applied sciences

MDPI

Article

One-Class LSTM Network for Anomalous Network Traffic Detection

Yanmiao Li [1], Yingying Xu [2], Yankun Cao [3], Jiangang Hou [4], Chun Wang [5], Wei Guo [3], Xin Li [2], Yang Xin [1], Zhi Liu [2,*] and Lizhen Cui [3,6,*]

1 School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China; yanmiao_li@hotmail.com (Y.L.); yangxin@bupt.edu.cn (Y.X.)
2 School of Information Science and Engineering, Shandong University, Qingdao 266237, China; xuyyedu@mail.sdu.edu.cn (Y.X.); 202112683@mail.sdu.edu.cn (X.L.)
3 School of Software, Shandong University, Jinan 250101, China; kunkun@sdu.edu.cn (Y.C.); guowei@sdu.edu.cn (W.G.)
4 School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China; houjiangang@126.com
5 Optical Advanced Research Center, Shandong University, Qingdao 266237, China; chunwang@sdu.edu.cn
6 Joint SDU-NTU Centre for Artificial Intelligence Research (C-FAIR), Shandong University, Jinan 250101, China
* Correspondence: liuzhi@sdu.edu.cn (Z.L.); clz@sdu.edu.cn (L.C.)

Abstract: Artificial intelligence-assisted security is an important field of research in relation to information security. One of the most important tasks is to distinguish between normal and abnormal network traffic (such as malicious or sudden traffic). Traffic data are usually extremely unbalanced, and this seriously hinders the detection of outliers. Therefore, the identification of outliers in unbalanced datasets has become a key issue. To help solve this challenge, there is increasing interest in focusing on one-class classification methods that train models based on the samples of a single given class. In this paper, long short-term memory (LSTM) is introduced into one-class classification, and one-class LSTM (OC-LSTM) is proposed based on the traditional one-class support vector machine (OC-SVM). In contrast with other hybrid deep learning methods based on auto-encoders, the proposed method is an end-to-end training network that uses a loss function such as the OC-SVM optimization objective for model training. A comprehensive experiment on three large complex network traffic datasets showed that this method is superior to the traditional shallow method and the most advanced deep method. Furthermore, the proposed method can provide an effective reference for anomaly detection research in the field of network security, especially for the application of one-class classification.

Keywords: one-class classification; anomaly detection; OC-LSTM; network traffic

Citation: Li, Y.; Xu, Y.; Cao, Y.; Hou, J.; Wang, C.; Guo, W.; Li, X.; Xin, Y.; Liu, Z.; Cui, L. One-Class LSTM Network for Anomalous Network Traffic Detection. *Appl. Sci.* **2022**, *12*, 5051. https://doi.org/10.3390/app12105051

Academic Editor: Carla Raffaelli

Received: 8 April 2022
Accepted: 16 May 2022
Published: 17 May 2022

1. Introduction

When analyzing real-world data, the common requirement is to determine which instances are completely different from the others. Such instances are called outliers, and the task of anomaly detection is to identify all such instances in a data-driven manner [1]. Generally, this is regarded as an unsupervised learning problem, and it is assumed that most training datasets consist of normal data, among which the anomalous samples are unknown. According to a recent study [2], unsupervised anomaly detection has proven to be very effective and plays a key role in a variety of applications, such as fraud detection, network intrusion prevention, and fault diagnosis [3–5]. One-class classification is a widely used and effective unsupervised technique that learns from samples belonging to a single class while treating samples belonging to other classes as anomalies. The most representative methods of this type are the one-class support vector machine (OC-SVM) [6] and support vector data description (SVDD) [7]. However, the performance of these shallow models is suboptimal,

especially on large quantities of high-dimensional data (e.g., network traffic data). It is still an open and challenging research problem to learn the inherent characteristics effectively when only one class of samples is given.

With the development of artificial intelligence in recent years, deep neural networks have been widely used in the field of information security due to their powerful feature learning capabilities [8–10]. In general, the training of a deep learning model requires a complete and detailed sample set that must contain labeled samples of all traffic types appearing in the network communication. Manually labeling such samples is time-consuming and laborious, especially for large-scale backbone network traffic [11]. In addition, this process may not cover all network anomaly types, such as complex network attack behaviors and advanced persistent threats. On the other hand, due to the low incidence rates of many network anomalies, it is usually difficult to collect enough anomaly data for training. Data imbalances seriously affect the performance of classifiers, making many models that are effective on balanced datasets perform poorly in terms of anomaly detection in real-world network traffic applications [12]. As the data in these domains are growing rapidly in size and dimensionality, people require effective and efficient ways to detect anomalies in large quantities of high-dimensional data [13].

Traditional unsupervised techniques commonly used in anomaly detection include k-means (k-means), isolation forest (IF), self-organizing maps (SOM), density-based noisy spatial clustering (density-based spatial clustering of applications with noise, DBSCAN), one-class support vector machine (OC-SVM), etc. [8]. However, traditional shallow unsupervised models still perform poorly on complex high-dimensional datasets. Especially when only normal type samples are given, the question of how to better learn their inherent properties is still an open and challenging research problem. To improve the performance of the unsupervised classification model and solve the problem of insufficient training data in network traffic anomaly detection, an interesting approach is to complete the one-class classification task through the use of a deep learning method. In this paper, LSTM is introduced into one-class classification, and OC-LSTM is proposed based on the traditional OC-SVM. In contrast with other hybrid deep learning methods based on auto-encoders, the proposed method is an end-to-end training network that uses a loss function such as the OC-SVM optimization objective for model training.

In a word, traditional shallow learning methods cannot adapt to the dynamic growth of network traffic. They cannot meet the requirements for intelligent analysis and the prediction of large-scale high-dimensional traffic data. Therefore, designing fast and efficient anomaly detection algorithms according to the characteristics of traffic data has become an urgent goal in the field of network security. In this paper, we study network intrusion detection based on the deep learning method in order to reduce the workload of security practitioners and enhance the network situational awareness of security systems, and provide strong support for the improvement of China's network security technology system and the construction of the network power strategy. The main contributions of this paper can be summarized as follows:

(1) In this paper we propose an unsupervised anomaly detection algorithm based on the one-class long short-term memory (OC-LSTM) network. The model is an end-to-end single-class neural network with a specially designed loss function equivalent to the optimization objective of a single-class SVM.
(2) By directly adopting the objective of representation learning for anomaly detection, OC-LSTM can directly process raw data without using unsupervised transfer learning for further feature extraction. This will help to discern complex anomalies in large datasets, especially when the decision boundary between normal and anomalous data is highly nonlinear.

The rest of the paper is structured as follows. In Section 2, a detailed overview of related research regarding one-class classification and anomaly detection is given. Section 3 outlines the main models of the proposed OC-LSTM method. The experimental setup, including the dataset used, the compared methods, and the evaluation metrics, is described in Section 4.

An in-depth discussion and analysis of the obtained experimental results regarding OC-LSTM and other state-of-the-art methods are the focus of Section 5. Section 6 provides the conclusions of this paper, as well as the main points and directions of future work.

2. Background and Related Work

Before introducing the OC-LSTM method, this paper briefly reviews one-class classification and presents existing deep learning-based unsupervised anomaly detection methods.

2.1. Anomaly Detection

In data mining and statistics literature, anomalies are also called abnormal data, outliers or deviant data. Anomalies can be caused by errors in the data, but also sometimes indicate the presence of new, previously unknown underlying processes. In fact, Hawkins defines an outlier as an observation that is so different from other observations that there is good reason to speculate that it is produced by a different mechanism. As shown in Figure 1, the regions R_1 and R_2 that contain most of the observations are considered normal data instance regions, whereas the regions R_3 and P_1, which are far from most of the data points, contain only a few data points, and are considered exceptions. Often caused by system failures, illegal operations, external attacks, etc., this situation often conveys and reveals valuable information and exciting insights that exist in the data. Anomaly detection is an indispensable part of various modern discriminant models and decision-making systems. Traditional anomaly detection methods are mainly divided into four categories: based on the statistical distribution, based on distance, based on density, and based on clustering.

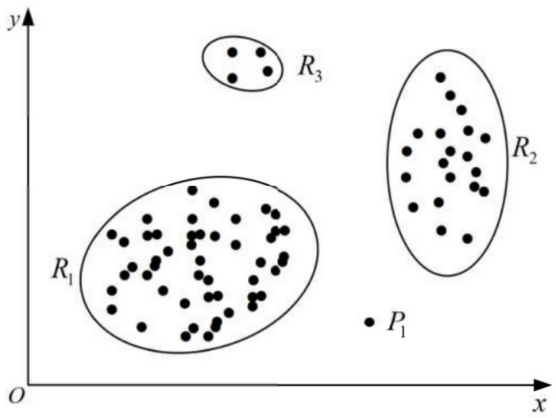


Figure 1. An example of two-dimensional data anomaly detection.

2.2. One-Class Classification

One-class classification refers to the classification learning model that only provides samples belonging to a certain class [14]. In contrast with the task of multiclass classification, which learns distinguishing features by comparing multiple samples of different classes, the key problem of one-class classification is how to effectively capture features related to a single class [15]. The one-class classification problem is described below. For a given training sample $\boldsymbol{x}$ from class A, the purpose is to learn the scoring function $f(\boldsymbol{x}) : \boldsymbol{x} \rightarrow R$. In R, a higher value indicates that the sample $\boldsymbol{x}$ is more likely to belong to class A. Therefore, for the test sample $\boldsymbol{x}'$, its score $f(\boldsymbol{x}')$ can be calculated and evaluated to determine whether it belongs to class A. In the one-class classification task, there are sufficient samples belonging to the target category and very few outliers; that is, the negative category sample portion is not available or not present. This property of the given dataset makes decision boundary detection a complex and challenging task [16].

There has been much work performed on one-class classification, usually focusing on feature fitting or feature mapping. Among them, the OC-SVM [6] is a one-class unsupervised approach that is widely used in document classification, disease diagnosis, fraud detection, etc. Intuitively, in OC-SVM, all data points are treated as instances with positive

labels, whereas the origin is treated as the only instance with negative labels, as shown in Figure 2a. Its main idea is to train a model (hyperplane) to achieve the maximum possible separation between the target data and coordinate origin, that is

$$\begin{aligned} &\min_{\omega,\rho} \tfrac{1}{2}\|\omega\|^2 + \tfrac{1}{vN}\sum_{i=1}^{N}\xi_i - \rho \\ &s.t.\ (\omega\cdot\phi(x_i)) \geq \rho - \xi_i,\ \xi_i \geq 0 \end{aligned} \tag{1}$$

where ω is the weight of the support vector, ρ is the distance between the hyperplane and the origin of the coordinates, and $v \in (0,\ 1]$ is used to control the trade-off between the maximum distance from the origin to the hyperplane and the number of data points allowed to cross the hyperplane. Training data $x_n \in X, i = 1, 2, \ldots, N$ are mapped to a high-dimensional feature space by the kernel function $\phi(\cdot)$, and their distance from the classified hyperplane is $\xi_i/\|\omega\|$. The decision function in the feature space is similar to the second-class SVM and is described by

$$f(x_i) = \mathrm{sgn}[\omega\cdot\phi(x_i) - \rho] \tag{2}$$

and most of the training set lies in the region $f(x) > 0$.

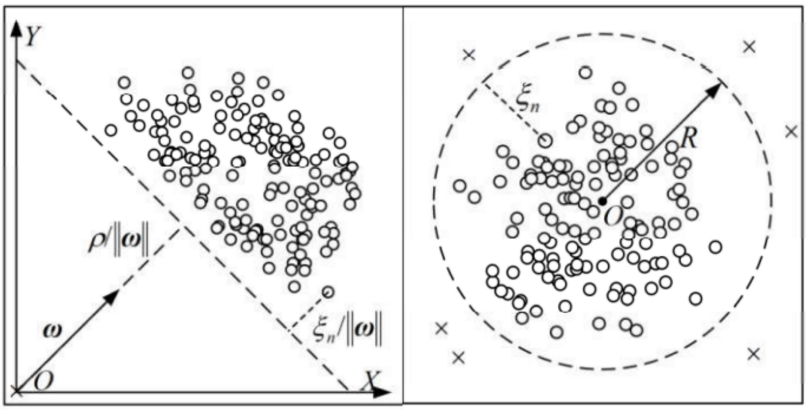


Figure 2. Schematic diagram of the classic single classification algorithm.

However, the performance of the OC-SVM on complex, high-dimensional datasets is not optimal. Based on the OC-SVM, the support vector data description (SVDD) [7] method was proposed to map an original image to a hypersphere instead of a hyperplane, as shown in Figure 2b. The primal problem in SVDD is

$$\begin{aligned} &\min_{R,\xi} R^2 + \tfrac{1}{vN}\sum_{i=1}^{N}\xi_i \\ &s.t.\ \|\phi(x_i) - a\|^2 \leq R^2 + \xi_i,\ \xi_i \geq 0 \end{aligned} \tag{3}$$

where a is the hypersphere center and R is the hypersphere radius. Again, slack variables $\xi_i \geq 0$ allow for a soft boundary, and a hyperparameter $v \in (0,\ 1]$ controls the trade-off between the penalties ξ_i and the volume of the sphere. The OC-SVM and SVDD are closely related and are still limited to complex datasets. These kernel-based approaches often fail in high-dimensional, data-rich scenarios due to their poor computational scalability and the curse of dimensionality.

2.3. Deep Learning for Unsupervised Anomaly Detection

Anomaly detection is an important topic in data science research [17]. The aim in unsupervised anomaly detection is to find a separation rule between anomalous and normal data without labels. In recent years, with the unprecedented success of deep neural

networks as feature extractors for processing image, audio, and text data [18], several hybrid models that combine deep learning and the OC-SVM have emerged [19,20]. The hybrid model uses a pretrained deep learning model to acquire rich representational features and is then fed into shallow anomaly detection methods, such as the OC-SVM. However, these hybrid OC-SVM methods are decoupled because their feature learning is task-agnostic and is not customized for anomaly detection [21].

In addition to hybrid approaches, another common method of anomaly detection is based on the use of deep auto-encoders such as the robust deep autoencoder (RDAE) [2] and robust convolution autoencoder (RCAE) [22]. The input data X of a deep autoencoder is decomposed into two parts, L_D and S, where L_D represents the latent representation of the hidden layer and S represents the noise and outliers that are difficult to reconstruct. Therefore, the optimization objective function is:

$$\begin{gathered} \min_{\theta,\, S} + \|L_D - D_\theta(E_\theta(L_D))\|_2 + \lambda \cdot \left\|S^T\right\|_{2,1} \\ s.t.\ \boldsymbol{X} = L_D + S \end{gathered} \tag{4}$$

Back-propagation and the alternating direction method of multipliers (ADMM) can be used to solve the above optimization problems, and the reconstruction error is employed as an anomaly score [23].

Auto-encoders have the objective of dimensionality reduction and do not target anomaly detection directly. However, the main difficulty of this approach is the question of how to choose the right degree of compression. Apart from auto-encoders, some deep models train neural networks by minimizing the volume of the hypersphere surrounding the data or the distance between the data and the hyperplane [20,24]. However, such experiments are still based on features extracted via deep auto-encoders or pre-trained models [25,26]. In our experiments, we carried out a detailed comparison between the proposed OC-LSTM and the abovementioned anomaly detection methods.

3. Materials and Methods

This section introduces a one-class LSTM (OC-LSTM) neural network model that employs the representation learning objective directly for anomaly detection. This makes it possible to discern anomalies in complex and large data sets, especially when the decision boundaries between the normal and anomalous data are highly nonlinear. We present the OC-LSTM objective, its optimization process, and the associated algorithm.

3.1. OC-LSTM Objective and Optimization Process

The OC-LSTM is a simple feed-forward network, which can be regarded as a neural architectural design with an OC-SVM-equivalent loss function. The optimization problem of the OC-SVM is shown in Equation (1), and it can be written as follows:

$$\min_{\omega,\rho} \frac{1}{2}\|\omega\|^2 + \frac{1}{vN}\sum_{i=1}^{N} \max(0,\ \rho - \omega \cdot \phi(\boldsymbol{x}_i)) - \rho \tag{5}$$

Assuming that a simple network consists of a hidden LSTM layer with a linear activation function $g(\cdot)$ and an output node, the scalar output from the hidden layer to the output node is $\boldsymbol{W}$, and the weight matrix from the input to the hidden layer is $\boldsymbol{V}$. The objective function of the network can be formulated as:

$$\min_{\boldsymbol{W},\boldsymbol{V},\rho} \frac{1}{2}\|\boldsymbol{W}\|^2 + \frac{1}{2}\|\boldsymbol{V}\|^2 + \frac{1}{vN}\sum_{i=1}^{N} \max(0,\ \rho - \boldsymbol{W} \cdot g(\boldsymbol{V}\boldsymbol{x}_i)) - \rho \tag{6}$$

Thus, it is possible to leverage the transfer learning features obtained using an LSTM layer by replacing $\omega\phi(\boldsymbol{x}_i)$ with $\boldsymbol{W}g(\boldsymbol{V}\boldsymbol{x}_i)$ [21].

However, the cost of this change is that the objective function becomes nonconvex and therefore the global optimal solution cannot be obtained. Fortunately, the alternating

minimization method can be used to optimize the objective [21]. Therefore, the optimization problems of $\boldsymbol{W}$, $\boldsymbol{V}$, and ρ can be defined as

$$\underset{\boldsymbol{W},\boldsymbol{V}}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{W}\|^2 + \frac{1}{2}\|\boldsymbol{V}\|^2 + \frac{1}{vN}\sum_{i=1}^{N}\max(0,\ \rho - \hat{y}_i) \tag{7}$$

$$\underset{\rho}{\operatorname{argmin}} \frac{1}{vN}\sum_{i=1}^{N}\max(0,\ \rho - \hat{y}_i) - \rho \tag{8}$$

where $\hat{y}_i = \boldsymbol{W}g(\boldsymbol{V}\boldsymbol{x}_i)$. Equation (7) can be optimized using the standard back-propagation (BP) algorithm as shown in Section 3.2, and Theorem 1 in [21] has proven that $v = \frac{1}{N}\sum_{i=1}^{N}(\rho - \hat{y}_i > 0)$, which means that the optimal value of ρ in Equation (8) is the v^{th} quantile of $\{\hat{y}_i\}_{i=1}^{N}$. Finally, the decision function can be defined as

$$f(\boldsymbol{x}_i) = \operatorname{sign}(\hat{y}_i - \rho),\ \ i = 1,\ \ldots,\ N \tag{9}$$

The proof process of (8) is as follows:

$$\begin{aligned}
&\underset{\rho}{\operatorname{argmin}}\ \tfrac{1}{vN}\sum_{i=1}^{N}\max(0,\ \rho - \hat{y}_i) - \rho \\
&= \underset{\rho}{\operatorname{argmin}}\ \tfrac{1}{vN}\sum_{i=1}^{N}\max(0,\ \rho - \hat{y}_i) - (\rho - \tfrac{1}{N}\sum_{i=1}^{N}\hat{y}_i) \\
&= \underset{\rho}{\operatorname{argmin}}\ \tfrac{1}{vN}\sum_{i=1}^{N}\max(0,\ \rho - \hat{y}_i) - \tfrac{1}{N}\sum_{i=1}^{N}(\rho - \hat{y}_i) \\
&= \underset{\rho}{\operatorname{argmin}}\ \sum_{i=1}^{N}\max(0,\ \rho - \hat{y}_i) - v\,.\,\tfrac{1}{N}\sum_{i=1}^{N}(\rho - \hat{y}_i) \\
&= \underset{\rho}{\operatorname{argmin}}\ \sum_{i=1}^{N}[\max(0,\ \rho - \hat{y}_i) - v\,.(\,\rho - \hat{y}_i)] \\
&= \underset{\rho}{\operatorname{argmin}}\ \sum_{i=1}^{N}\begin{cases} (1-v)\cdot(\rho - \hat{y}_i) & \rho - \hat{y}_i > 0 \\ -v\cdot(\,\rho - \hat{y}_i) & \rho - \hat{y}_i \le 0 \end{cases}
\end{aligned} \tag{10}$$

The derivative of (10) is obtained as:

$$F'(r) = \sum_{i=1}^{N}\begin{cases} 1 - v & r > \hat{y}_i \\ -v & r \le \hat{y}_i \end{cases} \tag{11}$$

Let $F'(r) = 0$; when $\rho - \hat{y}_i > 0$ we can get:

$$\begin{aligned}
&(1-v)\cdot\sum_{i=1}^{N}[\rho - \hat{y}_i > 0] = v\cdot\sum_{i=1}^{N}[\rho - \hat{y}_i \le 0] \\
\Rightarrow\ &(1-v)\cdot\sum_{i=1}^{N}[\rho - \hat{y}_i > 0] = v\cdot\sum_{i=1}^{N}[1 - (\rho - \hat{y}_i > 0)] \\
\Rightarrow\ &(1-v)\cdot\sum_{i=1}^{N}[\rho > \hat{y}_i] = v\cdot N - v\cdot\sum_{i=1}^{N}[\rho > \hat{y}_i] \\
\Rightarrow\ &\sum_{i=1}^{N}[\rho - \hat{y}_i > 0] = v\cdot N \\
\Rightarrow\ &\rho = v\cdot\sum_{i=1}^{N}\hat{y}_i
\end{aligned} \tag{12}$$

3.2. OC-LSTM Algorithm

The training process of the OC-LSTM model is shown in Algorithm 1. First initialize ρ in the third line. Then use the standard BP algorithm to train the parameters ($\boldsymbol{W}$, $\boldsymbol{V}$) of the

neural network in the sixth line. The seventh line updates the parameter ρ with the value of the v^{th} aliquot of $\{\hat{y}_i\}_{i=1}^{N}$. Equivalent to Equation (6), the cost function of the network is

$$C = \frac{1}{2}\|W\|^2 + \frac{1}{2}\|V\|^2 + \frac{1}{vN}\sum_{i=1}^{N}\max(0,\ \rho - \hat{y}_i) - \rho \tag{13}$$

where $\hat{y}$ is the network output value. Then, the value of ρ is a network parameter, which can be understood as the radius of the hypersphere. It is updated according to the solution of Equation (8). In the experimental section, the original data are used as the network input instead of the features extracted from the autoencoder. This indicates that the proposed OC-LSTM is an end-to-end one-class classification method, which is different from other deep-learning-based anomaly detection methods. When the network parameters converge, the classification results of the original data can be obtained through the decision function.

Algorithm 1 OC-LSTM algorithm

Input: Set of training data x_i, $i = 1, \ldots, N$
Output: Set of decision scores $f(x_i)$, $i = 1, \ldots, N$
Initialize=1.0
$t \leftarrow 0$
While (convergence not achieved) do
    Optimize Equation (7) using BP, and find $(W^{t+1},\ V^{t+1})$
    $\rho^{t+1} \leftarrow$ the v^{th} quantile of $\left\{\hat{y}_i^{t+1}\right\}_{i=1}^{N}$
    $t \leftarrow t + 1$
End while
Compute decision scores $S_i = \hat{y}_i - \rho$ for each x_i
If$(S_i \geq 0)$ **then**
    x_i is the normal data
else
    x_i is the anomalous data
Return$f(x_i) = sign(\hat{y}_i - \rho)$, $i = 1, \ldots, N$

4. Experimental Setup

This section introduces the experimental setup in detail, including the data used, the compared methods, and the detailed experimental implementation.

4.1. Compared Methods

In this study we selected three different types of detection algorithms—the shallow model, deep model, and the OC-LSTM algorithm presented in this paper. For each type, three representative and novel algorithms were selected for comparison, all of which are widely used in the task of anomaly detection, and these are described in detail below.

4.1.1. Shallow Baseline Models

(1) *OC-SVM/SVDD* as per the formulation in [6]. When the Gaussian kernel function is used, these two methods are equivalent and are asymptotically consistent density-level set estimators. For the OC-SVM/SVDD models, the kernel size is the reciprocal of the number of features, and the fraction of outliers $v \in (0,\ 1)$ is set according to the obtained outlier proportions.
(2) *Isolation Forest (IF)* as per the formulation in [27]. The amount of contamination is set according to the proportion of outliers in the dataset, and the number of base estimators in the ensemble is 100, as recommended in [24].
(3) *Kernel Density Estimation (KDE)* as per the formulation in [28]. The bandwidth of the Gaussian kernel is selected from $h \in \left\{2^{0.5},\ 2^{1},\ \ldots,\ 2^{5}\right\}$ using the log-likelihood score, and the best result is reported.

4.1.2. Deep Baseline Models

(1) *Deep Convolution Autoencoder (DCAE)* as per the formulation in [29]. The encoder part and decoder part of the DCAE architecture contain four portions. Every portion contains a convolutional layer, a batch normalization (BN) layer, an exponential linear unit (ELU) layer, and a down-sampling/up-sampling layer, which can provide a better representation of the convolutional filters. The DCAE is trained using a mean-squared error (MSE) loss function to enable the hidden layers to encode high-quality, nonlinear feature representations of the input data.

(2) *One-class neural network (OC-NN)* as per the formulation in [21]. A feed-forward neural network consisting of a single hidden layer with linear activation functions is trained with the OC-NN objective. The optimal value of the parameter $v \in (0, 1)$, which is equivalent to the percentage of anomalies in each dataset, is set according to the respective outlier proportions.

(3) *Soft-Bound SVDD* and *One-Class Deep SVDD* as per the formulation in [24]. For the encoder, we employed the same network architectures as those used for the DCAE models. An initial learning rate $\eta = 10^{-4}$ with a two-phase learning rate schedule was employed following the implementation used in [24].

4.1.3. One-Class LSTM (OC-LSTM)

Unlike one-class deep SVDD and other deep baseline models, OC-LSTM is an end-to-end training network and does not rely on a pretrained model. The original data were fed as input to a feed-forward neural network consisting of a single LSTM layer, along with linear activation functions, as recommended in [20] for producing the best results. The number of hidden units within each LSTM cell of the model $h \in \{32, 64, 128, 256\}$ was tuned via a grid search. Note that the main reason for using an LSTM layer instead of other types of layers is that the experiment in [21] proved that LSTM is a better feature extractor for structured data. The extracted features were then fed into a classification network that was characterized by a fully connected neural network. The fully connected layer (followed by a softmax regression layer) assigned a confidence score to each feature representation and the output of the classification network was the confidence score. The learning and drop-out rates were sampled from a uniform distribution in the range [0.01, 0.001] in order to obtain the best performance. The optimal value of the parameter $v \in [0, 1)$ was set according to the respective outlier proportions. The Adam optimizer was used to minimize the squared error loss, and the other parameters were set to their default values. The entire network was trained end-to-end using the loss function as described in Equation (6).

4.2. Datasets

Although the proposed method applies to any feature representation context, our main concern is large-scale network traffic data. We compared all methods on three widely used real-world datasets, as summarized in Table 1. Each of the data sets was further processed to create a good anomaly detection task, as described in the next section.

Table 1. Summary of the datasets used in the experiments.

Datasets		Instances	Anomalies	Features
NSL-KDD	KDDTrain+	67,343 (53.46%)	58,630 (46.54%)	41
	KDDTest+	9711 (43.08%)	12,833 (56.92%)	41
CIC-IDS 2017	CIC-DDoS	97,718 (43.29%)	128,027 (56.71%)	78
	CIC-PortScan	127,537 (44.52%)	158,930 (55.48%)	78
MAWILab	MAWILab-20200102	22,228,204 (97.06%)	673,825 (2.94%)	109
	MAWILab-20201203	25,195,651 (96.80%)	832,028 (3.20%)	109

(1) The NSL-KDD dataset is the benchmark dataset in the field of network security, and it can provide consistent and comparable evaluation results for different research

works [30]. Furthermore, the number of records in the NSL-KDD datasets is reasonable, and each record contains 41-dimensional features. The anomaly data mainly include thirty-nine types of network attacks across four categories.

(2) The CIC-IDS2017 dataset contains benign data and the most up-to-date common attacks, so it resembles true real-world data [31]. The data contain a total of five days of network traffic in July 2017. The implemented attacks include Brute-Force FTP, Brute Force SSH, DoS, Heartbleed, Web Attack, PortScan, Infiltration, Botnet, and DDoS. In this study we randomly selected Friday's traffic as the experimental dataset, which contained two types of abnormalities, DDoS and PortScan.

(3) MAWILab is a database that assists researchers in evaluating traffic anomaly detection methods [32,33]. The dataset has been updated daily since 2001 to include new traffic from upcoming applications and anomalies; this has been ongoing for over 20 years. In this study, the data collected on the first collection day in January and December 2020 were selected as the experimental dataset. We used the code provided in [1] to process the original network traffic and extract the inherent features. The processed data contained 109-dimensional features, and the data distribution is shown in the table above.

4.3. Evaluation Criteria

Anomaly detection is an unsupervised learning problem, and model evaluation in this scenario is challenging. To effectively measure and compare the performances of different models, the area under curve (AUC) was chosen as the main evaluation index, as it is the most commonly used metric for one-class problems. The AUC is defined as the area under the receiver operating characteristic (ROC) curve, which is independent of the selected threshold and can provide an assessment of the overall performance of the given classification model.

However, when the sample proportion changes, the insensitivity of the AUC makes it difficult to observe the changes in model performance. The precision-recall curve (PRC) is a useful measure of prediction success when the classes are very imbalanced. In practical applications, the data are usually severely unbalanced, so the PRC can help to understand the actual effects of the classifier and then improve and optimize the model on this basis. Therefore, in the experiment, the ROC was used to judge the advantages and disadvantages of the given classifier, whereas the results displayed by the PRC were used to measure the classifier's ability on unbalanced data.

5. Experimental Results and Discussion

In this section, the empirical results produced by the proposed OC-LSTM network on real-world datasets are presented and compared with those of several state-of-the-art baseline models, as illustrated in Section 4. For all datasets, all anomaly samples from the training set were removed for one-class training, and two standard metrics, the AUPRC and AUROC, were used to evaluate the predictive performance of each method according to the ground truth labels. Publicly available implementations in scikit-learn were used for the OC-SVM, IF, and KDE methods. For the DCAE, OC-NN, soft-bound deep SVDD, and one-class deep SVDD, the codes released by their respective authors were used for comparison. For our proposed OC-LSTM, TensorFlow [34] and Keras were used for the experiment. All the experimental results are the average performances obtained with the 10-fold cross-validation method.

5.1. One-Class Classification on NSL-KDD

NSL-KDD contained four different anomaly categories from which we could construct a one-class classification dataset. One of the categories was abnormal, and the samples of the other categories represented normal data. We used the original training and test split in the experiment, and only performed training with the training set examples from the respective normal class. We preprocessed all records with numeralization using a one-hot

encoder, rescaled them to [0, 1] via min-max-scaling, and finally obtained 121-dimensional experimental data. Figure 1 shows the error graph of the abnormal detection performance of different models. The optimizer used in the DCAE model was the stochastic gradient descent (SOD) recommended by the original author, so the performance of the model was not stable. The AUROC and AUPRC indicators in the detection results variances were large. It can be clearly seen from Figure 3 that the method proposed in this paper achieved excellent performance, reaching 96.86% ± 0.58% and 96.72% ± 0.34% on the AUROC and AUPRC metrics, respectively. In addition, the anomaly detection performance of the deep models was generally better than that of the shallow models.

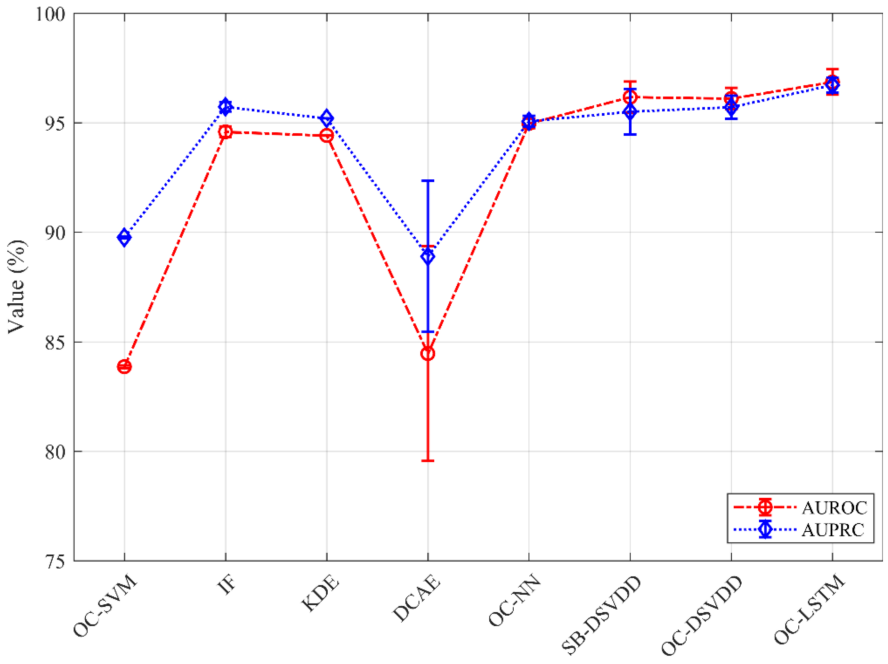


Figure 3. Anomaly detection performance of each model on NSL-KDD.

In order to evaluate the detection ability of each model for different anomaly (attack) types, in this study we combined normal types with different anomaly types to construct different single-classification tasks. Figure 4 shows the detection performance of different models for various types of anomalies, in which the R2L class of anomaly was difficult to identify, and the AUROC performance of the model was generally low. In addition, due to the serious unevenness in the number of normal and abnormal samples in the test set, the proportion of R2L abnormal samples was about 10%, whereas the proportion of U2R abnormal samples was less than 1%, so the AUPRC of each model in the abnormal types R2L and U2R was relatively poor, as shown in Figure 4c,d. However, the OC-LSTM model proposed in this chapter is stable in all four types of anomalies and significantly outperforms the detection performance of other shallow and deep single-classification models. AUROC indicators are 97.85% ± 0.23, 98.55% ± 0.15, 91.86% ± 1.33 and 98.01% ± 0.34, respectively. AUPRC indicators are 96.42% ± 0.37, 91.38% ± 0.96, 65.18% ± 3.84 and 42.24% ± 5.98, respectively. All experimental results were convincing, with shallow benchmark methods outperforming some of the deep models in the detection performance of individual anomaly types, whereas the deep single-class model showed more robust detection performance overall. It is worth noting that the detection performance of the shallow benchmark model IF, as shown in Figure 4b, was better than that of other deep single-classification models in relation to the Probe-type abnormalities.

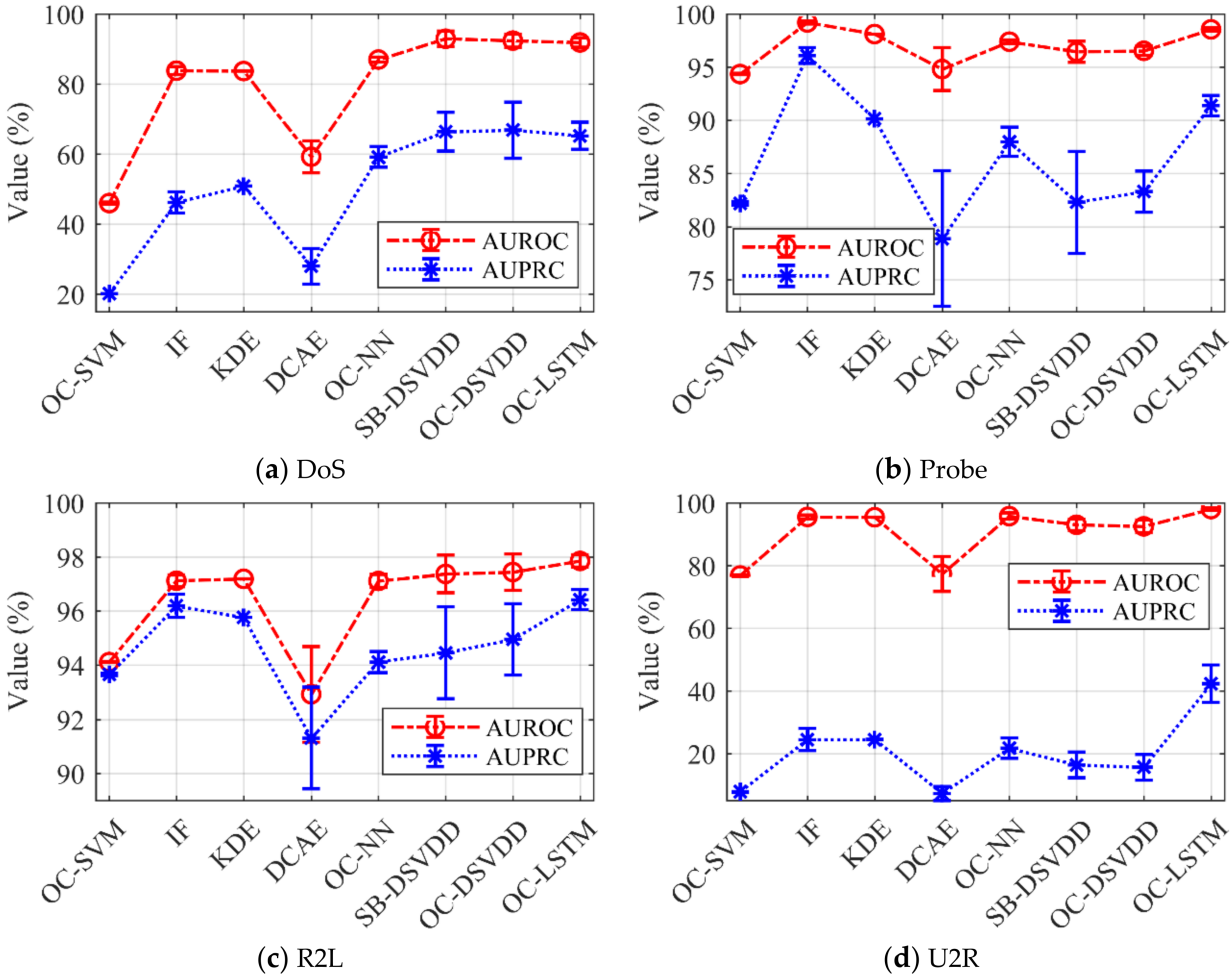


Figure 4. The detection performance of different models on NSL-KDD for various types of anomalies.

The results are presented in Tables 2 and 3, where "total" indicates all categories of attacks that were considered anomalous. Among these, the R2L anomaly class was difficult to identify, so the AUROC performance of each method was generally not good. In addition, due to the extremely uneven amount of positive and negative samples in the test set, the AUPRC performance for the R2L and U2R classes was particularly poor. However, the proposed OC-LSTM maintained high and stable detection performance in terms of both the AUROC and AUPRC, and it clearly outperformed both the shallow and deep competitors on NSL-KDD. These results are convincing, but for certain classes, the shallow baseline methods outperformed the deep models, whereas the deep models showed robust performances overall. It is interesting to note that the shallow IF method performed better than the deep methods on one of the four classes.

Table 2. Average AUROC values as percentages with StdDevs per method on NSL-KDD.

Anomaly Class	OC-SVM/ SVDD	IF	KDE	DCAE	OC-NN	Soft-Bound Deep SVDD	One-Class Deep SVDD	OC-LSTM
DoS	94.11 ± 0.02	97.12 ± 0.22	97.19 ± 0.00	92.92 ± 1.76	97.11 ± 0.24	97.37 ± 0.70	97.44 ± 0.67	**97.85 ± 0.23**
Probe	94.34 ± 0.07	**99.20 ± 0.13**	98.11 ± 0.00	94.81 ± 2.02	97.39 ± 0.16	96.44 ± 0.99	96.53 ± 0.44	98.55 ± 0.15
R2L	45.95 ± 0.28	83.80 ± 1.20	83.68 ± 0.03	59.19 ± 4.57	86.97 ± 0.70	**92.92 ± 2.22**	92.39 ± 1.86	91.86 ± 1.33
U2R	76.82 ± 0.17	95.51 ± 0.64	95.41 ± 0.01	77.29 ± 5.53	95.77 ± 1.00	93.00 ± 1.78	92.50 ± 2.05	**98.01 ± 0.34**
Total	83.86 ± 0.06	94.59 ± 0.25	94.42 ± 0.01	84.46 ± 4.91	94.97 ± 0.22	96.17 ± 0.71	96.11 ± 0.48	**96.86 ± 0.58**

Table 3. Average AUPRC values as percentages with StdDevs per method on NSL-KDD.

Anomaly Class	OC-SVM/ SVDD	IF	KDE	DCAE	OC-NN	Soft-Bound Deep SVDD	One-Class Deep SVDD	OC-LSTM
DoS	93.65 ± 0.04	96.19 ± 0.43	95.76 ± 0.00	91.31 ± 1.88	94.11 ± 0.39	94.45 ± 1.70	94.95 ± 1.32	**96.42 ± 0.37**
Probe	82.16 ± 0.18	**96.08 ± 0.74**	90.13 ± 0.01	78.88 ± 6.36	87.97 ± 1.37	82.28 ± 4.78	83.27 ± 1.94	91.38 ± 0.96
R2L	20.14 ± 0.09	46.14 ± 3.00	50.80 ± 0.06	28.00 ± 5.10	59.13 ± 2.97	66.34 ± 5.56	**66.82 ± 8.01**	65.18 ± 3.84
U2R	7.83 ± 0.06	24.42 ± 3.56	24.42 ± 0.04	7.20 ± 2.20	21.68 ± 3.28	16.34 ± 4.13	15.57 ± 4.10	**42.24 ± 5.98**
Total	89.77 ± 0.03	95.73 ± 0.22	95.20 ± 0.00	88.90 ± 3.45	95.07 ± 0.25	95.51 ± 1.04	95.71 ± 0.53	**96.72 ± 0.34**

5.2. One-Class Classification on CIC-IDS2017

Two sub-datasets from CIC-IDS2017, CIC-DDoS, and CIC-PortScan were selected as experimental data. We randomly selected 90% records from CIC-DDoS as the training set and the rest as the test set. At the same time, CIC-PortScan data were also used as a test set to measure the generalization abilities of the models, as this is quite important in the field of network security.

Table 4 illustrates that the OC-LSTM model performed significantly better than the existing state-of-the-art methods. It is evident that the ability of the OC-LSTM model to extract progressively rich representations of complex sequential data within the hidden LSTM layer of the feed-forward network induced better an anomaly detection performance. In addition, the LSTM model also showed great generalization performance on the CIC-PortScan dataset. In contrast, the generalization performance of conventional methods such as soft-bound deep SVDD and one-class deep SVDD was poor, which means that they may not be suitable for applications in complex and changeable large-scale network traffic anomaly detection tasks. Note that the IF and DCAE methods performed better on CIC-PortScan, mainly because of the difference caused by the different types of anomalies in the two datasets. In addition, due to the use of SGD optimization (as recommended in [24]), soft-bound and one-class deep SVDD exhibited higher standard deviations than those of the other methods. The PRC and ROC for each model on the CIC-DDoS dataset are shown in Figures 5 and 6, respectively.

Table 4. Average AUROC and AUPRC as percentages with StdDevs per method on CIC-IDS2017.

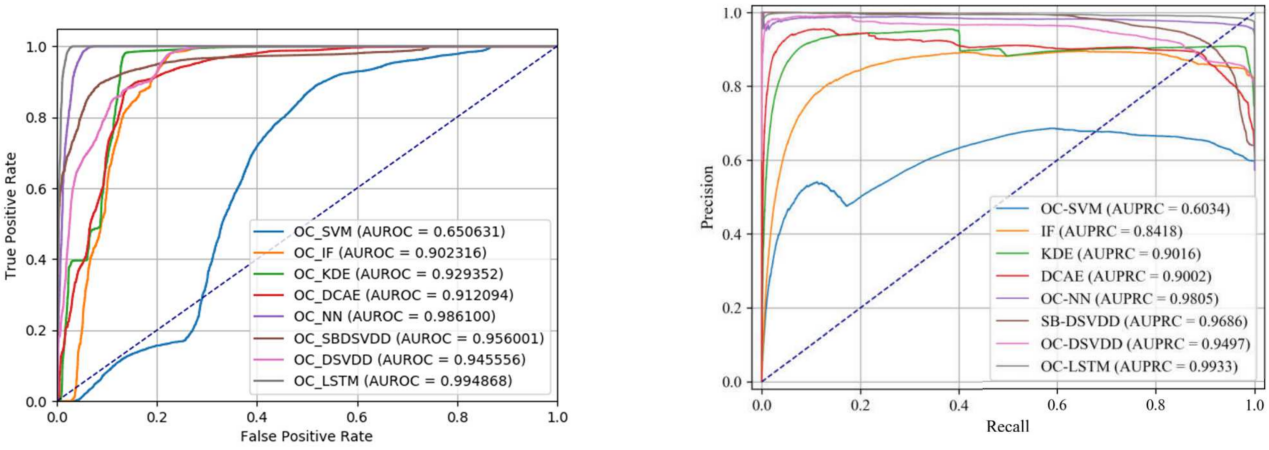
Method	CIC-DDoS		CIC-PortScan	
	AUROC	AUPRC	AUROC	AUPRC
OC-SVM/SVDD	66.46 ± 0.90	61.56 ± 0.79	71.92 ± 0.08	62.32 ± 0.06
IF	90.75 ± 1.92	84.95 ± 2.35	74.64 ± 5.42	67.11 ± 4.74
KDE	92.81 ± 0.34	89.78 ± 0.69	74.98 ± 0.20	67.79 ± 0.15
DCAE	90.62 ± 2.51	89.10 ± 5.12	94.75 ± 4.29	90.62 ± 7.37
OC-NN	99.07 ± 0.26	98.90 ± 0.41	98.53 ± 0.71	97.35 ± 1.48
Soft-Bound Deep SVDD	95.94 ± 2.83	96.56 ± 1.93	88.49 ± 7.78	82.47 ± 9.59
One-Class Deep SVDD	97.59 ± 1.12	97.72 ± 0.92	85.50 ± 4.91	79.25 ± 6.20
OC-LSTM	**99.24 ± 0.14**	**99.15 ± 0.16**	**99.66 ± 0.19**	**99.07 ± 0.39**

5.3. One-Class Classification on MAWILab

In the field of network security, people are mostly inclined to effectively detect anomalous traffic in large-scale networks in order to ensure privacy and security. In this experiment, we examined the performances of the proposed algorithms in detecting original network traffic. We considered the raw network traffic dataset MAWILab, for which we performed a series of preprocessing and feature engineering steps to obtain 109-dimensional structured data.

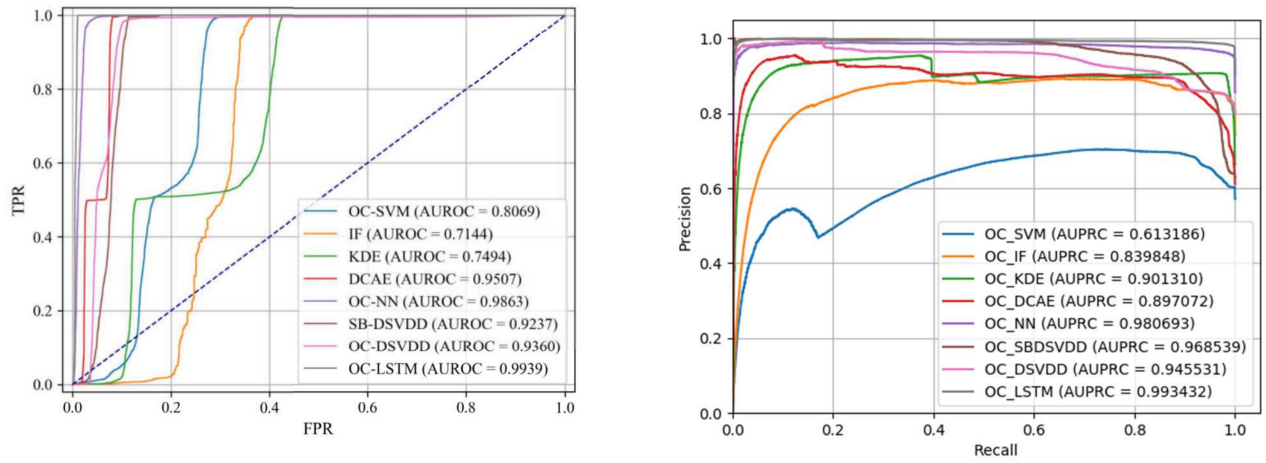
The experiment was performed on two data subsets separated by one year to measure the models' abilities to detect highly variable network anomalies. The ratio of the number of samples in the training set to the number of samples in the test set was nine to one, and the test set comprised anomaly instances with normal samples for the sake of having balanced data. Note that due to the large amount of data in the training set, the shallow baseline

models could not complete the training process in a reasonable time frame. Therefore, we randomly sampled the training set again and selected 10% of the data for the training of the shallow baseline models. This also reflects the superiority of the deep anomaly detection methods, that is, they can handle a large amount of complex data.



(**a**) ROC (**b**) PRC

Figure 5. The ROC and PRC for each model on the CIC-DDoS dataset.



(**a**) ROC (**b**) PRC

Figure 6. The ROC and PRC for each model on the CIC-PortScan dataset.

Table 5 presents the AUROC and AUPRC scores obtained by the various methods. The results on this dataset confirm that the performances of the deep methods were generally better than those of the shallow models, but individual shallow learning methods had more stable performance. The proposed OC-LSTM method undoubtedly outperformed all the deep models, with high AUROC and AUPRC values. Notably, the one-class deep SVDD method performed slightly better than its soft-boundary counterpart on both datasets.

Table 5. Average AUROC and AUPRC values as percentages with StdDevs per method on MAWILab.

Method	MAWILab-20200102		MAWILab-20201203	
	AUROC	**AUPRC**	**AUROC**	**AUPRC**
OC-SVM/SVDD	90.66 ± 0.23	83.59 ± 0.41	87.58 ± 0.30	80.19 ± 0.51
IF	94.58 ± 0.42	90.43 ± 0.86	89.35 ± 1.04	86.51 ± 1.29
KDE	92.17 ± 0.21	86.81 ± 0.55	89.81 ± 0.24	85.17 ± 0.31
DCAE	94.78 ± 1.04	92.49 ± 1.40	92.50 ± 1.31	90.45 ± 1.53
OC-NN	96.42 ± 0.30	94.90 ± 0.48	93.96 ± 0.21	92.71 ± 0.51
Soft-Bound Deep SVDD	94.89 ± 0.73	91.49 ± 1.46	90.96 ± 2.90	88.53 ± 1.60
One-Class Deep SVDD	95.00 ± 0.73	91.70 ± 1.26	91.79 ± 2.30	90.04 ± 1.80
OC-LSTM	**97.36 ± 0.49**	**96.70 ± 0.72**	**94.58 ± 0.47**	**93.23 ± 0.64**

5.4. The Anomaly Detection System

In order to verify the practicability of the above abnormal network traffic detection methods, our team designed a real-time dynamic monitoring system for network abnormality detection combined with a deep learning model and verified the effectiveness of the system in an ultra-large-scale high-speed network traffic environment.

The intelligent network anomaly detection real-time dynamic monitoring system used a browser/server (browser/server, B/S) structure. The system obtained traffic information by deploying security engines at various key points of the network. The system used an encrypted secure network to exchange information with the control center to achieve network data acquisition, analysis, and detection. At the same time, according to the results of the detection of abnormal traffic, the abnormal behavior that violates the security policy was merged and recorded or automatically filtered and blocked, and relevant information was sent to the control center in real time.

Based on the algorithm presented in this paper, the anomaly detection module in this system was constructed. The anomaly detection link was responsible for building a model to intelligently analyze the processed data and output the detection results. The network detection module implemented packet capture based on WinPcap and Tshark, and the feature engineering was implemented based on C++, and multi-threading was used to improve the processing speed of the system in the face of high-speed network traffic. The processed data contained 109-dimensional features. Based on the OC-LSTM algorithm proposed in this paper, the captured features were classified and trained, and then the classification model and prediction results were output and visualized. Among these, if the classification result was abnormal, a warning would be sent to the security center for blocking.

5.5. Discussion

In this study, we conducted experiments on three public datasets, and the proposed OC-LSTM algorithm achieved excellent results compared with other methods. As mentioned, on the NSL-KDD dataset, based on the total of all types of attacks, the method proposed in this paper achieved excellent performance, reaching 96.86% ± 0.58% and 96.72% ± 0.34% on the AUROC and AUPRC metrics, respectively. However, in the single-category classification, the IF algorithm obtained 99.20% ± 0.13% and 96.08% ± 0.74% on the AUROC and AUPRC metrics, respectively, when the data type was Probe. This shows that this shallow network model was not accurate in identifying all the data types. However, it achieved better results than our algorithm for the Probe data type. When the data type was R2L, the soft-bound deep SVDD algorithm achieved 92.92% ± 2.22% on AUROC, and the one-class deep SVDD achieves 66.82% ± 8.01% on AUPRC.

The OC-LSTM algorithm proposed in this paper achieved excellent results in relation to all data types on the other two data sets. However, there were also deficiencies in terms of specific data types. This shows that the algorithm still has room for improvement in identifying specific features. Furthermore, in designing a network anomaly detection system, to avoid this from happening, we adopted the form of an ensemble network. Some of the comparison methods and the OC-LSTM employed in this study were integrated

in the server, and different weights were assigned to them for joint training. In this way, the advantages of various methods could be integrated, thereby greatly improving the accuracy and usability of anomaly detection of the system.

6. Conclusions

In this study, a one-class LSTM (OC-LSTM) method was proposed for end-to-end anomaly traffic detection on large-scale networks, and it was trained using a loss function similar to the OC-SVM optimization target. The advantage of the OC-LSTM is that it constructs the hidden layer features for the special task of anomaly detection. The proposed approach is quite different from the recently proposed hybrid approach based on auto-encoders or pre-trained models, which use deep learning features as the input for the anomaly detector. A series of comprehensive experiments on three complex network security data sets were conducted to demonstrate the consistent ability of our method to work well on a variety of one-class classification applications, which proves that the proposed method has significantly better performance than the most existing state-of-the-art anomaly detection methods. In future work, we will continue to refine our model and system to improve the accuracy and usability of this anomaly detection approach.

Author Contributions: Conceptualization, Y.L. and Y.X. (Yingying Xu); methodology, Y.C. and X.L.; validation, J.H., C.W. and W.G.; writing—original draft preparation, Y.L.; writing—review and editing, Y.C. and Y.X. (Yang Xin); supervision, Z.L. and L.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Major Fundamental Research of Natural Science Foundation of Shandong Province under Grant ZR2019ZD05; Joint fund for Smart Computing of Shandong Natural Science Foundation under Grant ZR2020LZH013; open project of State Key Laboratory of Computer Architecture CARCHA202002; Qingdao Municipal Science and Technology Benefit People Demonstration and Guidance Special Project under Grant 21-1-4-sf-2-nsh; the Major Scientific and Technological Innovation Project in Shandong Province under Grant 2021CXG010506 and 2022CXG010504.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Chandola, V.; Banerjee, A.; Kumar, V. Outlier detection: A survey. *ACM Comput. Surv.* **2007**, *14*, 15.
2. Zhou, C.R.; Paffenroth, C. Anomaly detection with robust deep autoencoders. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017; pp. 665–674.
3. Rodríguez-Ruiz, J.; Mata-Sánchez, J.I.; Monroy, R.; Loyola-González, O.; López-Cuevas, A. A one-class classification approach for bot detection on Twitter. *Comput. Secur.* **2020**, *91*, 101715. [CrossRef]
4. Perera, P.; Patel, V.M. Learning deep features for one-class classification. *IEEE Trans. Image Process.* **2019**, *28*, 5450–5463. [CrossRef] [PubMed]
5. Li, D.; Deng, L.; Lee, M.; Wang, H. IoT data feature extraction and intrusion detection system for smart cities based on deep migration learning. *Int. J. Inf. Manag.* **2019**, *49*, 533–545. [CrossRef]
6. Schölkopf, B.; Platt, J.C.; Shawe-Taylor, J.; Smola, A.J.; Williamson, R.C. Estimating the support of a high-dimensional distribution. *Neural. Comput.* **2001**, *13*, 1443–1471. [CrossRef] [PubMed]
7. Tax, D.M.; Duin, R.P. Support vector data description. *Mach. Learn.* **2004**, *54*, 45–66. [CrossRef]
8. Mishra, P.; Varadharajan, V.; Tupakula, U.; Pilli, E.S. A Detailed Investigation and Analysis of Using Machine Learning Techniques for Intrusion Detection. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 686–728. [CrossRef]
9. Nisioti, A.; Mylonas, A.; Yoo, P.D.; Katos, V. From Intrusion Detection to Attacker Attribution: A Comprehensive Survey of Unsupervised Methods. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 3369–3388. [CrossRef]
10. Cui, Z.; Xue, F.; Cai, X.; Cao, Y.; Wang, G.; Chen, J. Detection of Malicious Code Variants Based on Deep Learning. *IEEE Trans. Ind. Inform.* **2018**, *14*, 3187–3196. [CrossRef]
11. Li, Y.; Xu, Y.; Liu, Z.; Hou, H.; Zheng, Y.; Xin, Y.; Zhao, Y.; Xin, Y.; Zhao, Y.; Cui, L. Robust detection for network intrusion of industrial IoT based on multi-CNN fusion. *Measurement* **2020**, *154*, 107450. [CrossRef]

12. Chalapathy, R.; Chawla, S. Deep learning for anomaly detection: A survey. *arXiv* **2019**, arXiv:1901.03407.
13. Aldweesh, A.; Derhab, A.; Emam, A.Z. Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues. *Knowl. Based Syst.* **2020**, *189*, 105124. [CrossRef]
14. Khan, S.S.; Madden, M.G. One-class classification: Taxonomy of study and review of techniques. *Knowl. Eng. Rev.* **2014**, *29*, 345–374. [CrossRef]
15. Krawczyk, B.; Galar, M.; Woźniak, M.; Bustince, H.; Herrera, F. Dynamic ensemble selection for multi-class classification with one-class classifiers. *Pattern Recogn.* **2018**, *83*, 34–51. [CrossRef]
16. Wan, M.; Shang, W.; Zeng, P. Double Behavior Characteristics for One-Class Classification Anomaly Detection in Networked Control Systems. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 3011–3023. [CrossRef]
17. Chalapathy, R.; Khoa, N.L.D.; Chawla, S. Robust Deep Learning Methods for Anomaly Detection. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual Event, CA, USA, 6–10 July 2020; pp. 3507–3508.
18. Xu, Y.; Liu, Z.; Li, Y.; Hou, H.; Cao, Y.; Zhao, Y.; Guo, W.; Cui, L. Feature data processing: Making medical data fit deep neural networks. *Future Gener. Comput. Syst.* **2020**, *109*, 149–157. [CrossRef]
19. Muhammad, S.; Cheol-Hong, K.; Jong-Myon, K. A hybrid feature model and deep-learning-based bearing fault diagnosis. *Sensors* **2017**, *17*, 2876.
20. Erfani, S.M.; Rajasegarar, S.; Karunasekera, S.; Leckie, C. High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning. *Pattern Recogn.* **2016**, *58*, 121–134. [CrossRef]
21. Chalapathy, R.; Menon, A.K.; Chawla, S. Anomaly detection using one-class neural networks. *arXiv* **2018**, arXiv:1802.06360.
22. Chalapathy, R.; Menon, A.K.; Chawla, S. Robust, deep and inductive anomaly detection. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases Springer, Skopje, Macedonia, 18–22 September 2017; pp. 36–51.
23. Chen, J.; Sathe, S.; Aggarwal, C.; Turaga, D. Outlier detection with autoencoder ensembles. In Proceedings of the 2017 SIAM International Conference on Data Mining, Houston, TX, USA, 27–29 April 2017; pp. 90–98.
24. Ruff, L.; Vandermeulen, R.; Goernitz, N.; Deecke, L.; Siddiqui, S.A.; Binder, A.; Kloft, M. Deep one-class classification. In Proceedings of the International Conference on Machine Learning, Vienna, Austria, 10–15 July 2018; pp. 4393–4402.
25. Oza, P.; Patel, V.M. One-Class Convolutional Neural Network. *IEEE Signal Proc. Let.* **2019**, *26*, 277–281. [CrossRef]
26. Schlachter, P.; Liao, Y.; Yang, B. Deep one-class classification using intra-class splitting. *arXiv* **2019**, arXiv:1902.01194.
27. Liu, F.T.; Ting, K.M.; Zhou, Z. *Isolation Forest*; IEEE: Piscataway, NJ, USA, 2008; pp. 413–422.
28. Parzen, E. On estimation of a probability density function and mode. *Ann. Math. Stat.* **1962**, *33*, 1065–1076. [CrossRef]
29. Masci, J.; Meier, U.; Cireşan, D.; Schmidhuber, J. *Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 52–59.
30. Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A.A. *A Detailed Analysis of the KDD CUP 99 Data Set*; IEEE: Piscataway, NJ, USA, 2009; pp. 1–6.
31. Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSp* **2018**, *1*, 108–116.
32. Fontugne, R.; Borgnat, P.; Abry, P.; Fukuda, K. Mawilab: Combining diverse anomaly detectors for automated anomaly labeling and performance benchmarking. In *Proceedings of the 6th International Conference*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 1–12.
33. Kim, C.; Sim, J.; Choi, J. Generating labeled flow data from MAWILab traces for network intrusion detection. In Proceedings of the ACM Workshop on Systems and Network Telemetry and Analytics, Phoenix, AZ, USA, 25 June 2019; pp. 45–48.
34. Abadi, M. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv* **2016**, arXiv:1603.04467.