Tri-training algorithm based on cross entropy and K-nearest neighbors for network intrusion detection

Jia Zhao[1], Song Li[1], Runxiu Wu[1], Yiying Zhang[2], Bo Zhang[3], Longzhe Han[1*]
[1]Nanchang Institute of Technology, School of Information Engineering
Nanchang 330099, China
*[e-mail: longzhehan@gmail.com]
[2]College of artificial intelligence, Tianjin University of Science & Technology,
Tianjin, 300457, China
[e-mail: yiyingzhang@tust.edu.cn]
[3]State grid smart grid research institute co., ltd
[*]Corresponding author: Longzhe Han

Received April 13, 2022; revised November 14, 2022; accepted December 5, 2022; published December 31, 2022

Abstract

To address the problem of low detection accuracy due to training noise caused by mislabeling when Tri-training for network intrusion detection (NID), we propose a Tri-training algorithm based on cross entropy and K-nearest neighbors (TCK) for network intrusion detection. The proposed algorithm uses cross-entropy to replace the classification error rate to better identify the difference between the practical and predicted distributions of the model and reduce the prediction bias of mislabeled data to unlabeled data; K-nearest neighbors are used to remove the mislabeled data and reduce the number of mislabeled data. In order to verify the effectiveness of the algorithm proposed in this paper, experiments were conducted on 12 UCI datasets and NSL-KDD network intrusion datasets, and four indexes including accuracy, recall, F-measure and precision were used for comparison. The experimental results revealed that the TCK has superior performance than the conventional Tri-training algorithms and the Tri-training algorithms using only cross-entropy or K-nearest neighbor strategy.

Keywords: network intrusion detection (NID), semi-supervised learning, Tri-training, cross entropy, K-nearest neighbors

1. Introduction

Currently, With the rapid development of network technology, network security is facing a huge threat, and the emergence of network intrusion detection technology (NID) has played an important role in network security. Traditional NID techniques achieve intrusion detection by comparing attacks identified in the feature code database, but this method has a high leakage rate and lag [1]. Recently, with the rapid development of artificial intelligence [2-5] and machine learning technologies [6-8], machine learning-based NID methods are gradually becoming a research hotspot.

The lack of labeled data is the difficulty of machine learning in network intrusion detection. How to use a small amount of tag data to detect network intrusion is the focus of research. Machine learning based NID methods can be classified into supervised, unsupervised and semi-supervised detection methods. In supervised NID algorithms, all training data need to be labeled, and a good model cannot be trained when the data does not have labeled categories or when the data identification features are not obvious [9]. The unsupervised NID algorithm can learn from unlabeled data, but it cannot obtain high learning precision [1]. Semi-supervised algorithms can effectively solve the shortcomings of supervised and unsupervised NID algorithms, and can obtain a model with high learning precision in a small amount of labeled data and a large amount of unlabeled data.

Semi-Supervised Learning (SSL) [10-11] methods mainly include Disagreement-Based Semi-supervised Learning [12], Generative Methods [13], Discriminative Methods [14] and Graph-Based methods [15]. Since the disagreement-based methods are less affected by model assumptions, loss function non-convexity and data size problems, it can meet most of the network intrusion data detection requirements and has good classification properties. Therefore, disagreement-based methods were used for NID in this study.

The disagreement-based methods originated from the co-training algorithm (Co-training) proposed by Blum and Mitchell [16]. The co-training algorithm used two different views to train the classifier, and improves the performance of the algorithm by expanding the training set for each other. Two assumptions are required for co-training: (1) sufficient redundancy of views; (2) conditional independence assumption. In practice, few data satisfy sufficient redundancy of views and condition independent. To compensate for the shortcomings of the co-training algorithm, Zhou *et al.* [17] proposed the Tri-training algorithm. Using three classifiers, it solves the problem of harsh conditions in the co-training algorithm and does not require sufficient redundancy in the data set. However, Tri-training algorithm can generate training noise due to mislabeling, and how to solve the noise problem effectively is the focus of scholars' attention.

To address the problems of Tri-training, Hu *et al.* [18] proposed a semi-supervised patent text classification method based on improved Tri-training algorithm,which makes three changes to Tri-training. The algorithm firstly uses three base classifiers with large differences to train the same data set instead of updating three training sets at the same time, secondly in the process of the untagged data are marked only when the three base classifier consistent and marking probability is greater than their respective probability threshold to put it in the marking of the training set, finally, the update of the training set for dynamic tracking, real-time updates to the same untagged data probability threshold, effectively reduce the noise of the training set through the above three data. Li *et al.* [19] proposed a novel semi-supervised adaboost technique based on improved Tri-training algorithm. The algorithm achieves noise rejection in labeled data by calculating the predicted probability of unlabeled data compared with the set probability threshold, and then using the calculated probability thresholds as

weights, the sum of the weights of the current marking errors is compared with the previous round, and if the sum of the current weights is smaller than the previous round, the base classifier is updated using the currently marked dataset. Zhang *et al.* [20] proposed a safety Tri-training algorithm based on cross entropy. It replaces the classification error rate with cross entropy, which effectively reduces the prediction bias of labeled noise on unlabeled data in Tri-training. Mo *et al.* [21] proposed a semi-supervised classification algorithm based on trapezoid network and improved three-training method. It improves the label confidence level on unlabeled data by calculating the classification difference of classifiers. The improved Tri-training algorithms mentioned above reduces the labeling noise to a certain extent, but uses a single strategy, which is not effective in improving the classification. Li *et al.* [22] pointed out in safe semi-supervised learning: a brief introductions algorithm that in some cases, learning with unlabeled data does not improve the performance of the algorithm, but degrades it by adding noise.

To address the problem of producing training noise due to mislabeling during Tri-training for NID, we propose the Tri-training algorithm based on cross entropy and K-nearest neighbors (TCK). The algorithm uses cross-entropy to replace the classification error rate. Cross-entropy can better identify the difference between the predicted distribution and the practical distribution of the model, and reduce the prediction bias of mislabeled data on unlabeled data; K-nearest neighbors are used to delete the mislabeled data and reduce the number of mislabeled data. Experimental results on the 12 sets of UCI datasets and the NSL-KDD network intrusion data set revealed that performances of the proposed algorithms are all improved in the four indexes (*accuracy*, *recall*, *F-measure* and *precision*).

In the second section, the paper introduces the basic ideas of the Tri-training algorithm, the concepts related to relative entropy and cross-entropy and the K-nearest neighbour idea. The third section introduces the principle of the algorithm and gives the pseudo-code of the algorithm. In the fourth section, the algorithm is tested experimentally and the results are analyzed. The fifth section summarizes the thesis.

2. Related work

This section introduces the basic idea of Tri-training algorithm, the concept of relative entropy and cross entropy, and the idea of K-nearest neighbor.

2.1 Tri-training algorithm

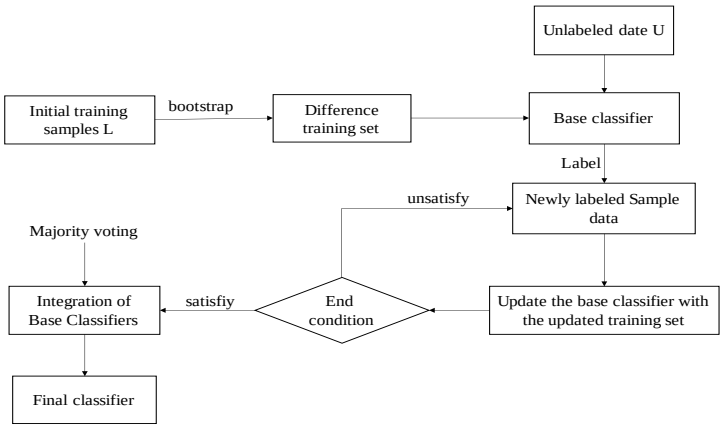

Fig. 1. Training flow chart

The Tri-Training algorithm uses three base classifiers to add pseudo-labels to the unlabeled data, The classification performance of the algorithm is improved by changing the training data set by labeling each other.. Assuming that there is a data set D, which includes a small amount of labeled data L and a large amount of unlabeled data U. L is Bootstrap sampled to train three base classifiers h_1, h_2 and h_3. Take the training process of h_1 as a column: x is any point in U, h_2 and h_3 predict x at the same time, if h_2 and h_3 have the same prediction result, i.e., $h_2(x) = h_3(x)$, then the labeled result $h_2(x)$ of x is put into a new training set $L_1^{'}$, $L_1^{'} = L \bigcup \{x \mid x \in U \ and \ h_2(x) = h_3(x)\}$, $L_1^{'}$ is the training set of h_1 in the next round. The training process of classifiers h_2 and h_3 is similar to h_1. The classification performance is improved by continuously updating the training set $L_i^{'}(i = 1,2,3)$ and the base classifier $h_i^{'}(i = 1,2,3)$. This process is repeated for the three classifiers until the classifiers h_1, h_2 and h_3 do not change, and finally the final classification results are decided by majority voting method. The training flow chart is shown in **Fig. 1**.

From the training process of the algorithm, it is clear that the algorithm is prone to noise when adding pseudo-labeling to unlabeled data. Angling *et al.* [23] demonstrated the learnability of the training set noise if the following conditions are satisfied. The training set noise is learnable:

$$m \geq \frac{2}{\epsilon^2 (1-2\eta)^2} \ln(\frac{2N}{\delta}) \tag{1}$$

where m is the training sample size, ϵ is the worst-case classification error rate, η (<0.5) is the upper limit of the classification noise rate, N is the number of hypotheses, and δ is the confidence level.

Suppose η_L represents the classification noise rate on dataset L, then the number of false marks in L is $\eta_L |L|$. e_i^t represents the classification error rate of h_j and h_k $(j,k \neq i)$ in round t. Suppose that in round t, the number of data sets that h_j and h_k jointly mark is z, and the number of data that both h_j and h_k correctly mark z is $z^{'}$, the $e_i^t = \frac{z - z^{'}}{z}$, therefore, the number of false marks in L^t is $e_i^t |L^t|$, then the classification noise rate η^t of round t can be defined as Eq (2).

$$\eta^t = \frac{\eta_L |L| + e_i^t |L^t|}{|L| + |L^t|} \tag{2}$$

Where L^t represents the training set marked h_i by h_j and h_k $(j,k \neq i)$ in round t.

Zhou et al. [17] proved that as long as the updated training set satisfies Eq (3), the performance of the classifier will be improved.

$$|L \bigcup L^t| (1 - 2\frac{\eta_L |L| + e_i^t |L^t|}{|L \bigcup L^t|})^2 > |L \bigcup L^{t-1}| (1 - 2\frac{\eta_L |L| + e_i^{t-1} |L^{t-1}|}{|L \bigcup L^{t-1}|})^2 \tag{3}$$

where L^{t-1} denote the training set in rounds $t-1$ where h_j and h_k $(j,k \neq i)$ are h_i labeled training set, e_i^{t-1} denotes the classification error rate of h_j and h_k $(j,k \neq i)$ in rounds $t-1$.

2.2 Relative entropy and cross entropy

Relative entropy, also known as KL (Kullback-Leibler) divergence or information divergence, represents an asymmetric measure of the difference between two probability distributions [24]. In information theory, relative entropy is the difference between the Shannon entropy of two probability distributions [25]. Let $P(X)$ and $Q(X)$ be the practical probability distribution and predicted probability distribution of the random variable X. The relative entropy is defined as [26]:

$$D_{KL}(P \| Q) = \sum_{x \in X} P(x) \lg \frac{P(x)}{Q(x)} \tag{4}$$

The smaller the relative entropy, the smaller the deviation of the practical probability distribution $P(X)$ of the model from the predicted probability distribution $Q(X)$. When the practical probability distribution is the same as the predicted probability distribution, $D_{KL}=0$.

The concept of cross-entropy was introduced by Rubinstein [27] to measure the variability between two probability distributions. Deform the Eq (4):

$$\begin{aligned} D_{KL}(P \| Q) &= \sum_{x \in X} P(x) \lg P(x) - \sum_{x \in X} P(x) \lg Q(x) \\ &= -H(P(x)) + [-\sum_{x \in X} P(x) \lg Q(x)] \end{aligned} \tag{5}$$

The relative entropy is split into the entropy $-H(P(x))$ of the practical distribution P and the cross-entropy:

$$H(P,Q) = -\sum_{x \in X} P(x) \lg Q(x) \tag{6}$$

The cross entropy has two important properties: (1) asymmetry, $H(P,Q)$ and $H(Q,P)$ are not equal; (2) non-negativity, the cross-entropy can only be greater than or equal to 0.

According to Eq (4), we can see that the relative entropy $D_{KL}(P \| Q)$ changes mainly due to the cross-entropy $H(P,Q)$, and the entropy of P remains unchanged. In the past machine learning process, the relative entropy D_{KL} is mainly used to determine the difference between the practical probability distribution $P(X)$ and the predicted probability distribution $Q(X)$. From Eq (6), it is clear that the difference between $P(X)$ and $Q(X)$ can be determined by cross entropy, and it is more convenient to calculate the differexnce between two probability distributions by using cross entropy than relative entropy.

Cross-entropy has been widely used in machine learning. Too *et al*. [28] proposed the incremental clustering algorithm based on cross entropy, the algorithm uses cross-entropy to map data points in a high-dimensional space to a low-dimensional space, to partition dynamic data. The experimental results show that this method has lower time complexity in large-scale data environments or dynamic working environments. Liu *et al*. [29] applied cross-entropy to the class imbalance problem, and proposed a new weighted cross-entropy as a loss function. The experimental results show that this method can effectively reduce the impact of noise on the classification results. Santosa [30] applied cross-entropy to a dual Lagrangian support vector machine (SVM), using cross-entropy to solve the Lagrangian SVM optimization problem to find the optimal or at least near-optimal Lagrangian multipliers as a solution, the experimental results show that the proposed algorithm has obvious advantages in terms of computation time and accuracy.

2.3 K-nearest neighbors classification

K-nearest neighbors were proposed by Cover and Hart [31] in 1968. K-nearest neighbors means that each data can be represented by its K nearest neighbors. The core idea is that if most of the K-nearest neighbors of a sample belong to a class in the feature space, then the sample also belongs to that class.

A distance-based measure is used to find the k nearest neighbors, such as Euclidean distance. Let two points or tuples be $X_1 = (x_{11}, x_{12}, \cdots, x_{1n})$ and $X_2 = (x_{21}, x_{22}, \cdots, x_{2n})$ respectively, then the Euclidean distance of the two points or tuples is:

$$dist(X_1, X_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2} \tag{7}$$

The K-nearest neighbor idea is widely used and has achieved good application results in the field of data mining. Wang *et al*. [32] proposed a new multi-label classification algorithm based on K-nearest neighbors and random walks. The algorithm uses the K-nearest neighbor idea to construct the edge set of the correlation between the vertex set of the random walk graph and the label of the training sample with the K-nearest neighbor training samples of the specific test data, which greatly reduces the time and space overhead. Xiao *et al*. [33] proposed a fast incremental learning algorithm for SVM based on K-nearest neighbors. The algorithm uses the K-nearest neighbor idea to extract the boundary vector set, and replaces the boundary vector set with the training set to train the SVM. Ren *et al*. [34] proposed an efficient density peak clustering algorithm based on hierarchical K-nearest neighbors and subcluster merging. The algorithm uses the K-nearest neighbor idea to divide the dataset into multiple layers so that the algorithm can obtain better clustering results. *Wu et al*. [35] proposed a density peaks clustering based on relative density estimating and multi cluster merging algorithm (DPC-RD-MCM), The algorithm redefines the local density by using the K-nearest neighbor idea. Through experiments on datasets with uneven density distribution, UCI datasets and complex morphological datasets, the DPC-RD-MCM algorithm can be used in data with uneven density distribution. Very good clustering effect is obtained on the complex morphological data set and UCI data set, and the clustering performance is higher than that of the comparison algorithm. *Zhao et al*. [36] proposed a density peak clustering algorithm based on mutual proximity. The new algorithm introduces the idea of k-nearest neighbors to calculate the local density, so as to ensure the relativity of the local density.

3. Tri-training algorithm based on cross-entropy and K-nearest neighbor (TCK)

3.1 Algorithm Principles

In order to better utilize the unlabeled data for learning and reduce the noise generated during the pseudo labeling process of Tri-training algorithm, this study proposes the TCK.

In semi-supervised learning, the predicted distribution $Q(X)$ obtained from the training data is expected to be as close to the practical distribution $P(X)$ as possible, but the practical distribution P is unknowable. Assuming that the training data is assumed to be obtained from independent homo-distributed sampling from real data, the predicted distribution Q is expected to be the least different from the training data distribution P'. Finding the least difference is equivalent to finding the cross-entropy $H(P', Q)$. The smaller the cross entropy value, the closer Q is to P'. Tri-training algorithms determine the difference between the practical and

predicted distributions by the classification error rate, and the cross entropy is a better measure model of the difference between the practical and predicted distributions than the classification error rate. In this study, the classification error rate in Tri-training algorithms is replaced by cross entropy.

Tri-training algorithms generate noise when learning from unlabeled data, while semi-supervised learning expects no noise or as little noise as possible. K-Nearest neighbors can identify the noise effectively, so the K-Nearest neighbors idea is used for noise processing. This is done as follows: suppose the pseudo labeling training set generated by the algorithm in round t is L^t, the initial training set of the algorithm is L, x is any point in L^t. Find k nearest neighbors of x and L. If the number of nearest neighbors same as the label x is greater than or equal to k', keep the data, otherwise delete the data. According to the literature [37], the best experimental results were obtained when k and k' were taken as 3 and 2. In the subsequent experiments of this study, this parameter was also used to set k and k'.

3.2 Procedures

Algorithm: TCK
Input: unlabeled data set U, labeled data set L, testing data set T.
Output: Classification result $h(x)$

1. *for* $i=1$ *to* 3 *do*
2. $S_i \leftarrow Bootstrap\ Sample(L)$
3. $h_i \leftarrow Learn(s_i)$
4. $e_i^{'}=0.5; L_i^{'}=0.5$
5. *end for*
6. *repeat until none of* $h_i(i=1\ to\ 3)$ *changes*
7. *for* $i=1$ *to* 3 *do*
8. $L_i \leftarrow \varnothing;\ S_i^{'} \leftarrow \varnothing;\ update_i \leftarrow False$
9. $e_i \leftarrow Measure\ Cross\ Entropy(\frac{H_j+H_k}{2})(j,k\neq i)$
10. *if* $(e_i \prec e_i^{'})$
11. *then for every* $x \in U$ *do*
12. *if* $h_j(x)=h_k(x)(j,k\neq i)$
13. *then* $L_i \leftarrow L_i \bigcup \{x, h_j(x)\}$
14. *end for*
15. *if* $(L_i^{'}=0)$
16. *then* $L_i^{'} \leftarrow [\frac{e_i}{e_i^{'}-e_i}+1]$
17. *if* $(L_i^{'} \prec \mid L_i \mid)$
18. *then if* $(e_i \mid L_i \mid \prec e_i^{'} L_i^{'})$
19. *then* $update_i \leftarrow True$

20. *else if* $L_i^{'} \succ \frac{e_i}{e_i^{'} - e_i}$
21. *then* $L_i \leftarrow Subsample(L_i, [\frac{e_i^{'} L_i^{'}}{e_i} - 1])$
22. $update_i \leftarrow True$
23. *end for*
24. *for* $i = 1$ *to* 3 *do*
25. if $update_i \leftarrow True$
26. *then for every* $l \in L_i$ *do*
27. *if* $(k^{'} \geq 2)$
28. *then* $S_i^{'} \leftarrow l$
29. *end for*
30. *then* $h_i \leftarrow Learn(L \bigcup S_i^{'}); e_i^{'} \leftarrow e_i; L_i^{'} \leftarrow | L_i^{'} |$
31. *end for*
32. *end repeat*
33. $Output: h(x) \leftarrow \arg\max \sum_{h_i(x)=y} 1$

Steps 1-5 of the pseudocode use Bootstrap sampling to obtain three different base classifiers, and initialize the error parameter $e_i^{'}$ and the pseudo labeling scale $L_i^{'}$. Step 9 calculates the cross entropy to estimate the classification error e_i of classifier h_i. The predicted distribution is first obtained by predicting the labeled data set L, and then the cross entropy H_j and H_k are calculated with the practical distribution of the labeled data set L by Eq(8) to obtain the error $e_i = (H_j + H_k)/2$; Step 10 determine whether the error e_i using cross entropy is smaller than the initial classification error $e_i^{'}$; Step 11 - 14 is h_j, h_k $(j, k \neq i)$ expands the h_i training set by adding pseudo labeling by voting, and when the two classifiers are labeled consistently, pseudo labels are added for unlabeled samples; Step 15-23 ensure that the training set of pseudo labels can improve the algorithm performance; Steps 24-29 use the K-nearest neighbors idea to identify and remove the noise in the pseudo labels; Finally, the majority voting method is used to predict the output classification label of the label category of the test data set.

4. Experimental results and analysis

4.1 Experimental data set

In order to verify the effectiveness of the algorithm proposed in this paper, this study uses 12 datasets [38] from the UCI machine learning database (see **Table 1**) and the network intrusion data set NSL-KDD for experiments. The NSL-KDD dataset is a commonly used network intrusion data set, improved from KDD-CUP99, NSL-KDD removes the redundant data of KDD-CUP99, the data has 41 attributes, the last column is labeled category, four categories of attacks are Dos: an attack that attempts to shut down traffic to and from a target system,

Probing: an attack that attempts to extract information from a network, U2R: an attack that starts with a regular user account and attempts to access a system or network as the super user (root), R2L: an attack that attempts to gain local access to a remote machine and a normal category Normal. To meet the experimental needs, the data set is divided into 3 parts: the test data set T, the labeled data set L, and the unlabeled data set U account for 20%, 20%, and 60%, respectively.

Table 1. UCI data set

Data set	Attribute	Size	Class
ionosphere	34	351	2
winewhite	11	4898	7
bupa	6	345	2
haberman	3	306	2
german	24	1000	2
heart	13	270	2
vehicle	18	946	4
cmc	9	1473.	3
ILPD	10	583	2
Iris	4	150	3
wdbc	30	569	2
dataR2	10	116	2

4.2 Experimental setup

To verify the effectiveness of the proposed algorithm and to test the effect of cross entropy and K-nearest neighbors strategies on the performance of Tri-training algorithms, four sets of experiments were done, namely Tri-training algorithm, Tri-training algorithm based on cross entropy (TCE), Tri-training algorithm based on K-nearest neighbors (TKNN) and Tri-training algorithm based on cross entropy and K-nearest neighbor (TCK). Tri-training is the benchmark algorithm. In this study, the algorithm performance is evaluated using four metrics: *accuracy*, *precision*, *recall* and *F-measure*. **Table 2** is the confusion matrix associated with the indexes.

Table 2. Confusion matrix

Actual class	Predicted class	
	yes	no
yes	TP	FN
no	FP	TN

In the confusion matrices, TP, TN denote the correctly classified positive and negative classes, FP, FN denote the incorrectly classified positive and negative classes. Among the indexes, *accuracy* is used to count the percentage of tuples correctly identified by the classifier; *precision* calculates the percentage of positive tuples to be actually positive; *recall* counts the percentage of positive tuples predicted to be positive; and *F-measure* is the harmonic mean of precision and recall. The closer the value of these metrics is to 1, the better the performance of the algorithm. The performance metrics are calculated as follows:

$$accuracy = \frac{TP + TN}{P + N} \tag{8}$$

$$precision = \frac{TP}{TP + FP} \tag{9}$$

$$recall = \frac{TP}{TP + FN} \tag{10}$$

$$F - measure = \frac{2 \times precision \times reacall}{precision + recall} \tag{11}$$

4.3 Experimental results analysis of UCI data set

The algorithms were evaluated on the basis of *accuracy*, *precision*, *recall*, *F-measure* by the UCI data set, and in the case of multi-classification problems, *precision* and *F-measure* were calculated using the weighted mean, and *recall* was calculated using the macro-average. The experimental results are shown in **Table 3-6**, and the best results are marked in bold.

Table 3. Accuracy

Data set	Tri-training	TCE	TKNN	TCK
ionosphere	**0.9257**	**0.9257**	0.9100	0.9114
winewhite	**0.5821**	0.5779	0.5733	0.5809
bupa	0.6203	0.6334	0.6420	**0.6478**
haberman	0.7083	0.7246	0.7283	**0.7311**
german	0.7110	**0.7445**	0.6950	0.7120
heart	0.7800	0.8037	0.7870	**0.8056**
vehicle	0.6900	0.7047	0.6994	**0.7148**
cmc	0.5170	0.5231	0.5306	**0.5353**
ILPD	0.6819	0.6836	0.6897	**0.6991**
Iris	0.9434	0.9467	**0.9600**	**0.9600**
wdbc	0.9608	0.9639	0.9573	**0.9654**
dataR2	**0.6174**	0.5913	0.6000	0.6016

Table 4. Precision

Data set	Tri-training	TCE	TKNN	TCK
ionosphere	0.8547	**0.8600**	0.7917	0.8191
winewhite	0.6901	0.6756	0.6850	**0.6920**
bupa	0.3767	0.3902	0.3390	**0.4200**
haberman	0.8980	0.9129	0.9326	**0.9392**
german	0.1370	**0.1770**	0.1224	0.1113
heart	0.6648	0.7016	0.6396	**0.7192**
vehicle	0.7618	0.7619	0.7623	**0.7673**
cmc	0.5487	0.5421	0.5447	**0.5592**
ILPD	0.8678	0.8745	0.8801	**0.8848**
Iris	0.9384	0.9475	0.9401	**0.9556**
wdbc	**0.9525**	0.9494	0.9331	0.9443
dataR2	**0.5294**	0.5075	0.4478	0.3989

Table 5. Recall

Data set	Tri-training	TCE	TKNN	TCK
ionosphere	0.9344	0.9181	**0.9582**	0.9343
winewhite	0.5010	0.4849	0.4697	**0.5127**
bupa	0.5681	0.5927	0.5835	**0.6137**
haberman	0.7698	0.7393	0.7729	**0.7738**
german	0.6644	0.6998	0.6389	**0.7022**
heart	0.7824	0.8275	0.8232	**0.8367**
vehicle	**0.6832**	**0.6832**	0.6575	0.6825
cmc	0.4920	0.4994	**0.5070**	0.4952
ILPD	0.7474	0.7351	0.7500	**0.7842**
Iris	**0.9594**	0.9517	0.9428	0.9527
wdbc	0.9440	0.9440	0.9450	**0.9497**
dataR2	0.5379	0.6148	0.4937	**0.6605**

Table 6. F-measure

Data set	Tri-training	TCE	TKNN	TCK
ionosphere	**0.8910**	0.8906	0.8700	0.8674
winewhite	**0.6118**	0.6074	0.6095	0.6084
bupa	0.4545	0.4556	0.4156	**0.4677**
haberman	0.8268	0.8102	0.8280	**0.8390**
german	0.2219	**0.2665**	0.2026	0.1791
heart	0.7390	0.7236	**0.7454**	0.7020
vehicle	0.7201	0.7140	0.6985	**0.7259**
cmc	0.5279	0.5283	0.5422	**0.5441**
ILPD	0.7951	0.8050	0.7956	**0.8055**
Iris	0.9308	0.9291	0.9331	**0.9418**
wdbc	0.9478	0.9423	0.9486	**0.9492**
dataR2	**0.5186**	0.5151	0.4439	0.4497

According to **Tables 3-6**, we can see that TCK has a clear advantage over the 12 UCI datasets. The data sets with superiority in 4 indexes (*accuracy*, *recall*, *F-measure* and *precision*) are 8, 8, 8 and 7 respectively, indicating that the TCK has different degrees of improvement for each metric. Tri-training achieved good results on *accuracy*, *recall*, *F-measure* and *precision* on only 3, 3, 2, and 3 datasets. On *accuracy*, Tri-training and TCE only have one data set tied for first place, and TKNN and TCK also only have one data set tied for first place.

To further analyze the performance of the four algorithms, their combined performance is analyzed from a statistical point of view. In this study, the Friedman test was introduced to test the rank mean of the four evaluation indicators *accuracy*, *recall*, *F-measure* and *precision*. The Friedman test is a significant difference test, and its rank mean value reflects the comprehensive performance of the algorithm. The larger the rank mean value, the better the comprehensive performance. The rank mean table of Friedman test is shown in **Table 7**.

Table 7. Rank mean values of indexes in four algorithms

Evaluation indicators	Tri-training	TCE	TKNN	TCK
Accuracy	1.88	2.46	2.13	**3.54**
Precision	2.33	2.58	1.92	**3.17**
Recall	2.17	2.25	2.17	**3.42**
$F-measure$	2.58	2.17	2.33	**2.92**
Mean	2.24	2.37	2.14	**3.26**

As shown in **Table 7**, TCK ranked first in all four indexes. TCK ranked first and TCE ranked second, Tri-training ranked third, and TKNN ranked fourth in the mean value of the four indexes. The reason for TKNN being in the last position is that it is not combined with cross entropy, which leads to a high initial classification error rate and eliminates the correct labels during noise removal, resulting in bad classification performance.

4.4 Experimental results analysis of NSL-KDD NID dataset

Due to the large size of NSL-KDD data, 10% of the NSL-KDD data set is selected for the experiment. *Accuracy* was used as an evaluation index in the experiments, the experimental results are shown in **Table 8**.

Table 8. Accuracy of the four algorithms in NSL-KDD

Data set	Tri-training	TCE	TKNN	TCK
Normal	0.9545	0.9626	0.9542	**0.9630**
Dos	0. 9913	0.9931	0.9933	**0.9966**
Probing	0. 9886	0.9861	0.9911	**0.9931**
U2R	0. 8043	0.8000	0.8108	**0.9000**
R2L	0. 9796	0.9812	0.9791	**0.9850**

As shown in **Table 8**, the NSL-KDD data set has the highest *accuracy* obtained by TCK algorithm in 4 classes of attack types and 1 class of normal classes. TCE ranked second in the categories Normal, R2L, and TKNN ranked second in the categories Dos, Probing, and U2R. As observed, the improvement is more obvious on U2R, which is due to the small U2R training data set. The improvement effect of the rest of the classes is a little weaker, because their training set has been able to train a better model due to the larger data size.

5. Conclusions

This paper presents a Tri-training network intrusion detection algorithm based on cross-entropy and K-nearest neighbors (TCK). Since the learning process of the Tri-training algorithm generates training noise due to mislabeling, the TCK algorithm replaces the classification error rate with cross entropy to reduce the difference between the practical distribution and the predicted distribution; K-nearest neighbors are used to remove the pseudo labeling noise. By examining the UCI data set with the network intrusion data set NSL-KDD, the TCK algorithm has significantly improved in four indexes such as *accuracy*, *recall*, *F-measure* and *precision* compared with Tri-training, TCE and TKNN algorithms, and has better detection effect in NID. The key to optimize the performance of Tri-training algorithm accurate identification and effective removal of noises, which shall be further investigated in the future.

Acknowledgement

This research was supported by the National Natural Science Foundation of China under Grant (Nos. 52069014, 61962036), the Jiangxi Province Department of Education Science and Technology Project under Grant (No. GJJ180940).

References

[1] W. H. Luo, C. D. Xu, “Network Intrusion Detection Based on Improved MajorClust Clustering,” *Netinfo Security*, vol. 20, no. 2, pp. 14-21, 2020. Article (CrossRef Link)

[2] J. Zhao, D. D. Chen, R. B. Xiao, Z. H. Cui, H. Wang and I. Lee, “Multi-strategy ensemble firefly algorithm with equilibrium of convergence and diversity,” *Applied Soft Computing*, vol. 123, no. 1, pp. 108938, Jul. 2022. Article (CrossRef Link)

[3] H. S. Wu and R. B. Xiao, “Flexible wolf pack algorithm for dynamic multidimensional knapsack problems,” *Research*, vol. 2020, pp. 1762107, Feb. 2020. Article (CrossRef Link)

[4] H. S. Wu, J. J. Xue, R. B. Xiao and J. Q. Hu, “Uncertain bilevel knapsack problem based on improved binary wolf pack algorithm,” *Frontiers of Information Technology & Electronic Engineering*, vol. 21, no. 9, pp. 1356-1368, Jun. 2020. Article (CrossRef Link)

[5] J. Zhao, L. Lv, H. Wang, H. Sun, R. X. Wu and Z. F. Xie, “Particle Swarm Optimization based on Vector Gaussian Learning,” *KSII Transactions on Internet and Information Systems*, vol. 11, no. 4, pp. 2038-2057, Apr. 2017. Article (CrossRef Link)

[6] L. Lv, X. D. Zhou, P. Kang, X. F. Fu, X. M. Tian, “Multi-Objective Firefly Algorithm with Hierarchical Learning,” *Journal of Network Intelligence*, vol. 6, no. 3, pp. 411-427, Aug. 2021.

[7] J. Zhao, W. P. Chen, R. B. Xiao, J. Ye, “Firefly algorithm with division of roles for complex optimal scheduling,” *Frontiers of Information Technology & Electronic Engineering*, vol. 22, no. 10, pp. 1311-1333, Oct. 2021. Article (CrossRef Link)

[8] L. Lv, J. Y. Wang, R. X. Wu, H. Wang, I. Lee, “Density peaks clustering based on geodetic distance and dynamic neighborhood,” *International Journal of Bio-Inspired Computation*, vol. 17, no. 1, pp. 24-33, Feb. 2021. Article (CrossRef Link)

[9] S. Y. Wu, J. Yu, X. P. Fan, “Intrusion Detection Algorithm Based on Tri-training,” *Computer Engineering*, vol. 38, no. 6, pp. 158-160, 2012. Article (CrossRef Link)

[10] J. W. Liu, Y. Liu, X. L. Luo, “Semi-supervised learning methods,” *Chinese Journal of Computers*, vol. 38, no. 8, pp. 1592-1617, 2015.

[11] O. Chapelle, B. Scholkopf and A. Eds, "Semi-Supervised Learning (Chapelle, O. et al., Eds.; 2006) [Book reviews]," *IEEE Transactions on Neural Networks*, vol. 20, no. 3, pp. 542-542, Mar. 2009. Article (CrossRef Link)

[12] Z. H. Zhou, “Disagreement-based Semi-supervised learning,” *Acta Automatica Sinica*, vol. 39, no. 11, pp. 1871-1878, 2013. Article (CrossRef Link)

[13] R. A. Fisher, “The use of multiple measurements in taxonomic problems,” *Annals of eugenics*, vol. 7, no. 2, pp. 179-188, Sep. 1936. Article (CrossRef Link)

[14] D. J. Miller, H. S. Uyar, “A mixture of experts classifier with learning based on both labelled and unlabelled data,” in *Proc. of the 9th International Conference on Neural Information Processing Systems (NIPS 1996)*, Cambridge, MA, USA, pp. 571–577, 1996.

[15] A. Blum, S. Chawla, “Learning from labeled and unlabeled data using graph mincuts,” in *Proc. of the 8th international conference on Machine learning (ICML 2001)*, San Francisco, CA, USA, pp. 19-26, 2001.

[16] A. Blum and T. Mitchell, “Combining labeled and unlabeled data with co-training,” in *Proc. of the eleventh annual conference on Computational learning theory (COLT 1998)*, New York, NY, USA, pp. 92–100, Jul. 1998. Article (CrossRef Link)

[17] Z. H. Zhou and M. Li, "Tri-training: exploiting unlabeled data using three classifiers," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 11, pp. 1529-1541, Nov. 2005. Article (CrossRef Link)

[18] Y. Q. Hu, Q. Y. Qiu, X. Yu, “Semi-supervised patent text classification method based on improved Tri-training algorithm,” *Journal of Zhejiang University (Engineering Science)*, vol. 54, no. 2, pp. 331-339, 2020. Article (CrossRef Link)

[19] D. M. Li, J. W Mao, S. Fuke, “A Novel Semi-supervised Adaboost Technique Based on Improved Tri-training,” in *Proc. of Australasian Conference on Information Security and Privacy(ACISP 2019)*, Cham, GERMANY, pp. 669-678, 2019. Article (CrossRef Link)

[20] Y. Zhang, R. R. Chen, J. Zhang, “Safe tri-training algorithm based on cross entropy,” *Journal of Computer Research and Development*, vol. 58, no. 1, pp. 60-69, 2021.
[21] J. W. Mo, P. Jia, “Semi-supervised classification model based on ladder network and improved tri-training,” *Acta Automatica Sinica*, vol. 48(08), 2022. Article (CrossRef Link)
[22] Y. F. Li, D. M. Liang, “Safe semi-supervised learning: a brief introduction,” *Frontiers of Computer Science*, vol. 13, no. 4, pp. 669-676, Jun. 2019. Article (CrossRef Link)
[23] D. Angluin, P. Laird, “Learning from noisy examples,” *Machine Learning*, vol. 2, no. 4, pp. 343-370, Apr. 1988. Article (CrossRef Link)
[24] S. Kullback, R. A. Leibler, “On information and sufficiency,” *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79-86, Mar. 1951.Article (CrossRef Link)
[25] I. Goodfellow, Y. Bengio, A. Courville, *Deep learning. Massachusetts*, USA : MIT press, 2016.
[26] D. J. C. MacKay, *Information theory, inference and learning algorithms*, Cambridge, UK: Cambridge university press, 2003.
[27] R. Y. Rubinstein, “Optimization of computer simulation models with rare events,” *European Journal of Operational Research*, vol. 99, no. 1, pp. 89-112, May. 1997. Article (CrossRef Link)
[28] G. Too, X. J. Cheng, F. B. Qin, "Incremental clustering algorithm via cross-entropy," *Journal of Systems Engineering and Electronics*, vol. 16, no. 4, pp. 781-786, Dec. 2005.
[29] H. Liu, Z. Liu, W. Jia, D. Zhang and J. Tan, “A Novel Imbalanced Data Classification Method Based on Weakly Supervised Learning for Fault Diagnosis,” *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 1583-1593, Mar. 2022. Article (CrossRef Link)
[30] B. Santosa, “Application of the Cross-Entropy Method to Dual Lagrange Support Vector Machine,” in *Proc. of the 5th International Conference on Advanced Data Mining and Applications(ADMA 2009)*, Beijing, CHINA, pp. 595-602, 2009. Article (CrossRef Link)
[31] T. Cover and P. Hart, “Nearest neighbor pattern classification,” *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21-27, Jan. 1967. Article (CrossRef Link)
[32] Z. W. Wang, S. K. Wang, B. T. Wan, “A novel multi-label classification algorithm based on K-nearest neighbor and random walk,” *International Journal of Distributed Sensor Networks*, vol. 16, no. 3, Mar. 2020. Article (CrossRef Link)
[33] H. Xiao, F. Sun, Y. Liang, “A Fast Incremental Learning Algorithm for SVM Based on K Nearest Neighbors,” in *Proc. of 2010 International Conference on Artificial Intelligence and Computational Intelligence(ICCAI 2010)*, Sanya, China, pp. 413-416, 2010. Article (CrossRef Link)
[34] C. Ren, L. Sun, Y. Yu and Q. Wu, “Effective Density Peaks Clustering Algorithm Based on the Layered K-Nearest Neighbors and Subcluster Merging,” *IEEE Access*, vol. 8, pp. 123449-123468, Jun. 2020. Article (CrossRef Link)
[35] R. X. Wu, S. H. Yin, J. Zhao, P. W. Li, B. H. Liu, “Density Peaks Clustering based on Relative Density Estimating and Multi Cluster Merging,” *Control and Decision*, 2022. Article (CrossRef Link)
[36] J Zhao, Z. F. Yao, L. Lv, T. H. Fan, “Density peaks clustering based on mutual neighbor degree,” *Control and Decision*, vol. 36, no. 3, pp. 543-552, Mar. 2021. Article (CrossRef Link)
[37] J. S. Sánchez, R. Barandela, A. I. Marqués, et al, “Analysis of new techniques to obtain quality training sets,” *Pattern Recognition Letters*, vol. 24, no. 7, pp. 1015-1022, Apr. 2003. Article (CrossRef Link)
[38] D. Dua, C, Graff, UCI Machine Learning Repository. [Online]. Available: http://archive.ics.uci.edu/ml

Jia Zhao received the B.S. degree in computer science and technology from Nanchang Institute of Aeronautical Technology, Nanchang, China, in 2004, and the M.E. degree in computer application technology from Nanchang Hangkong University, Nanchang, China, in 2011, and the Ph.D. degree in information and communication engineering from Hohai University, Nanjing, China, in 2020.He is currently a professor with the School of Information Engineering, Nanchang Institute of Technology, Nanchang, China. He is also the Director of Nanchang Key Laboratory of Big Data and Computational Intelligence. His research interests include big data analysis; artificial intelligence theory; deep learning and reinforcement learning.

Song Li is currently working towards the MSc degree at the School of Information Engineering, Nanchang Institute of Technology, China. His research interest focuses on Semi-supervised classification.

Runxiu Wu received his MA in Technology of Computer Application from the Nanchang University, Nanchang, China, and BA in Computer Software from the Jiangxi Normal University, Nanchang, China. Her research interests include swarm intelligence and rough set.

Yiying Zhang received the B.E. degree in Northeast Normal University in 1996 and the M.Ec. degree in Northeastern University in 2003 in China, and PhD degree in Korea University in 2010. His research interests include network security, wireless sensor network, Internet of Things, Smart Grid etc. He is currently a professor in Tianjin University of Science & Technology.

Bo Zhang received the B.S. degree from Hohai University in 2007, and the M.Sc. degree from Southeast University in 2012 and PhD degree in the School of Computer Science and Engineering, Nanjing University of Science and Technology in 2018. His research interests include cybersecurity in smart grid and network security situation awareness.

Longzhe Han received his Ph.D degree in the Dept. of Computer Science at Korea University in 2013. Currently, he is with the school of Information Engineering at Nanchang Institute of Technology as a professor. His research interests include cognitive radio networks, future Internet, network security, multimedia communications, machine-to-machine communication and heterogeneous network in 5G.