electronics

MDPI

Article

IMIDS: An Intelligent Intrusion Detection System against Cyber Threats in IoT

Kim-Hung Le [1,3,*,†], Minh-Huy Nguyen [2,3,†], Trong-Dat Tran [1,3] and Ngoc-Duan Tran [1,3]

[1] Faculty of Computer Networks and Communications, University of Information Technology, Ho Chi Minh City 70000, Vietnam; 18520263@gm.uit.edu.vn (T.-D.T.); 18520609@gm.uit.edu.vn (N.-D.T.)
[2] Faculty of Computer Science, University of Information Technology, Ho Chi Minh City 70000, Vietnam; 19520109@gm.uit.edu.vn
[3] Vietnam National University, Ho Chi Minh City 70000, Vietnam
* Correspondence: hunglk@uit.edu.vn
† These authors contributed equally to this work.

Abstract: The increasing popularity of the Internet of Things (IoT) has significantly impacted our daily lives in the past few years. On one hand, it brings convenience, simplicity, and efficiency for us; on the other hand, the devices are susceptible to various cyber-attacks due to the lack of solid security mechanisms and hardware security support. In this paper, we present IMIDS, an intelligent intrusion detection system (IDS) to protect IoT devices. IMIDS's core is a lightweight convolutional neural network model to classify multiple cyber threats. To mitigate the training data shortage issue, we also propose an attack data generator powered by a conditional generative adversarial network. In the experiment, we demonstrate that IMIDS could detect nine cyber-attack types (e.g., backdoors, shellcode, worms) with an average F-measure of 97.22% and outperforms its competitors. Furthermore, IMIDS's detection performance is notably improved after being further trained by the data generated by our attack data generator. These results demonstrate that IMIDS can be a practical IDS for the IoT scenario.

Keywords: intrusion detection system; Internet of Things; generative adversarial network

Citation: Le, K.-H.; Nguyen, M.-H.; Tran, T.-D.; Tran, N.D. IMIDS: An Intelligent Intrusion Detection System against Cyber Threats in IoT. *Electronics* **2022**, *11*, 524. https://doi.org/10.3390/electronics11040524

Academic Editors: Stavros Shiaeles, Bogdan Ghita and Nicholas Kolokotronis

Received: 10 December 2021
Accepted: 7 February 2022
Published: 10 February 2022

1. Introduction

Recent years have witnessed the proliferation of the Internet of Things, aiming to bring every physical object into the digital world, resulting in billions of IoT devices connected to the Internet. IoT is anticipated to provide innovative solutions to boost profit in several industries (e.g., healthcare, manufacturing, retailing, security, and transportation). According to a recent study [1], the total IoT market worldwide was worth around 389 billion US dollars in 2020 and is projected to reach more than one trillion US dollars by 2030. Moreover, the number of IoT devices deployed worldwide is forecast to triple during this duration and generate zillions of bytes of data crossing the Internet every second.

The explosion of IoT devices introduces non-trivial security challenges due to the lack of safety mechanisms and hardware security support. Indeed, the limited computational capability of IoT devices makes them insufficient for traditional protection methods. Thus, simplifying these methods is a common solution, but it may lead to serious security flaws at multiple levels [2]. For example, insecure physical interface, low-level Sybil, and spoofing attacks occur at the operating system level; replay attacks, sinkhole, and wormhole attacks occur at the network level. More seriously, attackers can exploit these flaws to create botnets and launch Distributed Denial of Service (DDoS) attacks [3] to disrupt Internet services by overwhelming the processing capability of the target with a flood of Internet traffic. For instance, the Mirai virus infected more than 65,000 IoT devices in its first 20 h after breaking out on 1 August 2016, and reached 200,000–300,000 infections [4]. In September 2016, the website of a security consultant named Brian Krebs was attacked with 620 Gbps of traffic

using the Mirai botnet. Meanwhile, a more significant DDoS attack targeted a French cloud service provider, OVH [5], peaking at 1.1 Tbps. Just one month later, in October 2016, a variant of Mira shut down hundreds of websites for several hours [6], including Twitter, Netflix, Reddit, and Github. After Mirai was released in August 2016, DDoS attacks [7] have occurred more frequently. The above examples indicate that IoT devices possess many vulnerabilities that attackers can exploit to compromise and create botnets for illegal purposes. According to reports from Security Today and Threat Post, the number of IoT devices in 2020 was around 31 billion, but half of them are vulnerable to severe attacks [8,9]. Therefore, an efficient and powerful defensive mechanism for cyber-attacks is necessary.

To mitigate security threats from IoT devices, deploying intrusion detection systems in the network is a potential solution. The IDS detects any signs of cyber-attacks by continuously monitoring inbound and outbound network traffic generated by IoT devices. Its attack detection method is classified into signature-based and anomaly-based. A signature-based attack detection adopts the rules created from typical patterns of already known attacks, and it identifies a threat if observed events match these rules. In contrast, an anomaly-based method trains a model using the normal behaviors of the system. This model is then used to detect the attack by calculating the difference between observed and learning behaviors. The major issue of these methods is that they only model a very limited number of system behaviors that is insufficient for IoT networks consisting of heterogeneous device types. Indeed, different device types may generate various network behaviors, resulting in degraded attack detection accuracy. Motivated by the success of deep learning in several fields, it is envisaged that deep learning-based IDS could overcome existing issues and increase the attack detection quality. Indeed, the deep learning model is good at learning complex and non-linear features in the network traffic, making it more efficient than other machine learning algorithms. However, training these models requires a massive amount of labeled data containing both normal and abnormal network traffic. The labeling process is time-consuming and laborious; thus, continuously maintaining the labeled datasets is impractical.Moreover, high-quality datasets for training the IDSs are hard to obtain, although they play an essential role in enhancing IDS detection performance. The main reason for this shortage is privacy concerns [10–12]. Following [13], enriching the network information for IDS datasets requires the neglect of privacy and security issues. Thus, these datasets are customized to remove the sensitive network information before publishing, leading to several defects in network flows. Moreover, most of the existing IDS datasets are out-of-date and lack labeled attacks [14]. For example, the worms attack in the UNSW-NB15 dataset, a well-known dataset used in our experiment, has only 174 samples, compared with 44,525 samples for the exploit attack. This significantly degrades the effectiveness of the training process.

The motivation of this work arises from a basic observation: it is non-trivial to enhance the attack detection quality while retaining the model's simplicity. This observation is described in several research papers about IDS, in which authors tend to introduce novel attack detection models to increase the detection accuracy regardless of their complexity, but we believe that high-quality training data also play a critical role in achieving an effective model. Thus, the main technical goal of this paper is to detect various cyber-attacks accurately by providing an efficient IDS along with an artificial method to generate valuable training data. In more detail, this paper introduces IMIDS, an effective intrusion detection system powered by a CNN model. IMIDS comprises two key components, the feature extractor and attack detection model , with the former capturing raw network packets and transforming them into network features, while the latter detects malicious activities from these features. We note that IMIDS could not only differentiate the normal and abnormal activities, but identify the type of cyber-attacks hidden behind these activities. To enhance the IMIDS's detection performance, we also propose a novel attack data generator leveraging a conditional generative adversarial network. This network is constituted of conditional generators that could learn the conditional distribution of samples in the dataset. The experimental results show that IMIDS could detect nine cyber-attacks with

an average F-measure of 97.22%. In addition, IMIDS's detection performance is notably improved after being further trained by the training data generated by our attack data generator. For example, the accuracy of detecting worms and analysis attacks increases from 35.58% to 70.94% and 49.12% to 83.64%, respectively. The main contributions of this work are summarized as follows:

- We present IMIDS, an intrusion detection system powered by a CNN model that can not only differentiate normal and abnormal activities but identify the type of cyber-attack hidden behind these activities. The evaluation results on two popular IDS datasets show that IMIDS could detect nine types of cyber-attacks with an average F-measure of 97.22% and 96.34%.
- We introduce a novel attack data generator leveraging a conditional generative adversarial network. In our evaluation, IMGAN's detection performance is notably improved after being further trained by the training data generated by our attack data generator. For example, the accuracy of detecting worms and analysis attacks increases from 35.58% to 70.94% and 49.12% to 83.64%, respectively.
- We perform an extensive experimental evaluation to show that IMIDS significantly improves the detection performance on two different intrusion datasets (UNSW-NB15 datasets and CICIDS2017) and outperforms its competitors.

The remainder of this paper is organized as follows: Section 2 reviews the most recent related literature. Section 3 delves into the details of the proposed IDS, while Section 4 presents an extensive analysis of the evaluation results. Finally, we draw our conclusions in Section 5.

2. Related Works

Cybersecurity has attracted more attention recently due to the evolution of information technology, so the domain of implementing intrusion detection systems and improving them with machine learning has been intensively studied [15]. Mojtaba et al. proposed Passban IDS, an intrusion detection system optimized by unsupervised learning to detect anomalies and applied on a limited hardware environment [16]. In [17], the authors introduced an IDS for online intrusion detection using AutoEncoders algorithms. The authors in [18] examined the ANN, as well as other classification algorithms, for the problem of network intrusion detection and suggested a solution based on an ensemble of classifiers utilizing connection-based methods. In [19], the authors presented a hierarchical approach consisting of multiple layers, such as an anomaly detection layer created by a series of ANN classifiers. The authors of [20] modified the backpropagation to speed up model training time. The authors in [21] employed multiple machine learning classifiers to detect several types of attack. To increase the detection performance, Moustafa et al. proposed a feature selection model that ignores all the irrelevant features and only keeps the significant ones before passing the data into the classification models [22]. The proposal uses a combination of the association rule mining technique and the central point of attributes values. The overall results showed improved performance (accuracy and false alarm reduction) and a significantly low processing time. Adopting the same idea of feature selection, the authors in [23] proposed a new model based on a customized generic algorithm and least squares support vector machine. The evaluation results show the low false-positive and high positive rates in anomalies classifications. A Reduced Error Pruning Tree algorithm was introduced in [24]. The proposed model contains a feature selection layer based on the user's requirements, a sub-layer to group network flows by protocol (TCP, UDP, or others), a layer of anomaly detection, and a layer to inspect the detected abnormalities.

The authors in [25] presented a cascade structure of ANN for multi-class detection (CANIDS). The model was evaluated using the UNSW-NB15 and KDD99 datasets, giving overall accuracy of 86.40% and 94,96%, respectively. Al-Zewairi et al. used deep learning classifiers based on a multilayer feedforward artificial neural network that is optimized by backpropagation and stochastic gradient descent in more detail [26]. Aiming to optimize

the ANN model, the authors in [27] reduced the costly resource expense of the algorithm by employing an effective feature selection algorithm. The evaluation results on the UNSW-NB15 and NSL-KDD datasets show that the proposed ANN model achieves an accuracy score of 95.45% and outperforms the state-of-the-art algorithms. To protect the model network, Nguyen et al. proposed an attack detection model applying a multi-class cascade of ANNs [28]. The results on the UNSW-NB15 dataset are approximately 95.84% accuracy, 83.40% precision, and 79.19% recall. To detect web application attacks, Moustafa et al. presented an anomaly-based detection system [29] consisting of a real-world network data collector, a dynamic feature selection module for web application data using association rule mining, and an optimized Outlier Gaussian Mixture classification module. In [30], the authors proposed an IDS to protect cyber-physical systems. The proposed IDS combines the hidden Markov model and beta mixture model to detect security threats. The model performance over the UNSW-NB15 dataset achieved a 95.89% detection rate, 96.32% accuracy, and a 3.82% false-positive rate. Chowdhury et al. presented a model based on the combination of simulated annealing and support vector machine [31]. The detection accuracy of the proposed model is around 98.76%, higher than the original SVM, which is only 88.03%. The authors in [32] evaluated several well-known algorithms, including decision and regression trees, naïve Bayes, and support vector machines, to reveal the lack of real-life evaluation of current IDS works. The evaluation results on the UNSW-NB15 and ISOT datasets indicate that the attack detection performance of current IDSs may vary in different evaluation environments.

Research on applying generative adversarial networks in networking security is still in the early stage. In [33], Chuanlong Yin et al. employed GAN to build Bot-GAN, a security framework to detect botnet attacks. In Bot-GAN, the generative model is used to improve the amount of data to bypass the original IDS acting as a discriminator. In [34], the authors proposed a GAN-based framework to evaluate cyber-attacks on the smart energy grid. The framework employs a GAN generator to generate abnormal traffic to evaluate the existing system defender. JooHwa Lee et al. presented a high-performance network intrusion detection system based on an autoencoder-conditional GAN (AE-CGAN) model [35]. The AE-CGAN uses an unsupervised learning model autoencoder and GAN to solve the imbalances between normal and abnormal traffic to increase the detection performance. Similarly, a Flow-WGAN model was introduced in [36] to generate new network traffic data from original datasets to enhance the amount of training data and protect the user's privacy. The authors in [37] proposed G-IDS to address the lack of training data for IDSs in cyber-physical system (CPS) contexts. In G-IDS, GAN is responsible for generating synthetic samples, which are then combined with real data to train the attack detection model. Ref. [38] presented a novel fog-based FID-GAN, an unsupervised intrusion detection system for CPSs using GAN to detect cyber-attacks with low latency requirements. To protect the ad hoc networks, Huang et al. proposed an imbalanced generative adversarial network-based IDS (IGAN-IDS) [39]. In IGAN-IDS, the authors added imbalanced data filters and convolutional layers to a typical GAN model to generate more data samples for minority classes. Zhao, Shuang et al. used GAN to generate adversarial attacks against black-box IDSs [40]. Studies in [41,42] have shown the importance of protection systems for IoT devices, including developing an effective IDS, along with ensuring data privacy. On the other hand, researching and testing IDSs on incomplete datasets may yield inaccurate results [43], as shown in Ref. [44] in the case wherein these datasets are unbalanced. To address imbalance in training data, researchers focus on feature selection specifically, as in [43,45,46].

Many research groups have followed the success of machine learning, integrating machine learning into NIDS to leverage attack detection quality. Refs. [42,47] provided a detailed overview of machine learning technologies adopted in cyber security over the last decade, including intrusion detection, malware detection, and spam detection, which cover both legacy and IoT systems. A brief assessment of machine learning models on IDS datasets has been presented in [48,49]. Ref. [21] proposed a two-layer network classifier:

the first used REP Tree and the JRip algorithm, while the second used Forest PA, which took initial features and first layer results in addition as inputs. Refs. [50,51] leveraged the success of convolutional neural networks in image and network environments and proposed an IDS based on the LeNet-5 CNN model, adopting normalization and one-hot encoding to improve the stability; the overall accuracy was 97.53% when evaluated on the KDD Cup99 dataset. Ref. [52] introduced a deep neural network (DNN)-based IDS. The authors in [53] implemented Apache Spark distributed computing, integrated with the Keras deep learning library and Apache Spark Machine Learning ensemble techniques, and the highest accuracy for multi-class classification was achieved by the DNN, which reached 97.01% on the UNSW-NB15 dataset and 99.56% on the CICIDS2017 dataset.

The authors in [45] have made comparisons between several machine learning and deep learning algorithms, including decision tree, random forest, K-nearest neighbors, logistic regression, naïve Bayes, support vector machine, and ANN. However, the authors focus more on the feature selection problem, so the above classification algorithms have not been evaluated thoroughly. A comparison of many machine learning models was performed in [54], and the results were evaluated on UNSW-NB15. Here, REPTree had the best overall accuracy of 87.37% and the lowest false alarm Rate. Ref. [55] provided a stacking ensemble of machine learning algorithms, including the first layer of random forest, logistic regression, and KNN, and the second layer optimized support vector machine. The result showed 94.27% accuracy in UNSW-NB15 binary classification and 82.22% accuracy in multi-class classification. Ref. [56] also proposed a two-stage network intrusion detection system on the UNSW-NB15 dataset. The first stage uses data mining techniques such as logistic regression, gradient boost machine, and support vector machine to perform binary classification; the second stage deploys multinomial classifiers for categorizing attack types using decision trees (C5.0), naïve Bayes, and multinomial support vector machines. The accuracy achieved for each model was 72.72%, 60.70%, and 70.21%, respectively. Ref. [57] proposed Dendron IDS, which evaluated decision trees and genetic algorithms to generate new detection rules. The experiments conducted on KDDCup99, NSL-KDD, and UNSW-NB15 yielded results of 98.85%, 97.55%, and 84.33%, respectively. The authors in [58] exploited incremental machine learning to build an effective network intrusion prevention system for IoT. The proposed system includes an online-cluster algorithm powered by the self-organizing incremental neural network and multiple support vector machines to classify the attacks. The evaluation results on the NSL-KDD dataset show that this system achieves 89.67% detection accuracy. Similarly, Ref. [59] used network profiling and a machine learning-based IDS to secure the IoT network. In more detail, the networking behaviors of connected IoT devices are defined as profiles. Any high deviation of these profiles is considered an attack sign and transferred to an attack detection model based on the MobileNet convolutional neural network. In the experiments on the testbed, the overall accuracy is reported at 98.35%.

To cope with novel cyber-attacks and increase the detection quality, the mentioned IDS works tend to employ highly complex machine learning models trained by public network traffic datasets. This complexity significantly affects the feasibility of these works because IDSs are mainly deployed on resource-constrained network devices (e.g., routers, switches, and firewalls) [60]. In addition, enhancing the detection accuracy of these models by generating more training data was not discussed. This paper introduces an effective intrusion detection system powered by a CNN model, namely IMIDS, and a novel attack data generator leveraging a conditional generative adversarial network. IMIDS could not only differentiate the normal and abnormal activities, but identify the type of cyber-attacks hidden behind these activities. Furthermore, IMIDS's detection performance is notably improved after being further trained by the training data generated by our attack data generator.

3. The IMIDS IDS

In this section, we explain how IMIDS works. First of all, the operation of IMIDS is divided into the training and prediction phases, the flow direction and key components of which are illustrated in Figure 1. In more detail, IMIDS is composed of the following components:

- Network capture block: It captures the raw network packets from network traffic and extracts network flows from them by using external libraries (e.g., afpacket [61], tshark [62]).
- Feature extraction block: This component is responsible for calculating network flow statistics and extracting network features from network flows. Its output is a set of network features compatible with the attack detection model.
- Train/load model block: In the prediction phases, the attack detection model, a ten-layer convolutional neural network, detects abnormal activity by monitoring current network traffic. In contrast, the training phase is used to offline update this model by retraining it with labeled network data consisting of 197 network features. After training, the model is temporarily saved in storage until the administrator triggers the deployment process, replacing the running model with the trained model. To maintain the detection accuracy, we frequently retrain the model with novel attack data via a web-based management interface. Indeed, it is updated offline. This means that the model is offline trained with novel collected network data after a certain duration. This model is then saved in model storage before deployment in the IDS device via a web-based management interface controlled by network administrators.

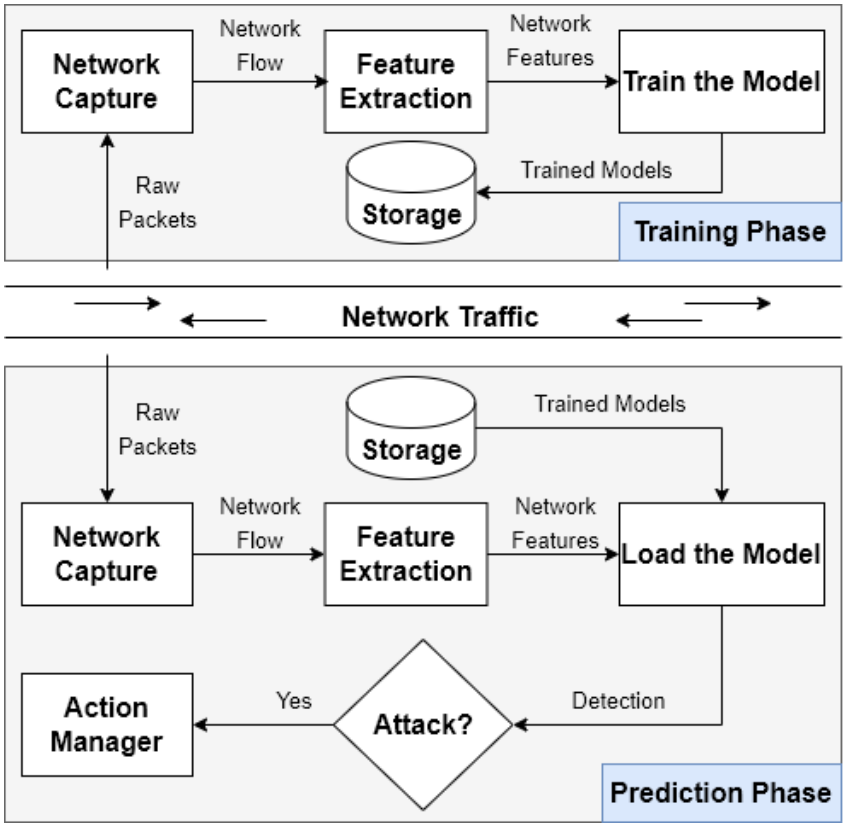

Figure 1. Main operational phases of IMIDS.

3.1. Attack Detection Model

Motivated by the success of the convolutional neural network (CNN) in several domains (e.g., computer vision, natural language processing), most well-known image classification models are built based on CNN architecture, such as Resnet, EfficientNet, and

MobileNet. In addition, compared with recurrent neural network (RNN) and deep neural network (DNN), CNN is much faster in classifying imagery data. Aiming to minimize the attack detection model's complexity while maintaining high detection accuracy, we designed IMIDS's attack detection model based on CNN architecture that has only ten layers and 890.826 parameters. The ten layers include convolutional, max-pooling, and fully connected layers, and its operation is separated into two phases:

- Feature extraction phase : The input layer receives network features with (197, 1) shape. After going through double 1D convolutional layers and the activation function ReLU, we obtain the feature map with 64 filters. This feature map passes a max-pooling layer to reduce the dimension from (197, 64) to (98, 64). It then goes through two convolutional layers and a max-pooling layer to receive a new feature map with shape (48, 128).
- Attack classification phase: The feature map is flattened to 6272 vector features and passes two fully connected layers to calculate the probability of 10-dimension outputs corresponding to 10 labels. The dropout value of the first fully connected layer is set at 0.5 to prevent overfit. The model architecture is shown in Figure 2.

The details of each layer and activation function are described below:

- Convolutional layer: This aims to extract the advanced features by using multiple kernels, which have the training weight and bias, to compute the feature maps. These maps of each kernel are the result of a region in the previous layer. Let (x, y) and (n, m) with $n \leq x$, $m \leq y$ denote the size of sample s and kernel w, respectively. The convolutional process is described as follows:

$$Conv_{x,y} = \sum_{i}^{n.m} W_i \cdot s_i \tag{1}$$

Then, *bias* is added to the output before applying the non-linear activation function h

$$Conv_{x,y} = h\left(\sum_{i}^{n.m} W_i \cdot s_i + bias\right) \tag{2}$$

In our model, the activation function h is the rectified linear units (ReLU) function, defined as:

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

- Fully connected layer: After passing the max-pooling layer, which reduces the dimensions of the previous output layer by keeping the maximum value in a block, the feature map is flattened and sent to the fully connected layer to compute the feature distribution. In this layer, every input unit connects to all activation units. To avoid overfitting, we drop 50% of connections by setting the weight of these connections to zero.
- Output layer: The output layer receives the result from the fully connected layer to compute the loss and update the weight of kernels. In our model, we employed categorical cross-entropy in the loss function, which is defined as:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} L_{y_i \in C_c} \log P_{model}[y_i \in C_c]$$

The term $L_{y_i \in C_c}$ is the indicator function of the ith observation belonging to the cth category, and the $P_{model}[y_i \in C_c]$ is the probability predicted by the model for this ith value. When there are more than two categories, the neural network outputs a vector of C probabilities, each giving the probability that the network input should be classified as belonging to the respective category.

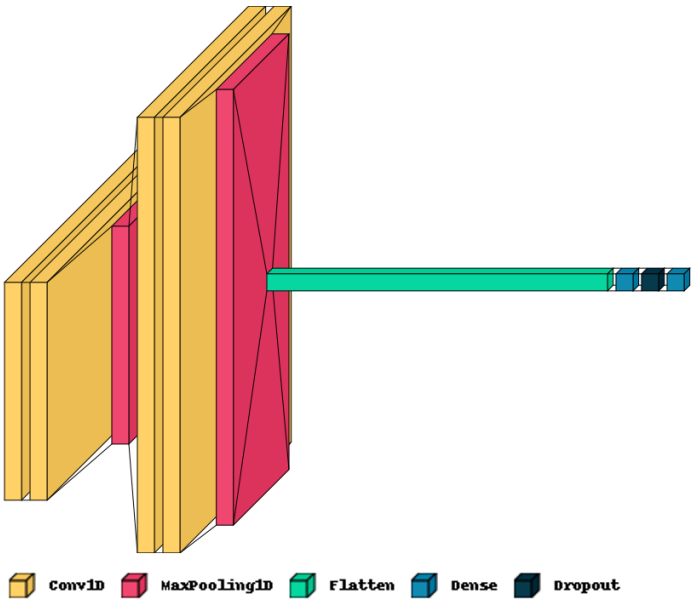

Figure 2. Structure of proposed convolutional neural network.

3.2. Attack Data Generator

Currently, high-quality datasets for training the IDSs are hard to obtain, although they play an essential role in enhancing IDS detection performance. This is because of privacy concerns. Thus, sensitive network information in these datasets is removed before publishing, leading to several defects in network flows. Moreover, most of the existing IDS datasets are out-of-date and lack labeled attacks. Meanwhile, the generative adversarial network, first introduced in 2014, has been widely used in synthetic data generation tasks due to its effectiveness and distinctive architecture. However, network datasets are formed under tables with discrete and continuous columns that challenge traditional GAN, such as vallinaGAN [63]. To solve this issue, we present a network data generator powered by the conditional GAN (ctGAN) that differs from other GAN models [64–67] in the generator component. Indeed, traditional GAN generators are fed by random data samples from the dataset, so they are inefficient with imbalanced datasets. In contrast, the ctGAN's generator is learned from the distribution of the dataset. In more detail, let K be the value of the i_{th} discrete column d_i, which is represented as generated data $\hat{r}$. The generator learns to represent the conditional distribution of rows or samples given the exact value and exact column denoted as $\hat{r} \sim P_G(row|D_i = K)$. The conditional generator should learn the real distribution of the dataset; then, we rewrite the distribution as:

$$P_G(row|D_i = K) = P(row|D_i = K)$$
$$\iff P_{row} = \sum_{k \in D_i} P_G(row|D_i = K)P(D_i = k)$$

With the aim of generating synthetic discrete tabular data, we propose generators in CTGAN shown in Figure 3 that include three key factors: conditional vector, generation loss, and training by sampling.

- Conditional vector: It is necessary to transform the input to a conditional vector before feeding it to the generator. Conditional vector is interpreted as the condition $(D_i = K)$. Given discrete columns $(D_1, D_2, \ldots, D_{|D|})$ and the mask vector $m_j = [m_j^1, m_j^2, \ldots, m_j^K]$, for $j = 1, 2, \ldots, |D|$, the condition can be defined as a one-hot vector as follows:

$$m_i^k = \begin{cases} 1 & if \ \ i = j \ and \ k = K \\ 0 & otherwise \end{cases}, for \ k = 1, 2, \ldots, |D_i|$$

Then, the conditional vector is

$$Ctvector = m_1 \oplus m_2 \oplus \cdots \oplus m_{|D|}$$

while $\oplus$ is the concatenation symbol. For example, we sample 2 discrete columns $D_1 = 1,2,3,4,5$ and $D_2 = 1,2,4,5$ and the condition is $(D_1 = 5)$, meaning $j = 1$ and $k = 5$ is defined as $m_1 = [0,0,0,1], m_2 = [0,0,0,0]$; hence, the $Ctvector = [0,0,0,1,0,0,0,0,0]$.

- Generator loss: During training process, the generator creates the one-hot discrete vector $\{\widehat{m}_1, \widehat{m}_2, \ldots, \widehat{m}_{|D|}\}$. Moreover, given the $Ctvector$ by the condition $(D_i = K)$ from the sample of input, the generator is penalized by loss cross-entropy between m_i and $\widehat{m}_i$, the average cross dimensions. The loss function is described as follows:

$$Loss(x, class) = -\log\left(\frac{\exp(x[class])}{\sum_j \exp(x[j])}\right)$$

$$= x[class] + \log\left(\sum_j \exp(x[j])\right)$$

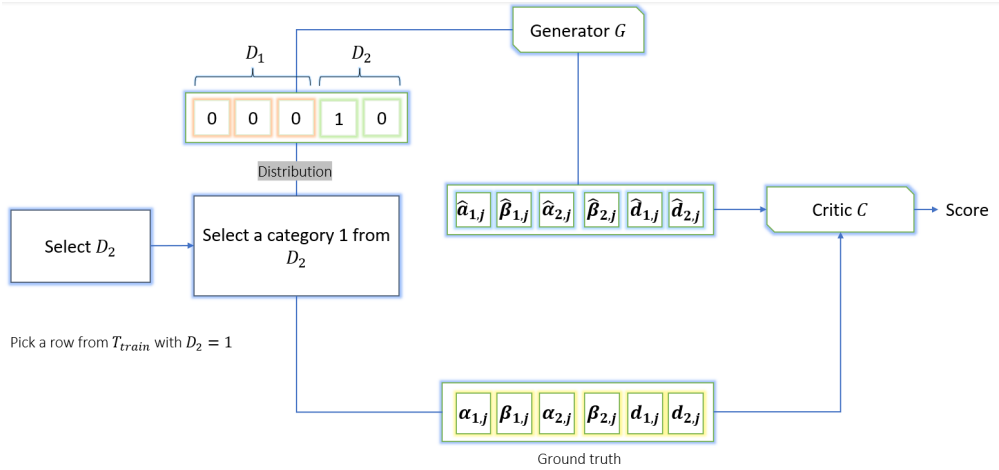

Figure 3. An overview of the attack data generator model.

4. Results and Discussion

In this section, we summarize and analyze the evaluation results of IMIDS in terms of attack detection and attack data generator performance. In more detail, we first introduce evaluated datasets and evaluation metrics in Section A. The detailed evaluation results and discussions are presented in Section B.

4.1. Experimental Datasets and Evaluation Metric

In the experiments, we employed UNSW-NB15 and CICIDS2017 because of their popularity, and they are the most up-to-date datasets about cyber security, aiming to minimize the drawbacks of popular datasets (e.g., KDDCUP, KDD99, NSL-KDD). The UNSW-NB15 dataset includes 2,540,044 records with 49 features, including packet-related features extracted from the packet headers and data fields and more complex features inferred from a sequence of packets, also called flow-based features. These features are divided into five categories: straight-from-headers information, basic flow features, content features, and time features. In UNSW-NB15, the network traffic is categorized into ten different types, including normal traffic, and nine attacks (e.g., fuzzers, analysis, backdoors, DoS, exploits, generic, reconnaissance, shellcode, and worms). In more detail, the normal traffic has 2,218,761 records and overwhelms attack traffic, which has 24,246, 2677, 2329, 16,535, 44,525, 21,5481, 13,987, 1511, and 174 records for fuzzers, analysis, backdoors, DoS, exploits, generic, reconnaissance, shellcode, and worms, respectively.

To evaluate the effectiveness of the attack detection model, we used several evaluation metrics, including the confusion matrix, accuracy, precision, recall, and F1 score. Let TP, TN, FP, and FN denote true positive, true negative, false positive, and false negative. These metrics are defined as follows:

- Accuracy is the ratio of correctly predicted packets over the total packets.

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

- Precision is the ratio of correctly predicted malicious packets over the total predicted malicious packets.

$$Precision = \frac{TP}{TP + FP}$$

- Recall represents the ratio of correctly predicted malicious packets over the total malicious packets.

$$Recall = \frac{TP}{TP + FN}$$

- F1 score measures the harmonic mean of precision and recall; it is a single weighted metric to evaluate detection performance.

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

4.2. Results and Discussion

Attack detection performance: The evaluation results of detecting nine attack types and two datasets in Table 1 show that all evaluation metrics of IMIDS are higher than 95%. In more detail, IMIDS achieves accuracy of 96.69% and 95.92% on average for UNSW-NB15 and CICIDS2017, respectively. To gain better insights into the detection performance for each cyber-attack, we illustrate the confusion matrix in Figure 4. From this figure, we note that our proposal could detect reconnaissance, generic, and shellcode attacks with accuracy of 80.24%, 97.27%, and 80.98%, respectively, whereas its detection accuracy for backdoor and analysis attacks is low and recorded at around 14.77% and 35.58%. By investigating further, we find that the low detection quality results from the lack of training samples, which total 2677 and 2329 for backdoors and analysis, respectively. This issue motivates us to develop a network attack data generator powered by a generative neural network.

Attack data generator performance: The proposed attack data generator aims to enhance the detection quality for attacks pre-trained by public datasets. As shown in Figure 5, the detection accuracy of backdoors, analysis, and worms is significantly improved after we retrain the attack detection model with 10.000 synthetic samples generated by our generator for each of these attack types. In detail, the accuracy of detecting analysis and backdoors attacks increases from 35.58% and 14.77% to 70.94% and 37.08%, respectively. To further investigate the changes in model performance after the additional training, we show the model's confusion matrix after training with synthetic samples of backdoors attacks in Figure 6. We can see minor changes in the detection performance for other attack types. For example, the detection accuracy of generic and worms attacks slightly changes from 98.38% and 49.52% to 97.36% and 54.55%, respectively. The largest change is found in the DoS attack, which decreases by 9.22% from 74.96% to 65.74%. This is because approximately 572 DoS attack samples are recognized as backdoor attacks. These experimental results demonstrate the effectiveness of our attack data generator in enhancing the attack detection quality.

Table 1. The attack detection performance of IMIDS.

Dataset	Accuracy	Recall	Precision	F1 Score
UNSW-NB15	0.9669	0.9828	0.9669	0.9722
CICIDS2017	0.9592	0.9592	0.9720	0.9634

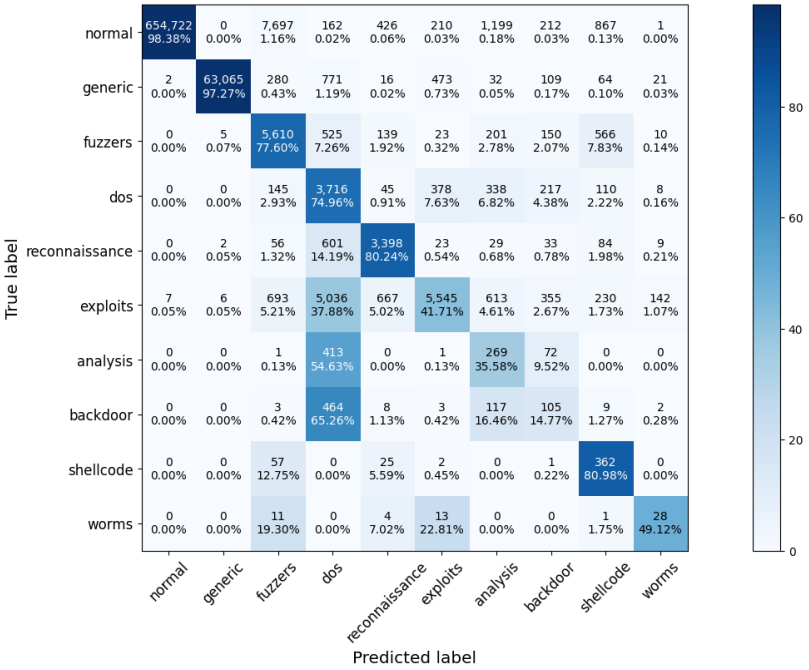

Figure 4. Identifying attack type confusion matrix, which shows how the predicted attacks compare against the actual ones.

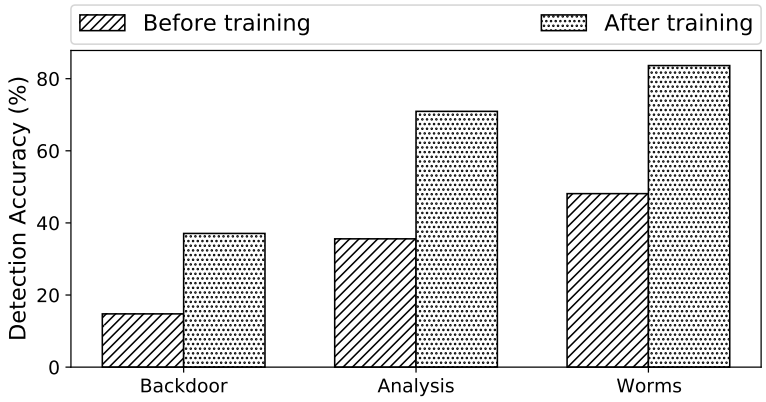

Figure 5. Comparing the attack detection accuracy of IMIDS before and after training with synthetic data generated by the proposed attack data generator.

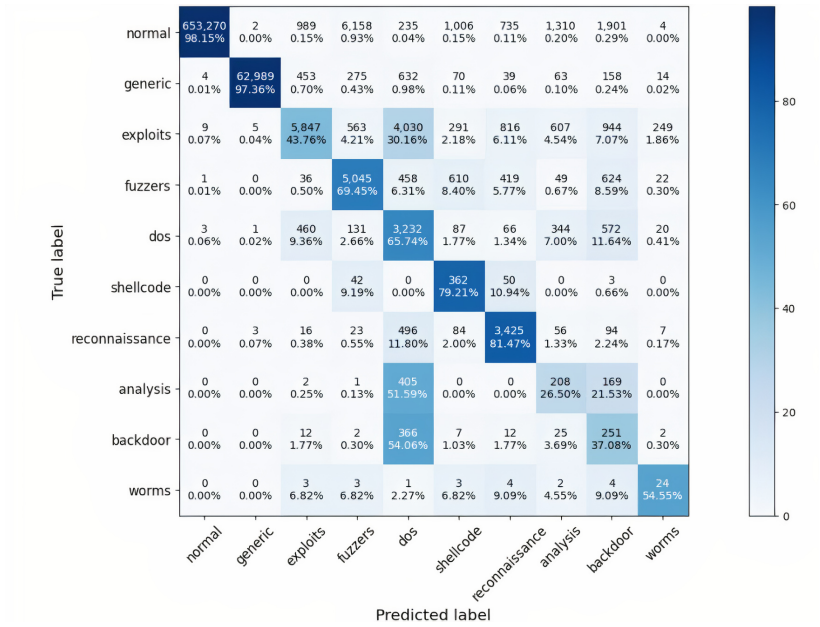

Figure 6. The attack detection model's confusion matrix after training with synthetic samples of backdoors attack.

Comparing baselines: We examine several deep learning models (e.g., support vector machine (SVM), k-nearest neighbors (KNN), linear discriminant analysis (LDA), artificial neural network (ANN)) and state-of-the-art IDSs. A comparative analysis of detection quality, shown in Tables 2 and 3, shows that IMIDS has comparable detection accuracy with the best of the competitors in each type of attack. For example, as reported in Table 2, GRU and ANN achieve the best accuracy in detecting generic and fuzzers attacks, with 97.93% and 78.78% accuracy; equivalently, the values of IMIDS are 97.27% and 77.60%. It should be noted that there are no existing IDSs that could effectively identify all attacks. On the other hand, IMIDS outperforms its competitors in detecting DoS, reconnaissance, analysis, shellcode, and worms. For instance, its accuracy in detecting analysis and worms attacks is 70.94% and 83.64%, whereas the best competitors yield values of 47.62% and 78.18%. As shown in Table 3, the detection performance of our proposed system outperforms existing IDSs. In more detail, the average detection accuracy of IMIDS is around 74.28%, whereas the best of the competitors is 63.61%. For each type of attack, IMIDS has a competitive result with the best of the competitors. For example, the proposed IDSs in [55,56] have the best accuracy in detecting generic and reconnaissance attacks, with 98.33% and 80.77% accuracy, whereas the values of IMIDS are 97.27% and 80.24%, respectively. In summary, the comparative analysis of detection quality reported in Tables 2 and 3 shows that our proposal could accurately detect various cyber-attacks and outperform its competitors.

Statistical significance: To evaluate the statistical significance of the proposed approach, five related IDSs over ten attack types were considered and subjected to experiments using the ANOVA test. The null hypothesis states that all IDSs are similar, and thus their population means should be equal. The test result in Table 4 shows that the ANOVA test rejected the null hypothesis because the p-value is 0.009 and lower than 0.05. In addition, the F-value is more significant than F-critical, which are reported at approximately 3.07 and 2.21, respectively. Therefore, we reject the null hypothesis, and the evaluation results are statistically significant.

Table 2. Comparing the attack classification accuracy between IMIDS and recent models for IDSs.

	ANN	LSTM	GRU	SVM	K-Nearest Neighbors	LDA	IMIDS
Normal	98.26%	98.44%	97.59%	97.79%	97.04%	97.72%	98.38%
Exploits	41.22%	46.73%	43.17%	43.93%	47.45%	42.49%	41.71%
Generic	97.38%	94.42%	97.93%	97.33%	97.31%	97.34%	97.27%
DoS	23.53%	54.42%	43.03%	69.60%	39.49%	44.91%	74.96%
Fuzzers	78.78%	77.55%	75.61%	69.45%	69.12%	58.58%	77.60%
Reconnaissance	77.38%	78.54%	76.61%	70.75%	77.87%	64.99%	80.24%
Backdoors	59.39%	14.91%	31.14%	5.97%	18.06%	30.69%	37.08%
Shellcode	57.76%	66.07%	41.16%	27.57%	63.93%	53.69%	80.98%
Analysis	27.91%	33.66%	34.80%	45.96%	47.62%	22.53%	70.94%
Worms	9.43%	34.43%	10.26%	9.09%	21.82%	78.18%	83.64%
Average Accuracy	57.10%	59.92%	55.13%	53.74%	57.97%	59.11%	74.28%

Table 3. Comparing the attack classification accuracy between IMIDS and recent articles.

	Rajagopal et al. [55]	Meftah et al. [56] Decision Trees	Meftah et al. [56] Naïve Bayes	Meftah et al. [56] SVM	Roy et al. [54]	Papa et al. [57]	IMIDS
Normal	91.83%	74.93%	64.54%	62.41%	100%	97.39%	98.38%
Exploits	85.07%	90.08%	24.97%	85.22%	92.84%	76.22%	41.71%
Generic	98.33%	96.96%	96.29%	96.24%	98.21%	81.37%	97.27%
DoS	25.07%	8.33%	0%	1.07%	10.23%	14.29%	74.96%
Fuzzers	60.98%	55.24%	36.28%	76.26%	88.91%	64.42%	77.60%
Reconnaissance	74.87%	80.77%	49.57%	68.44%	75.47%	46.07%	80.24%
Backdoors	10.79%	4.97%	22.47%	0%	10.60%	67.32%	37.08%
Shellcode	58.23%	60.84%	1.32%	0%	75.29%	36.39%	80.98%
Analysis	11.09%	0%	0%	0%	23.1%	20.45%	70.94%
Worms	37.5%	72.72%	38.64%	0%	61.54%	18.37%	83.64%
Average Accuracy	55.37%	54.53%	33.41%	38.96%	63.61%	52.23%	74.28%

Table 4. ANOVA results for accuracy rate of network attack classifiers.

Source of Variation	SS	df	MS	F	*p*-Value	F-Crit
Between groups	1.928515	9	0.214279	3.072905	0.009884	2.210697
Within groups	2.091956	30	0.069732			
Total	4.020471	39				

SS: sum of squared deviations about mean; df: degrees of freedom; MS: variance.

From these detection performance results, we can conclude that the proposed IDS has fairly high detection efficiency compared to previously introduced algorithms, and the data generated from ctGAN help to increase the efficiency of attack detection quite significantly, which opens up a novel direction in the field of network security.

5. Conclusions

In this paper, we propose a CNN-based IDS named IMIDS to accurately detect multiple types of cyber-attacks, and we construct a novel attack data generator powered by a generative neural network. We first detail the design of IMIDS, which comprises two key components: feature extractor and attack detection model. The feature extractor is responsible for extracting network features from captured network packets. The attack detection model is in charge of detecting malicious activities according to a novel convolutional neural network. To enhance the detection performance, we then retrained the detection model with several synthetic datasets about specific attacks (e.g., analysis, worms, and backdoors) created by our generator. The evaluation results on two popular IDS datasets

show that IMIDS could detect nine types of cyber-attacks, with an average F-measure of 97.22% and 96.34%. In addition, its detection performance is notably improved after being further trained by the training data generated by our attack data generator.

Author Contributions: Conceptualization, K.-H.L. and M.-H.N.; methodology, K.-H.L.; software, M.-H.N., N.-D.T. and T.-D.T.; validation, K.-H.L. and M.-H.N.; formal analysis, M.-H.N. and K.-H.L.; investigation, K.-H.L., M.-H.N., N.-D.T. and T.-D.T.; resources, K.-H.L.; data curation, M.-H.N., N.-D.T. and T.-D.T.; writing—original draft preparation, T.-D.T., N.-D.T. and M.-H.N.; writing—review and editing, K.-H.L.; visualization, M.-H.N.; supervision, K.-H.L.; project administration, K.-H.L.; funding acquisition, K.-H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by the fund supporting research activities from the University of Information Technology, Vietnam National University Ho Chi Minh City.

Data Availability Statement: Data available on request.

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

References

1. Holst, A. IoT Global Annual Revenue 2019–2030. 2021. Available online: https://www.statista.com/statistics/1194709/iot-revenue-worldwide/ (accessed on 19 October 2021).
2. Khan, M.A.; Salah, K. IoT security: Review, blockchain solutions, and open challenges. *Future Gener. Comput. Syst.* **2018**, *82*, 395–411. [CrossRef]
3. Sonar, K.; Upadhyay, H. A survey: DDOS attack on Internet of Things. *Int. J. Eng. Res. Dev.* **2014**, *10*, 58–63.
4. Antonakakis, M.; April, T.; Bailey, M.; Bernhard, M.; Bursztein, E.; Cochran, J.; Durumeric, Z.; Halderman, J.A.; Invernizzi, L.; Kallitsis, M.; et al. Understanding the mirai botnet. In Proceedings of the 26th {USENIX} Security Symposium ({USENIX} Security 17), Vancouver, BC, Canada, 16–18 August 2017; pp. 1093–1110.
5. Kolias, C.; Kambourakis, G.; Stavrou, A.; Voas, J. DDoS in the IoT: Mirai and other botnets. *Computer* **2017**, *50*, 80–84. [CrossRef]
6. Shaar, F.; Efe, A. DDoS attacks and impacts on various cloud computing components. *Int. J. Inf. Secur. Sci.* **2018**, *7*.
7. Salim, M.M.; Rathore, S.; Park, J.H. Distributed denial of service attacks and its defenses in IoT: A survey. *J. Supercomput.* **2020**, *76*, 5320–5363. [CrossRef]
8. The IoT Rundown For 2020: Stats, Risks, and Solutions. 2020. Available online: https://securitytoday.com/Articles/2020/01/13/The-IoT-Rundown-for-2020.aspx?Page=2 (accessed on 19 October 2021).
9. O'Donnell, L. More Than Half of IoT Devices Vulnerable to Severe Attacks. 2020. Available online: https://threatpost.com/half-iot-devices-vulnerable-severe-attacks/153609/ (accessed on 19 October 2021).
10. Ayodeji, A.; Liu, Y.K.; Chao, N.; Yang, L.Q. A new perspective towards the development of robust data-driven intrusion detection for industrial control systems. *Nucl. Eng. Technol.* **2020**, *52*, 2687–2698. [CrossRef]
11. Thakkar, A.; Lohiya, R. A survey on intrusion detection system: Feature selection, model, performance measures, application perspective, challenges, and future research directions. *Artif. Intell. Rev.* **2022**, *55*, 453–563. [CrossRef]
12. Thakkar, A.; Lohiya, R. A review of the advancement in intrusion detection datasets. *Procedia Comput. Sci.* **2020**, *167*, 636–645. [CrossRef]
13. Schurgot, M.R.; Shinberg, D.A.; Greenwald, L.G. Experiments with security and privacy in IoT networks. In Proceedings of the 2015 IEEE 16th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), Boston, MA, USA, 14–17 June 2015; pp. 1–6.
14. Cordero, C.G.; Vasilomanolakis, E.; Wainakh, A.; Mühlhäuser, M.; Nadjm-Tehrani, S. On generating network traffic datasets with synthetic attacks for intrusion detection. *ACM Trans. Priv. Secur. (TOPS)* **2021**, *24*, 1–39. [CrossRef]
15. Quincozes, S.E.; Albuquerque, C.; Passos, D.; Mossé, D. A survey on intrusion detection and prevention systems in digital substations. *Comput. Netw.* **2021**, *184*, 107679. [CrossRef]
16. Eskandari, M.; Janjua, Z.H.; Vecchio, M.; Antonelli, F. Passban IDS: An intelligent anomaly-based intrusion detection system for IoT edge devices. *IEEE Internet Things J.* **2020**, *7*, 6882–6897. [CrossRef]
17. Mirsky, Y.; Doitshman, T.; Elovici, Y.; Shabtai, A. Kitsune: An ensemble of autoencoders for online network intrusion detection. *arXiv* **2018**, arXiv:1802.09089.
18. Zhao, H.; Feng, Y.; Koide, H.; Sakurai, K. An ANN Based Sequential Detection Method for Balancing Performance Indicators of IDS. In Proceedings of the 2019 Seventh International Symposium on Computing and Networking (CANDAR), Nagasaki, Japan, 25–28 November 2019; pp. 239–244.
19. Golrang, A.; Golrang, A.M.; Yildirim Yayilgan, S.; Elezaj, O. A novel hybrid IDS based on modified NSGAII-ANN and random forest. *Electronics* **2020**, *9*, 577. [CrossRef]
20. Naoum, R.S.; Abid, N.A.; Al-Sultani, Z.N. An enhanced resilient backpropagation artificial neural network for intrusion detection system. *Int. J. Comput. Sci. Netw. Secur. (IJCSNS)* **2012**, *12*, 11.

21. Ahmim, A.; Maglaras, L.; Ferrag, M.A.; Derdour, M.; Janicke, H. A novel hierarchical intrusion detection system based on decision tree and rules-based models. In Proceedings of the 2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS), Santorini, Greece, 29–31 May 2019; pp. 228–233.
22. Moustafa, N.; Slay, J. A hybrid feature selection for network intrusion detection systems: Central points. *arXiv* **2017**, arXiv:1707.05505.
23. Gharaee, H.; Hosseinvand, H. A new feature selection IDS based on genetic algorithm and SVM. In Proceedings of the 2016 8th International Symposium on Telecommunications (IST), Tehran, Iran, 27–28 September 2016; pp. 139–144.
24. Belouch, M.; El Hadaj, S.; Idhammad, M. A two-stage classifier approach using reptree algorithm for network intrusion detection. *Int. J. Adv. Comput. Sci. Appl.* **2017**, *8*, 389–394. [CrossRef]
25. Baig, M.M.; Awais, M.M.; El-Alfy, E.S.M. A multiclass cascade of artificial neural network for network intrusion detection. *J. Intell. Fuzzy Syst.* **2017**, *32*, 2875–2883. [CrossRef]
26. Al-Zewairi, M.; Almajali, S.; Awajan, A. Experimental evaluation of a multi-layer feed-forward artificial neural network classifier for network intrusion detection system. In Proceedings of the 2017 International Conference on New Trends in Computing Sciences (ICTCS), Amman, Jordan, 11–13 October 2017; pp. 167–172.
27. Guha, S.; Yau, S.S.; Buduru, A.B. Attack detection in cloud infrastructures using artificial neural network with genetic feature selection. In Proceedings of the 2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), Auckland, New Zealand, 8–12 August 2016; pp. 414–419.
28. Nguyen, K.K.; Hoang, D.T.; Niyato, D.; Wang, P.; Nguyen, D.; Dutkiewicz, E. Cyberattack detection in mobile cloud computing: A deep learning approach. In Proceedings of the 2018 IEEE wireless communications and networking conference (WCNC), Barcelona, Spain, 15–18 April 2018; pp. 1–6.
29. Moustafa, N.; Misra, G.; Slay, J. Generalized outlier gaussian mixture technique based on automated association features for simulating and detecting web application attacks. *IEEE Trans. Sustain. Comput.* **2018**, *6*, 245–256. [CrossRef]
30. Moustafa, N.; Adi, E.; Turnbull, B.; Hu, J. A new threat intelligence scheme for safeguarding industry 4.0 systems. *IEEE Access* **2018**, *6*, 32910–32924. [CrossRef]
31. Chowdhury, M.N.; Ferens, K.; Ferens, M. Network intrusion detection using machine learning. In Proceedings of the International Conference on Security and Management (SAM), Las Vegas, NV, USA, 25–28 July 2016; The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp): Las Vegas, NV, USA, 2016; p. 30.
32. Bhamare, D.; Salman, T.; Samaka, M.; Erbad, A.; Jain, R. Feasibility of supervised machine learning for cloud security. In Proceedings of the 2016 International Conference on Information Science and Security (ICISS), Pattaya, Thailand, 19–22 December 2016; pp. 1–5.
33. Yin, C.; Zhu, Y.; Liu, S.; Fei, J.; Zhang, H. An enhancing framework for botnet detection using generative adversarial networks. In Proceedings of the 2018 International Conference on Artificial Intelligence and Big Data (ICAIBD), Chengdu, China, 26–28 May 2018; pp. 228–234.
34. Ahmadian, S.; Malki, H.; Han, Z. Cyber attacks on smart energy grids using generative adverserial networks. In Proceedings of the 2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP), Anaheim, CA, USA, 26–29 November 2018; pp. 942–946.
35. Lee, J.; Park, K. AE-CGAN model based high performance network intrusion detection system. *Appl. Sci.* **2019**, *9*, 4221. [CrossRef]
36. Han, L.; Sheng, Y.; Zeng, X. A packet-length-adjustable attention model based on bytes embedding using flow-wgan for smart cybersecurity. *IEEE Access* **2019**, *7*, 82913–82926. [CrossRef]
37. Shahriar, M.H.; Haque, N.I.; Rahman, M.A.; Alonso, M. G-ids: Generative adversarial networks assisted intrusion detection system. In Proceedings of the 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC), Madrid, Spain, 13–17 July 2020; pp. 376–385.
38. de Araujo-Filho, P.F.; Kaddoum, G.; Campelo, D.R.; Santos, A.G.; Macêdo, D.; Zanchettin, C. Intrusion detection for cyber–physical systems using generative adversarial networks in fog environment. *IEEE Internet Things J.* **2020**, *8*, 6247–6256. [CrossRef]
39. Huang, S.; Lei, K. IGAN-IDS: An imbalanced generative adversarial network towards intrusion detection system in ad-hoc networks. *Ad Hoc Netw.* **2020**, *105*, 102177. [CrossRef]
40. Zhao, S.; Li, J.; Wang, J.; Zhang, Z.; Zhu, L.; Zhang, Y. attackGAN: Adversarial Attack against Black-box IDS using Generative Adversarial Networks. *Procedia Comput. Sci.* **2021**, *187*, 128–133. [CrossRef]
41. Lohiya, R.; Thakkar, A. Application Domains, Evaluation Datasets, and Research Challenges of IoT: A Systematic Review. *IEEE Internet Things J.* **2020**, *8*, 8774–8798. [CrossRef]
42. Thakkar, A.; Lohiya, R. A review on machine learning and deep learning perspectives of IDS for IoT: Recent updates, security issues, and challenges. *Arch. Comput. Methods Eng.* **2021**, *28*, 3211–3243. [CrossRef]
43. He, H.; Garcia, E.A. Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.* **2009**, *21*, 1263–1284.
44. Thakkar, A.; Lohiya, R. Analyzing fusion of regularization techniques in the deep learning-based intrusion detection system. *Int. J. Intell. Syst.* **2021**, *36*, 7340–7388. [CrossRef]
45. Thakkar, A.; Lohiya, R. Attack classification using feature selection techniques: A comparative study. *J. Ambient. Intell. Humaniz. Comput.* **2021**, *12*, 1249–1266. [CrossRef]

46. Thakkar, A.; Lohiya, R. Role of swarm and evolutionary algorithms for intrusion detection system: A survey. *Swarm Evol. Comput.* **2020**, *53*, 100631. [CrossRef]
47. Shaukat, K.; Luo, S.; Varadharajan, V.; Hameed, I.A.; Xu, M. A survey on machine learning techniques for cyber security in the last decade. *IEEE Access* **2020**, *8*, 222310–222354. [CrossRef]
48. Shaukat, K.; Luo, S.; Chen, S.; Liu, D. Cyber Threat Detection Using Machine Learning Techniques: A Performance Evaluation Perspective. In Proceedings of the 2020 International Conference on Cyber Warfare and Security (ICCWS), Islamabad, Pakistan, 20–21 October 2020; pp. 1–6.
49. Shaukat, K.; Luo, S.; Varadharajan, V.; Hameed, I.A.; Chen, S.; Liu, D.; Li, J. Performance comparison and current challenges of using machine learning techniques in cybersecurity. *Energies* **2020**, *13*, 2509. [CrossRef]
50. Liu, Y.; Liu, S.; Zhao, X. Intrusion detection algorithm based on convolutional neural network. *DEStech Trans. Eng. Technol. Res.* **2017**. [CrossRef]
51. Lin, W.H.; Lin, H.C.; Wang, P.; Wu, B.H.; Tsai, J.Y. Using convolutional neural networks to network intrusion detection for cyber threats. In Proceedings of the 2018 IEEE International Conference on Applied System Invention (ICASI), Chiba, Japan, 13–17 April 2018; pp. 1107–1110.
52. Lohiya, R.; Thakkar, A. Intrusion detection using deep neural network with antirectifier layer. In *Applied Soft Computing and Communication Networks*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 89–105.
53. 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.
54. Roy, A.; Singh, K.J. Multi-classification of UNSW-NB15 Dataset for Network Anomaly Detection System. In *Proceedings of the International Conference on Communication and Computational Technologies*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 429–451.
55. Rajagopal, S.; Kundapur, P.P.; Hareesha, K.S. A stacking ensemble for network intrusion detection using heterogeneous datasets. *Secur. Commun. Netw.* **2020**, *2020*. [CrossRef]
56. Meftah, S.; Rachidi, T.; Assem, N. Network based intrusion detection using the UNSW-NB15 dataset. *Int. J. Comput. Digit. Syst.* **2019**, *8*, 478–487.
57. Papamartzivanos, D.; Mármol, F.G.; Kambourakis, G. Dendron: Genetic trees driven rule induction for network intrusion detection systems. *Future Gener. Comput. Syst.* **2018**, *79*, 558–574. [CrossRef]
58. Rose, J.R.; Swann, M.; Bendiab, G.; Shiaeles, S.; Kolokotronis, N. Intrusion Detection using Network Traffic Profiling and Machine Learning for IoT. In Proceedings of the 2021 IEEE 7th International Conference on Network Softwarization (NetSoft), Tokyo, Japan, 28 June–2 July 2021; pp. 409–415. [CrossRef]
59. Constantinides, C.; Shiaeles, S.; Ghita, B.; Kolokotronis, N. A Novel Online Incremental Learning Intrusion Prevention System. In Proceedings of the 2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS), Canary Islands, Spain, 24–26 June 2019; pp. 1–6. [CrossRef]
60. Sha, K.; Yang, T.A.; Wei, W.; Davari, S. A survey of edge computing-based designs for IoT security. *Digit. Commun. Netw.* **2020**, *6*, 195–202. [CrossRef]
61. Karlsson, M.; Töpel, B.; Fastabend, J.; Covalent, I. AF PACKET V4 and PACKET ZEROCOPY. In Proceedings of the Netdev Conference, Seoul, Korea, 8–10 November 2017; Volume 2.
62. Merino, B. *Instant Traffic Analysis with Tshark How-To*; Packt Publishing Ltd.: Birmingham, UK, 2013.
63. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 2672–2680.
64. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein generative adversarial networks. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 214–223.
65. Mirza, M.; Osindero, S. Conditional generative adversarial nets. *arXiv* **2014**, arXiv:1411.1784.
66. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A. Improved training of wasserstein gans. *arXiv* **2017**, arXiv:1704.00028.
67. Odena, A.; Olah, C.; Shlens, J. Conditional image synthesis with auxiliary classifier gans. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 2642–2651.