all malicious network traffic from the BoT-IoT, IoT network intrusion, MQTT-IoT-IDS2020, and IoT-23 datasets. The confusion matrices of the binary classification are presented in Table 12. Figure 3 shows the accuracy, precision, recall, and F1 score of binary classification. In Table 13, the CNNGRU model binary classification results are compared to those previously presented in other research articles. Our proposed binary classification model performs better than other deep learning models on all performance measures (accuracy, precision, recall, and F1 score).

Table 12. Confusion matrices binary CNNGRU model.

	Normal (Predicted)	**DDoS (Predicted)**
Normal (True)	45,450	0
DDoS (True)	0	45,056

	Normal (Predicted)	**DoS (Predicted)**
Norma l(True)	45,243	18
DoS (True)	9	4059

	Normal (Predicted)	**MITM (Predicted)**
Normal (True)	45,335	0
MITM(True)	0	1991

	Normal (Predicted)	**Mirai (Predicted)**
Normal (True)	45,371	0
Mirai (True)	1	18,354

	Normal (Predicted)	**MQTT-BF (Predicted)**
Normal (True)	45,223	5
MQTT-BF (True)	10	10,164

	Normal (Predicted)	**Sparta (Predicted)**
Normal (True)	45,156	73
Sparta (True)	49	16,470

	Normal (Predicted)	**Theft (Predicted)**
Normal (True)	45,241	6
Theft (True)	0	1265

	Normal (Predicted)	**Attack (Predicted)**
Normal (True)	45,332	12
Attack (True)	0	3723

	Normal (Predicted)	**C&C (Predicted)**
Normal (True)	45,102	36
C&C (True)	9	4558

	Normal (Predicted)	**FileDWNLD (Predicted)**
Normal (True)	45,117	8
FileDWNLD (True)	3	1649

	Normal (Predicted)	**HeartBeat(Predicted)**
Normal (True)	45,152	30
HeartBeat (True)	3	2555

	Normal (Predicted)	**Okiru (Predicted)**
Normal (True)	45,433	2
Okiru (True)	1	30,455

	Normal (Predicted)	**OS Scan (Predicted)**
Normal (True)	45,682	0
OS Scan (True)	0	27,265

	Normal (Predicted)	**Port Scan (Predicted)**
Normal (True)	45,435	0
Port Scan (True)	3	30,453

	Normal (Predicted)	**Torii (Predicted)**
Normal (True)	45,235	6
Torii (True)	3	4945

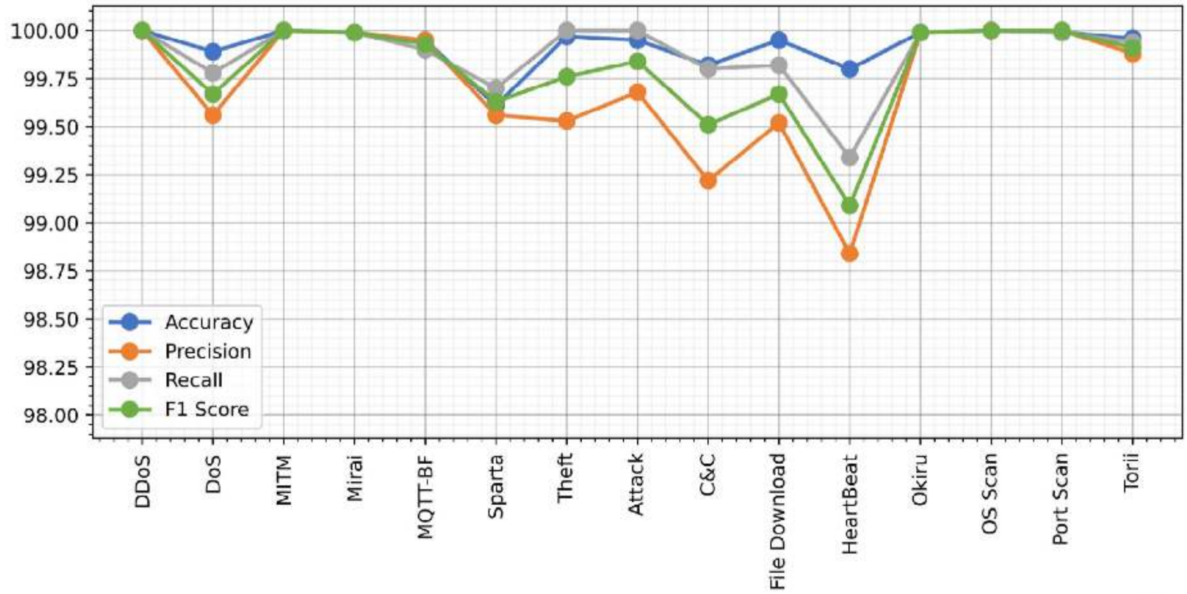


Figure 3. Performance metrics binary CNNGRU model.

Table 13. Binary classification comparison.

Article	Model	Accuracy%	Precision%	Recall%	F1 Score%
[19]	CNN	94.54	95.65	90.74	93.00
[56]	CNN	86.95	89.56	87.25	88.41
[65]	ANN	96.00	95.00	100.00	99.00
[68]	DNN	92.70	99.90	91.20	95.40
Proposed	**CNNGRU**	**99.96**	**99.90**	**99.95**	**99.93**

We classify IoT network traffic into four, five, ten, and sixteen categories using multiclass classification. Numerous publications in the literature concentrated exclusively on binary classification when designing an intrusion detection model using deep learning. Table 8 summarizes the proposed model average accuracy, precision, recall, and F1 score for multiclass classification. Our proposed model achieved high accuracy, precision, recall, and F1 score for BoT-IoT, IoT network intrusion, MQTT-IoT-IDS2020, IoT-23, IoT-DS-1, and IoT-DS-2 datasets. The CNNGRU model multiclass classification for the Bot-IoT dataset findings contrast with those previously published in other papers in Table 14. Our proposed multiclass classification model outperforms other deep learning models in accuracy, precision, recall, and F1 score. Previously referenced publications focused mainly on binary classification when creating an intrusion detection deep learning platform. We categorize IoT network traffic into four, five, ten, and sixteen classes using multiclass classification. In all datasets, our proposed CNNGRU model outperforms all existing implementations.

Table 14. Multiclass classification comparison.

Article	Model	Accuracy%	Precision%	Recall%	F1 Score%
[19]	CNN	98.43	98.00	98.00	98.00
[21]	CNN	99.20	85.20	88.25	91.16
[52]	FFN	98.07	99.03	99.03	99.03
[56]	CNN	86.00	60.00	56.00	54.00
[60]	FFN	99.80	99.79	99.79	99.79
[63]	CNN	98.02	97.71	98.39	98.05
[66]	FFN	99.73	99.86	98.67	98.77
[67]	LSTM	99.62	98.78	97.20	97.98
Proposed	**CNNGRU**	**99.92**	**99.91**	**99.91**	**99.91**

5. Conclusion

The effective application of deep learning has shown its capacity to recognize trends for various research fields. There are constantly more novel and evolving methods of launching cyberattacks. A network intrusion detection model for IoT networks has been proposed. It is built on a deep learning model that includes a convolutional neural network and gated recurrent units. We tested model effectiveness for multiclass and binary classification on six datasets of real-world network traffic. Our proposed multiclass and binary classification models demonstrated high accuracy, precision, recall, and F1 score compared to current classification techniques and recent deep learning models. The obtained results demonstrate the suggested technique's effectiveness.

In the future, we will investigate these attacks utilizing a variety of deep learning models and generative adversarial networks and compare the results to the current model.

Author Contributions: Writing—original draft preparation, I.U.; Review and editing, A.U. and M.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Makhdoom, I.; Abolhasan, M.; Lipman, J.; Liu, R.P.; Ni, W. Anatomy of Threats to the Internet of Things. *IEEE Commun. Surv. Tutor.* **2018**, *21*, 1636–1675. [CrossRef]
2. Tawalbeh, L.; Muheidat, F.; Tawalbeh, M.; Quwaider, M. IoT Privacy and Security: Challenges and Solutions. *Appl. Sci.* **2020**, *10*, 4102. [CrossRef]
3. Donnell, L.O. IoT Device Takeovers Surge 100 Percent in 2020. 2020. Available online: https://threatpost.com/iot-device-takeovers-surge/160504 (accessed on 20 May 2021).
4. Conti, M.; Dehghantanha, A.; Franke, K.; Watson, S. Internet of Things security and forensics: Challenges and opportunities. *Futur. Gener. Comput. Syst.* **2018**, *78*, 544–546. [CrossRef]
5. The Security and Privacy Issues that Come with the Internet of Things. 2020. Available online: https://www.businessinsider.com/iot-security-privacy?r=US&IR=T (accessed on 20 May 2021).
6. The Mirai Botnet Explained: How Teen Scammers and CCTV Cameras Almost brought down the Internet. 2018. Available online: https://www.csoonline.com/article/3258748/the-mirai-botnet-explained-how-teen-scammers-and-cctv-cameras-almost-brought-down-the-internet.html (accessed on 20 May 2021).
7. Analysis of Shamoon 2 Disk-Wiping Malware. 2017. Available online: https://logrhythm.com/blog/analysis-of-shamoon-2-malware/ (accessed on 22 May 2021).
8. The 11 Biggest Ransomware Attacks Of 2020. 2020. Available online: https://www.crn.com/slide-shows/security/the-11-biggest-ransomware-attacks-of-2020-so-far- (accessed on 21 May 2021).
9. Lee, K.; Kim, M.S.; Shim, P.; Han, I.; Lee, J.; Chun, J.; Cha, S. Technology advancement of laminate substrates for mobile, iot, and automotive applications. In Proceedings of the 2017 China Semiconductor Technology International Conference (CSTIC), Shanghai, China, 27–28 March 2017; pp. 1–4. [CrossRef]
10. Luh, R.; Marschalek, S.; Kaiser, M.; Janicke, H.; Schrittwieser, S. Semantics-aware detection of targeted attacks: A survey. *J. Comput. Virol. Hacking Tech.* **2016**, *13*, 47–85. [CrossRef]
11. Lashkari, A.H.; Gil, G.D.; Mamun, M.S.I.; Ghorbani, A.A. Characterization of Tor Traffic using Time based Features. In Proceedings of the 3rd International Conference on Information Systems Security and Privacy (ICISSP 2017), Porto, Portugal, 19–21 February 2017. [CrossRef]
12. Ullah, I.; Mahmoud, Q.H. Design and Development of a Deep Learning-Based Model for Anomaly Detection in IoT Networks. *IEEE Access* **2021**, *1*. [CrossRef]
13. Saha, S. A Comprehensive Guide to Convolutional Neural Networks. 2018. Available online: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53 (accessed on 19 May 2021).
14. Dickson, B. What are Convolutional Neural Networks (CNN). 2020. Available online: https://bdtechtalks.com/2020/01/06/convolutional-neural-networks-cnn-convnets (accessed on 18 May 2021).
15. Phi, M. Illustrated Guide to LSTM's and GRU's: A Step by Step Explanation. 2018. Available online: https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21 (accessed on 18 May 2021).
16. Olah, C. Understanding LSTM Networks. 2015. Available online: http://colah.github.io/posts/2015-08-Understanding-LSTMs (accessed on 18 May 2021).
17. Kostadinov, S. Understanding GRU Networks. 2017. Available online: https://towardsdatascience.com/understanding-gru-networks-2ef37df6c9be (accessed on 18 May 2021).
18. Kim, I.; Chung, T.-M. Malicious-Traffic Classification Using Deep Learning with Packet Bytes and Arrival Time. In Proceedings of the Future Data and Security Engineering. FDSE 2020, Quy Nhon, Vietnam, 25–27 November 2020; pp. 345–356. [CrossRef]
19. Hassan, M.M.; Gumaei, A.; Alsanad, A.; Alrubaian, M.; Fortino, G. A hybrid deep learning model for efficient intrusion detection in big data environment. *Inf. Sci.* **2020**, *513*, 386–396. [CrossRef]
20. Li, D.; Deng, L.; Lee, M.; Wang, H. IoT data feature extraction and intrusion detection system for smart cities based on deep migration learning. *Int. J. Inf. Manag.* **2019**, *49*, 533–545. [CrossRef]
21. Vinayakumar, R.; Alazab, M.; Srinivasan, S.; Pham, Q.-V.; Padannayil, S.K.; Simran, K. A Visualized Botnet Detection System Based Deep Learning for the Internet of Things Networks of Smart Cities. *IEEE Trans. Ind. Appl.* **2020**, *56*, 4436–4456. [CrossRef]
22. Kaur, G.; Lashkari, A.H.; Rahali, A. Intrusion Traffic Detection and Characterization using Deep Image Learning. In Proceedings of the IEEE 18th International Conference Dependable, Autonomic Secure Computing IEEE 18th International Conference Pervasive Intelligence Computing IEEE 6th Int. Conf. Cloud Big Data Computing IEEE 5th Cybe, Calgary, AB, Canada, 17–22 August 2020; pp. 55–62. [CrossRef]
23. Ullah, I.; Mahmoud, Q.H. A Two-Level Flow-Based Anomalous Activity Detection System for IoT Networks. *Electronics* **2020**, *9*, 530. [CrossRef]
24. Ullah, I.; Mahmoud, Q.H. A Two-Level Hybrid Model for Anomalous Activity Detection in IoT Networks. In Proceedings of the 16th IEEE Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 11–14 January 2019; pp. 1–6. [CrossRef]
25. Yang, J.; Lim, H. Deep Learning Approach for Detecting Malicious Activities over Encrypted Secure Channels. *IEEE Access* **2021**, *9*, 39229–39244. [CrossRef]
26. Ran, Z.; Zheng, D.; Lai, Y.; Tian, L. Applying Stack Bidirectional LSTM Model to Intrusion Detection. *Comput. Mater. Contin.* **2020**, *65*, 309–320. [CrossRef]

27. Ahmad, I.; Alsemmeari, R.A. Towards Improving the Intrusion Detection through ELM (Extreme Learning Machine). *Comput. Mater. Contin.* **2020**, *65*, 1097–1111. [CrossRef]
28. Ling, J.; Zhu, Z.; Luo, Y.; Wang, H. An intrusion detection method for industrial control systems based on bidirectional simple recurrent unit. *Comput. Electr. Eng.* **2021**, *91*, 107049. [CrossRef]
29. Kunang, Y.N.; Nurmaini, S.; Stiawan, D.; Suprapto, B.Y. Attack classification of an intrusion detection system using deep learning and hyperparameter optimization. *J. Inf. Secur. Appl.* **2021**, *58*, 102804. [CrossRef]
30. Sicato, J.C.S.; Singh, S.K.; Rathore, S.; Park, J.H. A comprehensive analyses of intrusion detection system for IoT environment. *J. Inf. Process. Systems* **2020**, *16*, 975–990. [CrossRef]
31. Wu, Z.; Wang, J.; Hu, L.; Zhang, Z.; Wu, H. A network intrusion detection method based on semantic Re-encoding and deep learning. *J. Netw. Comput. Appl.* **2020**, *164*, 102688. [CrossRef]
32. Sheu, J.-S.; Chen, I.-C.; Liao, Y.-S. Realization of Internet of Things Smart Appliances. *Intell. Autom. Soft Comput.* **2019**. [CrossRef]
33. Tariq, U.; Taiq, U.; Thapa, K. Intrusion Detection and Anticipation System (IDAS) for IEEE 802.15.4 Devices. *Intell. Autom. Soft Comput.* **2018**, 1–13. [CrossRef]
34. Aminanto, M.E.; Choi, R.; Tanuwidjaja, H.C.; Yoo, P.D.; Kim, K. Deep Abstraction and Weighted Feature Selection for Wi-Fi Impersonation Detection. *IEEE Trans. Inf. Forensics Secur.* **2017**, *13*, 621–636. [CrossRef]
35. Lin, S.Z.; Shi, Y.; Xue, Z. Character-Level Intrusion Detection Based On Convolutional Neural Networks. In Proceedings of the IEEE 2018 International Joint Conference on Neural Networks (IJCNN), Rio, Brazil, 8–13 July 2018; pp. 1–8. [CrossRef]
36. Zhou, Y.; Han, M.; Liu, L.; He, J.S.; Wang, Y. Deep learning approach for cyberattack detection. In Proceedings of the IEEE INFOCOM 2018—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Honolulu, HI, USA, 15–19 April 2018; pp. 262–267. [CrossRef]
37. Diro, A.; Chilamkurti, N. Leveraging LSTM Networks for Attack Detection in Fog-to-Things Communications. *IEEE Commun. Mag.* **2018**, *56*, 124–130. [CrossRef]
38. Pratomo, B.A.; Burnap, P.; Theodorakopoulos, G. Unsupervised Approach for Detecting Low Rate Attacks on Network Traffic with Autoencoder. In Proceedings of the 2018 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), Glasgow, UK, 11–12 June 2018; pp. 1–8. [CrossRef]
39. Roy, B.; Cheung, H. A Deep Learning Approach for Intrusion Detection in Internet of Things using Bi-Directional Long Short-Term Memory Recurrent Neural Network. In Proceedings of the 28th International Telecommunication Networks and Applications Conference (ITNAC), Sydney, Australia, 21–23 November 2018; pp. 1–6. [CrossRef]
40. Vinayakumar, R.; Alazab, M.; Soman, K.P.; Poornachandran, P.; Al-Nemrat, A.; Venkatraman, S. Deep Learning Approach for Intelligent Intrusion Detection System. *IEEE Access* **2019**, *7*, 41525–41550. [CrossRef]
41. Amouri, A.; Alaparthy, V.T.; Morgera, S.D. A Machine Learning Based Intrusion Detection System for Mobile Internet of Things. *Sensors* **2020**, *20*, 461. [CrossRef] [PubMed]
42. Nagisetty, A.; Gupta, G. Framework for Detection of Malicious Activities in IoT Networks using Keras Deep Learning Library. In Proceedings of the IEEE 3rd International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 27–29 March 2019; pp. 633–637. [CrossRef]
43. Otoum, Y.; Liu, D.; Nayak, A. DL-IDS: A deep learning–based intrusion detection framework for securing IoT. *Trans. Emerg. Telecommun. Technol.* **2019**. [CrossRef]
44. Zhang, Y.; Li, P.; Wang, X. Intrusion Detection for IoT Based on Improved Genetic Algorithm and Deep Belief Network. *IEEE Access* **2019**, *7*, 31711–31722. [CrossRef]
45. Lopez-Martin, M.; Carro, B.; Sanchez-Esguevillas, A.; Lloret, J. Shallow neural network with kernel approximation for prediction problems in highly demanding data networks. *Expert Syst. Appl.* **2019**, *124*, 196–208. [CrossRef]
46. Parker, L.R.; Yoo, P.D.; Asyhari, T.A.; Chermak, L.; Jhi, Y.; Taha, K. Demise: Interpretable deep extraction and mutual information selection techniques for IoT intrusion detection. In Proceedings of the 14th International Conference on Availability, Reliability, and Security, Canterbury, UK, 26–29 August 2019; pp. 1–10. [CrossRef]
47. Nguyen, T.D.; Marchal, S.; Miettinen, M.; Fereidooni, H.; Asokan, N.; Sadeghi, A.-R. DÏoT: A Federated Self-learning Anomaly Detection System for IoT. In Proceedings of the 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), Dallas, TX, USA, 7–9 July 2019; pp. 756–767. [CrossRef]
48. Li, F.; Shinde, A.; Shi, Y.; Ye, J.; Li, X.-Y.; Song, W.Z. System Statistics Learning-Based IoT Security: Feasibility and Suitability. *IEEE Internet Things J.* **2019**, *6*, 6396–6403. [CrossRef]
49. Rezvy, S.; Luo, Y.; Petridis, M.; Lasebae, A.; Zebin, T. An efficient deep learning model for intrusion classification and prediction in 5G and IoT networks. In Proceedings of the 2019 53rd Annual Conference on Information Sciences and Systems (CISS), Johns Hopkins Univrsity, Baltimore, MD, USA, 20–22 March 2019; pp. 1–6. [CrossRef]
50. Hwang, R.-H.; Peng, M.-C.; Nguyen, V.-L.; Chang, Y.-L. An LSTM-Based Deep Learning Approach for Classifying Malicious Traffic at the Packet Level. *Appl. Sci.* **2019**, *9*, 3414. [CrossRef]
51. Tian, Q.; Li, J.; Liu, H. A Method for Guaranteeing Wireless Communication Based on a Combination of Deep and Shallow Learning. *IEEE Access* **2019**, *7*, 38688–38695. [CrossRef]
52. Ge, M.; Fu, X.; Syed, N.F.; Baig, Z.; Teo, G.; Robles-Kelly, A. Deep Learning-Based Intrusion Detection for IoT Networks. In Proceedings of the 2019 IEEE 24th Pacific Rim International Symposium on Dependable Computing (PRDC), Kyoto, Japan, 1–3 December 2019; p. 256. [CrossRef]

53. Suwannalai, E.; Polprasert, C. Network Intrusion Detection Systems Using Adversarial Reinforcement Learning with Deep Q-network. In Proceedings of the IEEE 2020 18th International Conference on ICT and Knowledge Engineering (ICT&KE), Bangkok, Thailand, 18–20 November 2020; pp. 1–7. [CrossRef]
54. Lin, M.; Zhao, B.; Xin, Q. ERID: A Deep Learning-based Approach towards Efficient Real-Time Intrusion Detection for IoT. In Proceedings of the 2020 IEEE Eighth International Conference on Communications and Networking (ComNet), Hammamet, Tunisia, 27–30 October 2020; pp. 1–7. [CrossRef]
55. Almiani, M.; AbuGhazleh, A.; Al-Rahayfeh, A.; Atiewi, S.; Razaque, A. Deep recurrent neural network for IoT intrusion detection system. *Simul. Model. Pr. Theory* **2020**, *101*, 102031. [CrossRef]
56. Li, Y.; Xu, Y.; Liu, Z.; Hou, H.; Zheng, Y.; Xin, Y.; Zhao, Y.; Cui, L. Robust detection for network intrusion of industrial IoT based on multi-CNN fusion. *Measurement* **2020**, *154*, 107450. [CrossRef]
57. Rezaeipanah, A.; Afsoon, E.; Ahmadi, G. Improving the Performance of Intrusion Detection Systems Using the Development of Deep Neural Network Parameters. In Proceedings of the 10th IEEE International Conference on Computer and Knowledge Engineering (ICCKE), Square, Iran, 29–30 October 2020; pp. 278–283. [CrossRef]
58. Ferrag, M.A.; Maglaras, L.; Moschoyiannis, S.; Janicke, H. Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. *J. Inf. Secur. Appl.* **2020**, *50*, 102419. [CrossRef]
59. Tian, Z.; Luo, C.; Qiu, J.; Du, X.; Guizani, M. A Distributed Deep Learning System for Web Attack Detection on Edge Devices. *IEEE Trans. Ind. Inform.* **2019**, *16*, 1963–1971. [CrossRef]
60. Ge, M.; Syed, N.F.; Fu, X.; Baig, Z.; Robles-Kelly, A. Towards a deep learning-driven intrusion detection approach for Internet of Things. *Comput. Netw.* **2021**, *186*, 107784. [CrossRef]
61. Wang, B.; Su, Y.; Zhang, M.; Nie, J. A Deep Hierarchical Network for Packet-Level Malicious Traffic Detection. *IEEE Access* **2020**, *8*, 201728–201740. [CrossRef]
62. Alzahrani, H.; Abulkhair, M.; Alkayal, E. A Multi-Class Neural Network Model for Rapid Detection of IoT Botnet Attacks. *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*. [CrossRef]
63. Priyanga, S.; Krithivasan, K.; Pravinraj, S.; Shankar Sriram, V.S. Detection of Cyberattacks in Industrial Control systems using Enhanced Principal Component Analysis and Hypergraph based Convolution Neural Network (EPCA-HG-CNN). *IEEE Trans. Ind. Appl.* **2020**, *56*, 4394–4404. [CrossRef]
64. Ezeme, O.M.; Mahmoud, Q.H.; Azim, A. Design and Development of AD-CGAN: Conditional Generative Adversarial Networks for Anomaly Detection. *IEEE Access* **2020**, *8*, 177667–177681. [CrossRef]
65. Khamis, R.A.; Matrawy, A. Evaluation of Adversarial Training on Different Types of Neural Networks in Deep Learning-based IDSs. In Proceedings of the IEEE 2020 International Symposium on Networks, Computers, and Communications (ISNCC), Montreal, QC, Canada, 16–18 June 2020; pp. 1–6. [CrossRef]
66. Pecori, R.; Tayebi, A.; Vannucci, A.; Veltri, L. IoT Attack Detection with Deep Learning Analysis. In Proceedings of the 2020 IEEE International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020; pp. 1–8. [CrossRef]
67. Yin, C.; Zhang, S.; Wang, J.; Xiong, N.N. Anomaly Detection Based on Convolutional Recurrent Autoencoder for IoT Time Series. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, 1–11. [CrossRef]
68. Kishore, R.; Chauhan, A. Evaluation of Deep Neural Networks for Advanced Intrusion Detection Systems. In Proceedings of the 2020 IEEE 4th International Conference on Electronics, Communication, and Aerospace Technology (ICECA), Kerala, India, 5–7 November 2020; pp. 1–8. [CrossRef]
69. Wang, Z.; Zeng, Y.; Liu, Y.; Li, D. Deep Belief Network Integrating Improved Kernel-Based Extreme Learning Machine for Network Intrusion Detection. *IEEE Access* **2021**, *9*, 16062–16091. [CrossRef]
70. Kanna, P.R.; Santhi, P. Unified Deep Learning approach for Efficient Intrusion Detection System using Integrated Spatial–Temporal Features. *Knowl. Based Syst.* **2021**, *226*, 107132. [CrossRef]
71. Zhao, R.; Li, Z.; Xue, Z.; Ohtsuki, T.; Gui, G. A Novel Approach based on Lightweight Deep Neural Network for Network Intrusion Detection. In Proceedings of the 2021 IEEE Wireless Communications and Networking Conference (WCNC), Nanjing, China, 29 March–1 April 2021; pp. 1–6. [CrossRef]
72. Ullah, I.; Mahmoud, Q.H. A filter-based feature selection model for anomaly-based intrusion detection systems. In Proceedings of the 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 11–14 December 2017; pp. 2151–2159. [CrossRef]
73. Ullah, I.; Mahmoud, Q.H. A hybrid model for anomaly-based intrusion detection in SCADA networks. In Proceedings of the 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 11–14 December 2017; pp. 2160–2167. [CrossRef]
74. 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. [CrossRef]
75. Ullah, I.; Mahmoud, Q.H. A Technique for Generating a Botnet Dataset for Anomalous Activity Detection in IoT Networks. In Proceedings of the 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Online, 11–14 October 2020; pp. 134–140. [CrossRef]
76. Kang, H.K.K.H.; Ahn, D.H.; Lee, G.M.; Yoo, J.D.; Park, K.H. IoT Network Intrusion Dataset. Available online: https://ieee-dataport.org/open-access/iot-network-intrusion-dataset (accessed on 21 May 2021).
77. Ullah, I.; Mahmoud, Q.H. A Scheme for Generating a Dataset for Anomalous Activity Detection in IoT Networks. In Proceedings of the Advances in Artificial Intelligence. Canadian AI 2020, Vancouver, BC, Canada, 13–15 May 2020; pp. 508–520. [CrossRef]

78. Hindy, H.; Bayne, E.; Bures, M.; Atkinson, R.; Tachtatzis, C.; Bellekens, X. Machine Learning Based IoT Intrusion Detection System: An MQTT Case Study. In Proceedings of the International Networking Conference, Rhodes, Greece, 19–21 September 2020; pp. 73–84. [CrossRef]
79. Ullah, I.; Mahmoud, Q.H. IoT Intrusion Detection Datasets. 2021. Available online: https://sites.google.com/view/iotdataset1 (accessed on 20 May 2021).
80. Stratosphere Laboratory. A Labeled Dataset with Malicious and Benign IoT Network Traffic. Agustin Parmisano, Sebastian Garcia, Maria Jose Erquiaga. Available online: https://www.stratosphereips.org/datasets-iot23 (accessed on 22 May 2021).