mathematics

MDPI

Article

Detection of Unknown DDoS Attack Using Reconstruct Error and One-Class SVM Featuring Stochastic Gradient Descent

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

[1] Department of Electronic Engineering, National Kaohsiung University of Science and Technology, Kaohsiung 807618, Taiwan
[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 807378, Taiwan
* Correspondence: tuannt@ntu.edu.vn

Abstract: The network system has become an indispensable component of modern infrastructure. DDoS attacks and their variants remain a potential and persistent cybersecurity threat. DDoS attacks block services to legitimate users by incorporating large amounts of malicious traffic in a short period or depleting system resources through methods specific to each client, causing the victim to lose reputation, finances, and potential customers. With the advancement and maturation of artificial intelligence technology, machine learning and deep learning are widely used to detect DDoS attacks with significant success. However, traditional supervised machine learning must depend on the categorized training sets, so the recognition rate plummets when the model encounters patterns outside the dataset. In addition, DDoS attack techniques continue to evolve, rendering training based on conventional data models unable to meet contemporary requirements. Since closed-set classifiers have excellent performance in cybersecurity and are quite mature, this study will investigate the identification of open-set recognition issues where the attack pattern does not accommodate the distribution learned by the model. This research proposes a framework that uses reconstruction error and distributes hidden layer characteristics to detect unknown DDoS attacks. This study will employ deep hierarchical reconstruction nets (DHRNet) architecture and reimplement it with a 1D integrated neural network employing loss function combined with spatial location constraint prototype loss (SLCPL) as a solution for open-set risks. At the output, a one-class SVM (one-class support vector machine) based on a random gradient descent approximation is used to recognize the unknown patterns in the subsequent stage. The model achieves an impressive detection rate of more than 99% in testing. Furthermore, the incremental learning module utilizing unknown traffic labeled by telecom technicians during tracking has enhanced the model's performance by 99.8% against unknown threats based on the CICIDS2017 Friday open dataset.

Keywords: distributed denial of service (DDoS); deep learning; open-set recognition (OSR); one-class support vector machine; reconstruct error; incremental learning

MSC: 68T07

Citation: 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* **2023**, *11*, 108. https://doi.org/10.3390/math11010108

Academic Editors: Oliviu Matei and Rudolf Erdei

Received: 10 November 2022
Revised: 18 December 2022
Accepted: 20 December 2022
Published: 26 December 2022

1. Introduction

Since 20 September 2016, the Mirai malware has attacked Internet of Things (IoT) devices [1] and crippled half of U.S. network activity [2]. Distributed denial of service (DDoS) attacks have exploded and escalated trends over many years. With the COVID-19 pandemic breakout in 2020, people have isolated themselves from activities and become more dependent on the network, and DDoS attacks have also grown dramatically [3]. Because most businesses are service providers, they must be operated continuously, so failure caused by a hacked network or service will result in financial and reputational loss [4]. Along with the advancement of technology, DDoS attack techniques evolve

daily [5], and it is impossible to defend against new threats with old methods. In this situation, we require a mechanism that allows the existing intrusion detection system (IDS) to recognize unknown traffic characteristics to assist the telecom engineer in locating unseen attacks.

According to the distributed denial of service (DDoS) quarterly report conducted by Cloudflare, a content delivery network (CDN) provider, thousands of DDoS attacks are launched each month [6]. Although most of the attack traffic is below 500 Mbps, this volume is sufficient to interrupt several enterprise systems temporarily. Even every quarter, specific attacks up to 100 Gbps will occur, causing large-scale service disruptions and possibly data center closures, harming the service provider's finances and resulting in compensation.

In light of internet information activities' growth and the expansions of new services, DDoS attack tactics are also continually evolving. This is a significant challenge for traditional IDS systems, which must repeatedly be trained on attack patterns reported by telecommunications experts. However, according to a report by Cloudflare, most attacks are over within an hour, making it too late for telecom technicians to launch an investigation. Artificial intelligence technology has made pivotal advancements in recent years, and related research has been utilized in various disciplines, including cybersecurity. Many deep learning-based IDSs have been designed and exhibit high accuracy. The accuracy rate for identifying recognized conventional DDoS attacks can reach more than 90% in the relevant experiments [7–9]. However, if a traditional IDS encounters new types of attacks, the model does not consider them unknown, so they are incapable of being confronted. Given this, we need an IDS that can flag the unknown traffic to the telecom engineer for analysis at the start of the attack rather than evaluating whether it is good or bad, especially when the different characteristics between old and new threats are highly evident. The defensive system's reaction will be particularly crucial if it faces an attack with distinct essential elements. That indicates that the issue is no longer with the performance of the training procedure; perhaps the most straightforward approach is to update the training and test datasets. Nevertheless, the model's challenge is the unknown traffic, and the open set is not as simplistic as the closed one.

This paper proposes a novel IDS architecture using deep learning technology as a basis combined with the statistical value of the reconstruction error and the distribution of the output feature space to detect unknown traffic. The model backbone employs DHRNet [10] as the original architecture, enhanced with SLCPL (spatial location constraint prototype loss) [11] to centralize the outputs in different directions, whereas the feature space distribution modeling part is implemented by a one-class support vector machine (OC-SVM) [12] approximated by stochastic gradient descent (SGD). This study's architecture inherits advantages from DHRNet architecture: it can directly generate reconstruction errors to incorporate with SGD OC-SVM to identify unknown traffic and forward it to the telecom engineer for labeling. The incremental learning module uses labeled samples to enhance the defensive performance of the IDS.

The remainder of this paper is organized as follows: Section 2 provides a summary of related work. Section 3 describes the assumptions about the situation and the detection framework proposed in this paper. The experimental results are described in Section 4. Section 5 concludes this research and provides future prospects.

2. Related Work

2.1. IDS Based on Machine Learning and Deep Learning

Under the closed-set assumption where attacks are correlated with the dataset, there has been considerable research on implementing artificial intelligence technologies in IDS systems, such as random forest (RF), support vector machine (SVM), convolutional neural network (CNN), and long short-term memory (LSTM), which have achieved excellent performance [13]. However, IDSs based on these technologies are incapable of detecting unknown attacks. Some unsupervised learning-based approaches, such as autoencoders, can identify attacks by adjusting thresholds, but false-positive rates may reach up to 10% [14].

In recent years, IDS models utilizing CNN architecture, as researched by Chen et al. [7] and Kim et al. [8], have all achieved good accuracy of 94% or higher. In addition, CNN defense models employing CSV files and image reconstruction technologies developed by Kaur et al. [15] have also gained positive results. For the problem of unbalanced data during IDS training, M. Azizjon et al. categorized data using a 1D-CNN architecture [16]. Meanwhile, P. Toupas et al. [17] employed SMOTE ENN pseudo-sampling to make the data more balanced and incorporated the Yeo-Johnson transformation in the preprocessing step to alleviate the deformed data distribution. However, counterfeiting must be conducted with caution, as the properties of the imitation are dissimilar from the original distribution and may be confused with malicious traffic. Furthermore, some architectures employ LSTM and RNN [18] with reasonable accuracy, possibly exceeding 90%.

In an effort to broaden the scope of security, researchers have begun investigating layer 7 (L7) DDoS attacks, which aim at the application layer of the OSI model and endeavor to exploit web application features to disable and limit access to these services. M. Cirillo et al. [19] establish the technical circumstances under which the BotClusterBuster identification algorithm can predict the real botnet using an emulation dictionary along with individual clusters. L. Zhou et al. [20] propose a detection measurement for low-rate DDoS attacks based on the expected size of hypertext transfer protocol packets. The shrew DDoS attacks are another type of L7 DDoS that is periodic, bursty, and stealthy. By examining the frequency-domain characteristics of incoming data flows to a server, Yu Chen et al. [21] developed a new signal processing approach for identifying and detecting shrew DDoS.

To provide insight into the set of articles that share the proposed work's objective and to compare DL approaches for intrusion detection, we selected the recent research presented in Table 1. The first column provides a pointer to the source; the second column describes the dataset used; the third column highlights the problem coverage (close-set recognition or open-set recognition); and the fourth column offers a brief description of the surveyed technical, with "non-homogeneous" referring to a comparison.

Table 1. Notable work related to deep learning techniques implemented for DDoS detection.

Author	Dataset	Problem Coverage	Technical	Year
Chen et al. [7]	CICIDS2017	CSR	NIDS (network intrusion detection system) based on CNN. Detection models were trained using both extracted features and original network data.	2020
Kim et al. [8]	CSE-CIC-IDS 2	CSR	A convolutional neural network (CNN) model employed deep learning image techniques.	2019
Roopak et al. [9]	CICIDS2017	CSR	Four different deep learning models for classifying: MLP, 1D-CNN, LSTM, and CNN + LSTM.	2019
Hindy et al. [14]	NSLKDD and CICIDS2017	CSR	Autoencoder implementation for detecting zero-day attacks.	2019
Kaur et al. [15]	CICIDS2017 and CSE-CICIDS2018	CSR	A deep neural model CNN featured image learning to classify various attacks.	2020
Henrydoss et al. [22]	KDDCUP'99	OSR	Extreme value machine, derived from the statistical extreme value theory, is capable of kernel-free, nonlinear, variable bandwidth outlier detection in conjunction with incremental learning.	2017
Shieh et al. [23]	CICIDS2017	CSR, OSR	DDoS detection framework featuring bidirectional long short-term memory (BI-LSTM), Gaussian mixture model (GMM), and incremental learning.	2021

Table 1. *Cont.*

Author	Dataset	Problem Coverage	Technical	Year
Chapaneri et al. [24]	CICIDS2017	CSR, OSR	Multilevel Gaussian mixture model able to precisely classify network traffic into several classifications and detect novel attacks.	2021
Our	CICIDS2017, CICDDoS 2019	CSR, OSR	One-dimensional deep hierarchical reconstruction nets (1D-DHRNet) combined with spatial location constraint prototype loss (SLCPL), one-class SVM and SGD as a solution for open-set risks.	2022

2.2. Open-Set Recognition

Suppose that in closed-set training, we only guarantee no overfitting or underfitting between the training and test sets. However, open-set recognition makes the issue more challenging due to unknown patterns. Numerous researchers have gradually explored and investigated open-set recognition in recent years as A. Bendale et al. introduced the OpenMax class [25], stating that the model should reject the output and thus modify the number of output layers from N to N + 1 layers. The Weibull function was used to estimate the probability and subtract it from the total probability of 1 before passing it to the softmax. Additionally, the Weibull assessment is only employed for some samples from the distribution's poles. Then, the distances are calculated from the hypersphere's center, which is determined using the output of the mean activation vector in the feature space (MAV). If the distance exceeds the acceptable range, it is assumed that the sample does not belong to any class. This is called the OOD (out of distribution) method. In Bendale's research, Weibull curves and OpenMax are based on extreme theory, regularly utilized in image classification. The hypersphere distribution is determined using the CROSR architecture [10], which combines reconstruction and distribution. Simultaneously, the reconstructed hidden layer's output is employed to improve detection performance. The extreme theory is conceptually based on a spatial distribution approximating a probability density function. If the new sample falls beyond the acceptable range, it shows it is unknown. Meanwhile, for the distribution of the output space of the system trained with softmax loss, the model's primary goal is classification; therefore, the distance will remain near the boundary regardless of the distance between classes.

2.3. Deep Learning on Open-Set Recognition

The OSR deep learning classifier comprises two components: a closed-set classifier and an unknown detector that both utilize a deep classification-reconstruction network. While the known-class classifier makes use of a supervised learning-based prediction y, the unknown detector combines y with a reconstructive latent representation z. This enables unknown detectors to utilize a larger set of traits that may not be discriminatory for known classes. In addition, higher-level layers of supervised deep neural networks tend to lose input information, which may not be desirable for unknown recognition. To simultaneously provide effective y and z, we adapted deep hierarchical reconstruction nets (DHRNets) [10]. The basic concept of DHRNets is bottlenecked lateral connections, which can be exploited to simultaneously train rich representations for classifying and compact representations for detection of unknowns. DHRNets gain hierarchical latent representation via learning reconstruction of each intermediate layer in classification networks using latent representations, i.e., mapping to low-dimensional spaces.

In OSR, accurately classifying known classes equates to a decrease in empirical risk. Consequently, OSR must reduce not just the generalization risk but also the open-set risk, which relates to efficiently recognizing unknown classes. Deep neural networks excel at closed-set recognition due to their robust feature extraction capacity. However, when a conventional deep learning model is applied to OSR, there is a clear overlap between

known- and unknown-class features. The overlap of these features in the feature space causes the open-space risk. Xia et al. offer the SLCPL (spatial location constraint prototype loss) for OSR, which adds a constraint term to regulate the spatial placement of prototypes in the feature space to mitigate the two hazards simultaneously. This method not only reduces empirical risk effectively but also governs the cluster of known classes in the boundary of the feature space.

2.4. Unknown DDoS Detection

In recent years, besides the works utilizing extreme value theory [22], there are other approaches to identify unknown DDoS attacks by determining the input's distribution using the Gaussian mixture model (GMM) and related techniques [23,24]. Extreme vector machines (EVMs) are commonly used in research on extreme value theory to find samples whose feature spaces are out of distribution. J. Henrydoss et al. achieved excellent results on the KDD99, and the paper also mentions the reduction of severe minority classes and their redundant data. The study's limitation is that it is restricted to a specific dataset and unexpanded to other datasets. It could be because the property compatibility of various datasets differed.

Based on the distribution threshold of GMM [23], Shieh et al. used BI-LSTM as a deep learning framework to distinguish benign from malicious in binary classification and finally used OOD for unknown identification. Unlike EVM, this research employs BI -LSTM feature values as unknown identity characteristics rather than the original. The dataset is theoretically similar to the OpenMax implementation, but it applies GMM to fit the output feature distribution to recognize patterns that exceed the threshold. Chapaneri et al. deployed numerous GMMs to suit each input feature in another GMM-based investigation [24]. The input samples to the GMM employed in the study were raw data rather than deep learning model output characteristics. The CICIDS2017 dataset was used for the tests in the two GMM papers, and the findings demonstrated that it could be used to identify unknown traffic to some extent.

K. Yang et al. deployed an autoencoder featuring reconstruction error known as AE-D3F for threat detection [26]. The framework was tested on three distinct datasets and achieved a detection level of 82% with a false positive rate of 0%. The usage model was derived from publicly available datasets and trained with only benign traffic. Although the framework did not return unknown samples, it still obtained good detection results. In addition, numerous techniques employing generative adversarial networks (GAN) as IDS have emerged. These solutions are frequently more sophisticated in architecture and exceedingly difficult in training. Z. Lin et al. introduced the IDSGAN architecture [27], which utilizes the GAN network to resist malicious traffic directly targeting a defending system. In this study, adversarial samples significantly impacted the performance of conventional classifiers. R. Chauhan et al. deployed WGAN to overcome the initial GAN training problem [28] and demonstrated that adversarial attacks would negate the performance of the original trained model. GAN is almost an independent domain of cyber attack and defense. On the attack side, the created patterns of the adversary network can quickly render the defense model ineffective. On the defense side, it is essential to maintain the robustness of the discriminators to withstand malicious attacks. This paper is based on the OOD schema and reconstructed error detection to address the OSR problem.

3. Proposed Methodology

This article presents a framework incorporating 1D-DHRNet implicit reconstruction error and SLCPL loss function, a one-class support vector machine module (OC-SVM), and incremental learning as a solution to the OSR challenge in DDoS attack detection. Figure 1 depicts the functional diagram of the proposed framework.

This study's framework is constructed around the 1D-DHRNet model, which is used to discriminate between regular traffic and DDoS attacks. This model's loss function comprises two parts. The first component is the reconstruction error, and SSE is used

as the loss function for encoding and decoding restoration. The second part is SLCPL, located after the model's output to deal with open-set risk. For SLCPL, the loss decreases as the output of same-class samples becomes more concentrated and the distance between different-class samples increases. To enable the model to detect unknown samples, this study adopts the SGD OC-SVM approach to model the feature space generated by SLCPL and identify samples outside the distribution. When SGD OC-SVM is fitted, only samples of the same class correctly classified by the model are used; data scattered outside the fitting range of this class of models are considered outliers.

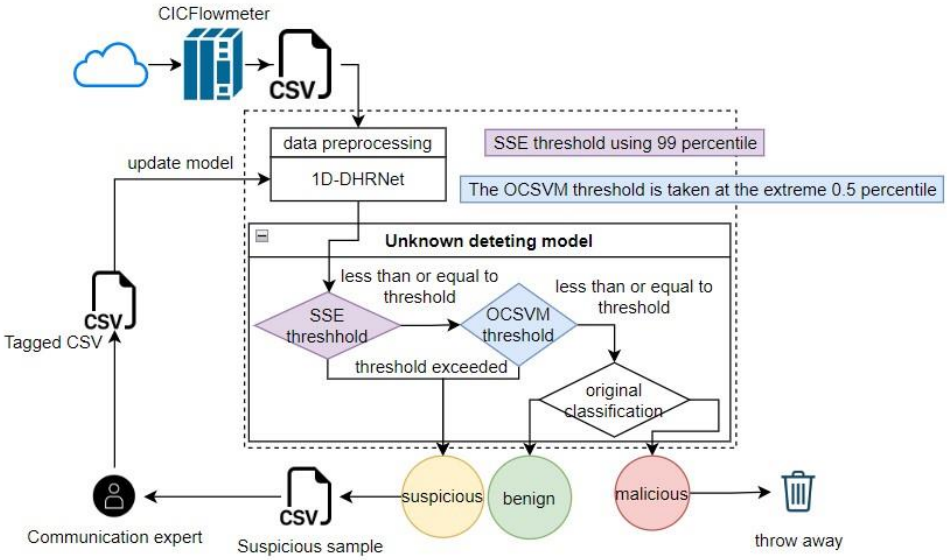

Figure 1. Proposed framework architecture.

In this research, DHRNet is preferred as the backbone. The network architecture concept proposed by Yoshihashi [10] for the image field as a classification network employs an encoder-decoder, and the reconstruction error is considered during training. The prominent feature of DHRNet is that procedural reconstruction errors for unseen samples are more significant than the training data. Due to the different data types, the dataset in this study uses numerical type; therefore, we refer to the concept of this architecture and reimplement it with 1D CNN, which is called 1D-DHRNet.

The potential danger in the OSR problem is that even though the unknown samples have different spatial distributions, the softmax function will still classify them into any category. This study will rely on SLCPL to control the object space to eliminate the above issue. The output of this method will centralize the distribution of samples to create more space for different samples with any class. As a necessary enhancement, this study incorporated SLCPL into the loss function of 1D-DHRNet. This framework's loss function consists of two parts. The first part employs reconstruction error and the loss function SSE (total squared error). The second component uses SLCPL, which follows the model's output. The smaller the SLCPL value, the more concentrated samples for the same type and the greater the distance between samples of different types.

The OOD method utilized in this paper is still insufficient to give the model the ability to recognize unknown samples. It is crucial to develop a technique to model the feature space produced by SLCPL and identify samples outside the distribution. The more straightforward the procedure, the better the data generated by this research. The solution that satisfies the criteria and operates quickly enough is SGD OC-SVM. This technique simulates stochastic gradient descent using the OC-SVM. This study uses the technique of modeling each classification to get SGD OC-SVM closer to the original single-class application method. Only samples from the same class with the correct classification are used when SGD OC-SVM is fitted. Therefore, while predicting, all samples dispersed outside of this class of models' fitting range are outliers.

3.1. The 1D Deep Hierarchical Reconstruction Nets (1D-DHRNet)

This study employs a modified DHRNet-based network architecture featuring 1D convolution, as shown in Figure 2. The fundamental idea behind the network architecture is to enable the model to perform feature learning of the categories as well as classification to recover as many embedded feature values as feasible in the reconstruction phase. Following the data stream, SLCPL calculates the output y from DHRNet to determine the inner- and interclass distances. The output x'1 is used for the SSE calculation, and the loss value is merged with the class distance data from the SLCPL to accomplish the sample classification.

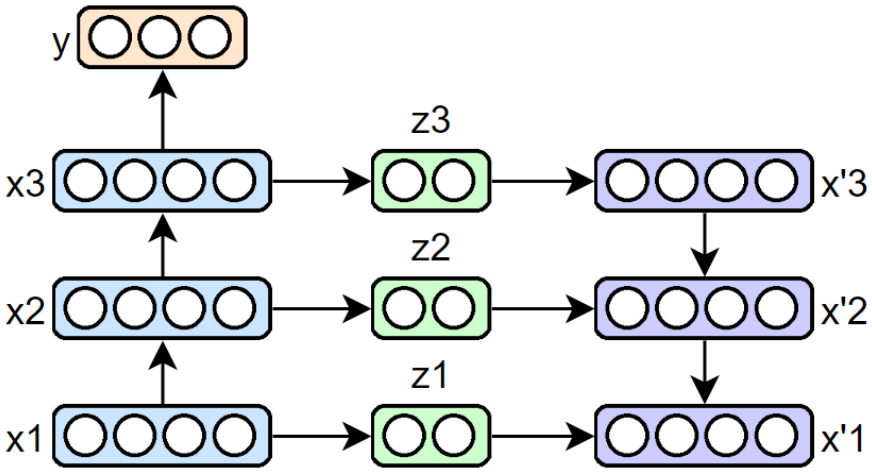

Figure 2. DHRNet conceptual architecture.

In Figure 3, the real flow of the model's data is generally portrayed. Prelu is utilized as the activation function in CNN encoders to enrich information display. Prelu maintains negative values and is linear; hence, no gradient vanishes. The main output y, with three neurons, is sent to SLCPL for classification and aggregation operations. Another output of the model is the z layer which is depicted in Figure 2. After converting each layer's values to convolution, they are compressed, deconvolved, and then converted back to the original data for error comparison and reconstruction.

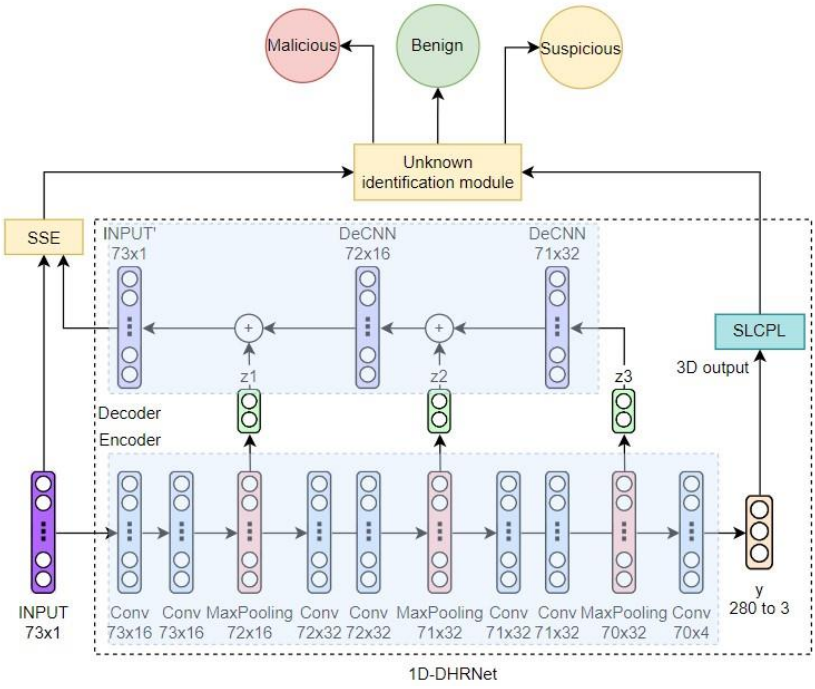

Figure 3. Framework's architecture.

3.2. Spatial Location Constraint Prototype Loss

SLCPL loss function is based on GCPL (generalized convolutional prototype learning). Both loss functions will generate large values when the model is classified correctly, but the output is not concentrated. Given k is the class being predicted, N is the number of known classes, and Θ is the embedding function (that is, the encoder CNN in the architecture of this article), $d\left(\Theta(x), O^k\right)$ is the Euclidean distance between the output of the embedding function and the center of the prototype O^k. The formula for GCPL loss is derived as (1):

$$l_G(x, y; \theta, O) = l(x, y; \theta, O) + \lambda pl(x; \theta, O) \tag{1}$$

The distance between classes is provided by $l(x, y; \theta, O)$. This loss uses the distance $d\left(\Theta(x), O^k\right)$ between the sample x and the prototype center that predicts the k class. To minimize the loss function, one can increase the value of the sample with other classes' prototype centers or reduce the distance from the predicted class prototype center as Formula (2):

$$\begin{aligned} l(x, y; \theta, O) &= -logp(y = k|x, \Theta, O) \\ &= -\log \frac{e^{-d(\Theta(x), O^k)}}{\sum_{i=1}^{N} e^{-d(\Theta(x), O^i)}} \end{aligned} \tag{2}$$

The constraint term $pl(x; \theta, O)$ is used to concentrate the distribution distance of the same class of samples. The distance formula is as in (3):

$$pl(x; \theta, O) = \|\Theta\left(x^k\right) - O^k\|_2^2, k = 1, \ldots, N \tag{3}$$

SLCPL is deduced additionally based on GCPL, as in (4). It can be found that this restriction is performed on the prototype center (5) as the SLCPL restriction item.

$$l_{SLC} = l(x, y; \theta, O) = l_G(x, y; \theta, O) + slc(O) \tag{4}$$

$$slc(O) = \frac{1}{N-1} \sum_{i=1}^{N} \left(r_i - \frac{1}{N} \sum_{j=1}^{N} r_j \right)^2 \tag{5}$$

In (5), $r_i = d\left(O^i, O_c\right)$, $O_c = \frac{1}{N} \sum_{i=1}^{N} O^i$. The r_i part is the distance between the center of the i-class prototype and the center point. The literature shows that the O_c implementation method here is helpful for optimization of the training process. By controlling the variance of these distances, the distance from the center point of each class to the coordinate origin can be limited. Then, the model can be manipulated to yield the original value of the output. The space near the point in this paper, $l(x, y; \theta, O)$ will be written as l_{SLC}. The conceptual diagram of the operation is shown in Figure 4, where the black dotted line is the decision boundary of softmax when making classification judgments.

3.3. Reconstruction Loss

This research uses reconstruction loss and SLCPL as multipurpose loss functions during training. Reconstruction loss will force the model to classify and reconstruct during training, and SLCPL will strengthen various types of intraclass distances during classification.

The loss part uses *SSE* (sum of squared for error), which is expressed in (6) as reconstruction errors, and the loss for each batch is (7).

$$l_{SSE} = SSELoss(s, y) = \sum_{t=1}^{n} (s_t - y_t)^2 \tag{6}$$

$$l_{SSEBatch} = \frac{SSELoss(s, y)}{BatchSize} = \frac{\sum_{t=1}^{n} (s_t - y_t)^2}{BatchSize} \tag{7}$$

where s are the original features, and y are the features after reconstruction.

Compared with MSE (mean squared error), SSE can make the model pay more attention to the restoration difference of a single feature in the training stage. Because the single sample error is no longer averaged but evolved, this will magnify the reconstruction error of a single feature item. The overall loss function formula is shown in Formula (8).

$$l_{Total} = l_{SSE} + l_{SLC} \tag{8}$$

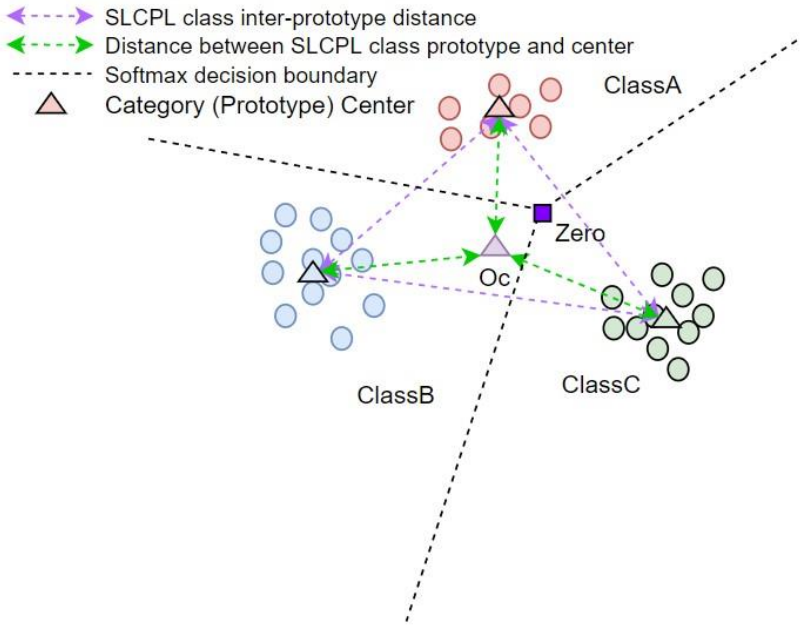

Figure 4. SLCPL feature space and softmax dividing line.

3.4. Unknown Identification Module

Under the OpenMax principle, a hypersphere is constructed for each category, with the average start vector as the center. The farthest Euclidean distances from the center will be used to fit the Weibull curve to accumulate the distribution function for extreme value estimation. Therefore, this study uses the same concept, using the 3D feature space output produced by SLCPL with centralized features, with a one-class support vector machine (OC-SVM) featuring the SGD variant for hypersphere construction. Compared with the radial basis kernel function version of the OC-SVM, the computational complexity of the SGD OC-SVM is much lower.

The OC-SVM algorithm aims to find a hypersphere that distinguishes positive samples from negative samples. This outcome can be regarded as an optimization problem. The gradient descent method utilizes all samples to update the gradient loss during calculation, so its computational complexity remains high. SGD is also based on gradient descent, but small sample batches are used for updating. Since the update parameters are solved in small batches, the degree of loss reduction can be observed to determine when to stop the iteration. This approximation can significantly reduce the time complexity.

For the SLCPL feature space approximation map of OC-SVM, refer to Figure 5, where the yellow area is the circled area of known classification, and the samples outside the yellow area will be regarded as unknown.

In the unknown identification module, this study uses a dual-index strategy for classification, and the strategy architecture is shown in Figure 6. The first detection indicator is the observation reconstruction error l_{SSE}. Both l_{SSE} and the 99th percentile method are used to remove the large reconstruction data. Then, the OC-SVM scheme based on SGD approximation is adopted, and the model output is screened by the 0.5 percentile of the upper and lower bounds, such as in Formulas (9) and (10). Only the samples within the 99th percentile of the reconstruction error and within the OC-SVM rules will be passed, and

the others will be aggregated and forwarded to telecommunication experts. The passed rules are shown in (11).

$$OCSVM_{classLL} = OCSVM_{classSC}\ 0.5\ percentile \tag{9}$$

$$OCSVM_{classHL} = OCSVM_{classSC}\ 99.5\ percentile \tag{10}$$

$$\begin{cases} l_{SSE} \leq 99\ percentile \\ OCSVM_{classSC} \geq OCSVM_{classLL} \\ OCSVM_{classSC} \leq OCSVM_{classHL} \end{cases} \tag{11}$$

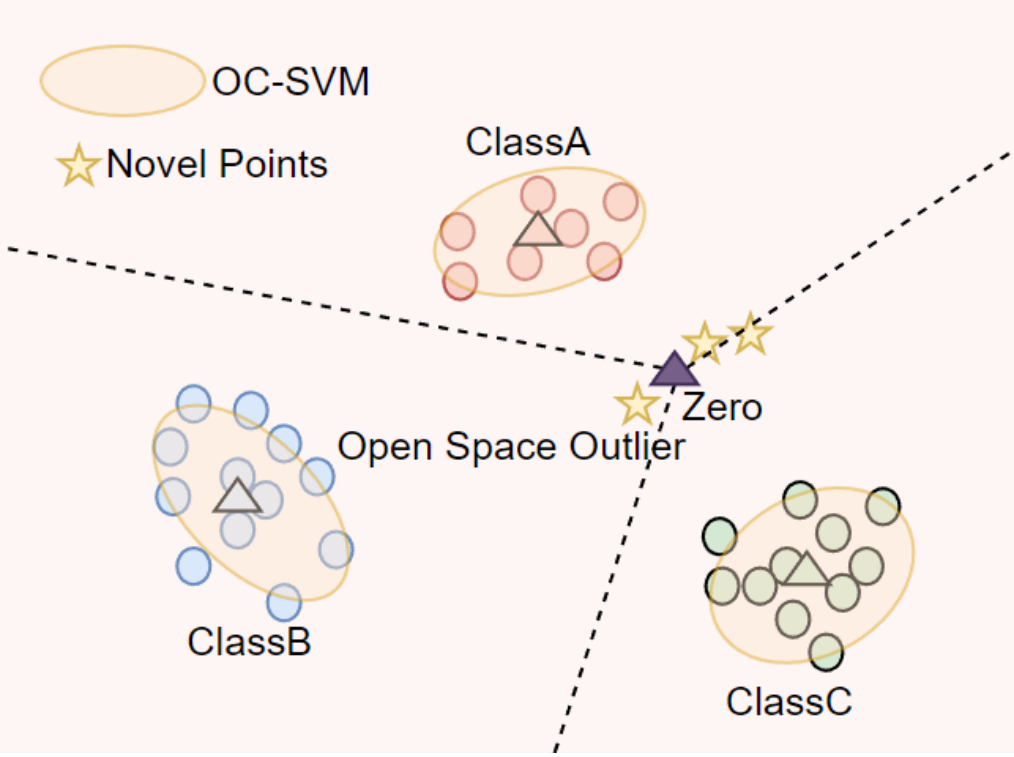

Figure 5. SLCPL feature space modeled by OC-SVM.

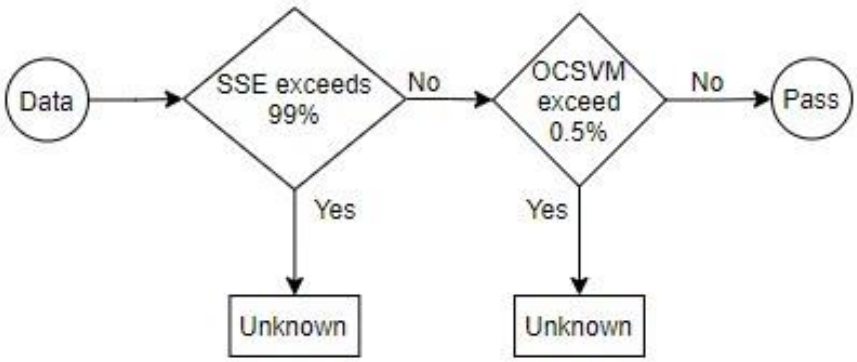

Figure 6. Unknown identification strategy.

3.5. Incremental Learning

The framework developed in this study has an unknown identification module that can capture unknown traffic. In the hypothetical situation, the captured traffic is reported to the communication experts to be marked and to let the model learn again. This study uses a fine-tuned strategy for the aforementioned purpose. In the architecture of a multiclass model, it is possible to make the model learn again by updating some framework modules. The

component that must be modified is the number of classifications of the SLCPL loss function, which allows the model to acquire new knowledge by adding new classifications and reduces the learning rate during training to prevent excessive forgetting of old knowledge.

4. Experiment

4.1. Dataset

We evaluated the proposed model on CICIDS2017 and CICDDoS2019 datasets. CICIDS2017 records 5 days of network attack traffic or normal traffic. DoS and DdoS occurred on 5 July 2017 and 7 July 2017. CICDDoS2019 is a popular dataset of amplification attacks in recent years. These two datasets contain lists of features and tags, and the signature list shows attack and normal traffic information. Table 2 lists the main attack vectors of the used datasets.

Table 2. Quantitative analysis of datasets.

Dataset	Label	Quantity	Ratio	The Total Number
CICIDS2017 Wednesday<training set>	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 Friday	BENIGN	51,496	35.117%	146,640
	DdoS	95,144	64.883%	
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%	
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%	

We used DoS attacks and normal traffic in the CICIDS2017 Wednesday dataset for model training to help give the model the ability to detect DoS attacks and normal traffic. The DdoS attacks in CICIDS2017 Friday and CICDDoS2019 were used as the unknown attacks in the experiment. The confusion matrix in Table 3 was used for the evaluation

metric, where TP is malicious traffic and is predicted as malicious traffic, TN is benign traffic and is predicted as benign traffic, FP is benign traffic and is predicted as malicious traffic, and FN is malicious traffic and is predicted as benign traffic.

Table 3. Confusion matrix.

Actual \ Predict	Malicious	Benign
Malicious	TP	FP
Benign	FN	TN

Performance indices include the confusion matrix, as shown in Table 3, and the accuracy, precision, and recall, as defined in (12)–(15), respectively. Precision attempts to answer the question of what proportion of positive identifications are correct. Recall concerns the proportion of actual positives that are identified correctly. Precision measures the percentage of identified instances that are correctly classified.

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

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

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

$$F_1 Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{15}$$

4.2. Framework

With certain efforts of investigation, we arrived at a 1D-DHRNet architecture with the configuration shown in Figure 7 and parameter settings as in Table 4. This experiment was carried out on a workstation, using Ubuntu 20.04 operating system, with AMD Ryzen 5700X 8C16T and 96 GB DDR4 memory, as well as Nvidia RTX3070 and Nvidia RTX2060 as computing acceleration devices, and the driver using NVIDIA Driver Server 510 version. Using VSCode and Conda as the development environment, the model framework part used Pytorch 1.11.0, sklearn with Python 3.9.12.

Table 4. Training parameters.

Parameter	Value
Optimizer	Adam
Weight_decay	3×10^{-5}
Learning rate	3×10^{-3}
Random seed	0, 42, 123, 222, 419, 844, 918, 1344, 65536, 815149
Training split ratio	0.8 train; 0.2 test
Batch size	256
OC-SVM nu	0.5
OC-SVM tol	1×10^{-7}

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv1d-1              [-1, 16, 73]              64
             PReLU-2              [-1, 16, 73]               1
            Conv1d-3              [-1, 16, 73]             784
             PReLU-4              [-1, 16, 73]               1
            Conv1d-5              [-1, 32, 72]           1,568
             PReLU-6              [-1, 32, 72]               1
            Conv1d-7              [-1, 32, 72]           3,104
             PReLU-8              [-1, 32, 72]               1
            Conv1d-9              [-1, 32, 71]           3,104
            PReLU-10              [-1, 32, 71]               1
           Conv1d-11              [-1, 32, 71]           3,104
            PReLU-12              [-1, 32, 71]               1
           Conv1d-13               [-1, 4, 70]             132
            PReLU-14               [-1, 4, 70]               1
           Linear-15                   [-1, 3]             843
            PReLU-16                   [-1, 3]               1
           Linear-17                   [-1, 6]              24
           Conv1d-18               [-1, 8, 70]             776
            PReLU-19               [-1, 8, 70]               1
           Conv1d-20               [-1, 8, 71]             776
            PReLU-21               [-1, 8, 71]               1
           Conv1d-22               [-1, 1, 72]              49
            PReLU-23               [-1, 1, 72]               1
           Conv1d-24              [-1, 32, 70]             800
           Conv1d-25              [-1, 32, 71]             800
           Conv1d-26              [-1, 16, 72]              64
  ConvTranspose1d-27              [-1, 32, 71]           2,080
            PReLU-28              [-1, 32, 71]               1
  ConvTranspose1d-29              [-1, 16, 72]           1,040
            PReLU-30              [-1, 16, 72]               1
  ConvTranspose1d-31               [-1, 1, 73]              33
            PReLU-32               [-1, 1, 73]               1
================================================================
Total params: 19,159
Trainable params: 19,159
Non-trainable params: 0
```

Figure 7. Deep network architecture.

We use ten different random seeds for 1D-DHRNet to train ten times, and used average results to verify that the model performed well in closed sets. The results in Table 5 show that the model works very effectively on the closed dataset.

Table 5. Training results on CICIDS2017 Wednesday.

Dataset	Accuracy	Precision	Recall	F_1 Score
CICIDS2017 Wednesday	0.99929	0.99943	0.99959	0.99951

4.3. Unknown Attack Detection and Analysis

4.3.1. Detection of Unknown Attack with First Stage 1D-DHRNet

After training on the CICIDS2017 Wednesday dataset, the 1D-DHRNet was capable of contending effectively against the conventional attack. The first test of 1D-DHRNet's defense against unknown attacks was conducted on the CICIDS2017 Friday dataset. The results of correlation comparison with the original dataset are shown in Table 6.

Table 6. Detecting results on unknown attack from CICIDS2017 Friday.

Dataset	Accuracy	Precision	Recall	F_1 Score
CICIDS2017 Wednesday	0.99929	0.99943	0.99959	0.99951
CICIDS2017 Friday	0.57859	0.98317	0.35662	0.52339

The precision of the experiment on the CICIDS2017 Friday was still maintained at 0.983, indicating that the model also has a certain generalization in defending unknown traffic. However, the accuracy score rapidly declines to 0.578, showing that the model's performance on new types of attacks is inadequate. Similar declines also occurred for recall and F1 scores. The experiment continued to be expanded with OSR datasets belonging to CICIDS2019; the results are shown in Table 7.

Table 7. Model's detecting results on each dataset.

Dataset	Accuracy	Precision	Recall	F_1 Score
CICIDS2017 Wednesday	0.99933	0.99946	0.99964	0.99955
CICIDS2017 Friday	0.57859	0.98317	0.35662	0.52339
DrDoS_LDAP	0.28329	0.99996	0.28277	0.44088
DrDoS_MSSQL	0.02268	0.99975	0.02225	0.04353
DrDoS_DNS	0.21147	0.99998	0.21095	0.34840
DrDoS_NetBIOS	0.00069	0.98435	0.00028	0.00055
DrDoS_NTP	0.01435	0.97831	0.00266	0.00531
DrDoS_UDP	0.00201	0.99666	0.00133	0.00266
DrDoS_SNMP	0.40499	1.00000	0.40482	0.57632
DrDoS_SSDP	0.00955	0.99979	0.00926	0.01836

The performance of the precision part is almost not degraded and remains at 0.99. The experiment on CICIDS2017 Friday and other unknown datasets reveals that the proposed framework did not convert many benign samples to malicious. Since benign samples in the CICIDS2017 Wednesday and CICIDS2017 Friday datasets differ, it is obvious that the model has a generalization capability for the benign classifier and does not suffer from an overfitting issue. Recall dropped significantly, indicating the model cannot provide correct answers for unknown attacks. At this time, the unknown identification module must be screened in the second stage to enhance the defense power of the overall structure.

4.3.2. Unknown Identification Module

This study uses the SLCPL method with the reconstruction error SSE and OC-SVM featuring SGD to identify unknown samples. The 99th percentile of the SSE value of the trained data is utilized as the detection threshold, and samples that fall outside the threshold are omitted. Then, the OC-SVM scheme based on SGD approximation is adopted, and the model output is screened by the 0.5 percentile of the upper and lower bounds. Only the samples within the 99th percentile of the reconstruction error and within the OC-SVM rules will be passed, and the others will be aggregated and forwarded to telecommunication experts. The concept is shown in Figure 8.

The evaluation index used for the unknown recognition module is the detection rate (DR) and the false positive rate (FPR). These two metrics are defined according to Formulas (16) and (17), respectively.

$$DR = \frac{Outlier}{C_{OTHER}} \tag{16}$$

$$FPR = \frac{Outlier}{C_{BENIGN}} \tag{17}$$

where *Outlier* is the number of data samples that exceed the threshold after being processed by the model, C_{BENIGN} is the number of benign samples, and C_{OTHER} is the number of nonbenign samples.

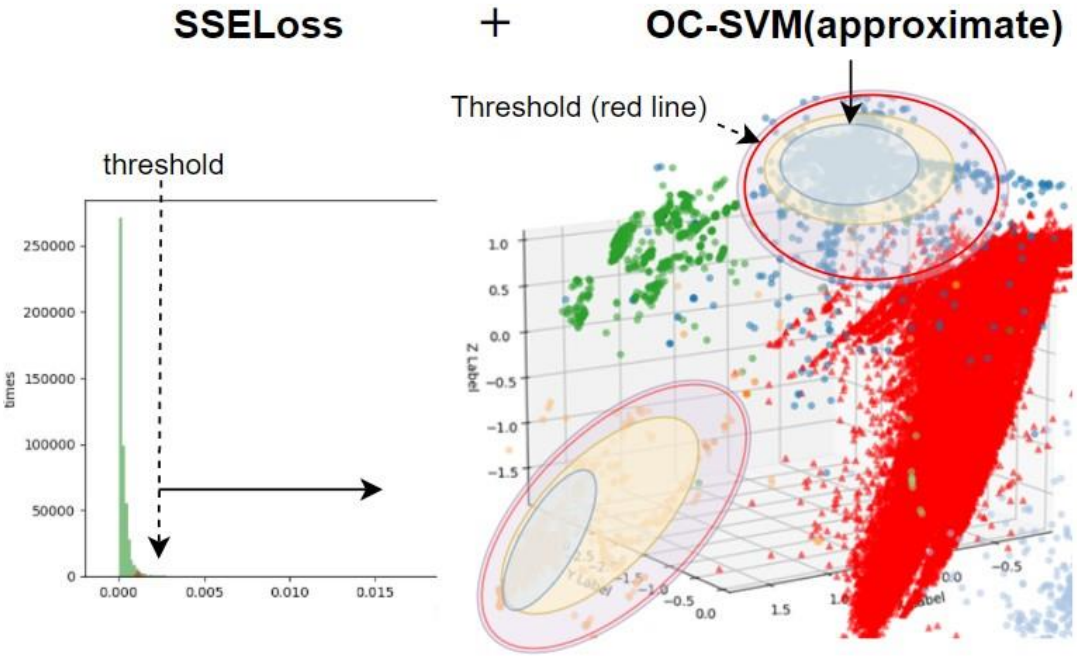

Figure 8. Unknown classification evaluation index.

4.3.3. Unknown Traffic Detection Result

Table 8 shows the framework's defense against unknown attacks with DR and FPR metrics. Noting that some data originate from the same network environment makes this FPR score more indicative. The remaining attack categories are primarily concerned with the DR rating, which reflects this model's efficiency against unknown threats.

Table 8. Unknown attack detection result.

Dataset	*DR*	*FPR*
CICIDS2017 Friday	0.99978	0.04635
CICDDoS 2019 LDAP	0.99996	0.38452 *
CICDDoS 2019 MSSQL	0.99982	0.36241 *
CICDDoS 2019 DNS	0.99775	0.44053 *
CICDDoS 2019 NetBIOS	0.99976	0.39355 *
CICDDoS 2019 NTP	0.99493	0.41334 *
CICDDoS 2019 UDP	0.99970	0.36541 *
CICDDoS 2019 SNMP	0.99983	0.41611 *
CICDDoS 2019 SSDP	0.99982	0.35302 *
CICDDoS 2019 SYN	0.99115	0.44216 *

* Indicates from different network environments, so the data are less representative.

The traffic belonging to the CICIDS2017 Friday dataset was recorded in the same period and network environment as training data, so the near-zero FPR value of this dataset may accurately represent this study framework's extraordinarily rare false alarm.

In addition, the DR value is 0.99978, indicating that the model can almost perfectly capture all DDoS network flows. In the performance of the attacks from CICDDoS 2019, it can be seen that the DR of this model for the attack reaches more than 0.99, which demonstrates the assistance of the unknown identification module. The model regards the vast majority of malicious traffic as unknown. Most attack traffic has been shut out through the control of reconstruction error. In the test of the CICDDoS2019 dataset, the benign traffic composition is not necessarily the same as CICIDS2017. Therefore, the FPR indicators of those tests are less informative.

4.3.4. Incremental Learning and the Post-Incremental Learning Results

After being detected by the unknown detection module, the unknown traffic can be forwarded to the communication engineer for analysis and marking, and finally sent to the incremental learning module for fine-tuning. Only new data are used in the incremental learning process, not the original training data, and this method is called fine-tuning. Although it will cause a slight performance degradation, it can still maintain a certain level for the old task and is more reasonable for the actual online operation situation. Regarding the incremental learning performance, the sorted table is shown in Table 9. In the "post-incremental learning" item in the table, the test also uses the training set data used in pretraining together with CICIDS2017 Wednesday to verify that the old knowledge is not excessively forgotten.

Table 9. Model's defending results after incremental learning.

Dataset	Test	Accuracy	Precision	Recall	F1 score
CICIDS2017 Wednesday	Raw performance	0.99929	0.99943	0.99959	0.99951
	After incremental learning	0.99933	0.99946	0.99964	0.99955
CICIDS2017 Friday	Raw performance	0.57859	0.98317	0.35662	0.52339
	After incremental learning	0.99864	0.99711	0.9997	0.9984
CICDDoS 2019 LDAP	Raw performance	0.28329	0.99996	0.28277	0.44088
	After incremental learning	0.99942	0.99986	0.999 48	0.9996 7
CICDDoS 2019 MSSQL	Raw performance	0.02268	0.99975	0.02225	0.04353
	After incremental learning	0.99909	0.99981	0.9992 3	0.99952
CICDDoS 2019 DNS	Raw performance	0.21147	0.99998	0.21095	0.3484
	After incremental learning	0.999	0.99924	0.99969	0.99946
CICDDoS 2019 NetBIOS	Raw performance	0.00069	0.98435	0.00028	0.00055
	After incremental learning	0.99845	0.99974	0.99860	0.99917
CICDDoS 2019 NTP	Raw performance	0.01435	0.97831	0.00266	0.00531
	After incremental learning	0.99360	0.99856	0.9935	0.99602
CICDDoS 2019 UDP	Raw performance	0.00201	0.99666	0.00133	0.00266
	After incremental learning	0.99928	0.99976	0.99945	0.99603
CICDDoS 2019 SNMP	Raw performance	0.40499	0.9999	0.40482	0.57632
	After incremental learning	0.99905	0.9999	0.99909	0.9995
CICDDoS 2019 SSDP	Raw performance	0.00955	0.99979	0.00926	0.01836
	After incremental learning	0.93473	0.9991	0.92807	0.96228
CICDDoS 2019 SYN	Raw performance	0.07925	0.99996	0.07899	0.14642
	After incremental learning	0.99892	0.99923	0.99946	0.99935

As indicated in Table 9, integrating the proposed framework can effectively solve the open-set recognition problem in detecting unknown attacks. With the help of traffic

engineers, labeled new instances are fed back to the proposed model for incremental learning. The good performance of CIC-DDoS2019/NTP and CIC-DDoS2019/LDAP is much more evident. With the aid of the suggested 1D-DHRNet-OCSVM framework and incremental learning strategy, all performance indicators return to acceptable levels. The updated model can then deal with both the old and new traffic correctly and efficiently.

4.3.5. Time Complexity of Proposed Framework

Another aspect to consider in this study is the time complexity of the proposed model. To ensure that the model can react promptly in a real-time environment, the information of training and predicting time on the CICIDS2017 Wednesday dataset is shown in Table 10. These data are averaged and derived from 30 independent executions.

Table 10. Training and predicting time on the CICIDS2017 Wednesday.

Dataset	Training Time (s)	Predicting Time (s)
CICIDS2017 Wednesday	179.31	4.39

Figure 9 also depicts the model's prediction time on 10 unknown datasets.

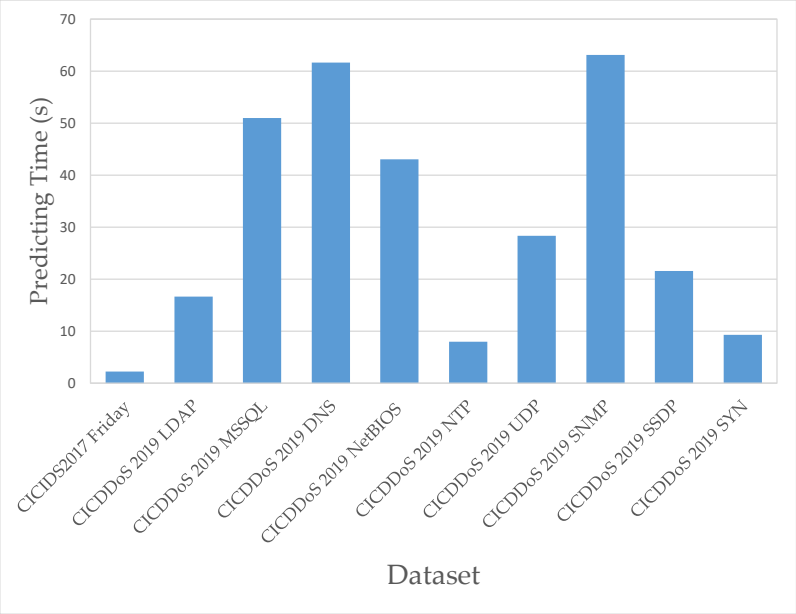

Figure 9. The 1D-DHRNet-OCSVM model's prediction time on 10 unknown datasets.

The average training time of the proposed model on the CICIDS2017 Wednesday dataset is 179.31 s, which is relatively fast and is adequate for a complicated deep learning model such as 1D-DHRNet-OCSVM. In terms of prediction time, the predicted execution times for the CICIDS2017 Wednesday and CICIDS2017 Friday datasets are as fast as 4.39 s and 2.26 s, respectively. Further, 1D-DHRNet-OCSVM also performed relatively fast prediction on two sets of CICDDoS 2019 SYN and CICDDoS 2019 NTP, with execution time under 10 s. For larger datasets, the prediction time ranges from 16.66 s for CICDDoS 2019 LDAP to 63.15 s for the largest dataset CICDDoS 2019 SNMP. All of these results are satisfactory, and it can be assumed that the 1D-DHRNet-OCSVM model meets the criteria for real-time detection.

5. Conclusions

According to existing research, the preponderance of training and testing studies only analyze known categories. Therefore, an intrusion detection system trained solely on datasets has weaknesses. Further, attacks having similar features to benign traffic is one

of its crucial limits. This study presents a hybrid network architecture that combines the characteristics of unsupervised and supervised networks. Concurrently, the reconstruction and classification errors are used for training in conjunction with the OOD solution to detect unknown attacks. The experimental results demonstrate that the proposed architecture can provide a closed-set training model, a technique for rejecting output or recognizing it as unknown, which depends on communications engineers for data labeling and incremental training for evolution. The architecture proposed in this study shows promise in facing unknown emerging attacks.

For the existing new attack methods proposed by Cloudflare, such as CLDAP or layer 7 (L7) DDoS attack, no dataset with relevant attack samples can retarget attacks with this type of attack. The L7 attack is the most challenging because its traffic may appear to originate from a legitimate source. Our future research direction will be adding additional expansion modules to the proposed framework to address this issue. It is hoped that after further verification of the performance of this research architecture, it can be applied to the internal network environment as the gatekeeper of enterprise network security.

Author Contributions: Conceptualization, C.-S.S.; methodology, C.-Y.C.; software, C.-Y.C.; validation, T.-T.N.; writing—original draft preparation, T.-T.N.; writing—review and editing, C.-S.S., M.-F.H.; visualization, C.-Y.C.; supervision, C.-S.S.; 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 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. The Internet of Stings. *The Economist*, October 2022. Available online: https://www.economist.com/science-and-technology/2016/10/08/the-internet-of-stings (accessed on 30 October 2022).
2. Newman, L.H. What We Know About Friday's Massive East Coast Internet Outage. *Wired*, October 2022. Available online: https://www.wired.com/2016/10/internet-outage-ddos-dns-dyn/ (accessed on 30 October 2022).
3. DDoS in the Time of COVID-19. *Resource Library*, October 2022. Available online: https://www.imperva.com/resources/resource-library/reports/ddos-in-the-time-of-covid-19/ (accessed on 30 October 2022).
4. DDoS Attack Against Dyn Managed DNS. October 2022. Available online: https://www.dynstatus.com/incidents/nlr4yrr162t8 (accessed on 30 October 2022).
5. New Variant of Mirai Embeds Itself in TalkTalk Home Routers. Imperva, Blog, December 2016. Available online: https://www.imperva.com/blog/new-variant-mirai-embeds-talktalk-home-routers/ (accessed on 30 October 2022).
6. DDoS Attack Trends for 2022 Q1. The Cloudflare Blog, April 2022. Available online: http://blog.cloudflare.com/ddos-attack-trends-for-2022-q1/ (accessed on 30 October 2022).
7. 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]
8. Kim, J.; Shin, Y.; Choi, E. An Intrusion Detection Model based on a Convolutional Neural Network. *J. Multimed. Inf. Syst.* **2019**, *6*, 4. [CrossRef]
9. 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. [CrossRef]
10. 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, 15–20 June 2019; pp. 4011–4020. [CrossRef]
11. Xia, Z.; Dong, G.; Wang, P.; Liu, H. Spatial Location Constraint Prototype Loss for Open Set Recognition. *arXiv* **2021**, arXiv:2110.11013. [CrossRef]
12. sklearn.linear_model.SGDOneClassSVM. scikit-learn, October 2022. Available online: https://scikit-learn/stable/modules/generated/sklearn.linear_model.SGDOneClassSVM.html (accessed on 30 October 2022).

13. Maseer, Z.K.; Yusof, R.; Bahaman, N.; Mostafa, S.; 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]
14. 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]
15. Kaur, G.; Lashkari, A.H.; Rahali, A. Intrusion Traffic Detection and Characterization using Deep Image Learning. In Proceedings of the 2020 IEEE International Conference on Dependable, Autonomic and Secure Computing, International Conference on Pervasive Intelligence and Computing, International Conference on Cloud and Big Data Computing, International Conference on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), Calgary, AB, Canada, 17–20 August 2020; pp. 55–62. [CrossRef]
16. 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]
17. Toupas, P.; Chamou, D.; Giannoutakis, K.; 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]
18. Laghrissi, F.; Douzi, S.; Douzi, K.; Hssina, B. Intrusion detection systems using long short-term memory (LSTM). *J. Big Data* **2021**, *8*, 65. [CrossRef]
19. Cirillo, M.; Mauro, M.; Matta, V.; Tambasco, M. Application-Layer DDOS Attacks with Multiple Emulation Dictionaries. In Proceedings of the ICASSP 2021—2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; pp. 2610–2614. [CrossRef]
20. Zhou, L.; Liao, M.; Yuan, C.; Haoyu, Z. Low-Rate DDoS Attack Detection Using Expectation of Packet Size. *Secur. Commun. Netw.* **2017**, *2017*, 3691629. [CrossRef]
21. Chen, Y.; Hwang, K.; Kwok, Y.-K. Filtering of shrew DDoS attacks in frequency domain. In Proceedings of the IEEE Conference on Local Computer Networks 30th Anniversary (LCN'05), Sydney, NSW, Australia, 17 November 2005; p. 793. [CrossRef]
22. Henrydoss, J.; Cruz, S.; Rudd, E.; 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]
23. 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]
24. Chapaneri, R.; Shah, S. Multi-level Gaussian mixture modeling for detection of malicious network traffic. *J. Supercomput.* **2021**, *77*, 4618–4638. [CrossRef]
25. Bendale, A.; Boult, T. Towards Open Set Deep Networks. *arXiv* **2015**, arXiv:1511.06233. [CrossRef]
26. 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]
27. Lin, Z.; Shi, Y.; Xue, Z. IDSGAN: Generative Adversarial Networks for Attack Generation Against Intrusion Detection. In *Advances in Knowledge Discovery and Data Mining*; Springer: Cham, Switzerland, 2022; pp. 79–91. [CrossRef]
28. 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, QC, Canada, 20–22 October 2020; pp. 1–6. [CrossRef]