

Article

# SOINN Intrusion Detection Model Based on Three-Way Attribute Reduction

Jing Ren <sup>1</sup>, Lu Liu <sup>1</sup>, Haiduan Huang <sup>2,\*</sup> , Jiang Ma <sup>1</sup>, Chunying Zhang <sup>1,3,4</sup>, Liya Wang <sup>1,3,4</sup>, Bin Liu <sup>5</sup> and Yingna Zhao <sup>6,7</sup>

<sup>1</sup> College of Science, North China University of Science and Technology, Tangshan 063210, China; renjing@ncst.edu.cn (J.R.); liulu\_hblg@ncst.edu.cn (L.L.); majing@ncst.edu.cn (J.M.); zchunying@ncst.edu.cn (C.Z.); wangliya@ncst.edu.cn (L.W.)

<sup>2</sup> College of Qian'an, North China University of Science and Technology, Tangshan 063210, China

<sup>3</sup> Tangshan Intelligent Industry and Image Processing Technology Innovation Center, North China University of Science and Technology, Tangshan 063210, China

<sup>4</sup> Hebei Engineering Research Center for the Intelligentization of Iron Ore Optimization and Ironmaking Raw Materials Preparation Processes, North China University of Science and Technology, Tangshan 063210, China

<sup>5</sup> Big Data and Social Computing Research Center, Hebei University of Science and Technology, Shijiazhuang 050091, China; liubin@hebust.edu.cn

<sup>6</sup> College of Material Science and Engineering, North China University of Science and Technology, Tangshan 063210, China; zhyn@ncst.edu.cn

<sup>7</sup> Hebei Province Laboratory of Inorganic Nonmetallic Materials, Tangshan 063210, China

\* Correspondence: huanghaiduan@ncst.edu.cn

**Abstract:** With a large number of intrusion detection datasets and high feature dimensionality, the emergent nature of new attack types makes it impossible to collect network traffic data all at once. The modified three-way attribute reduction method is combined with a Self-Organizing Incremental learning Neural Network (SOINN) algorithm to propose a self-organizing incremental neural network intrusion detection model based on three-way attribute reduction. Attribute importance is used to perform attribute reduction, and the data after attribute reduction are fed into a self-organized incremental learning neural network algorithm, which generalizes the topology of the original data through self-organized competitive learning. When the streaming data are transferred into the model, the inter-class insertion or node fusion operation is performed by comparing the inter-node distance and similarity threshold to achieve incremental learning of the model streaming data. The inter-node distance value is introduced into the weight update formulation to replace the traditional learning rate and to optimize the topological structure adjustment operation. The experimental results show that T-SOINN achieves high precision and recall when processing intrusion detection data.

**Keywords:** intrusion detection; incremental learning; attribute reduction; self-organizing neural networks



**Citation:** Ren, J.; Liu, L.; Huang, H.; Ma, J.; Zhang, C.; Wang, L.; Liu, B.; Zhao, Y. SOINN Intrusion Detection Model Based on Three-Way Attribute Reduction. *Electronics* **2023**, *12*, 5023. <https://doi.org/10.3390/electronics12245023>

Academic Editor: Andrei Kelarev

Received: 9 September 2023

Revised: 2 December 2023

Accepted: 7 December 2023

Published: 15 December 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

In the current 5G era, the Internet penetration rate is rapidly rising, the network environment is becoming more complex, and the network security problem is becoming increasingly serious. Traditional firewalls, data encryption, and other network security protection methods have had difficulty coping with the needs of network security protection [1]. Continuous improvements to the defensive capabilities of networks have become an increasingly important hot topic. Intrusion detection enhances a network's security defense capabilities by collecting information in the network, detecting abnormal behavior, and identifying dangers and alarms. The working principle of the intrusion detection system is to classify the collected network traffic data and to divide them into normal or abnormal types, in which the abnormal-type data can be refined into different attack categories [2]. The field of machine learning has been growing rapidly in recent years, and

a large number of machine learning algorithms have been applied to intrusion detection systems, such as support vector machine [3], decision tree [4], and naive Bayes [5].

However, with the advent of the era of big data, these traditional machine learning algorithms are having difficulty handling intrusion detection. Deep learning has shown a good feature extraction ability in the face of massive and high-dimensional data. Therefore, some scholars have applied deep learning algorithms such as convolutional neural network [6] and recurrent neural network to intrusion detection. For example, Henry [7] combined convolutional neural networks with a gated recurrent unit for intrusion detection, used different CNN-GRU combination sequences to optimize parameters, improved the dependence of deep learning on feature learning, and achieved a good classification effect of intrusion detection. Chen [8] conducted research experiments on recurrent neural networks, used the window-based instance selection algorithm to further streamline the data, then adjusted the parameters in the model, and finally used them for intrusion detection to improve the overall performance of intrusion detection. Compared with the above two kinds of neural networks, the self-organizing neural network can automatically adjust the structure and parameters of the network to adapt to the data mode and changes according to the distribution and characteristics of the intrusion detection data and can dynamically learn and adjust its own representation ability. Therefore, some scholars have considered the use of self-organizing incremental learning to handle intrusion detection data. Li [9] combined the possibility theory and SOINN algorithm to define the possibility membership to judge the sample category and formed the SOINN cluster input probability neural network to obtain the PS-PNN network, which finally improved the accuracy and recall rate of the intrusion detection system.

While self-organizing incremental neural networks can handle massive and high-dimensional dynamic data, this does not mean that the more data and attributes, the better. The attributes of many datasets are not conducive to the detection of network intrusion outcomes. At this point, an indicator, namely attribute importance, is used to select the attribute, which can detect the degree of association between the intrusion detection results and the attribute. Attribute importance is an indicator used in a rough set to measure the degree of correlation between conditional attributes and decision attributes and is often used for decision tree node division, such as in Wang [10], where the heuristic function was defined for decision tree division criteria. Attribute importance is also often used for attribute reduction. For example, in order to improve the operation efficiency and accuracy of a classification algorithm, Zhou [11] proposed a fast attribute reduction algorithm based on improving the importance of a neighborhood rough set, and Zheng [12] gave an evaluation strategy of attribute importance and designed an algorithm by integrating the multi-fork tree theory. All the above studies used attribute importance to reduce the data, and the results were significantly improved.

In the aforementioned studies, only properties in the positive domain were considered in the calculation of attribute importance, and properties in the boundary domain were not considered. The attributes in the boundary domain also contributed to the decimation result and it is obviously not appropriate to delete them directly, so the attributes in the boundary domain need to be taken into account. The decision boundary entropy proposed by Zhang [13] can not only consider the information in the boundary domain but also consider the information entropy, which can measure the uncertain information. Then, they defined a new measure of attribute importance, finally proposed a three-way attribute random selection algorithm to handle the dataset in random forest, and achieved good results in the intrusion detection datasets [14]. Based on the above analysis, the algorithm process was improved, and the three-attribute reduction algorithm was proposed and combined with the self-organized incremental neural network. Finally, the T-SOINN algorithm was obtained. The precision and recall, which are commonly used evaluation indicators of intrusion detection, were selected to validate the performance of the model. The results validate the advantages and disadvantages of T-SOINN.

The contributions of this article are as follows:

- The three-way attribute selection algorithm is improved, and a new attribute reduction method is proposed;
- The distance between nodes is introduced into the weight update formula, so that the weight update value is adjusted with the distance between neurons, to optimize the dynamic update operation of the topology structure;
- SOINN is used for network intrusion detection to obtain an incremental intrusion detection model and combined with three-way attribute reduction for the T-SOINN algorithm.

The structure of this paper is as follows: Section 1 provides an introduction. Section 2 provides the related work on incremental learning, attribute reduction, and three-way decisions. Section 3 briefly introduces incremental learning, the self-organization of an incremental neural network, and attribute reduction. Section 4 introduces the T-SOINN algorithm in detail. Section 5 presents the experimental verification algorithm, and finally, Section 6 provides a summary.

## 2. Related Work

### 2.1. Incremental Learning

The idea of incremental learning was first introduced in the 1980s. In 2001, Polikar [15] published the incremental learning algorithm of Learn++ and elaborated on the concept of incremental learning. The idea of incremental learning is a simulation of humans' learning ability and is a solution to some real-life problems [16]. Incremental learning is often used in image classification [17], target detection [18], and speech recognition [19]. In the field of network intrusion detection, not only is the volume of intrusion data large but also the categories are constantly updated, resulting in data that cannot be collected completely all at once. Incremental learning can not only continuously learn new knowledge from new samples but also save the majority of the learned knowledge, so it is gradually applied to intrusion detection systems. According to the current real-time update problem of intrusion detection data, the traditional machine learning algorithms support vector machine and K nearest neighbor have been integrated, the idea of incremental learning has been added, and an expansion of the knowledge base has been considered [20]. Models incorporating incremental ideas can not only handle problems in real time but also significantly improve accuracy. Liu [21] applied the incremental GHSOM algorithm to DDoS detection, dynamically generated the network structure, constrained the quantification error, and deleted the immature subnetwork structure in the process of continuous sample input. While this model achieves a high detection rate on static data, it also shows good detection performance on dynamic data.

Yang [22] used the incremental Growth-type Hierarchical Self-organization Mapping Neural Network model for intrusion detection and obtained the initial network structure by using static data training, and then, the dynamic data were input into the network structure and constantly adjusted to realize incremental learning. Liu [23] used stacked sparse autoencoders to realize sparse coding and feature extraction of known class attacks. SOINN used adaptive clustering to form the topological representation of original features and dynamically adjusted the topological structure to adapt to new data. This model effectively improves the accuracy of the incremental learning intrusion detection model.

### 2.2. Attribute Reduction

Although large amounts of dynamic data can be processed by incremental learning, this does not mean that more data are better and more properties are better. As the datasets and dimensions of intrusion detection increase, the neural network structure will become more and more complex and the difficulty of dataset training will gradually increase during the experiment. Therefore, some scholars have considered reducing the dimensionality of the original dataset when processing the data, and then training and classifying them in combination with related algorithms. Laghrissi [24] used the principal component analysis algorithm to reduce the original data, used mutual information to select the features, and finally used long- and short-term memory network classification to achieve high accuracy

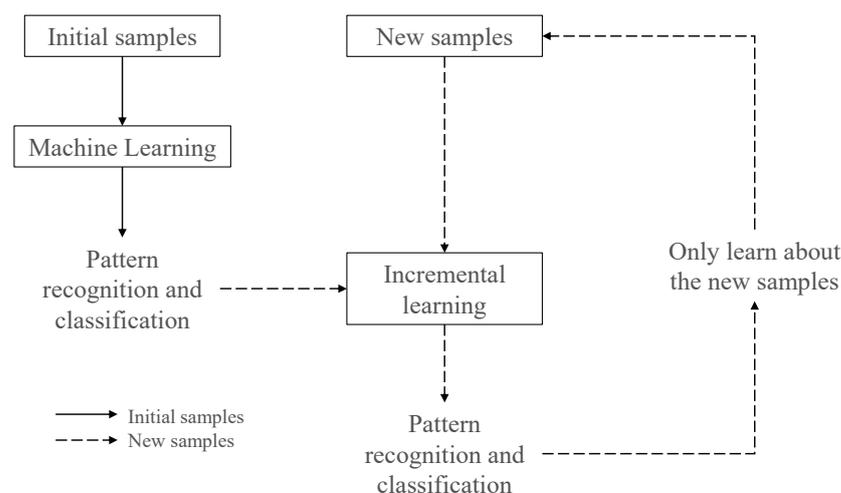
on the basis of improving the training time. In [25], decision tree was used to distinguish the original data, thus reducing the overall data amount; then, PCA was used to reduce the original data; and finally, the data were introduced into the improved DNN model for training. Luo [26] combined the genetic attribute reduction algorithm based on a rough set with the BP neural network, which effectively improved the convergence speed of the model, sped up the runtime, and significantly increased the detection rate.

Attribute reduction algorithms based on a classical rough set do not require prior knowledge. It is the most common approach for attribute reduction to find the rule of the problem by using only the information provided by the data itself. The attribute reduction algorithm based on the rough set includes the attribute reduction algorithm [27] based on the decision rough set model and the attribute reduction algorithm based on the attribute importance. The latter includes the attribute reduction algorithm based on attribute dependence [28], conditional information entropy [29], and mutual information [30]. There is uncertainty in the data information, and information entropy and approximate classification quality are measures of uncertain information. As a result, the attribute importance defined by combining information entropy, attribute dependence, and approximate classification quality is used to measure the attribute [13]. We improved the three-way attribute random selection algorithm in [13] to obtain a new reduction method, thus achieving the purpose of dimension reduction.

### 3. The Basics

#### 3.1. Incremental Learning

Incremental learning is a simulation of humans' learning ability and the solution to some real-life problems [16]. The use of incremental learning can improve some of the drawbacks of traditional machine learning: Traditional machine learning suffers from low recognition accuracy in the face of new data and is prone to forgetting old data when learning new data types. However, incremental learning, in which new knowledge is learned to selectively preserve and discard old knowledge, effectively improves the catastrophic forgetting problem. Furthermore, with current invasion detection datasets being large and have high characteristic dimensions, one-time complete learning will produce many problems, such as the high space requirements, the long learning time, new data being required after learning, the high consumption of data, etc.; however, incremental learning through dynamic partial read data training will further enhance the efficiency of the model. The incremental learning process [16] is shown in Figure 1.



**Figure 1.** Incremental learning process.

#### 3.2. Self-Organized Incremental Learning Neural Network

In 2006, Professor Shen Furao [31] first proposed the unsupervised clustering neural network model SOINN, which is a two-layer neural network structure with competitive

learning ability. The initial training data are received and the prototype neurons are generated adaptively using the first layer network structure. The second layer network then estimates the intra-class and inter-class distances based on the results from the first layer network. The prototype neurons are used as the initial data in this process, leading to the acquisition of a stable topological structure.

The SOINN algorithm uses a set of neurons distributed over the feature space to approximate the distribution of the input data [32], that is, the distribution of the original dataset is represented by a small number of neurons. During learning, neurons belonging to the same cluster are connected to form a topology similar to the form of a connected graph to represent the current distribution of the learned data. As new data continue to be input, the topology is adjusted and updated according to the distance between the new data and the surrounding neurons. When the difference between the new data and the original neuron is large, the inter-class insertion operation is performed to generate new categories, and when the difference is small, the node fusion operation is performed. Furthermore, noisy nodes are removed based on the number of winning neurons and the number of surrounding neighbor nodes, which ensures not only the incremental learning of SOINN but also the stability of the learning results.

### 3.3. Attribute Importance

In real datasets, the importance of different attributes may be different. The rough set method can be used to measure the importance of some attributes. Three-way decisions are used to select the attributes.

The decision table is  $DT = (Un, At, V, f)$ .  $Un$  is the universe, which is a non-empty finite set containing all instances.  $C = \{c_1, c_2, \dots, c_p\}$  denotes the conditional attributes,  $D = \{d_1, d_2, \dots, d_q\}$  denotes the decision attribute.  $At = C \cup D$  contains all attributes.  $V$  is a set of attribute values, and  $f : Un \times At \rightarrow V$  is a mapping. For any  $I \subseteq C \cup D$ , the indiscernibility relation  $IND(I)$  on  $Un$  is defined as  $IND(I) = \{(x, y) | (x, y) \in Un \times Un : \forall c \in I (f(x, c) = f(y, c))\}$ .

**Definition 1.** Given a decision table  $Dt = (Un, C \cup D, V, f)$ , for  $\forall I \subseteq C \cup D$  and  $X \subseteq Un$ , the lower and upper approximations of  $X$  are expressed as

$$\underline{X}_I = \cup\{[x]_I \in Un/IND(I) : [x]_I \subseteq X\} \tag{1}$$

$$\overline{X}_I = \cup\{[x]_I \in Un/IND(I) : [x]_I \cap X \neq \emptyset\} \tag{2}$$

$M^-_I(X) = \overline{X}_I - \underline{X}_I$  is called a boundary domain of  $X$  with respect to  $I$ , and  $L^-_I(X) = Un - \overline{X}_I$  is called the negative domain of  $X$  with respect to  $I$ .

**Definition 2.** (Approximate classification accuracy) Given a decision table  $Dt = (Un, C \cup D, V, f)$ , for  $\forall I \subseteq C \cup D$  and  $Un/IND(D) = \{D_1, D_2, \dots, D_q\}$ , the approximate classification accuracy  $\alpha_I(D)$  of  $D$  with respect to  $I$  is defined as

$$\alpha_I(D) = \frac{\left| \bigcup_{i=1}^m D_{iI} \right|}{\left| \bigcup_{i=1}^m \overline{D}_{iI} \right|} \tag{3}$$

Obviously, the approximate classification accuracy  $\alpha_I(D) \in [0, 1]$  describes the percentage of correct decisions among all possible decisions when classifying decision attribute  $D$  under  $I$ .  $\alpha_I(D) = 1$  means that  $I$  can completely describe decision attribute  $D$ .  $\alpha_I(D) = 0$  means that  $I$  cannot describe decision attribute  $D$  at all.

**Definition 3.** (Decision boundary entropy, DBE) [14] Given a decision table  $Dt = (Un, C \cup D, V, f), Un/IND(D) = \{D_1, D_2, \dots, D_q\}$ , for any  $I \subseteq C$ , the decision boundary entropy of  $D$  with respect to  $I$  is defined as

$$DBE(D, I) = (1 - \alpha_I(D)) \times \sum_{i=1}^m \frac{|M^-_I(D_i)|}{|Un|} \log_2 \left( \frac{|M^-_I(D_i)|}{|Un|} + 1 \right) \tag{4}$$

$\alpha_I(D)$  denotes the approximate classification accuracy, and  $M^-_I(D_i)$  denotes the boundary domain.

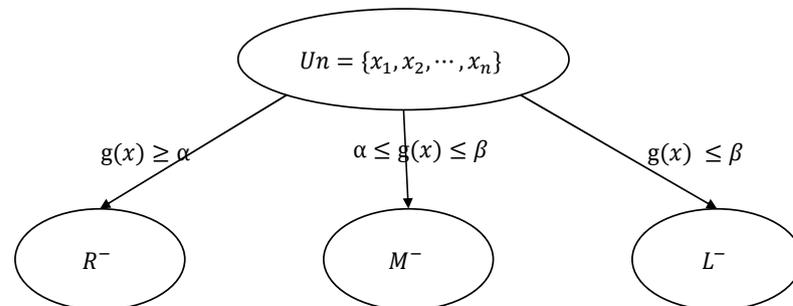
**Definition 4.** (DBE-based importance of an attribute) [14] Given a decision table  $Dt = (Un, C \cup D, V, f)$ , for any  $I \subset C, c \in C - I$ , the significance of attribute  $c$  with respect to  $I$  and  $D$  is defined as

$$IMP(c, I, D) = DBE(D, I - \{c\}) - DBE(D, I) \tag{5}$$

### 3.4. Three-Way Decisions

Three-way decisions based on the evaluation function give an uncertain division of element  $x \in X \subseteq Un$  in a certain target object set  $X$  according to the evaluation function  $g(x)$ . The, a threshold pair  $(\alpha, \beta)$ , where  $0 \leq \beta < \alpha \leq 1$ , is introduced. For any  $x \in X$ , when  $g(x) \geq \alpha$ , the element  $x$  is divided into the positive domain  $R^-(X)$  of the set  $X$ . When  $g(x) \leq \beta$ , the element  $x$  is divided into the negative domain  $M^-(X)$  of the set  $X$ . When  $\beta \leq g(x) \leq \alpha$ , the element  $x$  is divided into the delay domain  $L^-(X)$  of the set  $X$ . Then,  $X$  can be divided into three disjoint regions, as shown in Figure 2.

$$\begin{cases} R^- = \{x \in X \mid g(x) \geq \alpha\} \\ M^- = \{x \in X \mid \beta < g(x) < \alpha\} \\ L^- = \{x \in X \mid g(x) \leq \beta\} \end{cases} \tag{6}$$



**Figure 2.** Three-way decisions based on the evaluation function.

It should be noted that the function  $g$  evaluates the attributes, and by comparing parameters  $\alpha$  and  $\beta$ , the attributes are divided into the positive, boundary, and negative domains. This is explained in detail in Section 4.1.

## 4. T-SOINN Incremental Intrusion Detection Model

The T-SOINN model is divided into two parts: three-way attribute reduction and SOINN incremental clustering. The specific process is described as follows:

- (1) Preprocess the initial dataset: symbol digitization and K-means clustering discretization;
- (2) Calculate the importance of each attribute in the original dataset;
- (3) Set the attribute importance as the evaluation function and the threshold  $(\alpha, \beta)$  to realize three-way attribute reduction;
- (4) Select  $k/2$  attributes, and delete the rest to obtain a subset of the dataset;
- (5) Transfer the incremental dataset subset into SOINN for initialization;

- (6) Find the first and second winning neurons;
- (7) If  $d_1 > T_{s_1}$  or  $d_2 > T_{s_2}$ , insert new nodes;
- (8) Update the *age* values of any edges connected between two winning neurons;
- (9) Update the *age* values for all edges connected to the first winning neuron;
- (10) If  $d_1 < T_{s_1}$  and  $d_2 < T_{s_2}$ , update the weights of the two winning neurons;
- (11) Sum up the quantification error of the winning node;
- (12) Judge the *age* value of the connected edge, and delete redundant connections;
- (13) For each inputted  $\lambda$  sample, perform intra-class node insertion and denoising operations;
- (14) End the process.

The flow of the T-SOINN algorithm is shown in Figure 3.

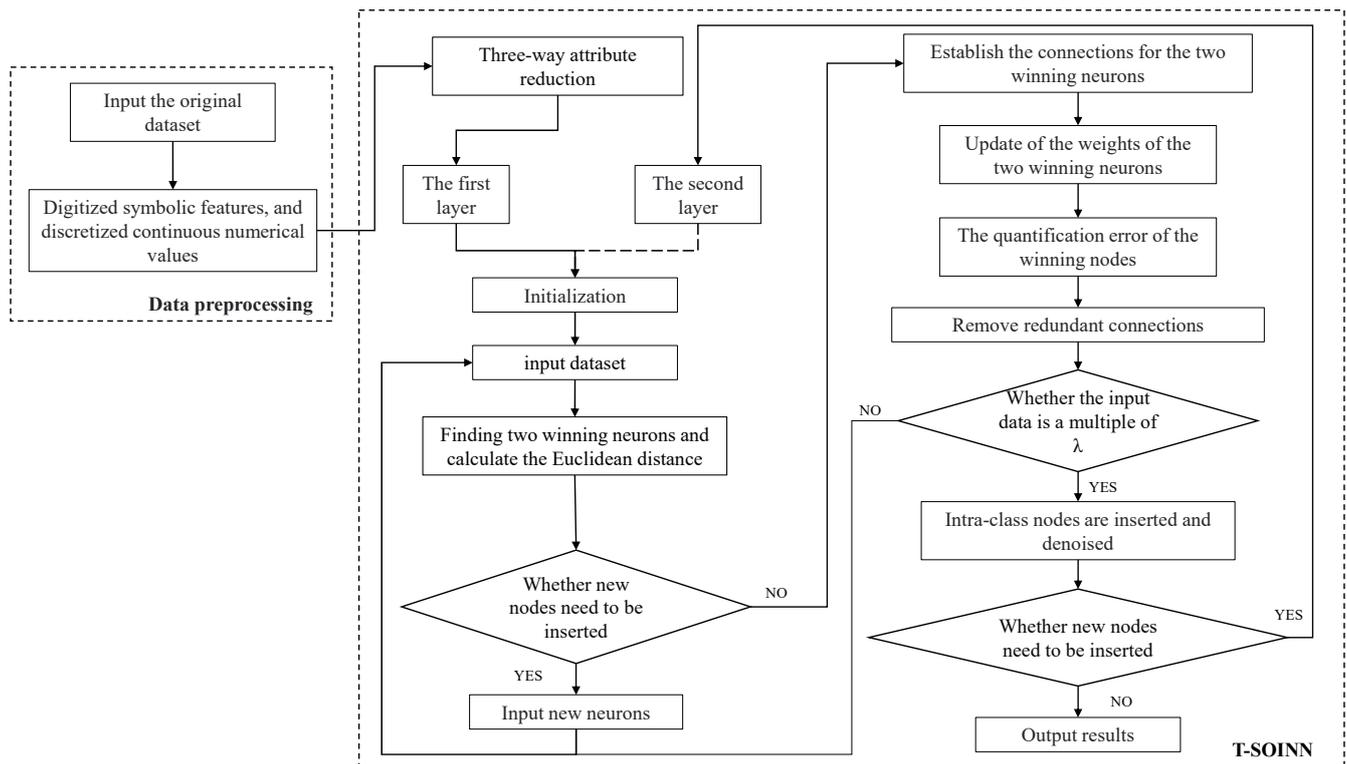


Figure 3. Flow chart of T-SOINN intrusion detection algorithm.

#### 4.1. Three-Way Attribute Reduction

The attribute importance based on DBE is set to the evaluation function in the three-way decisions that is  $g(c) = IMP(c, C, D)$ . The value of  $(\alpha, \beta)$  is then set for any  $c \in C$ :  $c$  is classified into the positive decision domain  $R^-$  if  $IMP(c, C, D) \geq \alpha$ ;  $c$  is classified into the negative decision domain  $L^-$  if  $IMP(c, C, D) \leq \beta$ ; and  $c$  is classified into the delay decision domain  $M^-$  if  $\alpha \leq IMP(c, C, D) \leq \beta$ .

Then,

$$\begin{cases} R^- = \{c \in C \mid IMP(c, C, D) \geq \alpha\} \\ M^- = \{c \in C \mid \beta < IMP(c, C, D) < \alpha\} \\ L^- = \{c \in C \mid IMP(c, C, D) \leq \beta\} \end{cases} \quad (7)$$

After dividing the attributes into three domains, the three-way attribute selection rules are defined as follows:

- (1)  $|R^-| > \frac{p}{2}$  is satisfied, the random  $\frac{p}{2}$  attributes in the positive domain are a subset of attributes.
- (2)  $|R^-| + |M^-| > \frac{p}{2}$  is satisfied, the random  $\frac{p}{2}$  attributes in the positive domain and the boundary domain are a subset of attributes.

- (3)  $|R^-| + |M^-| \leq \frac{p}{2}$  is satisfied, the  $|R^-| + |M^-|$  attributes in the positive and boundary domains, plus the random  $\frac{p}{2} - |R^-| - |M^-|$  attributes in the negative domain, constitute a subset of properties.

$p = |R^-| + |M^-| + |L^-|$  denotes the total number of attributes, and  $|R^-|$ ,  $|M^-|$ , and  $|L^-|$ , respectively, denote the number of attributes in the positive domain, the boundary domain, and the negative domain.

In the process of applying the three-way attribute selection rules, rule (1) is considered first, followed by rule (2) and rule (3).

The detailed steps of Three-way attribute reduction algorithm are shown in Algorithm 1.

---

**Algorithm 1** Three-way attribute reduction algorithm.

---

Input: Dataset  $Un = \{x_1, x_2, \dots, x_n\}$ , Decision table  $Dt = (Un, C \cup D, V, f)$ , Three-way attribute selection threshold  $(\alpha, \beta)$

Output: Dataset  $U^*$  containing attributes  $C'$

```

1:   Initialization  $R^- = \{\emptyset\}, M^- = \{\emptyset\}, L^- = \{\emptyset\}$ 
2:   For  $c \in C$  do
3:     Calculate  $g(c)$ 
4:     If  $g(c) > \alpha$  then
5:        $R^- \leftarrow c$ 
6:     Else
7:       If  $\beta < g(c) < \alpha$  then
8:          $M^- \leftarrow c$ 
9:       Else
10:         $L^- \leftarrow c$ 
11:      End if
12:    End if
13:  End for
14:  If  $|R^-| > p/2$  then
15:    draw  $p/2$  attributes from  $R^-$  to constitute an attributes subset  $C'$ 
16:  Else
17:    If  $|R^-| + |M^-| > p/2$  then
18:      draw  $\frac{p}{2}$  attributes from  $R^-$  and  $M^-$  to constitute an attributes subset  $C'$ 
19:    Else
20:      draw  $|R^-| + |M^-|$  attributes from  $R^-$  and  $M^-$ , draw  $\frac{p}{2} - |R^-| - |M^-|$  attributes
21:      from  $L^-$  to constitute an attributes subset  $C'$ .
22:    End if
23:  End if
24:   $U^*$  containing attributes  $C'$ 

```

---

When determining the number of attributes, if the number of attributes is too much, the effect of dimension reduction cannot be achieved; if the number of attributes is too small, the model accuracy may be affected. In reference [13], the authors take  $\sqrt{p}$  attributes to train a subtree and obtain a random forest. If it is obviously not appropriate to continue to select  $\sqrt{p}$  attributes, here, half of the original attributes are selected as the number of attributes after attribute selection.

#### 4.2. SOINN Incremental Clustering

- (1) The data subset after attribute reduction is received, and two samples are randomly selected to initialize the neuron set  $N = \{x_1, x_2\}$ , a set of neurons connected via the relationship  $M \subseteq N \times N$ . The two initialized neurons have no connection relationship, and their weights are, respectively,  $w_{x_1}, w_{x_2}$ .
- (2) Finding the winning neuron: set the newly received sample as  $\zeta$ , and the two most similar neurons to  $\zeta$  are sought as the first and second winning neurons,  $s_1, s_2$  respectively. The calculation formula are

$$s_1 = \operatorname{argmin}_{x \in N} |w_\zeta - w_{s_1}| \quad (8)$$

$$s_2 = \operatorname{argmin}_{x \in N \setminus \{x_1\}} |w_\zeta - w_{s_2}| \tag{9}$$

At this point, the winning number  $t_{s_1}$  of the first winning neuron is updated. If node  $s_1$  becomes the winning neuron for the first time, then  $t_{s_1}$  is denoted as 1; otherwise,  $t_{s_1}$  is added to 1.

- (3) The similarity thresholds  $T_{s_1}$  and  $T_{s_2}$  of the winning neurons  $s_1$  and  $s_2$  are then calculated. For any neuron  $\tau$ , if there is a connected neuron, it is proved that there is a neighbor neuron to this neuron and it forms a class with it. In this case, the similarity threshold is the maximum intra-class distance, and the calculation formula is

$$T_\tau = \max |w_\tau - w_j|, j \in N_\tau \tag{10}$$

If no neurons are connected to  $\tau$ , the similarity threshold is defined as the minimum inter-class distance, with the formula calculated as

$$T_\tau = \min |w_\tau - w_j|, j \in N \setminus \{\tau\} \tag{11}$$

- (4) According to the similarity threshold for performing the corresponding operation  $\zeta$ , the distance between  $\zeta$  and the first and second winning neurons are  $d_1$  and  $d_2$ , respectively. If  $d_1 > T_{s_1}$  or  $d_2 > T_{s_2}$ , the current  $\zeta$  differs greatly from the previously learned data, so  $\zeta$  is inserted into the new neurons to update the neuron ensemble  $N = N \cup \{\zeta\}$ . Next, return to step 2 to continue processing other new samples. The sample insertion diagram is shown in Figure 4.

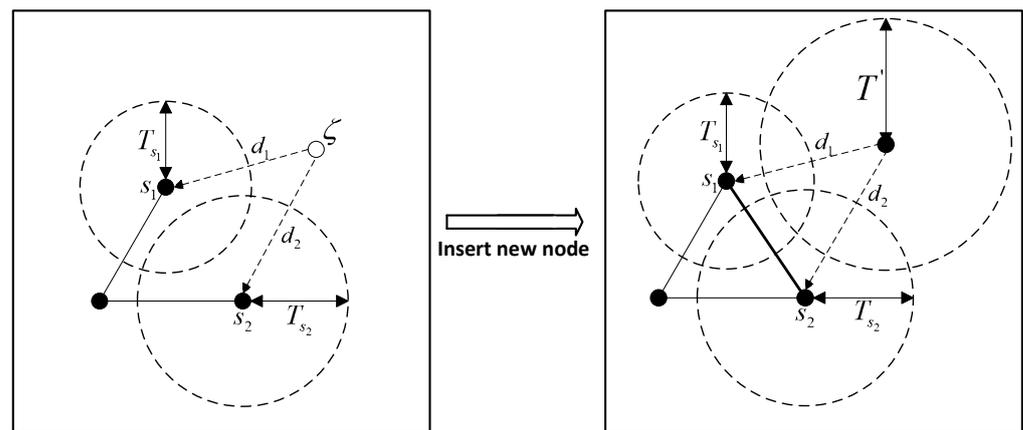


Figure 4. Inter-class insertion.

- (5) Update the parameter *age* for the connection edges. When the first and second winning neurons are activated simultaneously by the input sample  $\zeta$ , the two winning neurons are somewhat related. Thus, if there are connections between two winning neurons, then  $age_{s_1, s_2} + 1$ ; if there is no connection, the two winning neurons are connected and the parameter  $age_{s_1, s_2}$  is set to 1.

- (6) All edges connected to the first winning neuron have an *age* value plus one:

$$age_{s_1, i} = age_{s_1, i} + 1, \forall i \in N_i \tag{12}$$

- (7) The quantization error of the winning neuron is accumulated:

$$E_{s_1} = E_{s_1} + |\zeta - w_{s_1}| \tag{13}$$

- (8) If  $d_1 < T_{s_1}$  and  $d_2 < T_{s_2}$ , the current sample  $\zeta$  is similar to the neuron in topology and there is no need to generate new neurons. Next, we need to update the weights of the winning neuron based on  $\zeta$  and move toward  $\zeta$ , that is, the process of the fusing

sample  $\zeta$  into the topology. When the samples are fused, the winning neuron will move by updating its weights. A sample fusion diagram is shown in Figure 5.

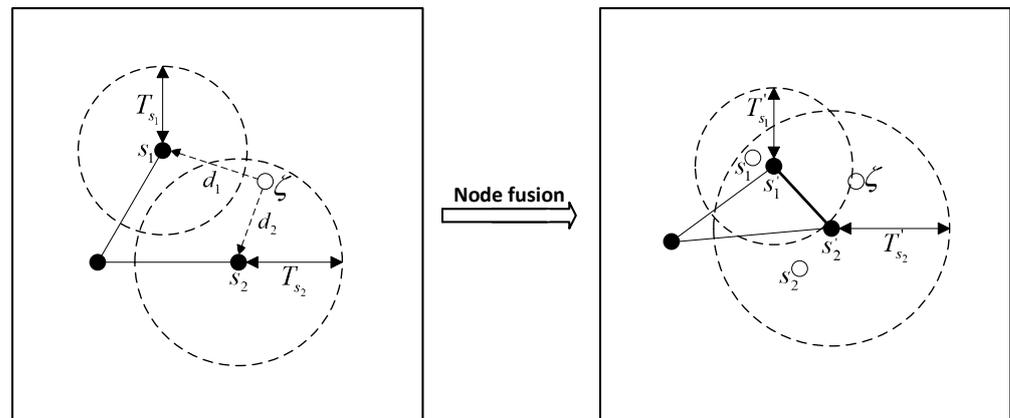


Figure 5. Node fusion.

Updating the weights: Since the coefficient of the original weight-updated value is a predefined fixed value and cannot be adjusted according to the actual situation, the coefficient is updated to

$$w_{s_1} = w_{s_1} + \frac{1}{t_{s_1} * (d_2 + 1)} (w_{\zeta} - w_{s_1}) \tag{14}$$

$$w_{s_2} = w_{s_2} + \frac{1}{t_{s_2} * (d_1 + 1) * (d_2 + 1)} (w_{\zeta} - w_{s_2}) \tag{15}$$

The coefficient of the weight change is the learning rate, and the multiplication is performed by adding one to the Euclidean distance so that the denominator of the updated coefficient is greater than one.  $t_{s_i}$  is the total number of wins for a node. Neurons with a high number of wins are highly represented, and neurons that are close to each other have a high probability of belonging to the same class and moving a relatively large distance. Therefore, the Euclidean distance is introduced into the coefficients of the node movement weights, so that the node movement value is changed by the number of node wins and the distance between nodes.

- (9) The learning cycle is  $\lambda$ ; after learning each  $\lambda$  sample, the denoising operation should be performed. If  $age_{(i,j)} > age_{max}$ , then the connection edge between neurons  $i$  and  $j$  is overaged, and the set  $M = M \setminus \{(i, j)\}$  of neuron relations is updated. If learning is not complete, go back to step 2 of Algorithm 1 and continue processing new samples.

If the value of  $\lambda$  is small, it is applicable to the datasets with many noise points, the noise nodes and connections can be found in time, and the deletion operation can be performed. Conversely,  $\lambda$  is suitable for datasets with less noisy points. When the dataset structure is more complex, the smaller  $age_{max}$  is, the more correct the topology. Generally, the value of  $age_{max}$  is between 50 and 100.

The dataset after three-way attribute reduction is used for SOINN incremental clustering, and the T-SOINN algorithm is obtained. The detailed steps of T-SOINN are shown in Algorithm 2.

**Algorithm 2.** T-SOINN algorithm.

Input: Dataset  $Un = \{x_1, x_2, \dots, x_n\}$ , Decision table  $Dt = (Un, C \cup D, V, f)$ , Three-way attribute selection threshold  $(\alpha, \beta)$

Output: The set  $N$  of neurons and the set  $M$  of connection relations

- 1: Algorithm 1 is used to reduce the attributes of the original dataset, obtain  $U^*$  containing attributes  $C'$
- 2: Initialize the set of neurons within a single competitive learning cycle  $N$ , The set  $M \subseteq N \times N$  of neuronal connection relations
- 3: Input  $\zeta$ , Computing the winning neurons  $s_1$  and  $s_2$
- 4:  $s_1 = \operatorname{argmin}|\zeta - W_\tau|, \tau \in N$
- 5:  $s_2 = \operatorname{argmin}|\zeta - W|_\tau, \tau \in N \setminus \{s_1\}$ ,  $W_\tau$  denotes the weights of  $\tau$
- 6: Let the set of neighbor neurons of  $\tau$  be  $N_\tau$
- 7: Calculate  $d_1, d_2$  and the similarity thresholds  $T_{s_1}$  and  $T_{s_2}$  of the winning neurons  $s_1$  and  $s_2$
- 8: If  $N_i \neq \emptyset$  then
- 9:      $T_\tau = \max|W_\tau - W_j|, j \in N_\tau$
- 10: Else
- 11:      $T_\tau = \min|W_\tau - W_j|, j \in N \setminus \{\tau\}$
- 12: End if
- 13: If  $d_1 > T_{s_1}$  or  $d_2 > T_{s_2}$
- 14:      $N = N \cup \{\zeta\}$
- 15:     If there is no connection between  $s_1$  and  $s_2$
- 16:          $C = C \cup \{(s_1, s_2)\}$
- 17:          $age_{(s_1, s_2)} = 1$
- 18:     Else
- 19:          $age_{(s_1, s_2)} = age_{(s_1, s_2)} + 1$
- 20:     End if
- 21:      $age_{s_1, i} = age_{s_1, i} + 1, \forall i \in N_i$
- 22:      $E_{s_1} = E_{s_1} + |\zeta - w_{s_1}|$
- 23: Else
- 24:      $d_1 < T_{s_1}$  and  $d_2 < T_{s_2}$ ,  $\zeta$  is fused into the topology
- 25:      $w_{s_1} = w_{s_1} + \frac{1}{t_{s_1} * (d_2 + 1)} (w_\zeta - w_{s_1})$
- 26:      $w_{s_2} = w_{s_2} + \frac{1}{t_{s_2} * (d_1 + 1) * (d_2 + 1)} (w_\zeta - w_{s_2})$ ,  $t$  is the number of node wins
- 27: End if
- 28: If the input sample is  $\lambda$  integer times,
- 29:     Perform intra-class node insertion and denoising operations
- 30: End if
- 31: If the input does not end
- 32:     Return to step 2 in algorithm 1 to continue
- 33: End if
- 34: Stop running the algorithm, and the set  $N$  and  $M$  of connection relations are output.

The used symbols and their meanings are shown in Table 1.

**Table 1.** Symbol table.

Symbol	Meaning
$\zeta$	Input new sample
$N$	Ensemble of neurons
$M$	The set of edge-connecting neurons
$T_\tau$	The similarity threshold of node $\tau$
$N_\tau$	The set of neighbor neurons of node $\tau$
$t_\tau$	The number of wins for node $t$
$w_\tau$	The weight of nodes in $\tau$
$age_{(i, j)}$	Age parameter of the connecting edge between nodes $i$ and $j$
$age_{\max}$	The biggest predefined age
$\lambda$	Execution of the denoising operation interval

## 5. Experimental Results and Analysis

### 5.1. Experimental Dataset

KDD CUP99 is selected as the experimental dataset. The dataset KDD-CUP99 contains network connectivity data collected from a simulated Air Force Local Area Network. This dataset has been widely used in experiments with different intrusion detection models and is a classical challenge for intrusion detection and machine learning research [33].

### 5.2. Experimental Environment and Evaluation Indicators

The experimental hardware environment: Windows 10 operating system, AMD Ryzen 53500U CPU with 8 GB of memory. Software environment: Python 3.6, Keras deep learning framework based on TensorFlow, to study the performance of T-SOINN model.

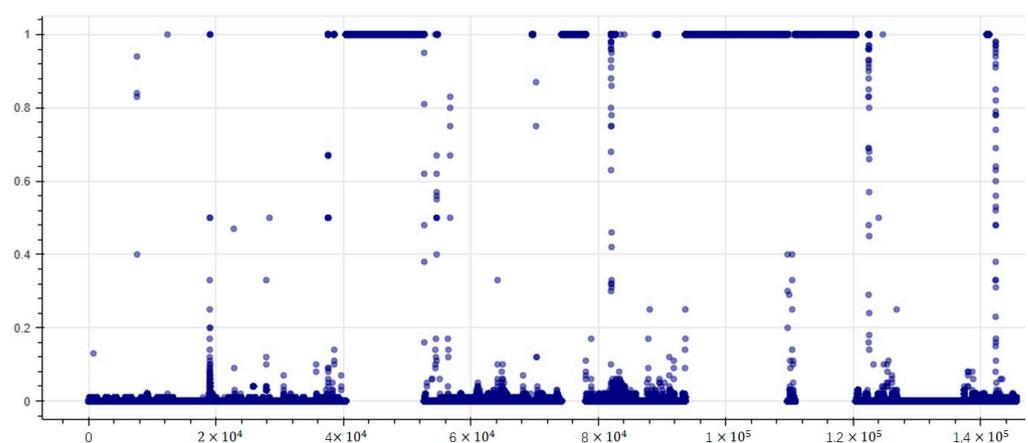
Since the experimental results all have a small range of fluctuations, the average of the ten experimental results is selected as the final detection for comparison across the different experiments. Precision, recall, F1-score, false positive rate (FPR), and false negative rate (FNR) were selected as the evaluation indicators of intrusion detection. For an intrusion detection model, the higher the precision and recall of the experiment and the lower the false positive rate and false negative rate, the better the performance of the intrusion detection model.

### 5.3. Data Preprocessing

Part of the features in the dataset are of the string type, and symbolic feature digitization is needed to convert the corresponding non-numeric type features into numeric identifiers [33].

The KDD-CUP99 dataset contains properties with character values. For example, the protocol\_type attribute has values for TCP, UDP, and ICMP, so it is necessary to convert these properties to numbers. Since the calculation method of attribute significance based on a rough set requires that the attribute values are discrete values, K-means clustering is used to discretize the data.

The 32 continuously valued attributes of the original dataset are visualized to obtain the distribution law of each attribute and thus to determine the number of clusters for each attribute. The 32 continuous attributes are visualized separately. Figure 6 shows the distribution diagram of attribute dst\_host\_srv\_error\_rate. The K value of each continuous attribute is determined using the results of the visual demonstration, and then, the K-means clustering discretization process is carried out.



**Figure 6.** Continuous attribute distribution diagram.

### 5.4. Experimental Comparison

#### 5.4.1. Parameter Analysis

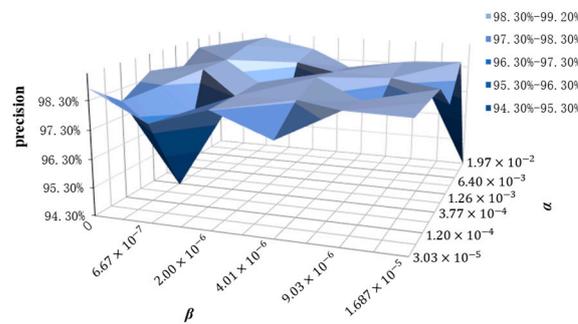
The importance of all attributes is calculated, and the importance of some attributes is shown in Table 2.

**Table 2.** Attribute importance of some attributes.

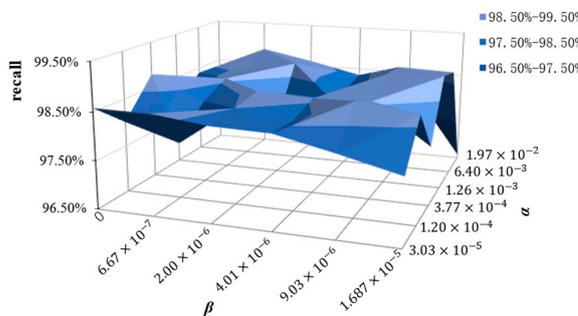
Name of Attribute	Importance of Attributes
srv_diff_host_rate	$1.316681 \times 10^{-3}$
dst_host_count	$1.969368 \times 10^{-2}$
dst_host_srv_count	$1.310848 \times 10^{-4}$
dst_host_same_srv_rate	$1.088951 \times 10^{-4}$
dst_host_same_srv_rate	$4.053212 \times 10^{-5}$
dst_host_same_src_port_rate	$2.014152 \times 10^{-5}$
dst_host_srv_diff_host_rate	$3.529692 \times 10^{-4}$

According to the partition attribute of  $(\alpha, \beta)$ , it is divided into a positive region, a negative region, and a boundary region. The differences in the importance of different attributes in Table 2 are relatively large. The value of attribute importance is divided into several equidistant intervals, and the average value of attribute importance in each interval is taken as the threshold value of  $(\alpha, \beta)$ , where  $\alpha > \beta$ . Due to the randomness of the three-way attribute reduction method, the selected attribute subset is different according to the difference of  $(\alpha, \beta)$ , and the model checking result also changes.

The changes in precision and recall under different  $\alpha$  and  $\beta$  are shown in Figures 7 and 8, respectively. When  $(\alpha, \beta) = (1.969368 \times 10^{-2}, 6.673866 \times 10^{-5})$ , the precision and recall of the model are optimal; when  $(\alpha, \beta) = (1.969368 \times 10^{-2}, 1.676748 \times 10^{-5})$ , the precision and recall of the model are the lowest. In this case, the original attribute is divided into three parts, and the part with the lowest attribute importance has the largest number. In the random selection process, the number of attributes with low importance is selected, so the overall accuracy and recall of the model are low, indicating that having many attributes with low importance will reduce the detection performance.



**Figure 7.** The precision changes under different  $(\alpha, \beta)$ .



**Figure 8.** The recall changes under different  $(\alpha, \beta)$ .

The changes in FPR and FNR under different  $\alpha$  and  $\beta$  are shown in Figures 9 and 10, respectively. From the results of the experiment, when  $(\alpha, \beta) = (1.969368 \times 10^{-2}, 6.673866 \times 10^{-5})$ , the precision and recall of the model are optimal and the FPR and FNR of the model are the lowest. The effect of the model reached an optimal level. At this time, most

attributes were selected from the positive domain and the boundary domain, while the number of attributes selected from the negative domain was small. That is, with more high-importance attributes, the T-SOINN model’s detection performance was better.

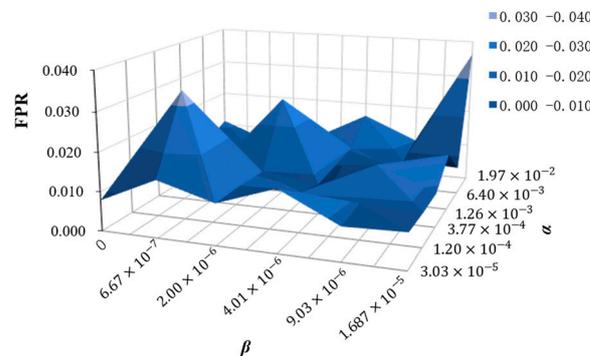


Figure 9. The FPR changes under different  $(\alpha, \beta)$ .

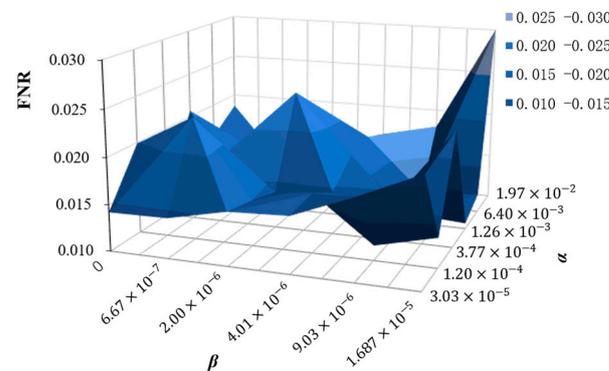


Figure 10. The FNR changes under different  $(\alpha, \beta)$ .

With the values of the above parameters, the data subset after attribute reduction was passed into SOINN for training. The value of  $age_{max}$  was between 50 and 100. The value of  $\lambda$  was between 100 and 200. The range of  $age_{max}$  and  $\lambda$  was divided equally.  $age_{max}$  took values of 50, 60, 70, 80, 90, and 100.  $\lambda$  took values of 100, 120, 140, 160, 180, and 200.

With different values of  $age_{max}$  and  $\lambda$ , the precision and recall changes in the model are shown in Figures 11 and 12 and the FPR and FNR changes in the model are shown in Figures 13 and 14, respectively. According to Figures 11–14, when  $age_{max} = 70$  and  $\lambda = 180$ , the precision and recall are the highest and the FPR and FNR are the lowest. According to an analysis of the above parameters,  $(\alpha, \beta) = (1.969368 \times 10^{-2}, 6.673866 \times 10^{-5})$ ,  $age_{max} = 70$ , and  $\lambda = 180$ .

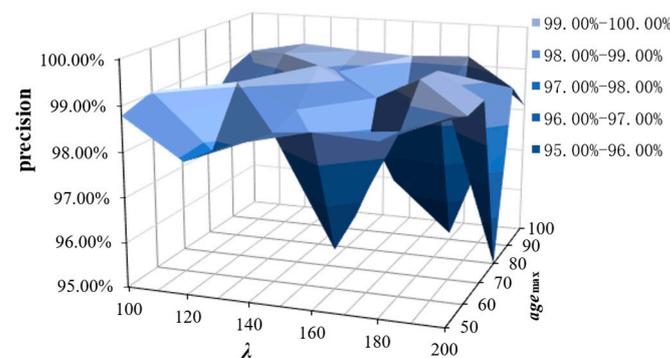


Figure 11. The precision changes under different  $age_{max}$  and  $\lambda$ .

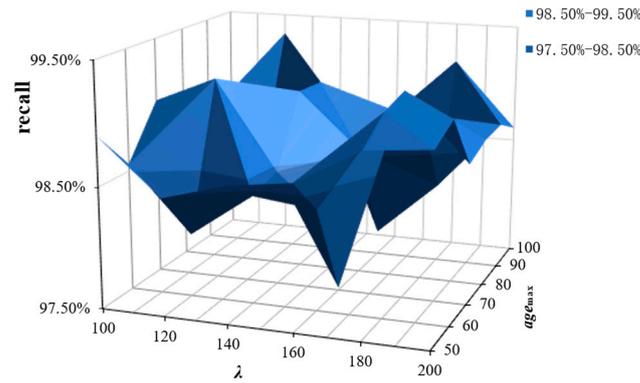


Figure 12. The recall changes under different  $age_{max}$  and  $\lambda$ .

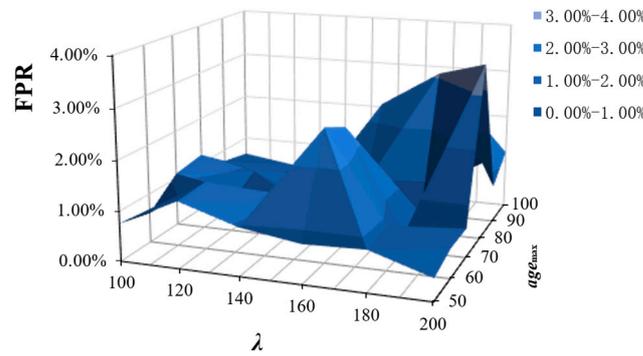


Figure 13. The FPR changes under different  $age_{max}$  and  $\lambda$ .

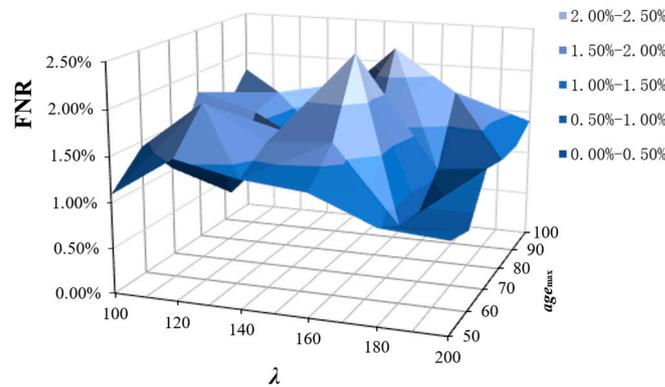
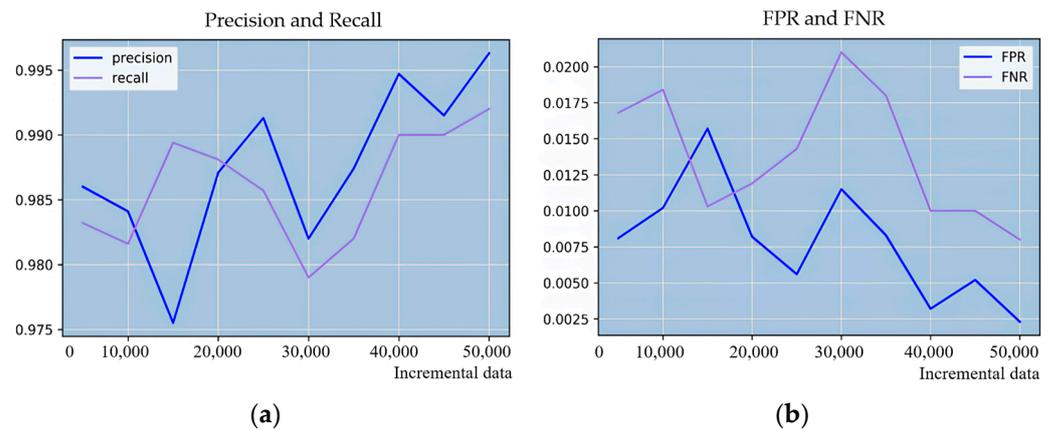


Figure 14. The FNR changes under different  $age_{max}$  and  $\lambda$ .

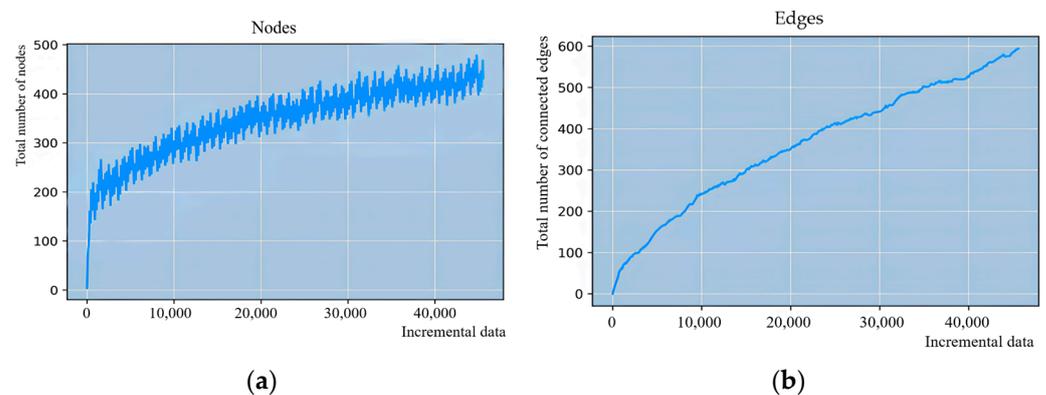
#### 5.4.2. Analysis of Experimental Results

The parameter values were obtained from the above analysis. The changes in precision, recall, false positive rate, and false negative rate of the model with the incrementally incoming stream data are shown in Figure 15. With the incremental data input, the precision and recall gradually show an upward trend after certain fluctuations, while the FPR and FNR show a downward trend. In other words, when the amount of training data is small, the model detection results fluctuate a lot, while the model detection performance gradually stabilizes as the amount of training data increases.



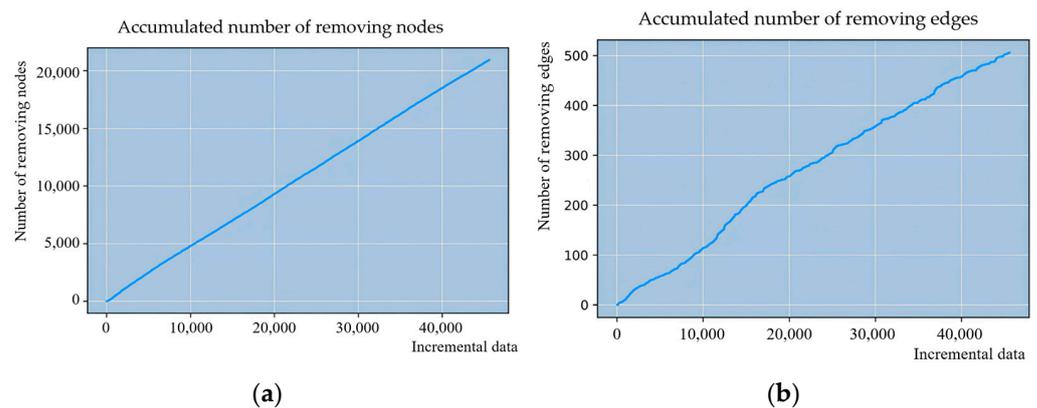
**Figure 15.** Incremental change in the evaluation index: (a) precision and recall; (b) FPR and FNR.

The total number of nodes is constantly changing during the training of the T-SOINN model. When the difference between the new sample and the data previously learned by the T-SOINN model is too large, that is,  $d_1 > T_{s_1}$  or  $d_2 > T_{s_2}$ , the new node is inserted and the total number of nodes increases. After  $\lambda$  cycles, intra-class insertion and denoising operations are needed for the entire SOINN structure. The intra-class node insertion operation requires selecting the two nodes with the largest quantization error among all nodes, inserting new nodes in the middle to reduce the quantization error of the neurons, and increasing the total number of nodes. When performing the denoising operation, the noise and outliers in the network structure and the associated connections should be removed, at which point the total number of nodes is reduced. The growth changes in the nodes and edges are shown in Figure 16. Figure 16a shows, as an incremental data input, the change in the total number of nodes during the topological structure adjustment process. Figure 16b shows the evolution of the total number of connected edges in the topology adjustment process with increasing data input.



**Figure 16.** The number of nodes and edges varies. (a) Nodes. (b) Edges.

The winning neurons  $s_1$  and  $s_2$  are activated simultaneously and make connections as new data are input. When the value of *age* reaches the maximum, then the current connection is removed and the number of connection edges decreases. In the denoising operation after  $\lambda$  cycles, the noise and outlier points and related connections in the network structure are deleted, and the number of connection edges decreases. The change in the number of deleted nodes and edges is shown in Figure 17. Figure 17a shows the change in the total number of deleted nodes in the topology adjustment process when incremental data are input. In Figure 17b, as the incremental data are input, a number of topological structure adjustments were removed during the changes.



**Figure 17.** Number of deleted nodes and edges. (a) Number of removed nodes. (b) Number of removed edges.

In order to verify the experimental effect, the T-SOINN model is compared with SOINN and LASSO-SOINN on the KDD-CUP99 dataset. Three-way attribute reduction is a dimensionality reduction method, and so is LASSO. LASSO is combined with the SOINN algorithm to obtain LASSO-SOINN, and it is applied to the KDD-CUP99 dataset. The performance comparison results of the three algorithms are shown in Table 3.

**Table 3.** Performance comparison.

Evaluation Index	T-SOINN	SOINN	LASSO-SOINN
precision	99.43%	97.45%	97.74%
recall	99.21%	97.73%	97.99%
FNR	0.79%	2.27%	2.01%
FPR	0.36%	1.63%	1.39%

If only the SOINN algorithm is used on the KDD-CUP99 dataset, the precision and recall are 97.45% and 97.73%, respectively. For SOINN, the precision and recall are increased by 0.31% and 0.26%, respectively, when LASSO is used for dimension reduction. Utilizing three-way attribute reduction for dimensionality reduction, the precision and recall are improved by 1.98% and 1.48%, respectively, which not only significantly improves the performance of SOINN but also outperforms LASSO. Similarly, using three-way attribute reduction for dimensionality reduction, not only are the FPR and FNR significantly reduced but also the performance is better than that of LASSO. According to the above analysis, the effectiveness of the three-way attribute reduction for optimizing the SOINN algorithm is evident, and T-SOINN is a good method for processing intrusion detection data.

## 6. Conclusions and Prospect

In view of the large scale of intrusion detection data that cannot be completely collected all at once, the T-SOINN incremental intrusion detection model was proposed. Through three-way attribute reduction, attribute importance was used as the evaluation function to calculate the attribute subset to achieve data dimensionality reduction. After receiving the dimensionality reduction data, SOINN generated a topological representation of the original data and achieved incremental learning by dynamically adjusting the topological structure. The results show that the precision and recall rate of the T-SOINN model are effectively improved.

In the future, we will further consider the optimization of the inter-class insertion operation, add three decisions to the inter-class insertion discrimination, and re-input the topological structure of the uncertain data combined with more information, so as to further improve the overall performance of the model. The comparison process of the similarity threshold and the Euclidean distance is relatively independent, and there are some errors.

Therefore, in the future, we can continue to enrich the comparison process and to improve the comparison content.

**Author Contributions:** Conceptualization, J.R.; data curation, L.L. and J.R.; methodology, H.H.; software, J.M.; validation, C.Z. and L.W.; formal analysis, B.L.; investigation, Y.Z.; resources, H.H.; data curation, J.R.; writing—original draft preparation, L.L.; writing—review and editing, J.M.; visualization, L.W.; supervision, B.L.; project administration, J.R.; funding acquisition, Y.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is supported by the Basic Scientific Research Business Expenses of Hebei Provincial Universities (JST2022001), the Tangshan Science and Technology Project (22130225G), the Innovation and Entrepreneurship Training Project for College Students in Hebei Province (S202210081055), S&T Program of Hebei (20310802D), and Tangshan Science and Technology Bureau (21130211D).

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

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data are openly available in a public repository. The datasets in this article are from KDD-CUP99 (Data Mining and Knowledge Discovery) on <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html> (accessed on 1 September 2022).

**Acknowledgments:** Thanks to all the projects that provided funding and experimental equipment for this paper.

**Conflicts of Interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

1. Wang, L. Network Intrusion Detection Method based on Improved Rough Set Attribute Reduction and K-means Clustering. *J. Comput. Appl.* **2020**, *40*, 1996–2002.
2. Liu, J.H.; Sun, X.; Jin, J. Intrusion Detection Model Based on Principal Component Analysis and Recurrent Neural Network. *J. Chin. Inf. Process.* **2020**, *34*, 105–112.
3. Bhati, B.S.; Rai, C.S. Analysis of Support Vector Machine-Based Intrusion Detection Techniques. *Arab. J. Sci. Eng.* **2020**, *45*, 2371–2383. [[CrossRef](#)]
4. Douiba, M.; Benkirane, S.; Guezzaz, A.; Azrou, M. An Improved Anomaly Detection Model for IoT security using decision tree and Gradient Boosting. *J. Supercomput.* **2023**, *79*, 3392–3411. [[CrossRef](#)]
5. Gu, J.; Lu, S. An Effective Intrusion Detection Approach Using SVM with Naïve Bayes Feature Embedding. *Comput. Secur.* **2021**, *103*, 102158. [[CrossRef](#)]
6. Jia, F.; Kong, L.Z. Intrusion Detection Algorithm Based on Convolutional Neural Network. *Trans. Beijing Inst. Technol.* **2017**, *37*, 1271–1275.
7. Henry, A.; Gautam, S.; Khanna, S.; Rabie, K.; Shongwe, T.; Bhattacharya, P.; Sharma, B.; Chowdhury, S. Composition of Hybrid Deep Learning Model and Feature Optimization for Intrusion Detection System. *Sensors* **2023**, *23*, 890. [[CrossRef](#)]
8. Chen, H.S.; Chen, J.J. Recurrent Neural Networks Based Wireless Network Intrusion Detection and Classification Model Construction and Optimization. *J. Electron. Inf. Technol.* **2019**, *41*, 1427–1433.
9. Li, S.S.; Li, Z.Y.; Lai, X.M.; Chen, H. Incremental Intrusion Detection Method based on Probabilistic Neural Networks. *Comput. Simul.* **2022**, *39*, 476–482.
10. Wang, R.; Liu, Z.R.; Ji, J. Decision Tree Algorithm Based on Attribute Significance. *Comput. Sci.* **2017**, *44*, 129–132.
11. Zhou, C.S.; Xu, J.C.; Qu, K.L.; Shen, K.L.; Zhang, L. A Fast Attribute Reduction Method based on Improved Attribute Importance in Neighborhood Rough Sets. *J. Northwest Univ.* **2022**, *52*, 745–752.
12. Zheng, W.B.; Li, J.J.; He, Q.H. Attribute Reduction Algorithm for Neighborhood Rough Sets with Variable Precision Based on Attribute Importance. *Comput. Sci.* **2019**, *46*, 261–265.
13. Zhang, C.; Ren, J.; Liu, F.; Li, X.; Liu, S. Three-way Selection Random Forest Algorithm based on Decision Boundary Entropy. *Appl. Intell.* **2022**, *52*, 13384–13397. [[CrossRef](#)]
14. Zhang, C.Y.; Wang, W.J.; Liu, L.; Ren, J.; Wang, L.Y. Three-Branch Random Forest Intrusion Detection Model. *Mathematics* **2022**, *10*, 4460. [[CrossRef](#)]
15. Polikar, R.; Upda, L.; Upda, S.S.; Honavar, V. Learn++: An Incremental Learning Algorithm for Supervised Neural Networks. *Syst. Man Cybern. Part C Appl. Rev. IEEE Trans.* **2001**, *31*, 497–508. [[CrossRef](#)]
16. Liu, T.Y. *Technology for Internet of Things Research on Incremental Intrusion Detection*; North China University of Technology: Beijing, China, 2021.

17. Belouadah, E.; Popescu, A.; Kanellos, I. A Comprehensive Study of Class Incremental Learning Algorithms for Visual Tasks. *Neural Netw.* **2021**, *135*, 38–54. [[CrossRef](#)]
18. Wei, X.; Liu, S.; Xiang, Y.; Duan, Z.; Zhao, C.; Lu, Y. Incremental Learning based Multi-Domain Adaptation for Object Detection. *Knowl.-Based Syst.* **2020**, *210*, 106420. [[CrossRef](#)]
19. García-Salinas, J.S.; Torres-García, A.A.; Reyes-García, C.A.; Villaseñor-Pineda, L. Intra-subject Class-Incremental Deep Learning Approach for EEG-based Imagined Speech Recognition. *Biomed. Signal Process. Control.* **2023**, *81*, 104433. [[CrossRef](#)]
20. Fu, Z.X.; Xu, Y.; Wu, Z.D.; Xu, D.D.; Xie, X.Y. SVM-KNN Network Intrusion Detection Method Based on Incremental Learning. *Comput. Eng.* **2020**, *46*, 115–122.
21. Liu, J.W.; Li, R.N.; Zhang, Y.; Liang, Y. Incremental GHSOM Algorithm for DDoS Attack Detection. *J. Nanjing Univ. Posts Telecommun.* **2020**, *40*, 82–88.
22. Yang, Y.H.; Huang, H.Z.; Shen, Q.N.; Wu, Z.H.; Zhang, Y. Research on Intrusion Detection Based on Incremental GHSOM. *Chin. J. Comput.* **2014**, *37*, 1216–1224.
23. Liu, Q.; Zhang, Y.; Zhou, W.X.; Jiang, X.T.; Zhou, W.N.; Zhou, M.G. Adaptive Class Incremental Learning-Based IoT Intrusion Detection System. *Comput. Eng.* **2023**, *49*, 169–174.
24. Laghrissi, F.; Douzi, S.; Douzi, K.; Hssina, B. Intrusion Detection Systems Using Long Short-Term Memory (LSTM). *J. Big Data* **2021**, *8*, 65. [[CrossRef](#)]
25. Alotaibi, S.D.; Yadav, K.; Aledaily, A.N.; Alkwai, L.M.; Dafhalla, A.K.Y.; Almansour, S.; Lingamuthu, V. Deep Neural Network-Based Intrusion Detection System through PCA. *Math. Probl. Eng.* **2022**, *2022*, 6488571. [[CrossRef](#)]
26. Luo, J.; Wang, H.; Li, Y.; Lin, Y. Intrusion Detection System based on Genetic Attribute Reduction Algorithm based on Rough Set and Neural Network. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 5031236. [[CrossRef](#)]
27. Gao, C.; Lai, Z.; Zhou, J.; Zhao, C.; Miao, D. Maximum Decision Entropy Based Attribute Reduction in Decision Theoretic Rough Set Model. *Knowl.-Based Syst.* **2018**, *143*, 179–191. [[CrossRef](#)]
28. Zhai, J.H.; Wan, L.Y.; Wang, X.Z. Attribute Reduction with Principle of Minimum Correlation and Maximum Dependency. *Comput. Sci.* **2014**, *41*, 148–150.
29. Liang, B.H.; Wu, Q.L. Attribute Reduction based on Information Entropy of Approximation Boundary Accuracy. *J. East China Norm. Univ.* **2018**, *2018*, 97–108.
30. Jia, P.; Dai, J.H.; Pan, Y.H. Novel Algorithm for Attribute Reduction based on Mutual Information Gain Ratio. *J. Zhejiang Univ.* **2006**, *40*, 1041–1044.
31. Shen, F.R.; Hasegawa, O. An Incremental Network for On-Line Unsupervised Classification and Topology Learning. *Neural Netw.* **2006**, *19*, 90–106.
32. Qiu, T.Y.; Shen, F.R.; Zhao, J.X. Review of Self-Organizing Incremental Neural Network. *J. Softw.* **2016**, *27*, 2230–2247.
33. Yu, H.H.; Zhou, F.Y.; Chen, M.M. Analysis of KDD-CUP99 Network Intrusion Detection Data Set based on Machine Learning. *Comput. Eng. Sci.* **2019**, *41*, 91–97.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.