*mathematics*

MDPI

*Article*

# Detection of Unknown DDoS Attack Using Convolutional Neural Networks Featuring Geometrical Metric

**Chin-Shiuh Shieh [1], Thanh-Tuan Nguyen [1,2,*] and Mong-Fong Horng [1,3,*]**

[1] Department of Electronic Engineering, National Kaohsiung University of Science and Technology, Kaohsiung 807618, Taiwan; csshieh@nkust.edu.tw
[2] Department of Electronic and Automation Engineering, Nha Trang University, Nha Trang 650000, Vietnam
[3] Ph.D Program in Biomedical Engineering, Kaohsiung Medial University, Kaohsiung 80708, Taiwan
* Correspondence: tuannt@ntu.edu.vn (T.-T.N.); mfhorng@nkust.edu.tw (M.-F.H.)

**Abstract:** DDoS attacks remain a persistent cybersecurity threat, blocking services to legitimate users and causing significant damage to reputation, finances, and potential customers. For the detection of DDoS attacks, machine learning techniques such as supervised learning have been extensively employed, but their effectiveness declines when the framework confronts patterns exterior to the dataset. In addition, DDoS attack schemes continue to improve, rendering conventional data model-based training ineffectual. We have developed a novelty open-set recognition framework for DDoS attack detection to overcome the challenges of traditional methods. Our framework is built on a Convolutional Neural Network (CNN) construction featuring geometrical metric (CNN-Geo), which utilizes deep learning techniques to enhance accuracy. In addition, we have integrated an incremental learning module that can efficiently incorporate novel unknown traffic identified by telecommunication experts through the monitoring process. This unique approach provides an effective solution for identifying and alleviating DDoS. The module continuously improves the model's performance by incorporating new knowledge and adapting to new attack patterns. The proposed model can detect unknown DDoS attacks with a detection rate of over 99% on conventional attacks from CICIDS2017. The model's accuracy is further enhanced by 99.8% toward unknown attacks with the open datasets CICDDoS2019.

**Keywords:** cybersecurity; distributed denial-of-service (DDoS); convolutional neural networks (CNN); geometrical metric; incremental learning; open-set recognition (OSR); machine learning; deep learning; unknown attack; CICIDS2017; CICDDoS2019

**MSC:** 68T07



**Citation:** Shieh, C.-S.; Nguyen, T.-T.; Horng, M.-F. Detection of Unknown DDoS Attack Using Convolutional Neural Networks Featuring Geometrical Metric. *Mathematics* **2023**, *11*, 2145. https://doi.org/10.3390/math11092145

Academic Editors: Chi-Yao Weng, Shoko Wakamiya, Chun-Ta Li and Cheng-Ta Huang

Received: 13 April 2023
Revised: 28 April 2023
Accepted: 29 April 2023
Published: 3 May 2023



## 1. Introduction

In recent years, the development of Artificial Intelligence (AI) technology has significantly contributed to various disciplines, including cybersecurity [1]. One significant issue in cybersecurity is the DDoS attacks, which has escalated over many years [2]. DDoS attacks disrupt legitimate users' access to services by injecting enormous volumes of malicious traffic quickly, costing the victims their reputation, resources, and potential clients [3]. The outbreak of the COVID-19 pandemic in 2020 has resulted in an increased reliance on network infrastructure, leading to a notable surge in DDoS attacks [4,5]. Since many businesses function as service providers, they must maintain uninterrupted operations. Consequently, any disruptions stemming from a compromised network or service can lead to significant financial and reputational damages [6]. According to Cloudflare, a vendor of Content Delivery Networks (CDN), a considerable number of DDoS attacks are initiated each month, as stated in their quarterly report on DDoS attacks [7]. Even though most malicious traffic records are beneath 500 Mbps, such a volume possesses the capability to cause temporary interruptions to multiple enterprise systems. Quarterly,

targeted assaults with a maximum capacity of 100 Gbps transpire, leading to extensive service interruptions and probable shutdowns of data centers and culminating in revenue losses for service providers.

The techniques employed in DDoS attacks are subject to constant evolution, as evidenced by the literature [8]. Employing outdated countermeasures is insufficient to protect against novel threats [9]. As a result, there is a need for an approach that facilitates the identification of unknown data attributes by the existing intrusion detection system (IDS). This mechanism would assist telecommunications technicians in detecting covert intrusion. In recent years, significant progress has been achieved in AI technology, and the associated work has been applied in various fields, consisting of cybersecurity [10]. Several different IDSs based on deep learning have been developed, and they all expose remarkable accuracy. Relevant experiments demonstrate that the accuracy of identifying standard DDoS may reach higher than 90% [11–13]. In the event that a conventional IDS is faced with novel forms of attacks, it does not classify them as unknown and inept in addressing them. Consequently, there is a necessity for an IDS that can promptly notify the telecom technician of any unfamiliar traffic for examination at the onset of an attack, instead of assessing its nature as positive or negative. This is particularly crucial when the distinction between previous and current threats is markedly apparent. The response of the defense system will be of utmost importance in the event of an assault characterized by discrete fundamental components. This suggests that the problem no longer pertains to the efficacy of the training process. One possible solution to address the issue at hand is to update both the training and test datasets. However, it is important to note that the model faces a significant challenge regarding unknown traffic, and the open set presents a more complex scenario than the closed set.

This study addresses the limitations of existing IDS architectures, which often struggle to detect unknown traffic in DDoS attacks, by proposing a novel IDS architecture that leverages deep learning technology. Our approach combines deep learning techniques and geometrical metrics to enhance accuracy and improved detection of unknown traffic. The model's backbone, CNN-Geo, is based on CNN architecture and incorporates a geometrical metric, which offers enhanced detection capabilities. Furthermore, the system's incremental learning module allows it to adapt to new attack patterns by incorporating newly labeled samples provided by telecom engineers, continuously improving its defensive performance. Given that it employs machine learning techniques such as supervised learning, CNN, geometrical metrics, and incremental learning modules to continuously enhance the model's performance by learning how to incorporate new information and adapt to new attack patterns, it is a smart and clever system. This framework's extraordinary level of intelligence allows it to detect unexpected DDoS attacks with high accuracy and maneuver around the restrictions of classical approaches.

The practical applications of our IDS architecture lie in its ability to protect networks and systems against DDoS attacks more effectively than classical approaches. The high detection rate of over 99% against conventional attacks from the well-known CICIDS2017 dataset demonstrates its efficacy. Moreover, the model's accuracy is further enhanced by 99.8% toward unknown attacks as tested on the recent CICIDDoS2019 open datasets. Our findings suggest that the proposed IDS architecture can significantly improve the detection and defense against DDoS attacks, ensuring the security and reliability of network systems in real-world applications.

The remainder of this paper is organized in the following manner: Section 2 offers an overview of relevant literature. Section 3 outlines the underlying assumptions and the proposed detection framework. Section 4 presents the experimental findings, while Section 5 concludes the study and discusses potential avenues for future research.

## 2. Related Work

### *2.1. MC-Based and DL-Based IDS*

Over the past several years, a significant amount of research has been carried out regarding the incorporation of AI technologies into IDS. The study has reported exceptional performance of Random Forest (RF), Support Vector Machine (SVM), CNN, and Long Short-Term Memory (LSTM), which have been extensively researched as reported in reference [14]. However, IDSs that rely on these technologies exhibit limitations in detecting novel attacks, thereby posing potential security threats. Unsupervised learning methodologies, such as autoencoders, have the capability to detect attacks through the modification of thresholds. Nonetheless, it is important to note that the occurrence of false positives may potentially escalate to ten percent [15]. Researchers have developed advanced IDS models utilizing deep learning architectures to address the limitations of traditional IDS systems. The utilization of CNN architectures in IDS models has been explored by Chen et al. [11] and Kim et al. [12], who have reported favorable outcomes with accuracy levels of 94% or greater. Furthermore, the utilization of CSV files and image reconstruction technologies developed by Kaur et al. [16] in CNN defense models has yielded favorable outcomes.

One of the major issues during IDS training is unbalanced data, which can lead to poor model performance. In order to tackle this matter, M. Azizjon and colleagues [17] employed a 1DCNN framework for data categorization, which has been demonstrated to enhance the efficacy of the model. In a previous study, Toupas et al. (2018) utilized SMOTE ENN pseudo-sampling to address imbalanced data and integrated Yeo–Johnson transformation during the preprocessing stage to mitigate the skewed data distribution [18]. It is essential to proceed with caution when utilizing synthetic data, as the characteristics of the simulation may differ from those of the initial distribution and could potentially be erroneous for malicious activity. In addition to CNN architecture, some deep learning architectures, such as LSTM and RNN, have been employed in IDS models and have shown reasonable accuracy, possibly exceeding 90% [19]. These architectures can capture temporal patterns and have been shown to be effective in detecting network intrusions.

Recent research in IDS systems has focused on the development of hybrid models that combine multiple DL architectures. For instance, Mu et al. [20] proposed a hybrid IDS model that integrates CNN, LSTM, and attention mechanisms. The model that was proposed demonstrated superior performance in comparison to conventional machine learning-based IDS models. Additionally, it exhibited a high level of accuracy in identifying both known and unknown attacks. Moreover, the recent development of explainable AI (XAI) techniques has enabled researchers to interpret the predictions of deep learning-based IDS models and enhance their transparency and trustworthiness [21]. For example, Sivamohan and Sridhar [22] proposed a novel XAI-based IDS model that uses an attention-based mechanism to visualize the feature importance and improve the interpretability of the model. These advancements have opened up new possibilities for the development of more robust and transparent IDS systems.

We propose to evaluate an overview of studies of general relevance to the purposes of our proposed work, as well as to compare different approach strategies in the intrusion detection context, based on similar datasets. Equivalently, we have compiled a series of recent works presented in Table 1. The first column indicates the source of each work; the second column describes the dataset applied in the study; the third column gives the characteristic of the problem (collection closed identity or set open identity); and the fourth column provides a summary assessment of the applied method, where the term "heterogeneous" refers to making a comparison between methods.

**Table 1.** Related research was conducted on applying machine learning and deep learning methodologies to detect DDoS attacks.

| Author | Dataset | Coverage of Problems | Technical | Year |
|---|---|---|---|---|
| Chen. J et al. [23] | KDD99, CICIDS2017 | CSR | Multichanel CNN and various MC frameworks to detect DDoS. | 2019 |
| Roopak et al. [13] | CICIDS2017 | CSR | Four distinct classification models based on deep learning: MLP, LSTM, CNN+LSTM, and 1D-CNN, | 2019 |
| Kurniabudi et al. [24] | CICIDS2017 | CSR | Information Gain, RF, Bayes Net, Random Tree, Naive Bayes, and J48 classifier algorithms. | 2020 |
| Swe et al. [25] | CICIDS2017, CSE-CIC-IDS 2018 | CSR | Slow DDoS attack types analysis, gain ratio, chi squared ranking methods, and various machine learning techniques for detection DDoS. | 2021 |
| Chen et al. [11] | CICIDS2017 | CSR | CNN-based network intrusion detection system (NIDS). Detection models were trained using both extracted features and original network data. | 2020 |
| Chapaneri et al. [26] | CICIDS2017 | CSR, OSR | Multilevel Gaussian mixture model capable of accurately classifying network traffic into multiple classifications. | 2021 |
| Shieh et al. [27] | CICIDS2017, CICDDoS 2019 | CSR, OSR | DDoS detection framework using Reconstruct Error and One-Class SVM Featuring Stochastic Gradient Descent. | 2022 |
| Najafimehr et al. [28] | CICIDS2017, CICDDoS 2019 | CSR, OSR | Clustering algorithm and statistical measures to label traffic and MC framework for DDoS detection. | 2022 |
| Proposed Model | CICIDS2017, CICDDoS 2019 | CSR, OSR | CNN-based framework to detect DDoS, Geometrical Metric calculate module and incremental learning as a solution for openset. | 2023 |

### *2.2. Open-Set Recognition*

Open-set recognition poses a more significant challenge than closed-set recognition since it deals with unknown patterns. In recent years, scholars have conducted investigations in this field. A. Bendale et al. have suggested the Openmax category, which alters the quantity of layer outputs from $N$ to $N + 1$ [29]. The Weibull function was employed for probability estimation, which was subsequently subtracted from the overall probability of 1 prior to being fed into the Softmax function. The Weibull analysis is exclusively applied to certain samples obtained from the extremities of the distribution. Subsequently, the distances from the center of the hyper-sphere are computed by utilizing the mean activation vector's output in the feature space. If the distance surpasses the permissible range, it is inferred that the sample does not pertain to any category. The aforementioned approach is commonly referred to as the Out-of-Distribution (OOD) method. Bendale's research is founded on highly theoretical principles and is frequently employed in the task of categorizing images. The Classification-Reconstruction learning for Open-Set Recognition (CROSR) architecture [30], which combines reconstruction and distribution, is used to determine the hypersphere distribution. Concurrently, the reconstructed hidden layer's output is utilized to improve the detection efficacy. The theoretical framework of the extreme theory is founded on the notion of a spatial distribution that closely approximates a probability density function. In the event that a recently acquired specimen is situated outside the acceptable range, it is classified as unknown.

Over the past few years, there have been several advancements in open-set recognition. For example, the Deep Dual Support Vector Data Description-based Autoencoder (Dual-SVDAE) algorithm was introduced by Zhang et al. [31]. It combines the strengths of deep learning and support vector data description to improve the ability to identify anomalies. The algorithm uses a deep autoencoder to create a latent space, which is then trained using

the support vector data description method to classify samples as belonging to a known or unknown class. Deep Support Vector Data Description achieved state-of-the-art results on several datasets, demonstrating its effectiveness in OSR. Another approach to open-set recognition is to use generative models to create data that closely resembles the training data but is not the same. This method creates a distribution that can be used to detect unknown samples. Variational Autoencoder with Outlier Detection (VAE-OD) is one such method that uses a variational autoencoder to generate data and then employs an outlier detection method to identify unknown samples [32].

### *2.3. Unknown DDoS Detection*

Detecting unknown Distributed Denial-of-Service (DDoS) attacks is a challenging task that various approaches have tackled. Extreme value theory has been utilized in several studies to identify unknown DDoS attacks [33]. However, Gaussian Mixture Model (GMM) and its associated methodologies have been employed to ascertain the distribution of the input [26,34]. In one study, J. Henrydoss et al. used Extreme Vector Machines (EVMs) to find samples whose feature spaces are out of distribution [33]. The study yielded outstanding outcomes on the KDD99 dataset; however, its scope is confined to a particular dataset and lacks scalability to other datasets.

In contrast, Shieh and colleagues employed a deep learning framework for binary classification, utilizing the distribution threshold of GMM and BI-LSTM [34]. The utilization of OOD for the purpose of unknown identification was implemented by the researchers. Specifically, the feature values of BI-LSTM were utilized as the defining characteristics of unknown identities. In contrast, Chapaneri and colleagues employed several GMMs to accurately model individual input features in their study [26]. The GMMs utilized in this research utilized unprocessed data as opposed to the characteristic outputs of deep learning models. The CICIDS2017 dataset was utilized in the experimentation of the two GMM papers. The results indicated that the dataset has the potential to detect unidentified traffic to a certain degree.

In their study, Yang and colleagues employed an autoencoder technique that incorporated a reconstruction error metric, known as AE-D3F, for the purpose of threat detection [35]. The efficacy of the framework was evaluated on three distinct datasets, yielding a detection rate of 82% and a false positive rate of 0%. Despite the absence of unknown samples in the framework, it yielded favorable outcomes in terms of detection. In contemporary times, Generative Adversarial Networks (GANs) have surfaced as a propitious technique for detecting DDoS attacks, as evidenced by sources [36,37]. The IDSGAN framework, as presented by Z. Lin and colleagues, employs a GAN network for the purpose of safeguarding the defending system against malevolent traffic that is aimed at it directly [36]. Chauhan et al. utilized Wasserstein GAN (WGAN) as a solution to the primary training issue encountered in GANs [37]. The authors exhibited that the efficacy of the initial trained model could be nullified by adversarial attacks.

### *2.4. Geometrical Metric*

In the realm of data analysis, geometric metrics have emerged as essential instruments for differentiating dataset distributions. Evaluating the quality of generative models is vital for scientific progress, and a multitude of quantitative metrics have been developed, each with its unique trade-offs. The Frechet Inception Distance (FID), a commonly used measure in image generation activities, was first introduced by Heusel et al. [38] and has shown an amazing correlation to human perception ratings. FID, nevertheless, is unable to adequately capture the full range of dataset characteristics because it can only produce a single value when contrasting two distributions. Precision and recall indicators should be used as measures of fidelity and diversity according to Sajjadi et al. [39]. Despite this, these metrics encounter limitations in real-world applications due to their lack of sensitivity to data fluctuations. To tackle these practical issues, Naeem et al. suggested the use of density and coverage metrics [40]. By adopting a carefully designed manifold estimation proce-

dure, these metrics provide a theoretically sound and empirically dependable approach for assessing fidelity and reliability in diverse situations. When applied to Generative Adversarial Network (GAN)-generated data, a widely used model in recent years, these innovative metrics have proven their effectiveness in distinguishing the distribution of image datasets.

This research introduces an Out-of-Distribution (OOD) framework that utilizes geometric metrics to identify unknown DDoS attacks. The OOD system, as proposed, demonstrates the ability to detect samples that do not align with any established class by integrating insights gleaned from previous research. Implementing these techniques improves the robustness and accuracy of the model in identifying novel attacks.

## 3. Proposed Methodology

We proposed a framework incorporating a CNN architecture to classify the conventional traffic. The OSR obstacle to identifying DDoS attacks is addressed by utilizing a geometrical metrics calculation module and an incremental learning approach, in conjunction with the aforementioned system. The operational illustration of the suggested structure is illustrated in Figure 1.

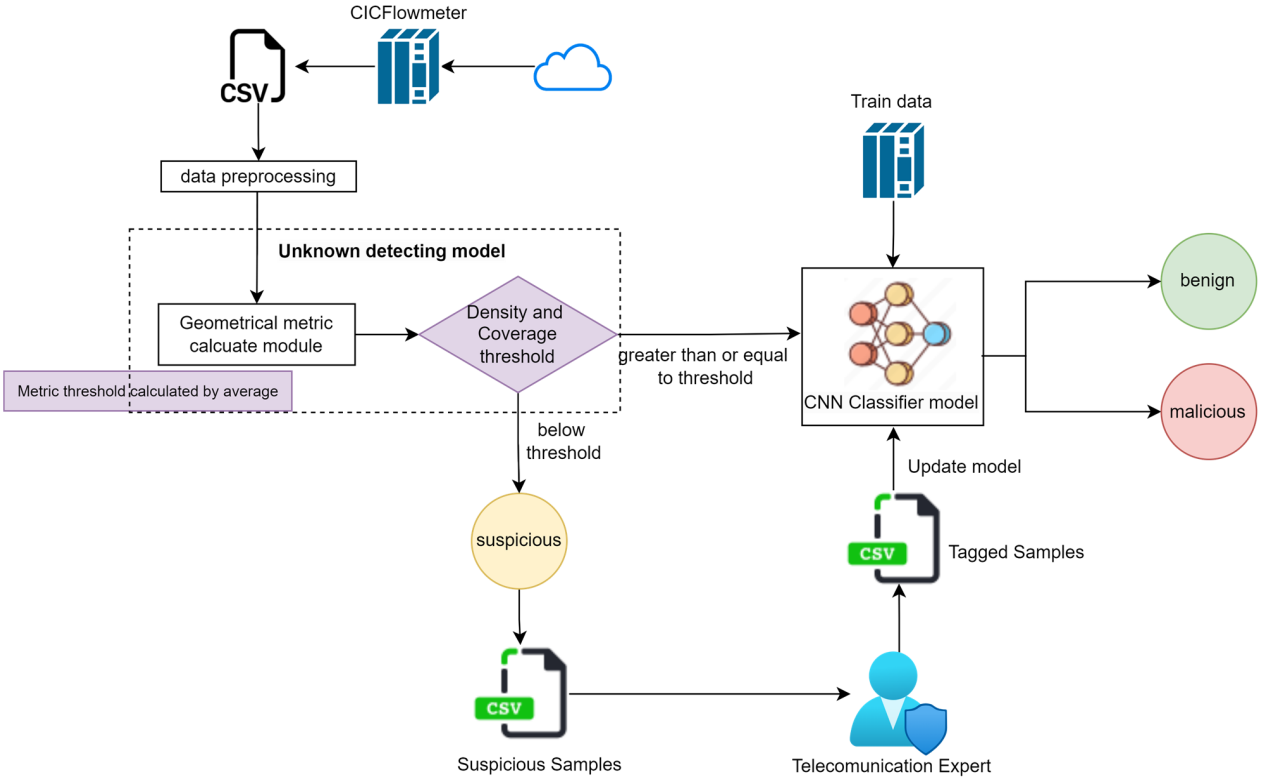


**Figure 1.** Proposed framework architecture.

In order to equip the model with the ability to identify unknown samples, the study adopted the Geometrical Metrics Calculate module approach, which calculates the metric threshold and enables the identification of samples that fall outside the distribution. Once the threshold is defined, the classification process proceeds only for the elements that satisfy the threshold condition, whereas the samples with the calculation results below the threshold are considered as outliers. The current study's utilization of the CNN model has several advantages, including its ability to recognize spatial and temporal patterns in the input data. Additionally, the sparse categorical cross-entropy loss function allows for a simpler optimization process, and the adopted coding approach mitigates issues of linear dependence between labels. Moreover, the approach of the calculating module uses

geometrical metrics aids in identifying data that deviate from the distribution, thereby augmenting the model's capability to detect unknown samples.

### *3.1. CNN Classifier*

The present investigation employed a CNN as the basis of the framework due to its aptitude for identifying patterns in data, particularly in the context of datasets with high dimensionality, as depicted in Figure 2. CNNs can acquire intricate features from unprocessed data, rendering them a suitable option for identifying unfamiliar traffic in Distributed DDoS attacks. The framework inputs a $9 \times 9$ matrix representing a network flow, and the output comprises two prediction levels corresponding to Benign and Attack classifications. The proposed classifier uses a CNN-based architecture with several convolutional layers followed by batch normalization, dropout, and fully connected layers. The number of filters and the filter size is progressively reduced, resulting in decreasing feature maps, which helps the model capture increasingly complex patterns in the network flow data. The batch normalization and dropout layers help to reduce overfitting and improve the convergence of the model during training. The model achieved promising results in accurately identifying different types of conventional DDoS attacks.

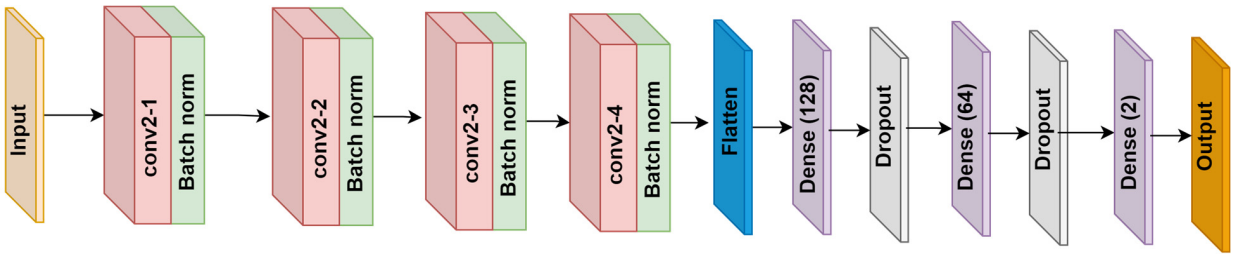


**Figure 2.** CNN architecture in block form.

### *3.2. Density and Coverage*

The ability to accurately assess the similarity between a real distribution $P(X)$ and a generative model $Q(Y)$ is crucial in machine learning applications. To achieve this objective, it is imperative to devise an algorithm that is capable of assessing the probability of the sets of samples $\{X_i\}$ and $\{Y_j\}$ originating from a typical distribution. The density and coverage metrics have been proposed as two metrics that can effectively assess the performance of generative models.

#### 3.2.1. Density Metric

*Density* is a metric that quantifies the degree to which the neighborhoods of real samples overlap with those of unknown samples. Specifically, density counts the number of real-sample neighborhood spheres $\{B(X_i; NND_k(X_i))\}_i$ that contain $Y_j$. Here, $B(x; r)$ represents the sphere in $R^D$ centered around $x$ with a radius of $r$, and $NND_k(X_i)$ denotes the distance from $X_i$ to the $k^{th}$ the nearest neighbor among $\{X_i\}$, excluding itself. The manifold consists of the superimposition of the neighborhood spheres $\{B(X_i; NND_k(X_i))\}_i$, and an expected likelihood of unknown samples is measured. The density metric is defined as formula (1) and illustrated according to Figure 3, where $k$ represents the $k$-nearest neighborhoods. By taking into account the degree to which unknown samples overlap with real samples in densely packed regions, the density metric is less vulnerable to the effects of outliers.

$$Density := \frac{1}{kM} \sum_{j=1}^{M} \sum_{i=1}^{N} 1_{Y_j \in B(X_i, NND_k(X_i))} \tag{1}$$

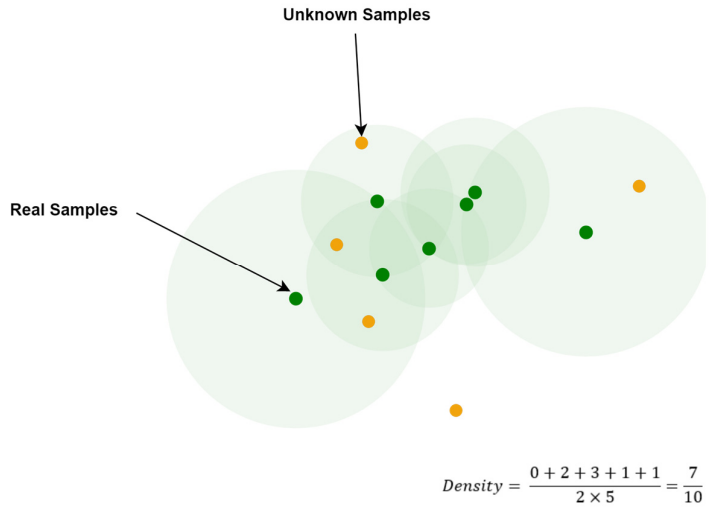


**Figure 3.** Illustration of density metric with $k = 2$.

The process of calculating density will be executed following Algorithm 1.

**Algorithm 1** Calculation of Density

**Input**: $D_R$: Dataset of real samples, $D_U$: Dataset of unknown samples, $N_R$: Number of real samples, $N_U$: Number of unknown samples, $k$ : Number of nearest neighbors to use for density calculation, $Count$: Array of number neighbourhood spheres of real sample that contain each unknown sample.
**Output**: density value

Real sample $\leftarrow r \in D_R$
Unknown sample $\leftarrow u \in D_U$
Define distance between the unknown sample $m$ and real sample $n$: $d_{mn} = distance\ (u_m, r_n)$
Define $k^{th}$ nearest neighbour distances for real sample $n$: $NND_n = nearest\_neighbour\_distances(r_n, k)$
$Count$ = array[$N_U$] of zeros
**for** $i$ in range $N_U$:
  **for** $j$ in range $N_R$:
    **if** $d_{ij} < NND_j$
      $Count[i] = Count[i] + 1$
  **end for**
**end for**
$Density = mean\ (Count)$
**return** $Density$

Figure 4 provides a detailed flowchart of Algorithm 1, illustrating the key stages and components involved.

### 3.2.2. Coverage Metric

*Coverage*, on the other hand, is a metric that aims to quantify diversity by measuring the extent to which unknown samples cover the real samples. In other words, coverage measures the ratio of real samples that are covered by unknown samples. To improve the accuracy of coverage, the nearest neighbor manifolds are built around the real samples instead of the unknown ones, as the former are less prone to outliers. Moreover, the manifold can only be computed per dataset instead of per model, reducing the heavy nearest neighbor computations in a recall. The coverage metric is defined as formula (2) [40],

illustrated by Figure 5, and represents the fraction of real samples whose neighborhoods contain at least one unknown sample. The coverage metric ranges from 0 to 1.

$$Coverage := \frac{1}{N}\sum_{i=1}^{N} 1_{\exists j: Y_j \in B(X_i, NND_k(X_i))} \tag{2}$$

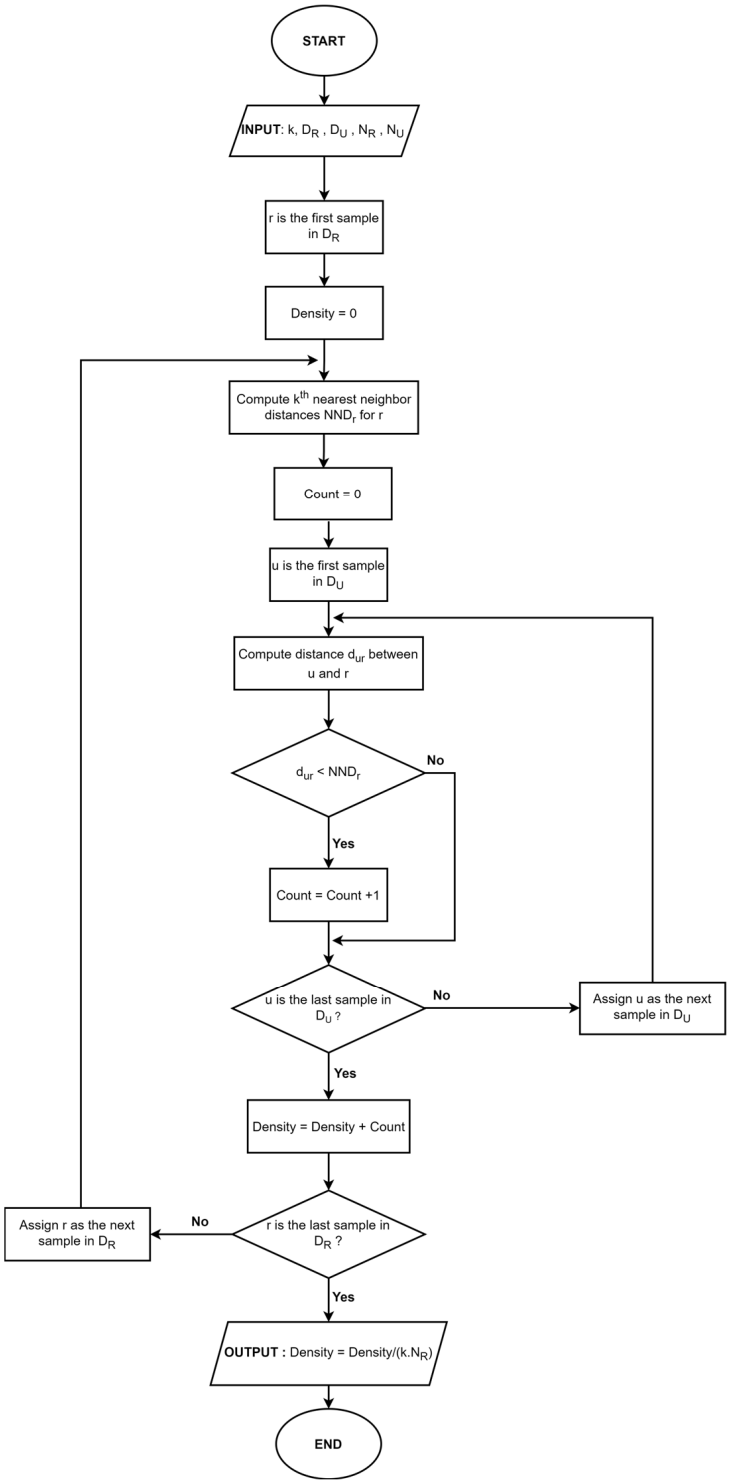


**Figure 4.** Detailed flowchart of Algorithm 1.

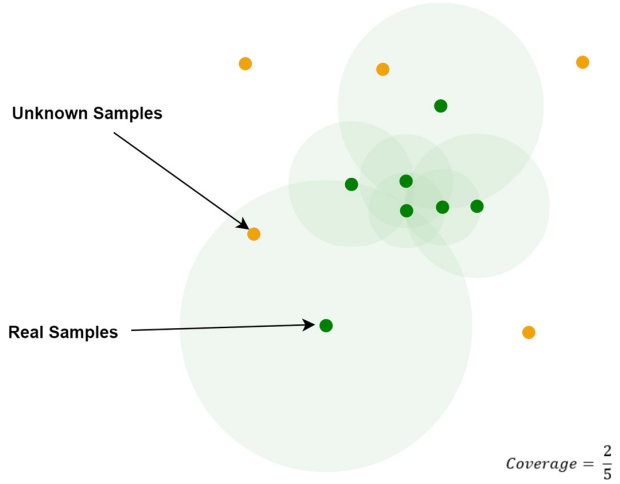


**Figure 5.** Illustration of Coverage metric with $k = 2$.

The process of calculating coverage will be executed following Algorithm 2.

**Algorithm 2** Calculation of Coverage

**Input**: $D_R$: Dataset of real samples, $D_U$: Dataset of unknown samples, $N_R$: Number of real samples, $N_U$: Number of unknown samples, $k$: Number of nearest neighbors to use for coverage calculation, $Count$: Array of number neighbourhood spheres of real sample that contain at least one unknown sample.
**Output**: coverage value

Real sample $\leftarrow r \in D_R$
Unknown sample $\leftarrow u \in D_U$
Define distance between the unknown sample $m$ and real sample $n$: $d_{mn} = distance\ (u_m, r_n)$
Define $k^{th}$ nearest neighbour distances for real sample $n$: $NND_n = nearest\_neighbour\_distances(r_n, k)$
$Count$ = array[$N_R$] of zeros
**for** $i$ in range $N_R$:
  **for** $j$ in range $N_U$:
    **if** $d_{ji} < NND_i$
      $Count[i] = 1$
      **break**
  **end for**
**end for**
$Coverage = mean\ (Count)$
**return** $Coverage$

Figure 6 provides a detailed flowchart of Algorithm 2, illustrating the key stages and components involved.

### 3.2.3. Density and Coverage Behavior Analysis

To verify the effectiveness of the density and coverage metrics, it is necessary to examine whether they reach their best values when the intended criteria are met. Analyzing the expected values $E[Density]$ and $E[Coverage]$ for identical real and unknown distributions reveals that these metrics approach 100% as the sample sizes $(N; M)$ and the number of neighborhoods $k$ increase. This analysis further leads to a systematic algorithm for selecting the hyperparameters $(k; N; M)$ for generative models. Specifically, the algorithm can be used to determine the optimal values of $k$, $N$, and $M$ that will maximize the effectiveness of the density and coverage metrics in assessing the similarity between the real and unknown

distributions. We derive the expected values of density and coverage under the identical real and unknown data in formulas (3) and (4) [40]:

$$E[Density] = 1 \tag{3}$$

$$E[Coverage] = 1 - \frac{(N-1)\cdots(N-k)}{(M+N-1)\cdots(M+N-k)} \tag{4}$$

As $M = N \to \infty : E[Coverage] = 1 - \frac{1}{2^k}$.

By taking into account the degree to which unknown samples overlap with real samples in densely packed regions and measuring the extent to which unknown samples cover the real samples, these metrics offer a comprehensive evaluation of the dataset's distribution. Additionally, by analyzing the expected values of density and coverage for identical real and unknown distributions, it is possible to develop a systematic algorithm for selecting the model's hyperparameters, thereby optimizing their performance.

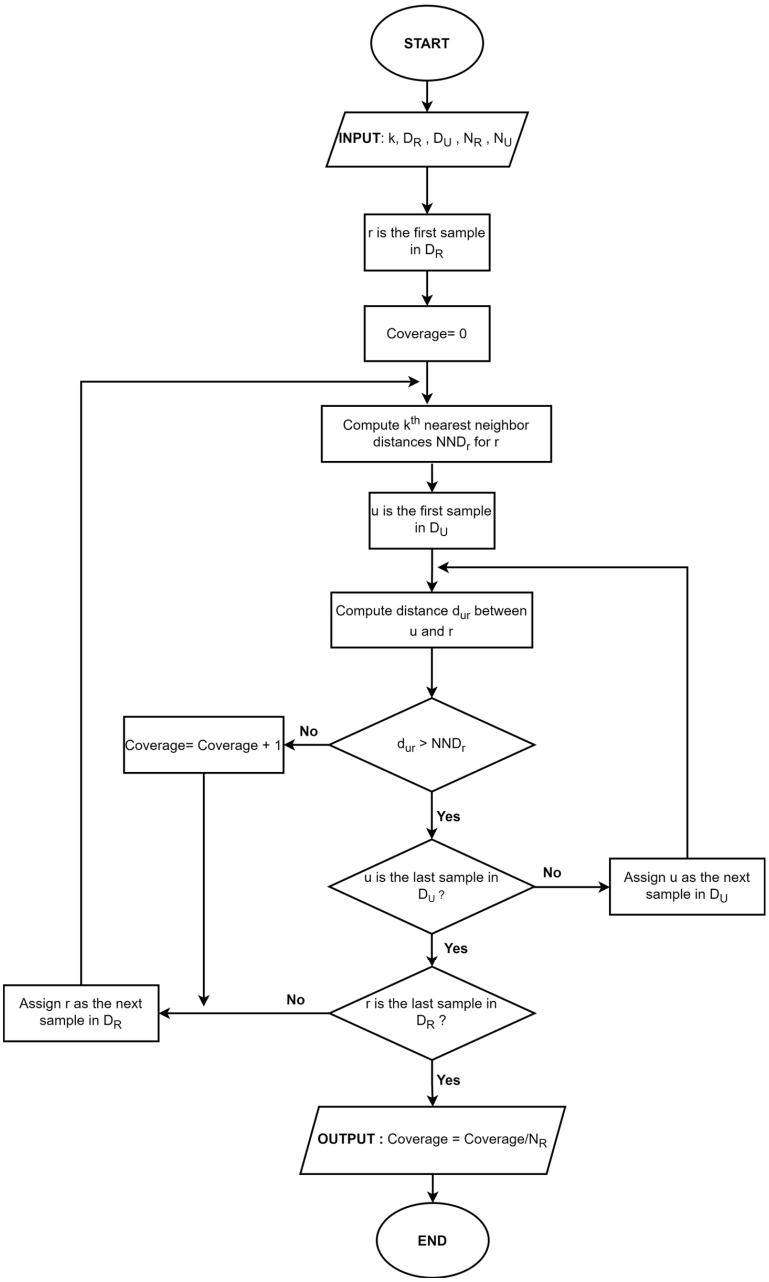


**Figure 6.** Detailed flowchart of Algorithm 2.

### *3.3. Unknown Identification Module*

The proposed Unknown Detecting Module is designed to address the challenge of identifying unknown attacks in the cybersecurity domain. The need for such a module arises due to the ever-evolving threat landscape and the difficulty in identifying and isolating unknown attacks. The proposed module is designed to work with the CICIDS2017-Wednesday dataset, which is a widely used benchmark dataset for network intrusion detection systems. To accomplish its goal, we divide the original data of the CICIDS2017-Wednesday dataset into batch samples with an element number of 10,000 and according to the ratio of the original labels in the dataset. By dividing the dataset into batches, we can assess the similarity between the batches and the baseline dataset without processing the entire dataset simultaneously. This improves the speed and efficiency of the module.

Subsequently, the density and coverage metrics are computed to determine the correlation between the data batches. The density and coverage metrics have been proposed as two metrics that can effectively assess the performance of generative models. Density quantifies the degree to which the neighborhoods of real samples overlap with those of unknown samples. In contrast, coverage aims to quantify diversity by measuring the extent to which unknown samples cover the real samples. We can assess the similarity between the batches and the baseline dataset by calculating the density and coverage metrics. This step allows us to determine the similarity of each batch to the baseline dataset, which is crucial for identifying unknown attacks. To evaluate outliers, we build a threshold for evaluating outliers from the average of all metric density and coverage correlation values as formula (5) and (6) [40].

$$D_{threshold} := \frac{2}{(N-1)N} \sum_{i=1}^{N} \sum_{j \neq i} Density\left(batch_i, batch_j\right) \tag{5}$$

$$C_{threshold} := \frac{2}{(N-1)N} \sum_{i=1}^{N} \sum_{j \neq i} Coverage\left(batch_i, batch_j\right) \tag{6}$$

where $N$ is the number of batchs.

The threshold is an important component of the proposed module, allowing us to distinguish between known and unknown attacks. When the metric density and coverage results of the data correlated with the baseline dataset fall below the threshold level, they will be considered outliers. This step enables us to identify unknown attacks that are not present in the baseline dataset and isolate them from the network. By combining the density and coverage metrics, we can effectively identify and isolate unknown attacks from the network. The proposed module can potentially serve as a valuable instrument in augmenting the cybersecurity of networks. The efficacy of the module lies in its ability to detect outliers. Furthermore, the module that has been suggested exhibits scalability, as it can be modified to function with additional datasets. The present study employs a double-index approach for categorization in the unidentified identification module. The schematic representation of the strategy architecture is depicted in Figure 7.

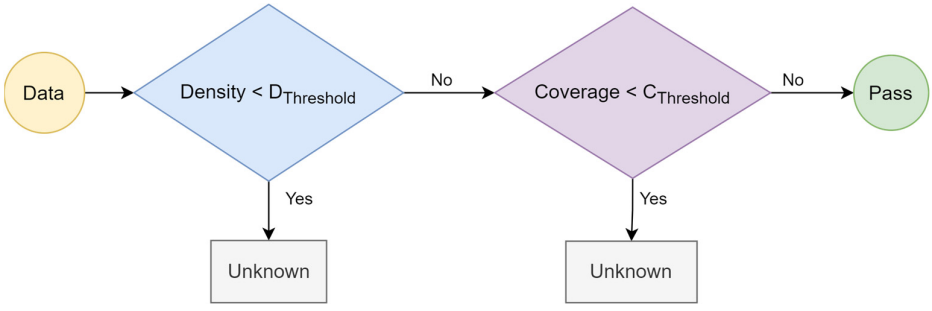


**Figure 7.** Unknown detecting strategy by using Geometrical metric threshold.

### *3.4. Incremental Learning*

The model will feature an identification module capable of detecting unknown samples. In the event of the detection of unidentified traffic, communication experts are alerted to label the data for subsequent model retraining. To this end, the proposed framework employs a fine-tuned strategy to update specific modules within the architecture of the model, thereby allowing for the acquisition of new knowledge by including additional classifications. Additionally, the model's learning rate during training is moderated to mitigate the risk of catastrophic forgetting of previously learned information.

## 4. Experiment

### *4.1. Dataset*

In this article, the performance of the suggested framework is thoroughly evaluated by two prominent network datasets: CICIDS2017 and CICDDoS2019. The CICIDS2017 dataset comprises network traffic logs spanning over five days, which capture various types of Denial-of-Service (DoS) and Distributed DoS (DDoS) attacks that occurred on 7 May 2017 and 7 July 2017. On the other hand, CICDDoS2019 is a widely used dataset that contains network traffic data of amplification attacks. Both datasets are characterized by a set of features and corresponding labels, where the label information indicates the presence of either benign network traffic or malicious. Specifically, the attack signatures in the dataset provide comprehensive information about the various types of network attacks, such as HTTP flood, TCP SYN flood, and UDP flood, among others. Table 2 in this study summarizes the primary attack vectors of the datasets above.

**Table 2.** The statistical examination of datasets.

| Dataset | Label | Quantity | Proportion | Total |
|---|---|---|---|---|
| CICIDS2017 Wed <train dataset> | BENIGN | 319,186 | 64.260% | 496,709 |
| | DoS Hulk | 159,049 | 32.021% | |
| | DoS GoldenEye | 7647 | 1.540% | |
| | DoS Slowloris | 5707 | 1.149% | |
| | DoS Slowhttptest | 5109 | 1.029% | |
| | HeartBleed | 11 | 0.002% | |
| CICIDS2017 Tuesday | BENIGN | 432,074 | 96.897% | 445,909 |
| | FTP-Patator | 7938 | 1.780% | |
| | SSH-Patator | 5897 | 1.323% | |
| CICDDoS2019 LDAP | BENIGN | 1602 | 0.073% | 2,181,530 |
| | DrDoS_LDAP | 2,179,928 | 99.927% | |
| CICDDoS2019 MSSQL | BENIGN | 1995 | 0.044% | 4,524,484 |
| | DrDoS_MSSQL | 4,522,489 | 99.956% | |
| CICDDoS2019 DNS | BENIGN | 3380 | 0.067% | 5,074,382 |
| | DrDoS_DNS | 5,071,002 | 99.933% | |
| CICDDoS2019 NetBIOS | BENIGN | 1705 | 0.042% | 4,094,978 |
| | DrDoS_NetBIOS | 4,093,273 | 99.958% | |
| CICDDoS2019 NTP | BENIGN | 14,337 | 1.178% | 1,216,976 |
| | DrDoS_NTP | 1,202,639 | 98.822% | |
| CICDDoS2019 UDP | BENIGN | 2151 | 0.069% | 3,136,794 |
| | DrDoS_UDP | 3,134,643 | 99.931% | |

**Table 2.** *Cont.*

| Dataset | Label | Quantity | Proportion | Total |
|---|---|---|---|---|
| CICDDoS2019 SNMP | BENIGN | 1502 | 0.029% | 5,161,365 |
| | DrDoS_SNMP | 5,159,863 | 99.971% | |
| CICDDoS2019 SSDP | BENIGN | 762 | 0.029% | 2,611,372 |
| | DrDoS_SSDP | 2,610,610 | 99.971% | |
| CICDDoS2019 SYN | BENIGN | 389 | 0.028% | 1,380,404 |
| | Syn | 1,380,015 | 99.972% | |

The proposed model will undergo training using the CICIDS2017 Wednesday dataset, which includes benign traffic and DoS attacks. This approach aims to enhance the model's capability to detect benign traffic and DoS attacks. Meanwhile, the CICIDS2017 Tuesday and CICDDoS2019 datasets were utilized as unseen traffic to evaluate the model's performance.

Evaluation metrics were gathered using the confusion matrix, as indicated in Table 3. The confusion matrix's parameters include True Positive (TP), which represents malicious traffic correctly identified, and True Negative (TN), which represents benign traffic correctly identified. False Positive (FP) represents benign traffic identified as malicious traffic, and False Negative (FN) represents malicious traffic mistakenly identified as benign traffic. This evaluation methodology is an essential aspect of the experiment and aims to accurately measure the model's effectiveness in distinguishing between benign and malicious traffic.

**Table 3.** Confusion Matrix.

| Actual \ Predict | Attack | Benign |
|---|---|---|
| Attack | TP | FP |
| Benign | FN | TN |

The evaluation of the proposed model was performed using the confusion matrix shown in Table 2, along with the commonly used performance metrics, namely accuracy, precision, recall, and F1 score, as defined in formulas (7), (8), (9), and (10), respectively. The precision metric assesses the proportion of true positive identifications out of all positive identifications, while recall refers to the ratio of correctly identified actual positives. The metric of accuracy evaluates the proportion of accurately classified instances, whereas the F1 score metric offers a balance between precision and recall.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{7}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{8}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{9}$$

$$\text{F}_1\text{Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{10}$$

### *4.2. Framework*

Following a thorough study process, a CNN architecture has been identified, featuring the configuration illustrated in Figure 8 and the parameter configurations outlined in Table 4. The experiment was carried out utilizing a workstation equipped with an Ubuntu

20.04 operating system, an AMD Ryzen 5700X 8C16T processor, and 96 GB DDR4 memory. Additionally, Nvidia RTX3070 devices were utilized for computing acceleration purposes. The driver component employs the NVIDIA Driver Server 510 version.

```
Model: "Classifier Model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 Input (InputLayer)          [(None, 9, 9, 1)]         0

 Conv2D-1 (Conv2D)           (None, 9, 9, 100)         500

 BatchNorm-1 (BatchNormaliza  (None, 9, 9, 100)        400
 tion)

 Conv2D-2 (Conv2D)           (None, 9, 9, 80)          32080

 BatchNorm-2 (BatchNormaliza  (None, 9, 9, 80)         320
 tion)

 Conv2D-3 (Conv2D)           (None, 9, 9, 60)          43260

 BatchNorm-3 (BatchNormaliza  (None, 9, 9, 60)         240
 tion)

 Conv2D-4 (Conv2D)           (None, 9, 9, 40)          21640

 BatchNorm-4 (BatchNormaliza  (None, 9, 9, 40)         160
 tion)

 Flatten (Flatten)           (None, 3240)              0

 Dense-1 (Dense)             (None, 128)               414848

 Dropout-1 (Dropout)         (None, 128)               0

 Dense-2 (Dense)             (None, 64)                8256

 Dropout-2 (Dropout)         (None, 64)                0

 Output (Dense)              (None, 2)                 130

=================================================================
Total params: 521,834
Trainable params: 521,274
Non-trainable params: 560
_________________________________________________________________
```

**Figure 8.** CNN classifier model architecture.

**Table 4.** Parameters configuration.

| Parameter | Value |
|---|---|
| Optimizer | Adam |
| Weight_decay | $3 \times 10^{-5}$ |
| Learning rate | 0.001 |
| Number of nearest neighbor | 5 |
| Random seed | 0; 42; 133; 207; 417; 830; 920; 1377; 65,536; 815; 123 |
| Training split ratio (train; test) | 0.8; 0.2 |
| Batch size | 512 |

For the numerical implementation, we used the Python programming language, version 3.9.12. The programming environment utilized in this study consisted of VSCode and Conda. The model framework relied on Tensorflow 2.12, a popular open-source machine learning library, which provided the necessary tools for building and training the CNN architecture. Additionally, we used the Scikit-learn (sklearn) library, a widely used Python library for machine learning and data science, to assist in data preprocessing, model evaluation, and other related tasks. To handle numerical computations efficiently,

we incorporated the NumPy library, a fundamental Python scientific computing package, which facilitated operations with multi-dimensional arrays and matrices. These tools and libraries enabled us to effectively implement and analyze the proposed CNN-Geo method in our study.

To ensure the robustness of the proposed CNN model, we conducted ten separate training runs with different random seeds and averaged the results. The model's performance was evaluated on a closed dataset, and the results presented in Table 5 demonstrate its effectiveness.

**Table 5.** Training outcomes for CICIDS2017 Wed.

| Dataset | Accuracy | Precision | Recall | $F_1$Score |
|---|---|---|---|---|
| CICIDS2017 Wed | 0.9983 | 0.9966 | 0.9954 | 0.9960 |

### *4.3. Unknown Attack Recognition and Evaluation*

#### 4.3.1. Identify Unknown Attack by CNN Classifier

Upon completing the CICIDS2017 Wednesday dataset training, the CNN exhibited commendable efficiency in countering conventional attacks. An initial assessment was performed on the CICIDS2017 Tuesday dataset to determine its efficacy in safeguarding against unknown attacks. Table 6 displays the outcomes and correlation analysis in relation to the initial dataset.

**Table 6.** Identifying unknown attack outcomes with CICIDS2017 Tuesday.

| Dataset | Accuracy | Precision | Recall | $F_1$Score |
|---|---|---|---|---|
| CICIDS2017 Wed | 0.9983 | 0.9966 | 0.9954 | 0.9960 |
| CICIDS2017 Tuesday | 0.9626 | 0.6737 | 0.6370 | 0.6528 |

The experimental findings reveal that the model maintains its accuracy in defending against unknown traffic, as evidenced by the consistent score of 0.9626 on the CICIDS2017 Tuesday dataset. Notably, the precision score plummets to 0.6737, indicating that the model's ability to detect novel kinds of attacks is inadequate. Additionally, comparable declines are observed in recall and F1 scores.

Delving further into this issue, it becomes clear that the discrepancy between the Accuracy metric and the other indices on the classification of the Tuesday dataset primarily stems from the imbalance and nature of the dataset. As indicated in Table 1, the BENIGN sample in CICIDS2017-Tuesday constitutes 96.897% of the data. The Confusion Matrix of the CICIDS2017-Tuesday data classification results, as depicted in Figure 9, reveals that the notably high True Negative (TN) index is responsible for the elevated Accuracy metric outcomes. However, the low True Positive (TP) rate significantly decreases precision, recall, and F1 scores.

Given that the CICIDS2017-Wednesday and CICIDS2017-Tuesday datasets were collected from the same network environment, the BENIGN samples from both sets display similarities, enabling the model to detect benign samples from the CICIDS2017-Tuesday dataset effectively. Nonetheless, the model struggles to identify new attack patterns from the CICIDS2017-Tuesday dataset, highlighting its limitations in addressing unknown attacks. The results of additional experiments conducted on OSR datasets associated with CICDDoS2019 are presented in Table 7.

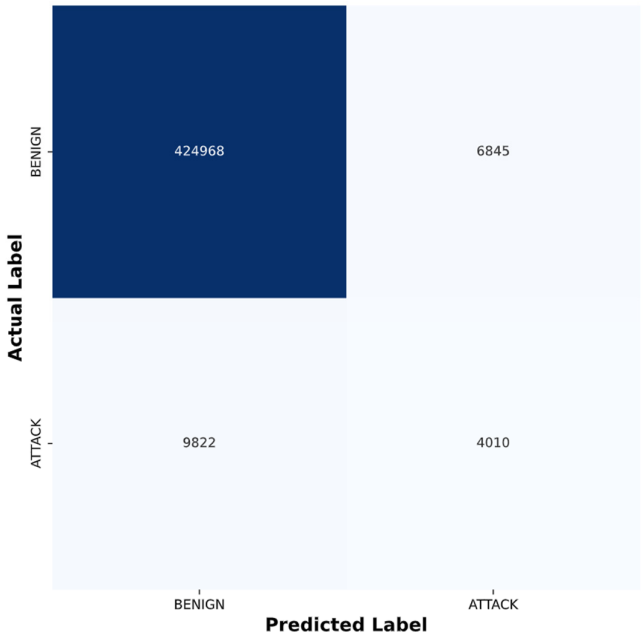


**Figure 9.** Confusion Matrix of Classification on CICIDS2017-Tuesday.

**Table 7.** Model's detection outcomes on each set.

| Dataset | Accuracy | Precision | Recall | $F_1$Score |
|---|---|---|---|---|
| CICIDS2017 Wed | 0.9983 | 0.9966 | 0.9954 | 0.9960 |
| CICIDS2017 Tuesday | 0.9626 | 0.6737 | 0.6370 | 0.6528 |
| CICDDoS 2019 LDAP | 0.0020 | 0.0003 | 0.1378 | 0.0007 |
| CICDDoS 2019 MSSQL | 0.0004 | 0.0239 | 0.1434 | 0.0001 |
| CICDDoS 2019 NetBIOS | 0.0003 | 0.0001 | 0.1446 | 0.0001 |
| CICDDoS 2019 Portmap | 0.0227 | 0.0161 | 0.1464 | 0.0074 |
| CICDDoS 2019 SYN | 0.0074 | 0.0070 | 0.1400 | 0.0024 |
| CICDDoS 2019 UDP | 0.0007 | 0.0001 | 0.1455 | 0.0002 |
| CICDDoS 2019 DNS | 0.0006 | 0.2001 | 0.1815 | 0.0002 |
| CICDDoS 2019 NTP | 0.0116 | 0.2023 | 0.1974 | 0.0046 |
| CICDDoS 2019 SNMP | 0.0003 | 0.0001 | 0.1839 | 0.0001 |
| CICDDoS 2019 SSDP | 0.0003 | 0.0001 | 0.1878 | 0.0001 |

The accuracy is very low for the datasets (except for the first two) presented in Table 6 because the model struggles to identify new and unknown attack patterns that were not present in the training data. When evaluating traffic from a disparate dataset such as CICDDoS2019, the model's performance indicators experienced a substantial reduction. This is because the attack patterns in the CICDDoS2019 dataset are different and unknown to the model, as it was not exposed to these patterns during training. The model's limitations in addressing unknown attacks become evident when faced with the challenge of identifying traffic originating from a different dataset. To improve the overall defense capabilities of the structure, it is imperative to screen the unknown identity module in the second stage.

#### 4.3.2. Unknown Identification Index

Outlier Detection Rate (ODR), which is defined by the formula (11), is used as the evaluation metric for determining the performance of the unknown detection component. This metric allows the assessment of the module's ability to identify outliers among incom-

ing data samples. By utilizing this metric, the performance of the unknown recognition module can be accurately quantified, and any areas of improvement can be identified for further optimization.

$$ODR = \frac{N_{Outlier}}{N} \tag{11}$$

where $N_{Outlier}$ is the number of observed samples that fall short of the threshold following analysis by the framework, and $N$ is the total amount of samples in the procedure.

#### 4.3.3. Outcome of Unknown Attack Detection

The model's efficiency to confront unknown attacks is reflected in Table 8, which presents the ODR metrics that demonstrate the model's ability to detect unseen threats.

**Table 8.** Outcome of unknown attack detection.

| Dataset | *ODR* |
|---|---|
| CICIDS2017 Tuesday | 0.7461 |
| CICDDoS 2019 LDAP | 0.9902 |
| CICDDoS 2019 MSSQL | 0.9871 |
| CICDDoS 2019 NetBIOS | 0.9867 |
| CICDDoS 2019 Portmap | 0.9944 |
| CICDDoS 2019 SYN | 0.9888 |
| CICDDoS 2019 UDP | 0.9861 |
| CICDDoS 2019 DNS | 0.9895 |
| CICDDoS 2019 NTP | 0.9892 |
| CICDDoS 2019 SNMP | 0.9897 |
| CICDDoS 2019 SSDP | 0.9870 |

The CICIDS2017-Tuesday dataset's traffic was captured within the same network environment and timeframe as the training data, resulting in an ODR of 0.7461 that implies some degree of similarity between the two datasets, yet the model still exhibits satisfactory performance. In relation to the model's efficacy in countering CICDDoS 2019 attacks, it is noteworthy that the ODR score surpassed 0.98, with LDAP exhibiting the highest ODR of 0.99. The findings indicate that the model can proficiently identify a significant portion of unidentified traffic, particularly in cases where the data display minimal correlation, using the unknown identifies module.

#### 4.3.4. Incremental Learning and the Outcomes Following

After being detected by an unknown identification component, the unidentified traffic is forwarded to telecommunications technician for analysis and labeling. It is then transmitted to a progressive learning module for further improvement. The fine-tuning process exclusively employs novel data, refraining from utilizing the initial training dataset. Despite causing a minor decline in performance, this method retains a satisfactory degree of competence for the preceding task and aligns more appropriately with real-world online operational scenarios. With respect to the performance of incremental learning, a sorted list is presented in Table 9. To enable a comprehensive assessment of incremental learning's efficacy, the performance metrics of the model before incremental learning, as presented in Table 7, are incorporated under the tag "before incremental learning". The "post incremental learning" entry within the table indicates that the evaluation also incorporates the pre-training dataset, employed alongside CICIDS2017 Tuesday, to substantiate that previously acquired knowledge is not excessively compromised.

**Table 9.** CNN-Geo's detection outcomes post incremental learning.

| Dataset | Test | Accuracy | Precision | Recall | F1 Score |
|---|---|---|---|---|---|
| CICIDS2017 Wed | Before incremental learning | 0.9983 | 0.9966 | 0.9954 | 0.9960 |
| | Post incremental learning | 0.9979 | 0.9962 | 0.9944 | 0.9953 |
| CICIDS2017 Tuesday | Before incremental learning | 0.9626 | 0.6737 | 0.6370 | 0.6528 |
| | Post incremental learning | 0.9996 | 0.9989 | 0.9944 | 0.9966 |
| CICDDoS 2019 LDAP | Before incremental learning | 0.0020 | 0.0003 | 0.1378 | 0.0007 |
| | Post incremental learning | 0.9999 | 0.9998 | 0.9995 | 0.9996 |
| CICDDoS 2019 MSSQL | Before incremental learning | 0.0004 | 0.0239 | 0.1434 | 0.0001 |
| | Post incremental learning | 0.9973 | 0.9952 | 0.9984 | 0.9968 |
| CICDDoS 2019 NetBIOS | Before incremental learning | 0.0003 | 0.0001 | 0.1446 | 0.0001 |
| | Post incremental learning | 0.9997 | 0.9981 | 0.9992 | 0.9987 |
| CICDDoS 2019 Portmap | Before incremental learning | 0.0227 | 0.0161 | 0.1464 | 0.0074 |
| | Post incremental learning | 0.9998 | 0.9976 | 0.9993 | 0.9985 |
| CICDDoS 2019 SYN | Before incremental learning | 0.0074 | 0.0070 | 0.1400 | 0.0024 |
| | Post incremental learning | 0.9957 | 0.9986 | 0.9999 | 0.9993 |
| CICDDoS 2019 UDP | Before incremental learning | 0.0007 | 0.0001 | 0.1455 | 0.0002 |
| | Post incremental learning | 0.9992 | 0.9842 | 0.9813 | 0.9836 |
| CICDDoS 2019 DNS | Before incremental learning | 0.0006 | 0.2001 | 0.1815 | 0.0002 |
| | Post incremental learning | 0.9990 | 0.9989 | 0.9963 | 0.9976 |
| CICDDoS 2019 NTP | Before incremental learning | 0.0116 | 0.2023 | 0.1974 | 0.0046 |
| | Post incremental learning | 0.9989 | 0.9988 | 0.9989 | 0.9989 |
| CICDDoS 2019 SNMP | Before incremental learning | 0.0003 | 0.0001 | 0.1839 | 0.0001 |
| | Post incremental learning | 0.9997 | 0.9987 | 0.9973 | 0.99800. |
| CICDDoS 2019 SSDP | Before incremental learning | 0.0003 | 0.0001 | 0.1878 | 0.0001 |
| | Post incremental learning | 0.9994 | 0.9974 | 0.9944 | 0.9953 |

Table 9 illustrates that integrating the suggested system adequately resolves the problem of OSR in detecting unfamiliar attacks. By leveraging the proficiency of telecommunication technicians, recently classified instances are reintegrated into the proposed model to facilitate incremental learning. The enhancement in performance for CICDDoS2019/LDAP and CICDDoS2019/PORTMAP is significantly evident. Moreover, the implementation of the recommended CNN-Geo framework in conjunction with the incremental learning approach ensures that all performance metrics revert to satisfactory levels. Consequently, the refined model can competently and elegantly handle established and emerging traffic patterns.

### *4.4. Comparative Analysis of the Proposed Method and Existing Approachs*

In the next stage of the analysis, we conduct a comprehensive comparison between the CNN-Geo and traditional ML algorithms. Many recent and related studies have suggested using conventional ML algorithms or a combination of innovative and combined methods to detect DDoS attacks. CNN-Geo was compared with the results of three MC algorithms found in the literature: Decision Tree [41], Random Forest [13], SVM [23]. To provide a more thorough assessment of our proposed method, we present an overall comparison in Table 10, highlighting the main performance differences between our method and the ML algorithms used in the aforementioned studies on dataset CICIDS2017 where the superior outcomes will be distinctly emphasized by displaying them in bold font.

**Table 10.** CNN-Geo's result in comparison with the traditional ML algorithms on CICIDS2017.

| Method | Accuracy | Precision | Recall |
|---|---|---|---|
| Decision Tree (2020) [41] | 0.0194 | 0.9938 | 0.0163 |
| Random Forest (2019) [13] | 0.0032 | 0.9967 | 0.00004 |
| SVM (2019) [23] | 0.0147 | 0.9621 | 0.0120 |
| CNN-Geo | **0.9979** | **0.9962** | **0.9944** |

Upon examining the results presented in Table 10, it is evident that the CNN-Geo outperforms traditional ML algorithms in terms of accuracy, precision, and recall. Specifically, the CNN-Geo achieves an accuracy of 0.9979, a precision of 0.9962, and a recall of 0.9944. These values are significantly higher than those of the other algorithms, demonstrating the superior performance of the CNN-Geo method for detecting DDoS attacks. In contrast, the Decision Tree, Random Forest, and SVM methods exhibit lower performance levels in comparison to CNN-Geo. The Decision Tree algorithm shows a relatively high precision of 0.9938 but suffers from low accuracy (0.0194) and recall (0.0163). The Random Forest algorithm, despite having a high precision of 0.9967, demonstrates the weakest performance in terms of accuracy (0.0032) and recall (0.00004). Lastly, the SVM method reports a precision of 0.9621, an accuracy of 0.0147, and a recall of 0.0120, indicating that it also struggles with detecting DDoS attacks effectively.

In order to further demonstrate the efficacy of the CNN-Geo method in handling not only conventional DDoS attacks but also effectively addressing out-of-sample or unknown attacks, we have conducted a comparative analysis of the performance of CNN-Geo against state-of-the-art approaches, including the Gaussian Mixture Model (GMM) [26], GMM-Bidirectional Long Short-Term Memory (GMM-BiLSTM) [34], Density-Based Spatial Clustering of Applications with Noise-Random Forest (DBSCAN-RF) [28], Density-Based Spatial Clustering of Applications with Noise-Support Vector Machine (DBSCAN-SVM) [28], and One-Dimensional Deep High-Resolution Network-One-Class Support Vector Machine (1D-DHRNet-OCSVM) [27]. These comparison models are all trained on the original CICIDS2017 dataset and subsequently tested on a distinct dataset that differs from the original training set. The comprehensive comparison of averaged results is presented in Table 11 where the superior outcomes will be distinctly emphasized by displaying them in bold font.

**Table 11.** CNN-Geo's result in comparison with the existing DL algorithms on unknown DDoS attack detection.

| Method | Accuracy | Precision | Recall |
|---|---|---|---|
| GMM (2021) [26] | - | 0.970 | 0.950 |
| GMM-BiLSTM (2021) [34] | 0.952 | 0.994 | 0.957 |
| DBSCAN-RF (2022) [28] | 0.148 | 0.998 | 0.145 |
| DBSCAN-SVM (2022) [28] | 0.314 | 0.998 | 0.312 |
| 1D-DHRNet-OCSVM (2022) [27] | 0.992 | **0.999** | 0.991 |
| **CNN-Geo** | **0.996** | 0.997 | **0.996** |

After conducting a thorough examination of the results presented in Table 11, it becomes apparent that the CNN-Geo method demonstrates a well-balanced performance in detecting unknown DDoS attacks when compared to existing state-of-the-art approaches. While the 1D-DHRNet-OCSVM [27] method achieves the highest precision of 0.999, its accuracy and recall values are slightly lower than those of the CNN-Geo method. Specifically, the CNN-Geo achieves an accuracy of 0.996, a precision of 0.997, and a recall of 0.996, surpassing the overall performance of GMM [26], GMM-BiLSTM [34], DBSCAN-RF [28], and DBSCAN-SVM [28]. This comparative analysis highlights the robustness and adaptability of the CNN-Geo approach in handling not only known but also out-of-sample or unknown DDoS attacks. It is important to note that while some of the other approaches may excel in certain performance metrics, the CNN-Geo method provides a more balanced and consistent performance across all evaluation criteria. By effectively addressing these emerging threats, our proposed method offers a significant contribution to enhancing the overall security and resilience of computer networks in the face of evolving DDoS attack scenarios.

## 5. Conclusions

Existing studies primarily focus on general categories, resulting in intrusion detection systems' limitations when detecting unknown attacks. This study presents the novel CNN-Geo framework, a hybrid network architecture combining unsupervised and supervised networks' features to address these challenges. Utilizing datasets such as CICIDS2017-Wed and CICIDDoS2019, the framework effectively detects unknown cyber-attacks by employing DL techniques and geometrical metric calculating during training alongside the incremental learning solution. Our comprehensive comparison of CNN-Geo with traditional ML algorithms and state-of-the-art approaches demonstrates its superior performance in detecting conventional and unknown DDoS attacks. The experimental results validate the proposed architecture's effectiveness, achieving a detection rate of more than 99% for conventional attacks in the CICIDS2017-Wed dataset and enhancing the framework's efficiency to 99.8% in confronting unknown attacks in the recent CICIDDoS2019 unseen datasets. CNN-Geo demonstrates the adaptability to address evolving threats by leveraging telecommunications technicians for traffic defining and incrementally learning. The verified benefits of this research lie in the enhanced detection capabilities of unknown traffic in DDoS attacks and the framework's ability to incorporate new information and adapt to new attack patterns, making it a powerful and intelligent solution for intrusion detection systems.

The CNN-Geo system was initially developed to provide protection against L3, L4 DDoS attacks. Moreover, it is currently incapable of mitigating the latest attack techniques, such as Connection-less Lightweight Directory Access Protocol (CLDAP) or L7 DDoS attacks, as proposed by Cloudflare. The utilization of this particular attack is prevalent due to the lack of a dataset that encompasses corresponding attack patterns. The L7 attack poses a significant challenge due to the potential for its traffic to originate from a natural source. An avenue for enhancing the efficacy of the model is to integrate deep learning models with metaheuristic optimization algorithms such as Particle Swarm Optimization

(PSO). Integrating deep learning and PSO can potentially optimize the model, resulting in an enhanced and flexible intrusion detection system. Subsequent academic pursuits will encompass supplementary modules aimed at tackling those matters. The expectation is that following the confirmation of the efficacy of this research framework, it can be implemented within an intranet setting as a cybersecurity solution for enterprises.

**Author Contributions:** Conceptualization, C.-S.S.; methodology, T.-T.N.; software, T.-T.N.; validation, T.-T.N.; writing—original draft preparation, T.-T.N.; writing—review and editing, C.-S.S.; visualization, T.-T.N.; supervision, M.-F.H.; project administration M.-F.H. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was partly supported by the National Science and Technology Council, Taiwan with grant numbers 111-2221-E-992-066 and 109-2221-E-992-073-MY3.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data supporting the reported results are available upon request.

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

## References

1. Nishant, R.; Kennedy, M.; Corbett, J. Artificial intelligence for sustainability: Challenges, opportunities, and a research agenda. *Int. J. Inf. Manag.* **2020**, *53*, 102104. [CrossRef]
2. de Neira, A.B.; Kantarci, B.; Nogueira, M. Distributed denial of service attack prediction: Challenges, open issues and opportunities. *Comput. Netw.* **2023**, *222*, 109553. [CrossRef]
3. Lazenby, S. DDoS Attacks in the Financial Industry—INETCO. Oct. 2022. Available online: https://www.inetco.com/blog/ddos-attacks-in-the-financial-industry/ (accessed on 10 April 2023).
4. DDoS in the Time of COVID-19. Resource Library, Oct. 2022. Available online: https://www.imperva.com/resources/resource-library/reports/ddos-in-the-time-of-covid-19/ (accessed on 30 October 2022).
5. Irwin, L. DDoS Attacks Soar as Organisations Struggle with Effects of COVID-19. IT Governance Blog En, Oct. 2020. Available online: https://www.itgovernance.eu/blog/en/ddos-attacks-soar-as-organisations-struggle-with-effects-of-covid-19 (accessed on 27 April 2023).
6. Pallardy, C. DDoS Attacks on US Airport Websites and Escalating Cyberattacks. InformationWeek, Oct. 2022. Available online: https://www.informationweek.com/security-and-risk-strategy/understanding-ddos-attacks-on-us-airport-websites-and-escalating-critical-infrastructure-cyberattacks (accessed on 10 April 2023).
7. Cloudflare DDoS Threat Report for 2022 Q4. The Cloudflare Blog, Jan. 2023. Available online: http://blog.cloudflare.com/ddos-threat-report-2022-q4/ (accessed on 10 April 2023).
8. Gaurav, A.; Gupta, B.B.; Alhalabi, W.; Visvizi, A.; Asiri, Y. A comprehensive survey on DDoS attacks on various intelligent systems and it's defense techniques. *Int. J. Intell. Syst.* **2022**, *37*, 11407–11431. [CrossRef]
9. DDoS Attack against Dyn Managed DNS. October. 2022. Available online: https://www.dynstatus.com/incidents/nlr4yrr162t8 (accessed on 30 October 2022).
10. Mittal, M.; Kumar, K.; Behal, S. Deep learning approaches for detecting DDoS attacks: A systematic review. *Soft Comput.* **2022**. [CrossRef] [PubMed]
11. Chen, L.; Kuang, X.; Xu, A.; Suo, S.; Yang, Y. A Novel Network Intrusion Detection System Based on CNN. In Proceedings of the 2020 Eighth International Conference on Advanced Cloud and Big Data (CBD), Taiyuan, China, 5–6 December 2020; pp. 243–247. [CrossRef]
12. Kim, J.; Shin, Y.; Choi, E. An Intrusion Detection Model based on a Convolutional Neural Network. *J. Multimed. Inf. Syst.* **2019**, *6*, 165–172. [CrossRef]
13. Roopak, M.; Tian, G.Y.; Chambers, J. Deep Learning Models for Cyber Security in IoT Networks. In Proceedings of the 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 7–9 January 2019; pp. 0452–0457.
14. Maseer, Z.K.; Yusof, R.; Bahaman, N.; Mostafa, S.A.; Foozy, C.F.M. Benchmarking of Machine Learning for Anomaly Based Intrusion Detection Systems in the CICIDS2017 Dataset. *IEEE Access* **2021**, *9*, 22351–22370. [CrossRef]
15. Hindy, H.; Atkinson, R.; Tachtatzis, C.; Colin, J.-N.; Bayne, E.; Bellekens, X. Utilising Deep Learning Techniques for Effective Zero-Day Attack Detection. *Electronics* **2020**, *9*, 1684. [CrossRef]

16. Kaur, G.; Habibi Lashkari, A.; Rahali, A. Intrusion Traffic Detection and Characterization using Deep Image Learning. In Proceedings of the 2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), Falerna, Italy, 12−15 September 2020; pp. 55–62. [CrossRef]
17. Azizjon, M.; Jumabek, A.; Kim, W. 1D CNN based network intrusion detection with normalization on imbalanced data. In Proceedings of the 2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), Fukuoka, Japan, 19–21 February 2020; pp. 218–224. [CrossRef]
18. Toupas, P.; Chamou, D.; Giannoutakis, K.M.; Drosou, A.; Tzovaras, D. An Intrusion Detection System for Multi-class Classification Based on Deep Neural Networks. In Proceedings of the 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), Boca Raton, FL, USA, 16–19 December 2019; pp. 1253–1258. [CrossRef]
19. Laghrissi, F.; Douzi, S.; Douzi, K.; Hssina, B. Intrusion detection systems using long short-term memory (LSTM). *J. Big Data* **2021**, *8*, 1–16. [CrossRef]
20. Mu, J.; He, H.; Li, L.; Pang, S.; Liu, C. A Hybrid Network Intrusion Detection Model Based on CNN-LSTM and Attention Mechanism. In *Frontiers in Cyber Security*; Cao, C., Zhang, Y., Hong, Y., Wang, D., Eds.; Communications in Computer and Information Science; Springer: Singapore, 2022; pp. 214–229. [CrossRef]
21. Nwakanma, C.I.; Ahakonye, L.A.C.; Njoku, J.N.; Odirichukwu, J.C.; Okolie, S.A.; Uzondu, C.; Nweke, C.C.N.; Kim, D.-S. Explainable Artificial Intelligence (XAI) for Intrusion Detection and Mitigation in Intelligent Connected Vehicles: A Review. *Appl. Sci.* **2023**, *13*, 1252. [CrossRef]
22. Sivamohan, S.; Sridhar, S.S. An optimized model for network intrusion detection systems in industry 4.0 using XAI based Bi-LSTM framework. *Neural Comput. Appl.* **2023**, 1–17. [CrossRef]
23. Chen, J.; Yang, Y.; Hu, K.; Zheng, H.; Wang, Z. DAD-MCNN: DDoS Attack Detection via Multi-channel CNN. In Proceedings of the 2019 11th International Conference on Machine Learning and Computing, in ICMLC '19, New York, NY, USA, 22–24 February 2019; Association for Computing Machinery: New York, NY, USA, 2019; pp. 484–488. [CrossRef]
24. Kurniabudi; Stiawan, D.; Darmawijoyo; Bin Idris, M.Y.; Bamhdi, A.M.; Budiarto, R. CICIDS-2017 Dataset Feature Analysis With Information Gain for Anomaly Detection. *IEEE Access* **2020**, *8*, 132911–132921. [CrossRef]
25. Swe, Y.M.; Aung, P. A Slow DDoS Attack Detection Mechanism using Feature Weighing and Ranking. In Proceedings of the 11th Annual International Conference on Industrial Engineering and Operations Management, Singapore, 7–11 March 2021.
26. Chapaneri, R.; Shah, S. Multi-level Gaussian mixture modeling for detection of malicious network traffic. *J. Supercomput.* **2020**, *77*, 4618–4638. [CrossRef]
27. Shieh, C.-S.; Nguyen, T.-T.; Chen, C.-Y.; Horng, M.-F. Detection of Unknown DDoS Attack Using Reconstruct Error and One-Class SVM Featuring Stochastic Gradient Descent. *Mathematics* **2022**, *11*, 108. [CrossRef]
28. Najafimehr, M.; Zarifzadeh, S.; Mostafavi, S. A hybrid machine learning approach for detecting unprecedented DDoS attacks. *J. Supercomput.* **2022**, *78*, 8106–8136. [CrossRef] [PubMed]
29. Bendale, A.; Boult, T.E. Towards Open Set Deep Networks. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1563–1572. [CrossRef]
30. Yoshihashi, R.; Shao, W.; Kawakami, R.; You, S.; Iida, M.; Naemura, T. Classification-Reconstruction Learning for Open-Set Recognition. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 4011–4020.
31. Zhang, F.; Fan, H.; Wang, R.; Li, Z.; Liang, T. Deep Dual Support Vector Data description for anomaly detection on attributed networks. *Int. J. Intell. Syst.* **2021**, *37*, 1509–1528. [CrossRef]
32. Gouda, W.; Tahir, S.; Alanazi, S.; Almufareh, M.; Alwakid, G. Unsupervised Outlier Detection in IOT Using Deep VAE. *Sensors* **2022**, *22*, 6617. [CrossRef] [PubMed]
33. Henrydoss, J.; Cruz, S.; Rudd, E.M.; Gunther, M.; Boult, T.E. Incremental Open Set Intrusion Recognition Using Extreme Value Machine. In Proceedings of the 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), Cancun, Mexico, 18–21 December 2017; pp. 1089–1093. [CrossRef]
34. Shieh, C.-S.; Lin, W.-W.; Nguyen, T.-T.; Chen, C.-H.; Horng, M.-F.; Miu, D. Detection of Unknown DDoS Attacks with Deep Learning and Gaussian Mixture Model. *Appl. Sci.* **2021**, *11*, 5213. [CrossRef]
35. Yang, K.; Zhang, J.; Xu, Y.; Chao, J. DDoS Attacks Detection with AutoEncoder. In Proceedings of the NOMS 2020—2020 IEEE/IFIP Network Operations and Management Symposium, Budapest, Hungary, 20−24 April 2020; pp. 1–9. [CrossRef]
36. Lin, Z.; Shi, Y.; Xue, Z. IDSGAN: Generative Adversarial Networks for Attack Generation Against Intrusion Detection. In *Advances in Knowledge Discovery and Data Mining*; Gama, J., Li, T., Yu, Y., Chen, E., Zheng, Y., Teng, F., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2022; pp. 79–91. [CrossRef]
37. Chauhan, R.; Heydari, S.S. Polymorphic Adversarial DDoS attack on IDS using GAN. In Proceedings of the 2020 International Symposium on Networks, Computers and Communications (ISNCC), Montreal, Canada, 20−22 October 2020; pp. 1–6. [CrossRef]
38. Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; Hochreiter, S. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Long Beach, CA, USA, 2017.
39. Sajjadi, M.S.M.; Bachem, O.; Lucic, M.; Bousquet, O.; Gelly, S. Assessing Generative Models via Precision and Recall. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc: Montreal, Canada, 2018.

40. Naeem, M.F.; Oh, S.J.; Uh, Y.; Choi, Y.; Yoo, J. Reliable Fidelity and Diversity Metrics for Generative Models. In Proceedings of the 37th International Conference on Machine Learning, Virtual Event, 13–18 July 2020; pp. 7176–7185. Available online: https://proceedings.mlr.press/v119/naeem20a.html (accessed on 10 April 2023).
41. Morfino, V.; Rampone, S. Towards Near-Real-Time Intrusion Detection for IoT Devices using Supervised Learning and Apache Spark. *Electronics* **2020**, *9*, 444. [CrossRef]