electronics

MDPI

Article

CNN-Based Network Intrusion Detection against Denial-of-Service Attacks

Jiyeon Kim [1], Jiwon Kim [2], Hyunjung Kim [2], Minsun Shim [2] and Eunjung Choi [2,*]

[1] Center for Software Educational Innovation, Seoul Women's University, Seoul 01797, Korea; jykim07@swu.ac.kr

[2] Department of Information Security, Seoul Women's University, Seoul 01797, Korea; jiwonmik@swu.ac.kr (J.K.); light0826@swu.ac.kr (H.K.); mshim1007@swu.ac.kr (M.S.)

* Correspondence: chej@swu.ac.kr; Tel.: +82-2-970-5339

Received: 28 April 2020; Accepted: 26 May 2020; Published: 1 June 2020

Abstract: As cyberattacks become more intelligent, it is challenging to detect advanced attacks in a variety of fields including industry, national defense, and healthcare. Traditional intrusion detection systems are no longer enough to detect these advanced attacks with unexpected patterns. Attackers bypass known signatures and pretend to be normal users. Deep learning is an alternative to solving these issues. Deep Learning (DL)-based intrusion detection does not require a lot of attack signatures or the list of normal behaviors to generate detection rules. DL defines intrusion features by itself through training empirical data. We develop a DL-based intrusion model especially focusing on denial of service (DoS) attacks. For the intrusion dataset, we use KDD CUP 1999 dataset (KDD), the most widely used dataset for the evaluation of intrusion detection systems (IDS). KDD consists of four types of attack categories, such as DoS, user to root (U2R), remote to local (R2L), and probing. Numerous KDD studies have been employing machine learning and classifying the dataset into the four categories or into two categories such as attack and benign. Rather than focusing on the broad categories, we focus on various attacks belonging to same category. Unlike other categories of KDD, the DoS category has enough samples for training each attack. In addition to KDD, we use CSE-CIC-IDS2018 which is the most up-to-date IDS dataset. CSE-CIC-IDS2018 consists of more advanced DoS attacks than that of KDD. In this work, we focus on the DoS category of both datasets and develop a DL model for DoS detection. We develop our model based on a Convolutional Neural Network (CNN) and evaluate its performance through comparison with an Recurrent Neural Network (RNN). Furthermore, we suggest the optimal CNN design for the better performance through numerous experiments.

Keywords: intrusion detection systems; denial of service; deep learning; convolutional neural network; recurrent neural network

1. Introduction

As cyberattacks evolve, attackers are exploiting unknown vulnerabilities and bypassing known signatures. One of the most representative network solutions is an intrusion detection system (IDS). There are two types of IDSs. One is misuse detection that detects attacks based on known signatures, and the other is anomaly detection which detects abnormal attacks based on normal use patterns. While misuse detection is difficult to detect unknown attacks, anomaly detection has the advantage of being able to detect unknown attacks. However, the anomaly detection has high false alarms because it is challenging to define a variety of normal use patterns. Deep Learning (DL) is a technique that compensates for these weaknesses by learning its own features through a deep neural network. Employing DL into IDS can compensate for the drawbacks of IDS. In other words, Machine Learning

(ML) and DL learns an intrusion set by itself and determines the normal use patterns, so that it can reduce the false alarms. In this paper, we employ DL into our IDS study to detect Denial-of-Service (DoS) attacks. The KDD CUP 1999 dataset (KDD) developed by Defense Advanced Research Projects Agency (DARPA) is the most used dataset for IDS evaluation [1]. KDD classifies attacks into four broad categories, such as DoS, User to Root (U2R), Remote to Local (R2L) and Probing. KDD was generated by injecting these kinds of attacks into each category. Numerous IDS studies have been using KDD as a dataset since machine learning is actively employed into IDS studies. Most of these studies perform binary classification that classifies the entire KDD into attack and benign. They also carry out multiclass classification to classify the KDD into the four categories. In this work, we focus on individual attacks injected into KDD. Rather than distinguishing attacks from benign samples or classifying into the four attack categories, we classify individual attacks belonging to the same category by finding out the fine differences through DL. Among the four categories of KDD, only the DoS category has sufficient samples to train each attack. We use the DoS samples in not only KDD but also CSE-CIC-IDS 2018 for our development. CSE-CIC-IDS 2018 contains advanced DoS attacks including DoS attacks on an application layer. We develop a DL-based detection model for DoS attacks in the two datasets. Our model is based on a Convolutional Neural Network (CNN) and we perform binary classification and multiclass classification using the CNN-based model. Finally, we evaluate its performance by compared to a model based on a Recurrent Neural Network (RNN). Furthermore, we suggest a way of improving the performance of our model through numerous experiments. The remainder of this paper is organized as follows. We briefly review the two datasets that we use and investigate the trends of IDS studies employing machine learning and DL in Section 2. We design our CNN-based IDS model against DoS and evaluate our model under various scenarios in Sections 3 and 4, respectively. We also compare its performance with an RNN model in Section 5 and the conclusion is finally in Section 5.

2. Related Works

2.1. IDS Datasets

KDD dataset is the dataset which is developed in 1998 to evaluate the performance of IDS in DARPA [1]. It is the most used dataset in IDS studies since 1999 [2]. After MIT Lincoln Laboratory built military network environment with Air Force LAN (Local Area Network), they generated a variety of attacks and TCP/IP data [3] to simulate the LAN of the United States Air Force (USAF). Each record of the data has 41 network parameters and all data belong to one of the categories among 4 types of attacks (DoS, U2R, R2L, Probing). DoS is a service denial attack that exhausts the network resources and disturbs normal connections. U2R is an attack that accesses to a victim system and obtains an administrator access. After getting the access, it abuses the system. R2L is an attack that attempts to access a remote system to obtain a victim's account. Probing is an attack that analyzes a victim system to get information about the system. The actual injected attacks for each category differ according to the type of KDD. There are three types of KDD. The first one is the whole KDD which has the largest samples generated by injecting 22 types of attacks. The second one is the 10% KDD that is separated about 10% from the whole KDD. The last one is the corrected KDD which is injected 15 types of attacks more than the whole KDD. We use the corrected KDD in this paper for the multiclass classification of attacks. The attacks injected to the corrected KDD are as shown in Table 1. We randomly and exclusively divide the training and testing samples by 70 to 30 for each experiment, so the training and testing sets are independent each other.

ISCXIDS 2012 is a dataset created by the Information Security Centre of Excellence at the University of New Brunswick [4]. As the DoS attack evolved not only in the network layer which was relatively easy to detect but also in application layer that is difficult to detect, ISCXIDS 2012 includes both types of attacks. In ISCXIDS 2012, there are several attacks including DoS attacks, such as HTTP DoS and DDoS using an IRC Botnet. In CICIDS 2017, Quadratic Discriminate, Slowloris, Hulk, Goldeneye, and LOIT were added from ISCXIDS 2012. Furthermore, in CSE-CIC-IDS 2018, Heartleech, LOIC UDP, TCP,

and HTTP are added from CICIDS 2017. Therefore, CSE-CIC-IDS 2018 is an advanced data set that includes all previous ISCXIDS 2012 and CICIDS 2017.

Table 1. Type of attacks in corrected KDD.

Classification	Name of Attack	Num. of Samples
Denial of Service (DoS)	Neptune, smurf, pod, teardrop, land, back, apache2, udpstorm, processtable, mail-bomb	229,853
User to Root (U2R)	buffer-overflow, load-module, perl, rootkit, xterm, ps, sqlattack	70
Remote to Local (R2L)	guess-password, ftp-write, imap, phf, multihop, spy, warezclient	16,347
Probing	port-sweep, ip-sweep, nmap, satan, saint, mscan	4166

In this paper, we use 'corrected KDD' which is the most widely and commonly used for IDS studies. Furthermore, we use CSE-CIC-IDS 2018 which is the most up-to-date IDS dataset and consists of advanced DoS attacks such as Slowloris and Slowhttptest.

2.2. Trends of IDS Studies

There are several works that studied about IDS [5]. Jing-Xin et al. [6] use Artificial Neural Network (NN) to Network IDS (NIDS) and they propose the NIDS prototype. Manso et al. [7] propose IDS based on Software Defined Network (SDN). The proposed IDS detects DDoS attacks and informs to SDN controller. Karim et al. [8] study about experimental performance of Snort-based IDS (S-IDS) in network. Xu et al. [9] suggest Distributed Denial-of-Service (DRDoS) detection and defense model based on Deep Forest model (DDDF). In particular, they focus on attacks in Internet of Things (IoT) devices and big data environment. Also, there are several studies about anomaly detection schemes for Industrial Wireless Sensor Networks (IWSNs) based on machine learning [10]. Zhang et al. [11] suggest Hierarchical Intrusion Detection System (HIDS) based on statistical preprocessing and NN classification. Koc et al. [12] show that Hidden Naive Bayes (HNB), which is one of the data mining models, can be used in IDS. Hodo et al. [13] suggest analysis about threat of IoT based on Artificial Neural Network (ANN) to detect DoS/DDoS attacks. They especially focused on classification about normal and threat patterns. Chung et al. [14] show hybrid IDS using intelligent dynamic swarm-based rough set (IDS-RS) or feature selection and simplified swarm optimization or intrusion data classification. They use KDD as dataset and find the proposed model can increase the performance. Aydin et al. [15] suggest hybrid IDS by putting two IDS systems together which is misuse detection and anomaly detection. Al-Jarrah et al. [16] use Time Delayed Neural Network (TDNN) structure to maximize the recognition rate of network attacks. In addition, Karthick et al. [17] propose IDS based on problematic classifier and Hidden Markov Model (HMM). Wahab et al. [18] point out the problem of maximizing the detection of Virtual Machine-based DDoS attacks in a cloud system and propose its trust model. They also propose defense and detection mechanisms especially for the cloud-based systems in the further study [19]. Chen et al. [20] propose a Low-rate Denial-of-Service (LDoS) attack detection model using Hilbert-Huang and trust evaluation.

2.3. Trends of IDS Studies Based on Machine Learning and Deep Learning

Numerous IDS studies that employ machine learning have used KDD dataset [21,22]. Sabhnani et al. [23] evaluate the performance of a comprehensive set of pattern recognition and machine learning algorithms in four attack categories of KDD. They employ MLP (Multilayer Perceptron), K-means clustering, and Gaussian classifier and suggest the optimal algorithm showing the high detection accuracy by 4 types of attacks. The experimental result shows that DoS and U2R are the most accurate when applying K-means clustering. R2L and probing have the highest accuracy when using Gaussian classifier and MLP, respectively. Mulay et al. [24] suggest a model combining Support Vector Machine (SVM) and a decision tree. They evaluate the proposed model using KDD and then show that the combined model has a higher accuracy and reduces training and testing time than that of a model

with an SVM or decision tree. Further works on KDD improve the performance of intrusion detection by the kernel type of SVM [25,26]. Hasan et al. [25] analyze the type of kernel with a best performance for an SVM-based intrusion detection. They generate new datasets called KDD99Train+ and KDD99Test+ by preprocessing duplicated data belonging to both training and testing datasets. Using the newly generated datasets, they found out that the ability of the SVM classification depends on the type of kernel and hyperparameter setting. Yao et al. [26] propose an enhanced SVM model of weighted kernel functions based on the characteristics of training dataset. According to the experimental evaluation, they find out that the performance of the proposed model is better than the existing SVM model.

Kim et al. [27] compare the detection accuracy using DL as well as machine learning, such as SVM, decision tree, NN, and CNN models. They carry out binary classification that classifies KDD into benign and attack. They also perform multiclass classification that classifies the dataset into the 4 categories. While all the four models have high accuracy in the binary classification, the performance of intrusion detection in the multiclass classification varies depending on the type of models. According to the experimental results, the performances of decision tree and NN are lower than that of SVM and CNN. Yin et al. [28] perform the binary classification and multiclass classification based on RNN which is one of DL models. The experimental results show that the accuracy of binary classification is higher than that of multiclass classification. In addition, they find out that hyperparameters such as hidden nodes and learning rate affect the detection accuracy.

Further studies [29,30] that improve the performance of KDD classification with the proposed model have been addressed. Sheikhan et al. [29] propose a three-layers RNN architecture, which classifies features as input and attack types, as a misused-based IDS. They compare the proposed model with other machine learning methods in terms of Detection Rate (DR), False Alarm Rate (FAR) and Cost Per Example (CPE). Their experimental results show that the proposed model improves the classification rate, especially in R2L attacks. They also present better DR and CPE when compared to Multilayer Perceptron and Elman-based intrusion detectors. Bontemps et al. [30] propose a new collective anomaly detection model based on Long-short Term Memory RNN (LSTM-RNN). They show that various output reactions depend on the number of inputs of LSTM-RNN and the proposed model is effective in detecting group anomalies.

Numerous studies [31–33] that detect attacks in binary and multiple categories based on CNN have also been addressed. Khan et al. [31] point out the disadvantages of using machine learning algorithms to obtain intrusion detection models. They also propose ways to combine CNN-based network intrusion detection models with soft max algorithms. They evaluate the proposed model using KDD and the experimental results show that the model is more efficient in detecting intrusions compared to the SVM and Deep Belief Network (DBN) algorithms. Tavallaee et al. [34] select some records from KDD and propose new dataset called 'NSL-KDD'. Several pieces of research use this NSL-KDD. Li et al. [32] propose image conversion methods using NSL-KDD data and analyze how CNN models automatically learn the transformed intrusion data. They find out that the CNN model is sensitive to the transformation of data images and can be used for intrusion detection techniques. Gao et al. [35] propose IDS-combined incremental Extreme Learning Machine (I-ELM) with an Adaptive Principle Component (A-PCA) using NSL-KDD and UNSW-NB15 dataset. Chu et al. [36] also use NSL-KDD to detect the attack. Upadhyay et al. [33] use KDD with randomly selected 36 features from 41 number of KDD features. They transform the dataset into/1 × 6 size of images and then store the remaining features in different variables to train the CNN model. The experimental results show that the proposed model results in less than 2% errors in detecting intrusions.

Fares et al. [37] study to achieve higher detection rates and lower false alarm rates using Niyaz et al. [38] employ Self-Taught Learning (STL) algorithm to develop an IDS. Tang et al. [39] suggest IDS in SDN based on Deep Neural Network (DNN) by using NSL-KDD. Ingre et al. [40] employ ANN for intrusion detection using NSL-KDD. The proposed model consists of tansig transfer function, Levenberg-Marquardt (LM), and BFGsquasi-Newton Backpropagation (BGFS) algorithm. The experimental results show that the performance of binary classification with the LM and BFGS

algorithms is higher than that of multiclass classification with five categories. Also, Erol et al. [41] propose IDS based on ANN by using KDD. In addition, Ibrahim et al. [42] use Distributed Time-Delay ANN to model the network IDS. Tan et al. [43] suggest IDS by using Synthetic Minority Oversampling Technique (SMOTE) to balance the dataset and the random forest algorithm to train the classifier for intrusion detection. Farnaaz et al. [44] also use forest algorithm to build their IDS. Ye [45] uses Principal Component Neural Network (PCNN) and Multiclass SVM (MSSVM) algorithm to detect key features in network intrusion signals with KDD. In addition, Ali et al. [46] develop Fast Learning Network (FLN) based on Particle Swarm Optimization (PSO) to solve the problem of IDS in different approach. They name the proposed model as PSO-FLN and show that PSO-FLN has higher testing accuracy compared to meta-heuristic algorithm for training ELM and FLN classifier. Yang et al. [47] propose the LM-Back Propagation (BP) NN model to increase performance of IDS in IoT. Compared to PSO-BP model and BP NN model, the proposed model shows higher DR and less false alarms. Seo [48] proposes a data preprocessing technique to control the ratio of learning data in sparse classes to increase the performance of the model. He also evaluates his suggestion based on k-nearest Neighbor, SVM, and decision tree, and the experimental results show that the performance with preprocessed data has a higher accuracy than that of with the original data. Amma et al. [49] propose a DoS detection model based on Deep Radial Intelligence (DeeRaI) with Cumulative Incarnation (CUI). They use NSL-KDD and UNSW NB15 as datasets.

There are numerous IDS studies using the ISCX dataset. Koay et al. [50] propose a novel multi-classifier system based on novel entropy and machine learning classifier using ISCXIDS 2012. Idhammad et al. [51] propose semi-supervised DDoS detection based on entropy estimation, co-clustering, information gain ratio, and extra-trees ensemble classifier. Yassin et al. [52] suggest K-means clustering and Naive Bayes combined KMC + NBC-based IDS. Soheily-Khah et al. [53] propose K-means, random forest combined kM-RF-based hybrid intrusion detection. In addition, Faker et al. [54] propose IDS based on DNN, Random Forest, and Gradient Boosting Tree classification using CICIDS 2017. Zhang et al. [55] propose an IDS called DCF-IDS by combining DL network and gcForest (deep random forest). Zhou et al. [56] analyzed CSE-CIC-IDS 2018 using 6 types of ML algorithms such as Random Forest, Naive Bayes, Decision Tree, Neural Network (MLP), Quadratic Discriminant, and K-Neighbors. Kim et al. [57] propose a CNN-based IDS using CSE-CIC-IDS 2018. Chadza et al. [58] carry out predicting intrusions using HMM. They use CSE-CIC-IDS 2018 and evaluate three initialization techniques such as uniform, random, and count-based.

We focus on the DoS category in not only KDD which is the most widely used IDS dataset but also CSE-CIC-IDS 2018, the most up-to-date IDS dataset.

3. Designing IDS Model Based on CNN

In this Section, we explain the training and testing datasets we use, and design our IDS model based on CNN.

3.1. DoS Datasets

The samples of DoS attacks in KDD and CSE-CIC-IDS 2018 are as shown in Table 2.

The smurf attack which has the largest samples in KDD is an attack that exhausts network resources by transmitting massive Internet Control Message Protocol (ICMP) packets to a victim system. The attack takes place by broadcasting with a forged IP address to the victim system. Neptune attack, which has the second largest samples, is a SYN flooding attack which induces imperfect TCP session so that it exhausts resources of the victim server. Except Smurf and Neptune attacks, several attacks also belong to the DoS category of KDD. However, the samples are not enough to make reliable training models. In CSE-CIC-IDS 2018, there are several DoS attacks as shown in Table 2. These DoS attacks are more advanced DoS attacks than that of KDD. We perform the binary and multiclass classifications on DoS attacks belonging to CSE-CIC-IDS 2018 as well as KDD. We observe whether our DL model

classifies the minute features of DoS attacks which are in the same category as well as features of attack and benign classes.

Table 2. Number of DoS samples in KDD and CSE-CIC-IDS 2018.

Dataset	Classification	Total
KDD	Benign	60,591
	Neptune Attack	58,001
	Smurf Attack	164,091
CSE-CIC-IDS 2018	Benign	9,108,759
	DoS-Hulk Attack	461,912
	DoS-SlowHTTPTest	139,890
	DoS-GoldenEye	41,508
	DoS-Slowloris	10,990
	DDoS-LOIC-HTTP	576,191
	DDoS-HOIC	686,012

3.2. Creating the Attack Images

KDD consists of 41 traffic features and 1 feature which determines where each data belongs to. In the 41 number of traffic features, 38 of them are represented in numerical features and 3 of them are represented in symbolic features. We transform the symbolic data to numerical data to unify all data formats. The 3 features of the symbolic type are the protocol type of a TCP/IP layer, the service type of a target system and flag type which shows the connection state of the session. There are three types of the protocol type such as ICMP, TCP, and UDP. These protocols are transformed to 3-dimensional vector (1,0,0), (0,1,0) and (0,0,1) through one-hot encoding. Likewise, 67 types of the service including HTTP and FTP are transformed to 67-dimensional vector and the 9 features of the flag type are transformed to 9-dimensional vector. We finally generate the 79-dimensional vector through these transformations. When this 79-dimensional vector is combined with 38 features that have the original numerical features, the 117-dimensional vector is finally generated. In addition, we rescale all the numerical features to be between 0 to 255 to convert the 117-dimensional vector into images with 13×9 pixels. Each color channel of the image should be represented with the value between 0 to 255. We then feed these images to our CNN model. The reason we convert the numerical samples into images is that CNN is a DL model for image training.

In this paper, we generate two types of image datasets. One is an RGB set which has 3 color channels (Red, Green, and Blue) and the other one is a grayscale set that has a single channel. An RGB image is an overlaid structure of the three types of color images and is converted into an array of $M \times N \times 3$ pixels finally. M and N are the number of columns and rows, respectively [59]. We observe how accuracy varies depending on the type of image. When the 13×9 pixel of image is transformed to grayscale and RGB, $13 \times 9 \times 1$ and $13 \times 9 \times 3$ images will be generated, respectively.

CSE-CIC-IDS 2018 consist of 78 numerical features including destination port, type of protocol, and flow duration. We rescale these features and transform into grayscale and RGB images with 13×6. Figure 1 shows the steps of creating the attack images explained above.

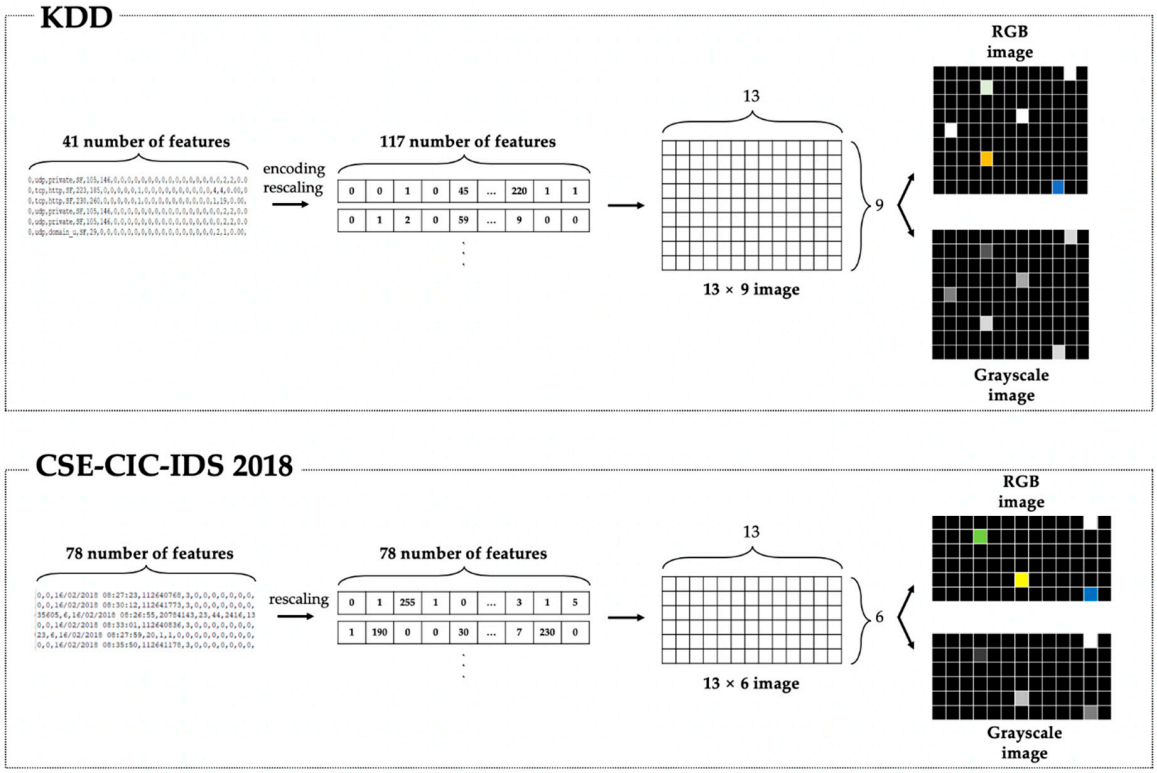


Figure 1. Steps of creating attack images.

3.3. Designing CNN Model

CNN is the most widely used DL model for image recognition, consisting of a convolution layer that extracts the features of the image and a fully connected layer that determines which class the input image belongs to. The convolution layer extracts the unique features of the image while keeping I/O and spatial information of the image and reduces the size of the feature data by adding a pooling layer to the convolution layer. An image is processed based on the following equation:

$$L' = \frac{L - K + 2P}{S} + 1 \tag{1}$$

L refers to the length of input image. K and P refer to the kernel size and zero which is filled by the level of dimension of both ends. Finally, S refers to a stride of the kernel on a convolution layer.

While multiple convolution layers may more effectively learn images with complex features, the number and performance of the convolutional layers are not always proportional. Because a correlation between the number of convolutional layers and its performance depends on the characteristics of the input images, we need to find out the optimal design through various designs and learning. We design our models considering hyperparameters such as the type of images (grayscale or RGB), the number of convolutional layers and the size of kernel, the number of weights used to design a hidden layer in the convolution layer. Figure 2 shows the structure of our CNN model. In addition, we develop our model using Python programming language with Tensorflow [60].

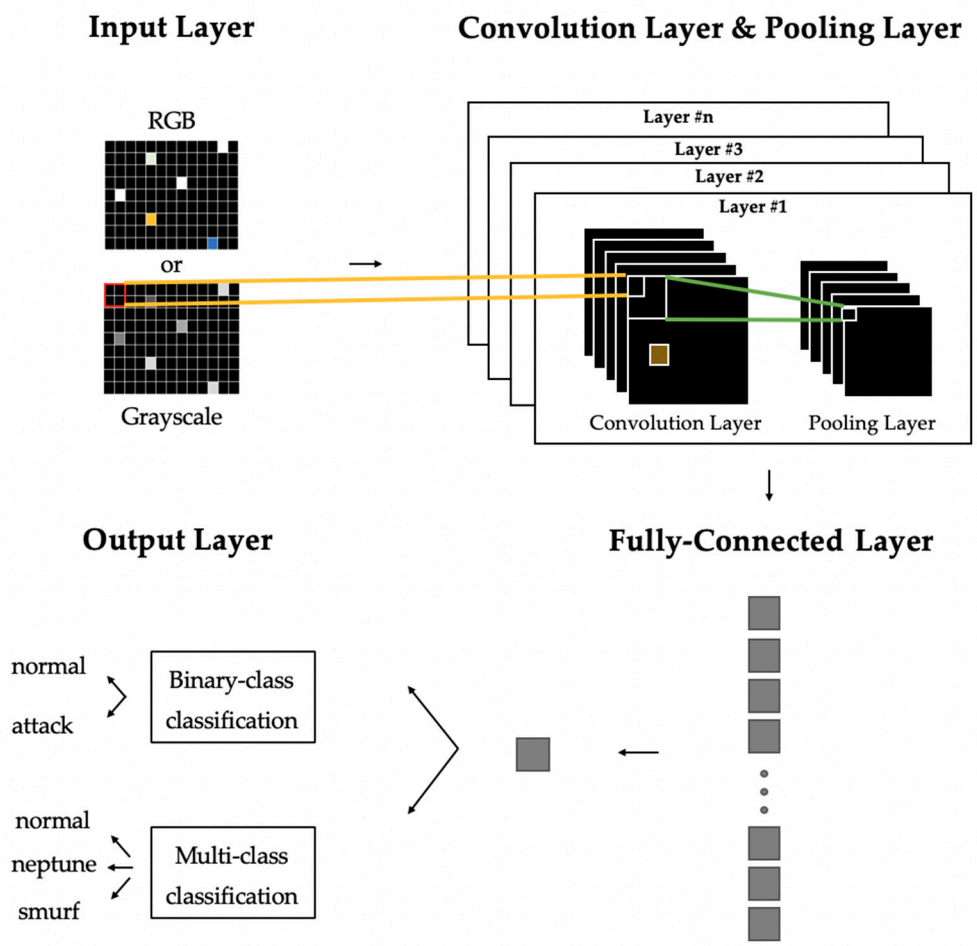


Figure 2. Design of our CNN model.

4. Experimental Evaluation

In this Section, the DoS dataset described in Section 3 is trained based on our CNN model and the performances of binary and multiclass classification are evaluated.

4.1. Scenario

The proposed CNN model receives grayscale or RGB images as its input. The CNN model is also possible to change two more parameters such as the number of convolutional layer and size of kernel as described in Section 3. We call these parameters as hyperparameters and create 18 kinds of scenarios considering the hyperparameters as shown in Table 3.

The CNN model consists of 1, 2, or 3 convolutional layers, and the number of kernels corresponding to the number of neurons per layer increases by a multiple of 2. In addition, the kernel size is usually set to 3×3. However, we set 3×3 as a median value and do experiment on sizes of 2×2 and 4×4 to find out the optimal size. The kernel generates a feature map by moving over the image as much as stride which is designated value. We set the stride to 1 [61] to extract the feature densely. Figure 3 shows examples of our CNN design. We chose 6 scenarios (RGB-1, GS-1, RGB-5, GS-5, RGB-9 and GS-9) that can show various CNN designs with a different number of layers, kernels, and color channels.

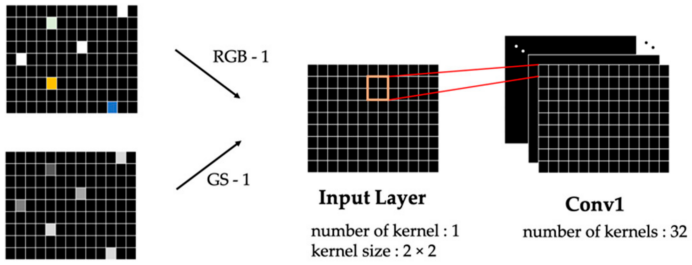
Experiments are performed with binary and multiclass classifications. Table 4 shows the detection classes for each classification.

Table 3. 18 number of scenarios considering hyperparameters.

No.	Scenario	Num. of Conv. Layer	Kernel Size	Num. of Kernels		
				Conv. Layer1	Conv. Layer2	Conv. Layer3
1	RGB-1	1	2 × 2	32	-	-
2	RGB-2	2	2 × 2	32	64	-
3	RGB-3	3	2 × 2	32	64	128
4	RGB-4	1	3 × 3	32	-	-
5	RGB-5	2	3 × 3	32	64	-
6	RGB-6	3	3 × 3	32	64	128
7	RGB-7	1	4 × 4	32	-	-
8	RGB-8	2	4 × 4	32	64	-
9	RGB-9	3	4 × 4	32	64	128
10	GS-1	1	2 × 2	32	-	-
11	GS-2	2	2 × 2	32	64	-
12	GS-3	3	2 × 2	32	64	128
13	GS-4	1	3 × 3	32	-	-
14	GS-5	2	3 × 3	32	64	-
15	GS-6	3	3 × 3	32	64	128
16	GS-7	1	4 × 4	32	-	-
17	GS-8	2	4 × 4	32	64	-
18	GS-9	3	4 × 4	32	64	128

Table 4. Classes for Binary and Multiclass classifications.

Class	KDD		CSE-CIC-IDS 2018	
	Binary	Multiclass	Binary	Multiclass
1	benign	benign	benign	benign
2	attack	smurf	attack	DoS-Hulk
3	-	neptune	-	DoS-SlowHTTPTest
4	-	-	-	DoS-GoldenEye
5	-	-	-	DoS-Slowloris
6	-	-	-	DDoS-LOIC-HTTP
7	-	-	-	DDoS-HOIC



1 Convolution Layer

(a)

Figure 3. *Cont.*

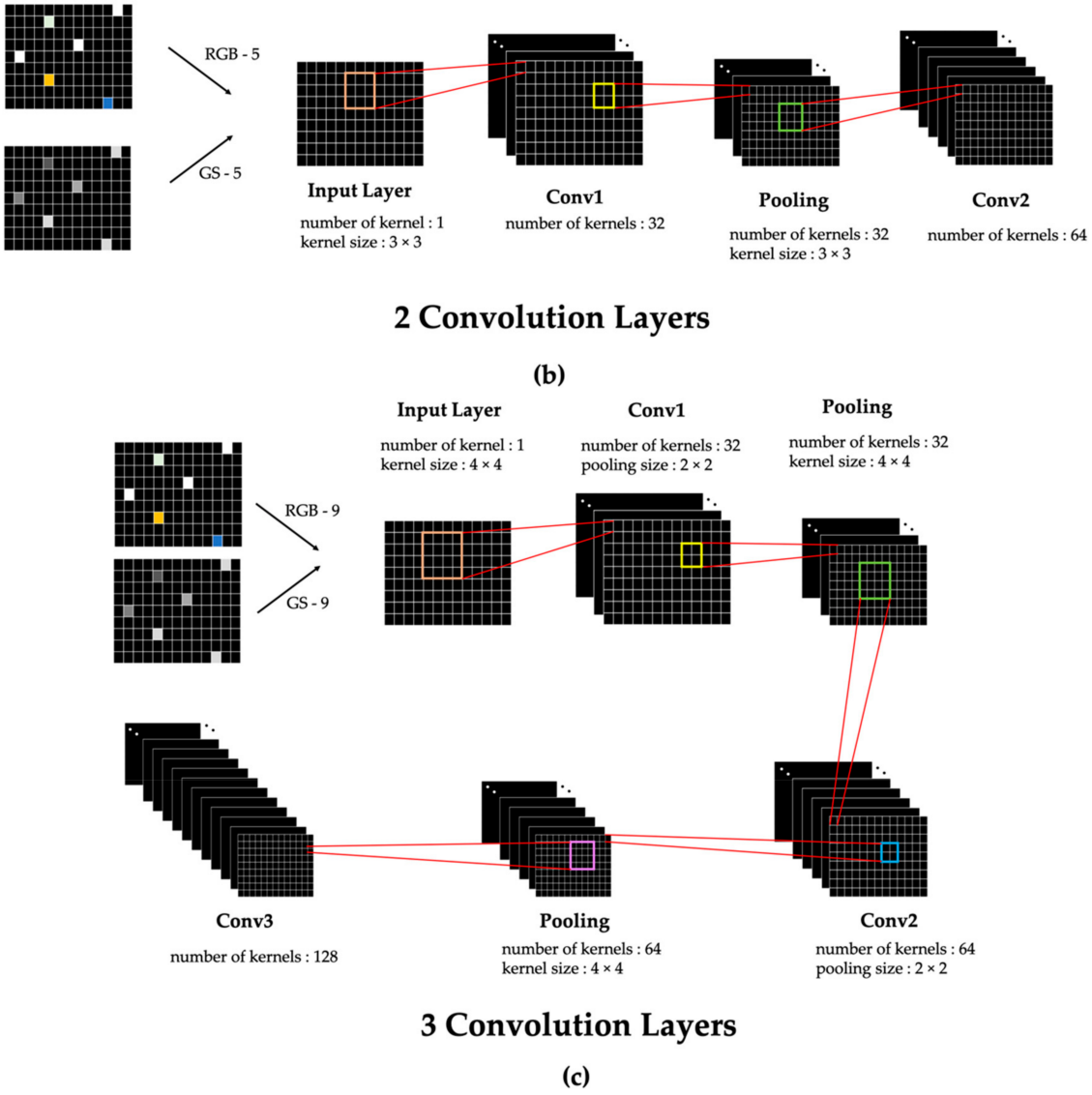


Figure 3. Example CNN designs: (**a**) RGB-1 and GS-1 scenarios with a single convolution layer and 2 × 2 size of kernel, (**b**) RGB-5 and GS-1 scenarios with 2 convolution layers and 3 × 3 size of kernel, (**c**) RGB-9 and GS-9 scenarios with 3 convolution layers and 4 × 4 size of kernel.

4.2. Evaluation of Binary Classification

The experimental results of binary classification for 18 experimental scenarios show that most of scenarios have more than 99% of accuracy. To evaluate the performance of the proposed model, we calculate F1-score. F-score is an index that implies both precision and recall. F1-score is the value that is given a weighted beta value of 1 for precision when calculating the F-score. F1-score is defined as following Equation (2).

$$\text{F1} - \text{score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \text{ where precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \text{ and, recall} = \frac{\text{TP}}{\text{FN} + \text{TP}} \quad (2)$$

True Positive (TP) is the number of samples which are properly classified as benign. False Negative (FN) is the number of samples falsely detect benign data as attack. False Positive (FP) refers to the number of samples that incorrectly predict an attack as benign. True Negative (TN) indicates the number of samples which are properly detected as an attack.

In deep learning-based learning, the results may vary slightly from experiment to experiment. However, the accuracy of this paper shows the average value of the results tested five times for each scenario, so that even a slight difference can reveal differences in characteristics according to the hyperparameters. In case of KDD, especially in binary classification, RGB scenarios show the highest performance in order of RGB-3, RGB-8, and RGB-6. For the grayscale scenarios, the performance is high in order of GS-8, GS-6, and GS-3. In the case of CSE-CIC-IDS 2018, RGB scenarios show the highest performance in order of RGB-8, RGB-9, and RGB-6. The detection performance with the grayscale images, the performance is high in order of GS-8, GS-3, GS-6, and GS-9, i.e., regardless of the RGB and grayscale images in binary classification, when the kernel size is 2×2 or 3×3, the performance of the scenario of three convolutional layers is the best. When the kernel size is 4×4, the scenario of two convolutional layers has the best performance. A more detailed analysis of binary classification based on hyperparameters is as follows.

4.2.1. RGB Vs Grayscale

Table 5 shows a comparison of the number of correct one, the number of wrong one and accuracy of the 9 RGB scenarios (RGB 1~9) and the grayscale scenarios (GS 1~9), respectively in case of KDD. Given that the RGB scenario is more accurate than the grayscale scenario on all graphs, we can see that generating an RGB image of DoS is a way to improve the detection performance.

Table 5. Experimental results of 18 scenarios for KDD.

RGB Scenarios	TP and TN	FP and FN	Accuracy	GS Scenarios	TP and TN	FP and FN	Accuracy
RGB-1	282,596	87	0.999693	GS-1	282,502	181	0.999359
RGB-2	282,607	76	0.999731	GS-2	282,580	103	0.999637
RGB-3	282,664	19	0.999932	GS-3	282,620	63	0.999778
RGB-4	282,589	94	0.999667	GS-4	282,582	101	0.999642
RGB-5	282,642	41	0.999856	GS-5	282,602	81	0.999712
RGB-6	282,648	35	0.999875	GS-6	282,623	60	0.999788
RGB-7	282,614	69	0.999755	GS-7	282,537	146	0.999484
RGB-8	282,661	22	0.999922	GS-8	282,646	37	0.999868
RGB-9	282,630	53	0.999814	GS-9	282,590	93	0.999670

In the experimental result of CSE-CIC-IDS 2018, all the scenarios with RGB images are more accurate than greyscale scenarios as shown in Figure 4.

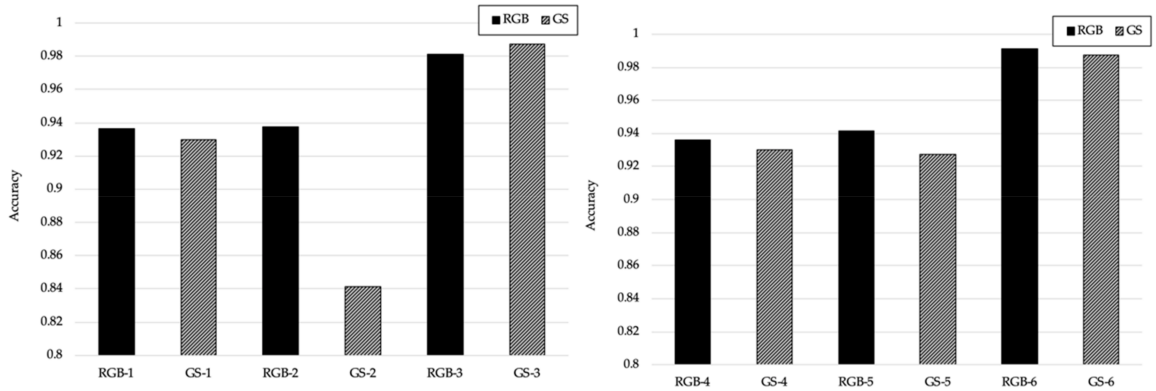


Figure 4. *Cont.*

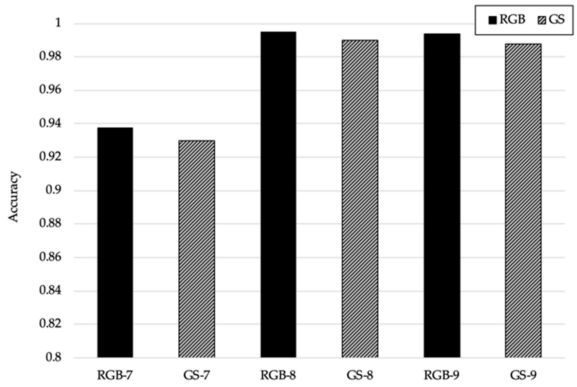


Figure 4. Accuracy Comparison of RGB and grayscale scenarios of CSE-CIC-IDS 2018.

4.2.2. Number of Convolutional Layer

Table 6 shows the experimental results in binary classification of KDD. When the kernel is 2×2 and 3×3, we can see the more convolutional layers (1 L, 2 L, 3 L) the more accuracy is increased. Thus, the more layers, the better performance by extracting features more accurately. The 4×4 kernel shows that both the RGB and grayscale show the highest performance when there are two convolutional layers.

Table 6. Accuracy by Number of Convolution Layers in binary classification for KDD.

Kernel Size	RGB Scenarios	Num. of Conv. Layer	Accuracy	Kernel Size	GS Scenarios	Num.of Conv. Layer	Accuracy
2×2	RGB-1	1	0.999693	2×2	GS-1	1	0.999359
	RGB-2	2	0.999731		GS-2	2	0.999637
	RGB-3	3	0.999932		GS-3	3	0.999778
3×3	RGB-4	1	0.999667	3×3	GS-4	1	0.999642
	RGB-5	2	0.999856		GS-5	2	0.999712
	RGB-6	3	0.999875		GS-6	3	0.999788
4×4	RGB-7	1	0.999755	4×4	GS-7	1	0.999484
	RGB-8	2	0.999922		GS-8	2	0.999868
	RGB-9	3	0.999813		GS-9	3	0.999670

Because the number and performance of the convolutional layers are not always proportional, we can determine that when the kernel is 4×4, it must be composed of two convolutional layers to achieve better performance. Indeed, the RGB-8 has the second highest accuracy among the 9 RGB scenarios, and the GS-8 shows the highest accuracy among the grayscale scenarios. Thus, the kernel size 4×4 and the two convolutional layers are the combination of hyperparameters with the best performance. In Figure 5, similar graphs with kernel size 4×4 are shown in the experimental result of CSE-CIC-IDS 2018.

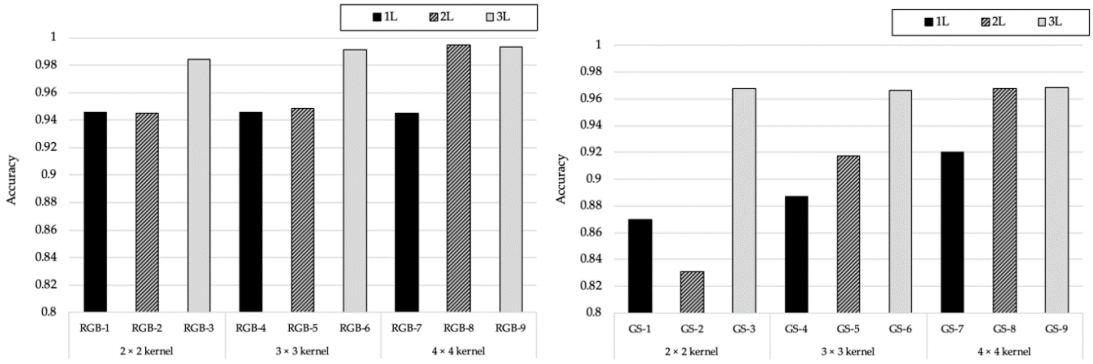


Figure 5. Comparison of Accuracy by Number of Convolution Layers in binary classification of CSE-CIC-IDS 2018.

4.2.3. Kernel Size

Table 7 shows a comparison of the accuracy on the kernel sizes of 2 × 2, 3 × 3 and 4 × 4 for KDD.

Table 7. Accuracy by the kernel size in binary classification for KDD.

Num. of Conv. Layer	RGB Scenarios	Kernel Size	Accuracy	Num. of Conv. Layer	GS Scenarios	Kernel Size	Accuracy
1	RGB-1	2 × 2	0.999693	1	GS-1	2 × 2	0.999359
	RGB-4	3 × 3	0.999667		GS-4	3 × 3	0.999642
	RGB-7	4 × 4	0.999755		GS-7	4 × 4	0.999484
2	RGB-2	2 × 2	0.999731	2	GS-2	2 × 2	0.999637
	RGB-5	3 × 3	0.999856		GS-5	3 × 3	0.999712
	RGB-8	4 × 4	0.999922		GS-8	4 × 4	0.999868
3	RGB-3	2 × 2	0.999932	3	GS-3	2 × 2	0.999778
	RGB-6	3 × 3	0.999875		GS-6	3 × 3	0.999788
	RGB-9	4 × 4	0.999813		GS-9	4 × 4	0.999670

For RGB scenarios, there is no pattern of constant shape (e.g., positive or negative) for accuracy according to the kernel size when there are two or three convolutional layers. Similarly, the same pattern is not visible in grayscale scenarios, indicating that kernel size is not a parameter that affects accuracy alone compared to the type of image or the number of convolutional layers. Similar to KDD, there is no particular pattern with CSE-CIC-IDS 2018 in the scenarios of 4 × 4 kernel size as shown in Figure 6.

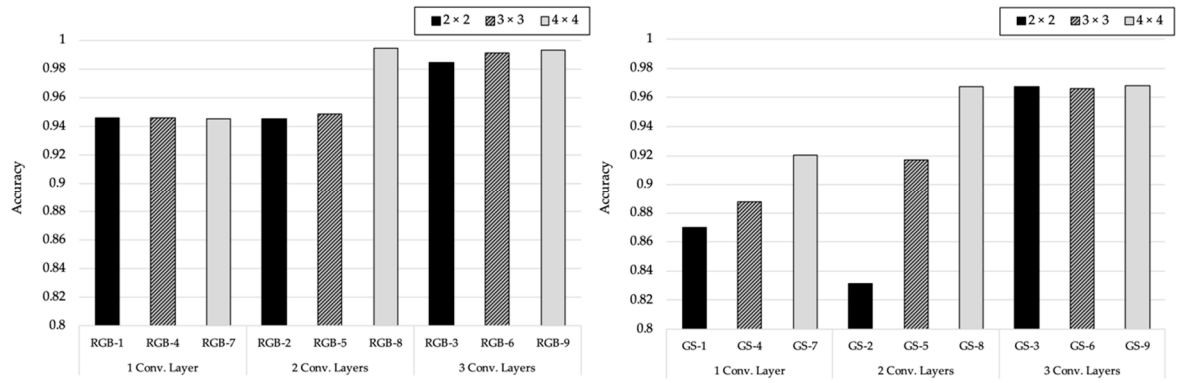


Figure 6. Comparison of Accuracy by size of kernel in case of CSE-CIC-IDS 2018.

4.3. Analysis of the Accuracy in Multiclass Classification

Like binary classifications, we compare the accuracy of scenarios according to parameter settings in multiclass classifications of KDD and CSE-CIC-IDS 2018. The experimental results of KDD show high accuracy in order of RGB-3, RGB-5, and RGB-6 for RGB, and GS-8, GS-5, and GS-6 for grayscale. In case of CSE-CIC-IDS 2018, the result show high accuracy in order of RGB-8, RGB-9 and RGB-6 for RGB, and GS-9, GS-3, and GS-6 for grayscale. In other words, both RGB and grayscale images show better performance when there are two and three convolutional layers than when there is one. A closer look at the multiple classification results based on hyperparameters is as follows.

4.3.1. RGB Vs Grayscale

Table 8 shows how the accuracy varies depend on the RGB and grayscale images for KDD. It means other hyperparameter values are all same except the number of color channels (RGB or grayscale). Same as the results of the binary classification for KDD, RGB images are more accurate than grayscale images, but GS-8 has higher accuracy than RGB images (RGB-8). Thus, this scenario is most likely to be affected by the combination of the number of convolutional layers and the size of kernel. However, since all other scenarios are more accurate with RGB images, we can determine that RGB images show higher performance in the multiclass classification than grayscale images.

Table 8. Accuracy Comparison of RGB and grayscale in multiclass classification of KDD.

RGB Scenarios	Accuracy	GS Scenarios	Accuracy
RGB-1	0.999691	GS-1	0.999481
RGB-2	0.999767	GS-2	0.999611
RGB-3	0.999960	GS-3	0.999755
RGB-4	0.999719	GS-4	0.999538
RGB-5	0.999889	GS-5	0.999825
RGB-6	0.999830	GS-6	0.999823
RGB-7	0.999566	GS-7	0.999476
RGB-8	0.999778	GS-8	0.999835
RGB-9	0.999781	GS-9	0.999455

In the experimental results of CSE-CIC-IDS 2018, all the scenarios with RGB images are more accurate than that of grayscale scenarios as shown in Figure 7.

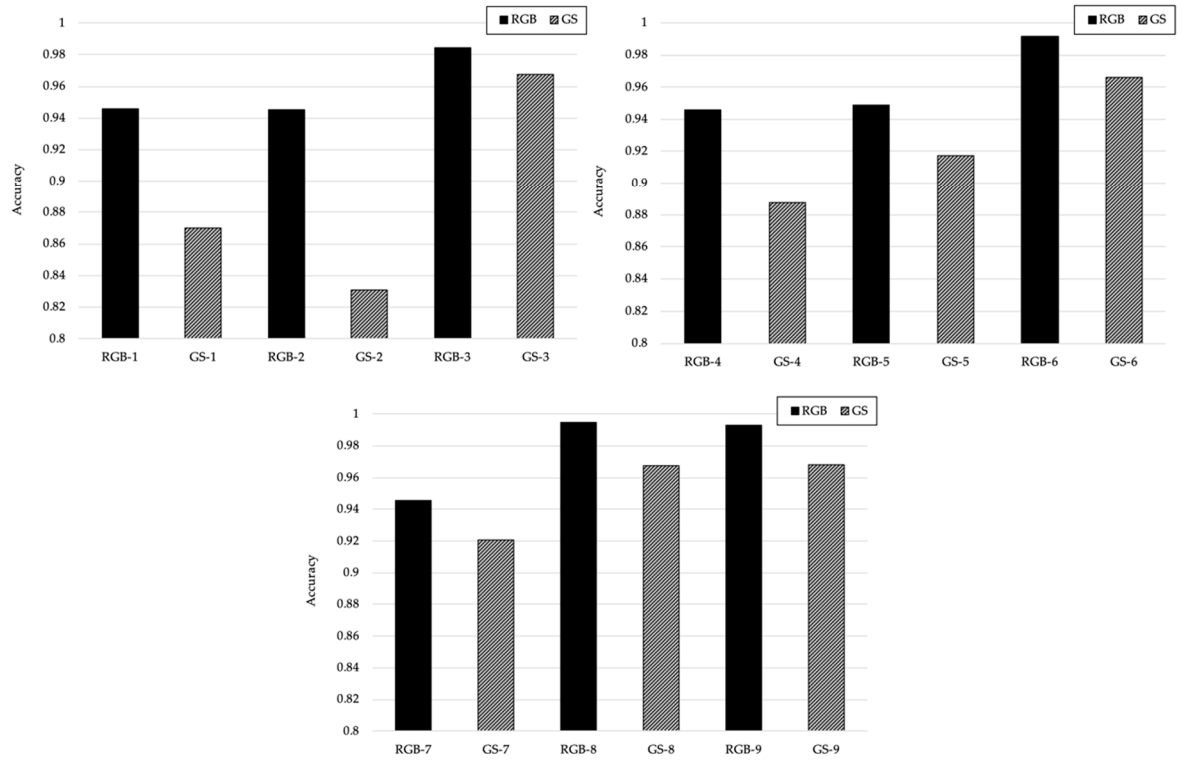


Figure 7. Accuracy Comparison of RGB and grayscale in multiclass classification of CSE-CIC-IDS 2018.

4.3.2. Number of Convolutional Layers

Table 9 shows experimental results of multiclass classification for KDD. This table compares the accuracy of the number of convolution layers.

In Table 9, the graphs of RGB scenarios and grayscale scenarios show that the higher the number of layers (1 L, 2 L, 3 L), the higher the accuracy, when the kernel size is 2×2. However, when the kernel sizes are 3×3 and 4×4, we can see that the accuracy is not proportional to the number of convolutional layers. In the multiclass classification, the accuracy is proportional to performance only when the kernel size is 2×2, while the number of convolutional layers is proportional to performance when the kernel sizes are 2×2 and 3×3 in the binary classification. Thus, we can say that the number of convolutional layers has a lower impact on performance in the multiclass classification.

Table 9. Comparison of Accuracy by Number of Convolution Layers in multiclass classification in case of KDD.

Kernel Size	RGB Scenarios	Num. of Conv. Layer	Accuracy	Kernel Size	GS Scenarios	Num. of Conv. Layer	Accuracy
2 × 2	RGB-1	1	0.999691	2 × 2	GS-1	1	0.999481
	RGB-2	2	0.999767		GS-2	2	0.999611
	RGB-3	3	0.999960		GS-3	3	0.999755
3 × 3	RGB-4	1	0.999719	3 × 3	GS-4	1	0.999538
	RGB-5	2	0.999889		GS-5	2	0.999825
	RGB-6	3	0.999830		GS-6	3	0.999823
4 × 4	RGB-7	1	0.999566	4 × 4	GS-7	1	0.999476
	RGB-8	2	0.999778		GS-8	2	0.999835
	RGB-9	3	0.999781		GS-9	3	0.999455

When the kernels are 2 × 2 and 3 × 3, the accuracies of both RGB and grayscale are much higher with scenarios for three convolutional layers than one or two convolutional layers. When the kernel size is 4 × 4, the accuracies of two and three convolutional layers are high except one convolutional layer. We can expect that the bigger kernel size, the better performance. Figure 8 shows a graph comparing the accuracy of the same scenario with all environments except for the number of convolutional layers in case of CSE-CIC-IDS 2018.

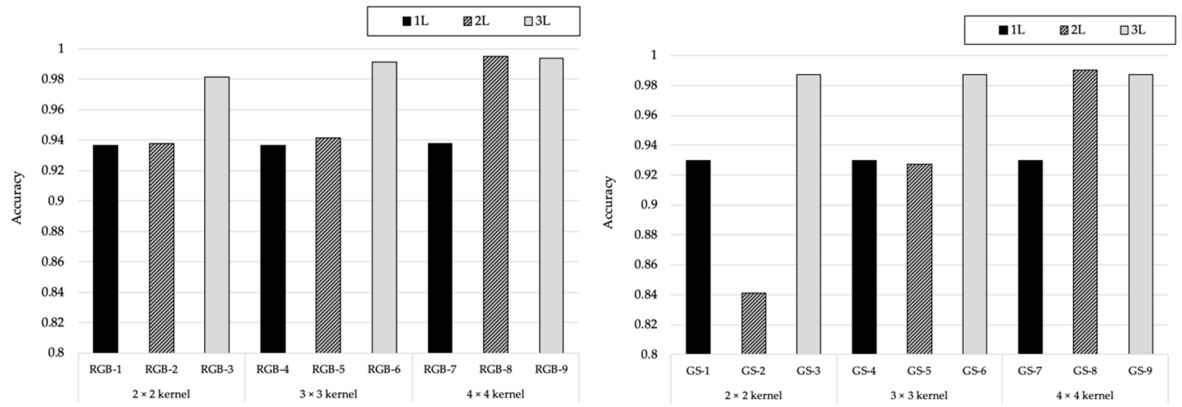


Figure 8. Comparison of Accuracy by Number of Convolution layers for CSE-CIC-IDDS 2018.

4.3.3. Kernel Size

Table 10 shows the results of accuracy according to the kernel size in case of KDD.

Table 10. Comparison of Accuracy by size of kernel in multiple classification in case of KDD.

Num. of Conv. Layer	RGN Scenarios	Kernel Size	Accuracy	Num. of Conv. Layer	GS Scenarios	Kernel Size	Accuracy
1	RGB-1	2 × 2	0.999691	1	GS-1	2 × 2	0.999481
	RGB-4	3 × 3	0.999719		GS-4	3 × 3	0.999538
	RGB-7	4 × 4	0.999566		GS-7	4 × 4	0.999476
2	RGB-2	2 × 2	0.999767	2	GS-2	2 × 2	0.999611
	RGB-5	3 × 3	0.999890		GS-5	3 × 3	0.999825
	RGB-8	4 × 4	0.999778		GS-8	4 × 4	0.999835
3	RGB-3	2 × 2	0.999959	3	GS-3	2 × 2	0.999755
	RGB-6	3 × 3	0.999830		GS-6	3 × 3	0.999823
	RGB-9	4 × 4	0.999781		GS-9	4 × 4	0.999455

Like there is no specific pattern in the binary classification, no pattern is found to correlate the kernel size and performance in multiclass classifications. In the experimental results of CSE-CIC-IDS 2018, when the kernel size is 2 × 2 and 3 × 3 there is no particular pattern of accuracy similar to KDD, as shown in Figure 9. However, when the kernel size is 4 × 4, the accuracy is much higher compare to the others (2 × 2 and 3 × 3). In particular, as we mentioned in Section 4.3.2, the accuracy is higher regardless of the number of the convolutional layers when kernel size is big.

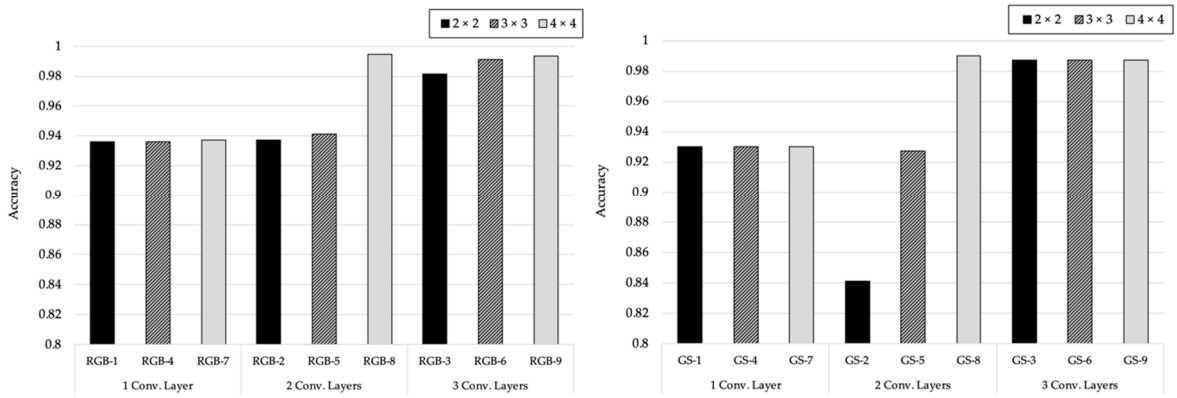


Figure 9. Comparison of Accuracy by size of kernel in case of CSE-CIC-IDS 2018.

5. Discussion

Experiments with the binary and multiclass classification for the proposed CNN model show that both have achieved higher accuracy than 99% in Section 4. Here we detect the DoS attacks using an RNN model to compare its performance with the proposed model. RNN is developed as a way to extend the NN into sequential data and the hidden nodes form a circular structure. In this experiment, we design a simple RNN model using Keras with five embedding vectors and a sigmoid activation function as hyperparameters. Table 11 shows the precision, recall and F1-score for each class of binary and multiclass classifications using RNN in case of KDD. Precision is the ratio of testing samples that are ground truth among the samples that the model classifies as true. Recall is the ratio of testing samples that are ground truth to predict as true. F1-score is a value represented in one number considering both precision and recall.

Table 11. Accuracy of the RNN model in binary and multiple classification in case of KDD.

Classification		Precision	Recall	F1-Score
binary-class	benign	0.99	1.00	0.99
	attack	1.00	1.00	1.00
multiclass	Benign	0.77	0.94	0.85
	Neptune	0.92	0.71	0.80
	Smurf	1.00	1.00	1.00

The RNN model has 99% accuracy in binary classification, almost the same as that of our CNN model. In multiclass classification, however, the RNN model has 100% accuracy in the smurf detection while the accuracies of the neptune and benign are 80% and 85%, respectively.

The interesting thing is that classifying Smurf and Neptune attacks does not cause much misdetections, while there are many misdetections in distinguishing benign from neptune attacks. That is why the accuracy of the RNN model is lower than that of CNN in the multiclass classification.

In the experimental results of CSE-CIC-IDS 2018, the accuracies of both RNN-based binary and multiclass classifications are significantly lower than that of CNN-based detection accuracy as shown in Table 12. In binary classification, about 2% of detection accuracy is lower in benign detection than in attack detection. Even in the multiclass classification, the accuracy of benign detection is only 73.5%. However, in the detection of attacks, DoS-GoldenEye, DDoS-LOIC-HTTP, and DoS-Slowloris have high accuracy of 97%, 95%, and 89%, respectively. However, in other attacks, their accuracies are less than 50%. DoS-Hulk and DDoS-HOIC result in much false positives and DoS-SlowHTTPTest is often incorrectly detected as DoS-Hulk.

Table 12. Accuracy of the RNN model in case of CSE-CIC-IDS 2018.

Classification		Precision	Recall	F1-Score
binary-class	benign	0.8175	0.8225	0.82
	attack	0.6	0.8475	0.8475
multiclass	benign	0.7275	0.77	0.735
	DoS-Hulk	0.37	0.51	0.43
	DoS-SlowHTTPTest	0.79	0.05	0.09
	DoS-GoldenEye	0.91	0.99	0.95
	DoS-Slowloris	0.84	0.93	0.89
	DDoS-LOIC-HTTP	1	0.94	0.97
	DDoS-HOIC	0.44	0.52	0.47

Experimental results show that the accuracy of CSE-CIC-IDS 2018 is generally lower than that of KDD. This is because our KDD model divides samples into 3 categories which are benign, Smurf and Neptune while the CSE-CIC-IDS 2018 model divides samples into 7 categories such as benign and 6 advanced DoS attacks. From the experimental results with the RNN model, furthermore, we can find out that advanced DoS attacks not only do not have novel characteristics compared to traditional DoS attacks, but also that the characteristics do not appear to be time-series features.

6. Conclusions

We develop a CNN-based model for the detection of DoS attacks using KDD and CSE-CIC-IDS 2018. There are 4 types of attack categories in KDD, such as DoS, U2R, R2L, and Probing. Most of deep learning-based KDD studies have carried out binary classifications that distinguish benign and attack across the entire category. These studies have also performed multiclass classification that distinguish the 4 categories in KDD.

We focus on one category of DoS and perform detection for different attacks in the same category. We also used the most up-to-date IDS dataset which contains advanced DoS attacks such as DoS-Hulk, DoS-SlowHTTPTest, DoS-GoldenEye, DoS-Slowloris, DDoS-LOIC-HTTP, and DDoS-HOIC. We have generated two types of intrusion image, RGB and grayscale. We have designed our CNN model considering the number of convolutional layers and the size of kernel. To evaluate our model, we created 18 scenarios considering hyperparameters, such as the type of image, the number of convolutional layers, and the kernel size mentioned above. We performed the binary and multiclass classifications for each scenario, and then suggested the optimal scenarios that have higher performance. Our experimental results have shown that RGB images in both binary and multiclass classifications have higher accuracy than that of grayscale images. In addition, we found out that both RGB and grayscale images performed best with three convolutional layers when the kernel sizes are 2×2 and 3×3. When the kernel size is 4×4, deploying two convolutional layers has the highest accuracy. In multiclass classification, there was generally high performance when there was more than one convolutional layer. However, the best model should be found through various hyperparameter setting, because the number and performance of convolutional layers are not proportional. The kernel size has not been found to have a significant impact on both binary and multiclass classifications. We performed a comparison with the RNN model to verify the performance of the proposed model. For KDD, while the CNN model showed 99% or more results in binary and multiclass classifications, the RNN showed 99% accuracy in binary classification and 93% in multiclass classifications. For CSE-CIC-IDS 2018, the CNN model showed 91.5% of accuracy on average while the RNN model showed 65% of accuracy on average. In other words, the CNN model proposed in this paper was able to identify specific DoS attacks with similar characteristics compared to the RNN model. As a future work, multiclass classifications will also be carried out for attacks belonging to other categories in KDD and CSE-CIC-IDS 2018. Furthermore, our model will be used for other intrusion datasets to improve the performance.

We believe our findings can be used for various fields, such as national defense, industry, and healthcare that require advanced intrusion detection techniques.

Author Contributions: Conceptualization, methodology, validation, funding acquisition, and project administration and writing—original draft, J.K. (Jiyeon Kim); Investigation, data acquisition, software, implementation, and writing—original draft preparation, J.K. (Jiwon Kim) and H.K.; Investigation, visualization and writing—review and editing, M.S. and E.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2018R1D1A1B07050543). This research was partially supported by the MISP (Ministry of Science, ICT & Future Planning), Korea, under the National Program for Excellence in SW) (2016-0-00022) supervised by the IITP (Institute of Information & communications Technology Planing & Evaluation) (2016-0-00022).

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

References

1. KDD. KDD CUP. Available online: https://kdd.ics.uci.edu/databases/kddcup99/task.html (accessed on 17 March 2020).
2. Özgür, A.; Erdem, H. A review of KDD99 dataset usage in intrusion detection and machine learning between 2010 and 2015. *Peer. J. Preprints* **2016**, *4*, e1954v1.
3. Paliwal, S.; Gupta, R. Denial-of-service, probing & remote to user (R2L) attack detection using genetic algorithm. *Int. J. Comput. Appl.* **2012**, *60*, 57–62.
4. Shiravi, A.; Shiravi, H.; Tavallaee, M.; Ghorbani, A.A. Toward developing a systematic approach to generate benchmark datasets for intrusion detection. *Comput. Secur.* **2012**, *31*, 357–374. [CrossRef]
5. Anwar, S.; Mohamad Zain, J.; Zolkipli, M.; Inayat, Z.; Khan, S.; Anthony, B., Jr.; Chang, V. From intrusion detection to an intrusion response system: Fundamentals, requirements, and future directions. *Algorithms* **2017**, *10*, 39. [CrossRef]
6. Jing-Xin, W.; Zhi-Ying, W.; Kui, D. A network intrusion detection system based on the artificial neural networks. In Proceedings of the 3rd international conference on Information security, Shanghai, China, 14–16 November 2004; pp. 166–170.
7. Manso, P.; Moura, J.; Serrao, C. SDN-Based Intrusion Detection System for Early Detection and Mitigation of DDoS Attacks. *Information* **2019**, *10*, 106. [CrossRef]
8. Karim, I.; Vien, Q.T.; Le, T.; Mapp, G. A comparative experimental design and performance analysis of snort-based intrusion detection system in practical computer networks. *Computers* **2017**, *6*, 6. [CrossRef]
9. Xu, R.; Cheng, J.; Wang, F.; Tang, X.; Xu, J. A DRDoS Detection and Defense Method Based on Deep Forest in the Big Data Environment. *Symmetry* **2019**, *11*, 78. [CrossRef]
10. Ramotsoela, D.; Abu-Mahfouz, A.; Hancke, G. A survey of anomaly detection in industrial wireless sensor networks with critical water system infrastructure as a case study. *Sensors* **2018**, *18*, 2491. [CrossRef]
11. Zhang, Z.; Li, J.; Manikopoulos, C.N.; Jorgenson, J.; Ucles, J. HIDE: A hierarchical network intrusion detection system using statistical preprocessing and neural network classification. In Proceedings of the IEEE Workshop on Information Assurance and Security, West Point, NY, USA, 5–6 June 2001; pp. 85–90.
12. Koc, L.; Mazzuchi, T.A.; Sarkani, S. A network intrusion detection system based on a Hidden Naïve Bayes multiclass classifier. *Expert Syst. Appl.* **2012**, *39*, 13492–13500. [CrossRef]
13. Hodo, E.; Bellekens, X.; Hamilton, A.; Dubouilh, P.L.; Iorkyase, E.; Tachtatzis, C.; Atkinson, R. Threat analysis of IoT networks using artificial neural network intrusion detection system. In Proceedings of the 2016 International Symposium on Networks, Computers and Communications (ISNCC), Hammamet, Tunisia, 11–13 May 2016; pp. 1–6.
14. Chung, Y.Y.; Wahid, N. A hybrid network intrusion detection system using simplified swarm optimization (SSO). *Appl. Soft Comput.* **2012**, *12*, 3014–3022. [CrossRef]
15. Aydin, M.A.; Zaim, A.H.; Ceylan, K.G. A hybrid intrusion detection system design for computer network security. *Comput. Electr. Eng.* **2009**, *35*, 517–526. [CrossRef]

16. Al-Jarrah, O.; Arafat, A. Network Intrusion Detection System using attack behavior classification. In Proceedings of the 2014 5th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, 1–3 April 2014; pp. 1–6.
17. Karthick, R.R.; Hattiwale, V.P.; Ravindran, B. Adaptive network intrusion detection system using a hybrid approach. In Proceedings of the 2012 Fourth International Conference on Communication Systems and Networks (COMSNETS 2012), Bangalore, India, 3–7 January 2012; pp. 1–7.
18. Wahab, O.A.; Bentahar, J.; Otrok, H.; Mourad, A. Resource-Aware Detection and Defense System Against Multi-Type Attacks in the Cloud: Repeated Bayesian Stackelberg Game. *IEEE Trans. Dependable Secure Comput.* **2019**. [CrossRef]
19. Wahab, O.A.; Bentahar, J.; Otrok, H.; Mourad, A. Optimal load distribution for the detection of VM-based DDoS attacks in the cloud. *IEEE Trans. Dependable Secure Comput.* **2017**. [CrossRef]
20. Chen, H.; Meng, C.; Shan, Z.; Fu, Z.; Bhargava, B.K. *A Novel Low-Rate Denial of Service Attack Detection Approach in ZigBee Wireless Sensor Network by Combining Hilbert-Huang Transformation and Trust Evaluation*; IEEE Access: Piscataway, NJ, USA, 2019; Volume 7, pp. 32853–32866.
21. Chang, R.I.; Lai, L.B.; Su, W.D.; Wang, J.C.; Kouh, J.S. Intrusion detection by backpropagation neural networks with sample-query and attribute-query. *Int. J. Comput. Intell. Res.* **2007**, *3*, 6–10. [CrossRef]
22. Staudemeyer, R.C.; Omlin, C.W. Extracting salient features for network intrusion detection using machine learning methods. *S. Afr. Comput. J.* **2014**. [CrossRef]
23. Sabhnani, M.; Serpen, G. Application of Machine Learning Algorithms to KDD Intrusion Detection Dataset within Misuse Detection Context. In Proceedings of the International Conference on Machine Learning; Models, Technologies and Applications, Las Vegas, NV, USA, 23–26 June 2003; pp. 209–215.
24. Mulay, S.A.; Devale, P.R.; Garje, G.V. Intrusion detection system using support vector machine and decision tree. *Int. J. Comput. Appl.* **2010**, *3*, 40–43. [CrossRef]
25. Al Mehedi Hasan, M.; Nasser, M.; Pal, B. On the KDD'99 dataset: Support vector machine based intrusion detection system (ids) with different kernels. *Int. J. Electron. Commun. Comput. Eng.* **2013**, *4*, 1164–1170.
26. Yao, J.T.; Zhao, S.; Fan, L. An enhanced support vector machine model for intrusion detection. In Proceedings of the International Conference on Rough Sets and Knowledge Technology, Chongqing, China, 24–26 July 2006; pp. 538–543.
27. Dong-Hoon, K.; Kim, J.-J.; Insoo, S. Studies on Intrusion Detection based on ML using KDD99CUP. In Proceedings of the Symposium of the Korean Institute of communications and Information Sciences, Jeju Island, Korea, 16–18 October 2019; pp. 861–862.
28. Yin, C.; Zhu, Y.; Fei, J.; He, X. *A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks*; IEEE: Piscataway, NJ, USA, 2017; pp. 21954–21961.
29. Sheikhan, M.; Jadidi, Z.; Farrokhi, A. Intrusion detection using reduced-size RNN based on feature grouping. *Neural Comput. Appl.* **2012**, *21*, 1185–1190. [CrossRef]
30. Bontemps, L.; Cao, V.L.; Mcdermott, J.; Le-Khac, N.A. Collective anomaly detection based on long short-term memory recurrent neural networks. In Proceedings of the International Conference on Future Data and Security Engineering, Can Tho City, Vietnam, 23–25 November 2016; pp. 141–152.
31. Khan, R.U.; Zhang, X.; Alazab, M.; Kumar, R. An Improved Convolutional Neural Network Model for Intrusion Detection in Networks. In Proceedings of the 2019 Cybersecurity and Cyberforensics Conference (CCC), Melbourne, Australia, 8–9 May 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 74–77.
32. Li, Z.; Qin, Z.; Huang, K.; Yang, X.; Ye, S. Intrusion detection using convolutional neural networks for representation learning. In Proceedings of the International Conference on Neural Information Processing, Guangzhou, China, 14–18 November 2017; pp. 858–866.
33. Upadhyay, R.; Pantiukhin, D. Application of convolutional neural network to intrusion type recognition. In Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics, Udupi, India, 13–16 September 2017.
34. Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A. A detailed analysis of the KDD CUP 99 data set. In Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Otawa, ON, Canada, 8–10 July 2009; IEEE: Piscataway, NJ, USA, 2009; pp. 1–6.
35. Gao, J.; Chai, S.; Zhang, B.; Xia, Y. Research on Network Intrusion Detection Based on Incremental Extreme Learning Machine and Adaptive Principal Component Analysis. *Energies* **2019**, *12*, 1223. [CrossRef]

36. Chu, W.L.; Lin, C.J.; Chang, K.N. Detection and Classification of Advanced Persistent Threats and Attacks Using the Support Vector Machine. *Appl. Sci.* **2019**, *9*, 4579. [CrossRef]
37. Fares, A.H.; Sharawy, M.I.; Zayed, H. Intrusion detection: Supervised machine learning. *J. Comput. Sci. Eng.* **2011**, *5*, 305–313. [CrossRef]
38. Niyaz, Q.; Sun, W.; Javaid, A.; Alam, M. A deep learning approach for network intrusion detection system. In Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS), New York, NY, USA, 3–5 December 2015; pp. 21–26.
39. Tang, T.A.; Mhamdi, L.; McLernon, D.; Zaidi, S.A.R.; Ghogho, M. Deep learning approach for network intrusion detection in software defined networking. In Proceedings of the 2016 International Conference on Wireless Networks and Mobile Communications (WINCOM), Fez, Morocco, 26–29 October 2016; pp. 258–263.
40. Ingre, B.; Yadav, A. Performance analysis of NSL-KDD dataset using ANN. In Proceedings of the 2015 International Conference on Signal Processing and Communication Engineering Systems, Vijayawada, India, 2–3 January 2015; pp. 92–96.
41. Erol, S.E.; Benzer, R. An Application of Artificial Neural Network Based Intrusion Detection System. In Proceedings of the 5th International Management Information Systems Conference, Ankara, Turkey, 24–26 October 2018.
42. Ibrahim, L.M. Anomaly network intrusion detection system based on distributed time-delay neural network (DTDNN). *J. Eng. Sci. Technol.* **2010**, *5*, 457–471.
43. Tan, X.; Su, S.; Huang, Z.; Guo, X.; Zuo, Z.; Sun, X.; Li, L. Wireless Sensor Networks Intrusion Detection Based on SMOTE and the Random Forest Algorithm. *Sensors* **2019**, *19*, 203. [CrossRef]
44. Farnaaz, N.; Jabbar, M.A. Random forest modeling for network intrusion detection system. *Procedia Comput. Sci.* **2016**, *89*, 213–217. [CrossRef]
45. Ye, K. Key Feature Recognition Algorithm of Network Intrusion Signal Based on Neural Network and Support Vector Machine. *Symmetry* **2019**, *11*, 380. [CrossRef]
46. Ali, M.H.; Al Mohammed, B.A.D.; Ismail, A.; Zolkipli, M.F. *A New Intrusion Detection System Based on Fast Learning Network and Particle Swarm Optimization*; IEEE Access: Piscataway, NJ, USA, 2018; pp. 20255–20261.
47. Yang, A.; Zhuansun, Y.; Liu, C.; Li, J.; Zhang, C. *Design of Intrusion Detection System for Internet of Things Based on Improved BP Neural Network*; IEEE Access: Piscataway, NJ, USA, 2019; pp. 106043–106052.
48. Seo, J.H. A study on the performance evaluation of unbalanced intrusion detection dataset classification based on machine learning. *J. Korean Inst. Intell. Syst.* **2017**, *27*, 466–474. [CrossRef]
49. Amma, B.N.G.; Selvakumar, S. Deep Radial Intelligence with Cumulative Incarnation approach for detecting Denial of Service attacks. *Neurocomputing* **2019**, *340*, 294–308.
50. Koay, A.; Chen, A.; Welch, I.; Seah, W.K. A new multi classifier system using entropy-based features in DDoS attack detection. In Proceedings of the 2018 International Conference on Information Networking (ICOIN), Chiang Mai, Thailand, 10–12 January 2018; pp. 162–167.
51. Idhammad, M.; Afdel, K.; Belouch, M. Semi-supervised machine learning approach for DDoS detection. *Appl. Intell.* **2018**, *48*, 3193–3208. [CrossRef]
52. Yassin, W.; Udzir, N.I.; Muda, Z.; Sulaiman, M.N. Anomaly-based intrusion detection through k-means clustering and naives bayes classification. In Proceedings of the 4th International Conference on Computing and Applied Informatics, Kuching, Sarawak, Malaysia, 28–30 August 2013; pp. 298–303.
53. Soheily-Khah, S.; Marteau, P.F.; Béchet, N. Intrusion detection in network systems through hybrid supervised and unsupervised machine learning process: A case study on the iscx dataset. In Proceedings of the 2018 1st International Conference on Data Intelligence and Security (ICDIS), Island, TX, USA, 8–10 April 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 219–226.
54. Faker, O.; Dogdu, E. Intrusion detection using big data and deep learning techniques. In Proceedings of the 2019 ACM Southeast Conference, Kennesaw, GA, USA, 18–20 April 2019; pp. 86–93.
55. Zhang, X.; Chen, J.; Zhou, Y.; Han, L.; Lin, J. *A Multiple-Layer Representation Learning Model for Network-Based Attack Detection*; IEEE Access: Piscataway, NJ, USA, 2019; pp. 91992–92008.
56. Zhou, Q.; Pezaros, D. Evaluation of Machine Learning Classifiers for Zero-Day Intrusion Detection-An Analysis on CIC-AWS-2018 dataset. *arXiv* **2019**, arXiv:1905.03685.
57. Kim, J.; Shin, Y.; Choi, E. An Intrusion Detection Model based on a Convolutional Neural Network. *J. Mult. Inform. Syst.* **2019**, *6*, 165–172. [CrossRef]

58. Chadza, T.; Kyrakopoulos, K.G.; Lambotharan, S. Contemporary Sequential Network Attacks Prediction using Hidden Markov Model. In Proceedings of the 2019 17th International Conference on Privacy, Security and Trust (PST), Fredericton, NB, Canada, 26–28 August 2019; pp. 1–3.
59. MATLAB. Available online: https://www.mathworks.com/help/matlab/ref/image.html (accessed on 20 January 2020).
60. Tensorflow. Available online: https://www.tensorflow.org (accessed on 17 March 2020).
61. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.