Hindawi
Security and Communication Networks
Volume 2021, Article ID 6638134, 13 pages
https://doi.org/10.1155/2021/6638134

Research Article

Representativeness-Based Instance Selection for Intrusion Detection

Fei Zhao,[1] **Yang Xin**,[1,2] **Kai Zhang**,[1] **and Xinxin Niu**[1,2]

[1]*National Engineering Laboratory for Disaster Backup and Recovery, Information Security Center, School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2]*Guizhou Provincial Key Laboratory of Public Big Data, Guizhou University, Guiyang 550025, China*

Correspondence should be addressed to Yang Xin; yangxin@bupt.edu.cn

Received 26 November 2020; Revised 24 January 2021; Accepted 26 February 2021; Published 13 March 2021

Academic Editor: Entao Luo

With the continuous development of network technology, an intrusion detection system needs to face detection efficiency and storage requirement when dealing with large data. A reasonable way of alleviating this problem is instance selection, which can reduce the storage space and improve intrusion detection efficiency by selecting representative instances. An instance is representative not only in its class but also in different classes. This representativeness reflects the importance of an instance. Since the existing instance selection algorithm does not take into account the above situations, some selected instances are redundant and some important instances are removed, increasing storage space and reducing efficiency. Therefore, a new representativeness of instance is proposed and considers not only the influence of all instances of the same class on the selected instance but also the influence of instances of different classes on the selected instance. Moreover, it considers the influence of instances of different classes as an advantageous factor. Based on this representativeness, two instance selection algorithms are proposed to handle balanced and imbalanced data problems for intrusion detection. One is a representative-based instance selection for balanced data, which is named RBIS and selects the same proportion of instances from each class. The other is a representative-based instance selection for imbalanced data, which is named RBIS-IM and selects important majority instances according to the number of instances of the minority class. Compared with other algorithms on the benchmark data sets of intrusion detection, experimental results verify the effectiveness of the proposed RBIS and RBIS-IM algorithms and demonstrate that the proposed algorithms can achieve a better balance between accuracy and reduction rate or between balanced accuracy and reduction rate.

1. Introduction

Along with the continuous development of network technology and 5G, smart systems are becoming more and more common in all fields of human life, such as finance, agriculture, and education. However, smart systems have become the target of many new attacks, which not only cause significant financial damage and personal information leakage but also hinder the large-scale deployment of smart systems in practice. As intrusion detection technology can effectively protect smart systems and detect attacks, the development of intrusion detection technology has attracted the attention of countries all over the world [1, 2]. From the perspective of classification, the main goal of building an intrusion detection system (IDS) is to train a classifier that can distinguish between normal and intrusive data from the original network data set.

The IDS based on machine learning has become an important part of IDS [3], which directly uses a large amount of network data to detect attacks. These network data can result in wasting time and storage space for IDS. Moreover, the redundant data and noise in these data can affect the performance of IDS. But, instance selection is used for IDS to select important data from the original data to achieve two goals. One is to reduce the number of instances required by IDS in the training phase, thereby saving time and reducing the amount of calculation for training the classifier; the other is that through effective instances, the performance of the trained classifier can be effectively improved [4–6].

In recent years, many instance selection techniques have been proposed to improve the performance of IDS [7–15]. However, in terms of the factors and application areas of instance selection, there are mainly four problems in existing instance selection algorithms.

Firstly, only the influence of a portion of instances is taken into account for selecting the instance [7–9]. For instance, the instance selection algorithm based on partition and cluster center (PCIS) [7] selects representative instance by considering K nearest neighbor instances of the same class; The binary nearest neighbor tree algorithm (BNNT) [8] and the constraint nearest neighbor-based instance reduction algorithm (CNNIR) [9] select representative instance by K nearest neighbor instances of the selected instance. As these instance selection algorithms only consider the influence of a portion of instances and ignore the influence of remaining instances, some important instances are not selected.

Secondly, the influence of instances of different classes is regarded as an adverse factor selecting the instance. For the instances of the same class, the instance selection algorithm based on a ranking procedure (ISAR) [10] and the ranking-based instance selection algorithm (RIS) [11] only select instances representing the same class and remove instances representing different classes. Some selected instances are not representative because the influence of instances of different classes is considered as an adverse factor.

Thirdly, some instance selection algorithms use sampling to select the instance. Instance selection algorithm based on hierarchical data topology [12] uses hierarchical sampling to deal with large-scale problems of data sets. This algorithm combines the random subset selection (RSS) with the topology-based selection (TBS) to select important instances, which is a subset of original instances. Since sampling is used to select instance, some important instances are still removed, which contain the information of or original instances.

Finally, in the field of intrusion detection, only a few algorithms are used to deal with imbalanced data, and most of the instance selection algorithms are used to deal with the problem of balanced data [13–15]. Data imbalance is known as instance imbalance. For the binary classification problem, under normal circumstances, the proportion of positive and negative instances should be relatively close, and many existing classification models are based on this assumption. However, in some specific scenarios, the proportion of positive and negative instances may vary greatly, which reduces the accuracy of minority class, which has smaller instances. Therefore, instance selection algorithms, which deal with imbalanced data, need to be strengthened.

Given the above four problems, the factors considered for instance selection include: (1) the influence of all instances of the same class on the selected instance; (2) the influence of instances of different classes on the selected instance; (3) the influence of different classes of instances as an advantageous factor; and (4) the instance selection algorithm should be applied to the balanced and unbalanced domains for intrusion detection. As the existing instance selection algorithm does not take into account the above four factors, some selected instances are redundant and some important instances are removed, increasing storage space and reducing efficiency. Therefore, for the first three factors, we propose a new concept of representativeness of instance. This concept is used to express the importance of the instance. Considering the fourth factor, we propose two representativeness-based instance selections, which are named RBIS and RBIS-IM. RBIS algorithm is used to handle balanced data and select the same proportion of instances from each class. And RBIS-IM algorithm is used to deal with imbalanced data and select important majority instances according to the number of instances of the minority class. Finally, the experimental results verify the effectiveness of proposed algorithms. Two algorithms can reduce the size of the training set while maintaining or even increasing accuracy (ACC) and balanced accuracy (BA).

The main contributions of this paper are as follows:

(1) A new concept of instance representativeness is proposed to represent the importance of an instance. In terms of instance representativeness, we consider not only the representativeness of the instance within its class but also the representativeness of the instance within different classes. The two representativenesses are advantageous factors;

(2) To deal with balanced data problem, the RBIS algorithm, which is based on instance representativeness, is designed to select the same proportion of normal instances and attack instances to improve intrusion detection efficiency. Compared with other algorithms on the benchmark data sets of intrusion detection, RBIS algorithm can achieve a better balance between accuracy and reduction rate.

(3) To handle imbalanced data problem, the RBIS-IM algorithm, which is based on instance representativeness, is designed to select the same number of normal instances and attack instances. Compared with other algorithms on the benchmark data sets of intrusion detection, RBIS-IM algorithm can achieve a better balance between balanced accuracy and reduction rate.

The paper is structured as follows. In Section 2, we introduce the basic concepts of instance selection technique. Section 3 reports a new concept of instance representativeness and two representativeness-based instance selection algorithms that are used with regard to balanced and imbalanced problems, respectively. Experimental results with two representative-based instance selection algorithms are shown in Section 4. Finally, conclusions with a discussion on future work are presented in Section 5.

2. Instance Selection Technique

In this section, the basic concepts of the instance selection technique are introduced. The instance selection is to select important instances and eliminate redundant instances from the original data. These selected instances can contain the total effective information of the original data. Suppose X

represents the original data; S represents the selected instances; so S is the subset of X, i.e., $S \subset X$ and $|S| \ll |X|$ .Using the instance subset S for IDS can improve detection efficiency and reduce storage requirements. According to the distribution of instances and selection strategies, instances with different locations play different roles in the classification process. In general, these algorithms are divided into three categories: condensation, edition, and hybrid.

The condensation algorithm considers that instances close to the boundary play an important role in the classification process, just like SVM. It preserves the boundary instances by deleting the interior instances of each class [16–18]. In the field of intrusion detection, nature-inspired instance selection technique (NIIS) [19] and instance selection technique based on cuckoo search and bat algorithm (CSBAIS) [20] are proposed to improve the training speed and accuracy of the support vector machine (SVM). The NIIS algorithm applies the lower polling algorithm and social spider algorithm to select instances near the boundary. CSBAIS algorithm uses cuckoo search and bat algorithm to select instances near the boundary. But these algorithms remove some important internal instances too.

The edition algorithm is the opposite of the condensation algorithm. It tends to smooth the class boundary by deleting the boundary instances [21–23]. The instance selection algorithm based on K-means and K-nearest neighbor (KMKNNIS) [24] is proposed to select important internal instances. Those instances near the boundary are removed. The penalty-reward-based instance selection method [25] is to select instances by removing noise and boundary instances. These algorithms can ignore some critical boundary instances.

Finally, the hybrid algorithm combines the condensation algorithm with the edition algorithm to obtain a smaller subset and an acceptable accuracy in the testing set [9, 26–28]. PCIS [7] algorithm applies the partition and cluster center to select the instance. First, the algorithm only considers the influence of k instances of the same class on the selected instances and does not consider the influence of all instances of the same class. Second, the algorithm only uses the class center instances of different classes and does not use the information of all instances of different classes. Third, the instance information of different classes is regarded as adverse information. ISAR [10] and RIS [11] algorithms select important instances by sorting the instances. In the process of sorting instances, although the influence of all instances of different classes is considered, it is regarded as adverse information. BNNT algorithm uses the binary nearest neighbor tree to select the instance [8]. The algorithm only considers the k nearest neighbor instances of the selected instance and does not consider the influence of remaining instances. Moreover, the algorithm needs to delete internal instances to select instances. The CNNIR algorithm uses the constraint nearest neighbor to select the instance [9]. The algorithm does not consider the influence of remaining instances.

To sum up, there are mainly four factors in the instance selection process: (1) the influence of all instances of the same class on the selected instance; (2) the influence of instances of different classes on the selected instance; (3) the influence of different classes of instances as an advantageous factor; and (4) the instance selection algorithm should be applied to the balanced and imbalanced domains for intrusion detection. Since the above four factors are not taken into account in the existing instance selection algorithm, some selected instances are redundant and some important instances are removed, increasing storage space and reducing efficiency. Therefore, we propose two algorithms to select important instances without deleting internal instances, which can handle balanced and imbalanced data problems. Meanwhile, the proposed algorithms consider not only the influence of all instances of the same class on the selected instances but also the influence of instances of different classes and take the influence of instances of different classes as an advantageous factor.

3. Proposed Algorithms

In this section, we introduce the proposed representativeness-based instance selection algorithms. In the first subsection, we introduce a new instance representativeness. In the next two subsections, two representativeness-based algorithms are introduced, which are used to deal with balanced and imbalanced data problems.

3.1. Proposed Instance Representativeness. The key factor of instance selection is to decide which instance is representative, which makes the selected instance subset representativeness of the original data. Selecting representative instances, we should consider not only the representativeness of the selected instance category but also the representativeness of different categories. In other words, the instance selected has the information of its category and different categories. And the influence of instances of different categories is seen as an advantageous factor.

Suppose that X is a training instance set containing normal and attack categories, $X = \{(x_1, c_1), \ldots, (x_n, c_2)\}$. X has n instances; x_i is a d-dimensional instance; c expresses the classes of instances and $c = \{c_1, c_2\}$; c_1 is the class of normal instances X_n and c_2 is the class of attack instances X_a; X is composed of X_n and X_a.

The representation of any instance x_i in the training set X is as follows:

$$R(x_i, c) = \{Q(x_i, c_r)\} * \{Q(x_i, c_p)\}. \quad (1)$$

The first half of formula (1) represents the representativeness of instance x_i in its category; the second half shows the representativeness of instance x_i in different categories; $c_r, c_p \subset \{c_1, c_2\}$ and $r \neq p$; c_r represents the category of instance x_i; c_p is a different category from instance x_i.

To realize $Q(x_i, c_r)$ or $Q(x_i, c_p)$ in formula (1), the Euclidean distance $d(x_i, x_j)$ can be used to represent the relation of two instances. The representativeness between an instance and a class is inversely proportional to the sum of its Euclidean distances of the instance and remaining instances of the same class. And the representativeness of instances of different categories is considered.

Thus, formula (1) is transformed into the following form:

$$R(x_i, c) = \left\{ \frac{1}{\left(\sum_{c_i = c_j, j=1}^{n_i} d(x_i, x_j) \right)} \right\} * \left\{ \frac{1}{\sum_{c_i \neq c_j, j=1}^{n_j} d(x_i, x_j)} \right\}, \tag{2}$$

where n_i is the number of instances in the same category as x_i; n_j is the number of instances in a different category from x_i. The expression c_i shows the category of instance x_i; the expression $c_i = c_j$ shows that instances x_i and x_j are the same category; the expression $c_i \neq c_j$ shows that instances x_i and x_j are different categories; if x_i and x_j are the same class, $i \neq j$.

Calculating the representativeness of instance $R(x_i, c)$, three factors are considered: (1) the influence of all instances of the same class on the selected instance; (2) the influence of instances of different classes on the selected instance; and (3) the influence of different classes of instances as an advantageous factor. The proposed representativeness of instance reflects the importance of instance. In Section 4.3, compared with other algorithms on the benchmark data sets of intrusion detection, experimental results verify the effectiveness of the representativeness of instance $R(x_i, c)$.

3.2. Representativeness-Based Instance Selection for Balanced Data. To handle balanced data problem, a representativeness-based instance selection algorithm is proposed to select representative instances, which is called RBIS, to improve accuracy (ACC) and reduce reduction rate (RR) for IDS. Through the RBIS algorithm, the same proportion of instances for each class is selected. Algorithm 1 shows the pseudo-code of the RBIS algorithm.

In Algorithm 1, original instances X are composed of normal instances X_n and attack instances X_a.S is the set of selected instances from original instances X;S_n is the set of selected normal instances from X; S_a is the set of selected attack instances from X; the parameter t is the ratio of selected instances by cross-validation or validation set. Firstly, in lines 3–5 of Algorithm 1, the representativeness of each instance is calculated. According to normal instances X_n and attack instances X_a, S_n and S_a are initialized. Secondly, according to representativeness $R(x_i, c)$, representativeness $R(x_i, c)$ and training set X are sorted in descending order (lines 6 and 7). Meanwhile, S_n and S_a are sorted in descending order. Thirdly, from line 8 to line 11, according to the cross-validation or validation data, 1-NN is used as the classifier. The parameter t with the best accuracy is selected and the range of parameter t is [0, 1]. In Section 4.3, the selection process of parameter$t$ is shown by Figures 1 and 2. According to parameter t, the first $|S_n| * t$ instances and the first $|S_a| * t$ instances are selected in S_n and S_a, respectively. Finally, according to S_n and S_a, S is determined.

Figure 3 with two dimensions is used to demonstrate the instance selection process of the RBIS algorithm. Figure 3(a) shows two types of original data, which are normal and attack instances. The circle is "Class One," which represents the normal instance; the square is "Class Two," which represents the attack instance. And there are 10 normal instances and 10 attack instances. According to their representativeness, the instances of each class are ranked in Figure 3(b). The numbers around the graph indicate the degree of representation of the instance. The smaller the number, the more representative the instance is. For example, in normal instances, the Number "1" is the most representative and the Number "10" is the least representative. In Figure 3(c), according to the parameter t, the same proportion of instances are selected in each class. When the parameter t is 0.6, the first six instances of each class are selected.

The RBIS algorithm is based on the representativeness $R(x_i, c)$ of instance. The selected instances of RBIS algorithm contain the information of original data. The efficiency of the RBIS algorithm is related to the accuracy (ACC) and reduction rate (RR). Compared with other algorithms on the benchmark data sets of intrusion detection, experimental results, which are shown in Section 4.3, prove that the RBIS algorithm is effective and achieves a better balance between accuracy and reduction rate. As the same proportion of instances for each class is selected, the RBIS algorithm can handle the balanced data problem.

According to Algorithm 1 and formula (2), the time complexity of the proposed algorithm is mainly related to the calculation of instance distance between the same and different classes. Therefore, the time complexity of the algorithm is $O(N^2) + O(M)$, where N represents the total number of training instances and M represents the number of experiments conducted by the classifier when selecting the parameter t. As $O(M)$ is far less than $O(N^2)$, the time complexity of RBIS is $O(N^2)$.

3.3. Representativeness-Based Instance Selection for Imbalanced Data. To solve the imbalanced data problem, a representativeness-based instance selection algorithm is proposed, which is called RBIS-IM. Through the RBIS-IM algorithm, the same number of instances for each class is selected to improve balanced accuracy (BA) and reduce reduction rate (RR) for IDS.

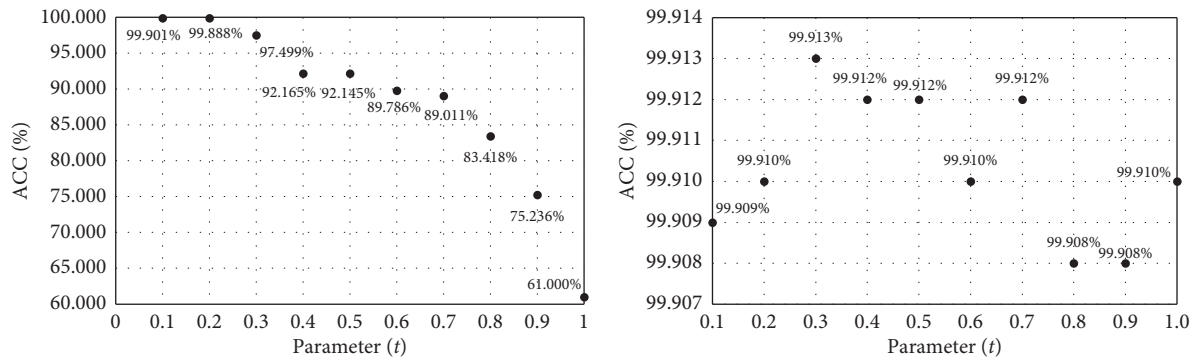
Algorithm 2 shows the pseudo-code of the RBIS-IM algorithm. Like Algorithm 1, Algorithm 2 is based on the representativeness of instance. Original instances X are composed of normal instances X_n and attack instances X_a. X_n and X_a are called the majority class and the minority class, respectively. The difference in the number between X_n and X_a is huge. S is the set of selected instances from original instances X; S_n is the set of selected normal instances from X; S_a is the set of selected attack instances from X; the parameter t is the ratio of selected instances by cross-validation or validation set.

In the process of instance selection, the number of selected instances of the majority class not only depends on the number of instances of the minority class but also is the same as that selected of the minority class. Firstly, in lines 3–5 of Algorithm 2, the representativeness of each instance is calculated. According to X_n and X_a, S_n and S_a are initialized. Secondly, according to representativeness,

Input: X: Training data set; t: the Ratio of selected instance by cross-validation or validation set; X_n: the Set of normal instances; X_a: the Set of attack instances.
Output: $S = S_n \cup S_a$; S: Set of selected instances from X; S_n: Set of selected normal instances from X_n; S_a: Set of selected attack instances from X_a

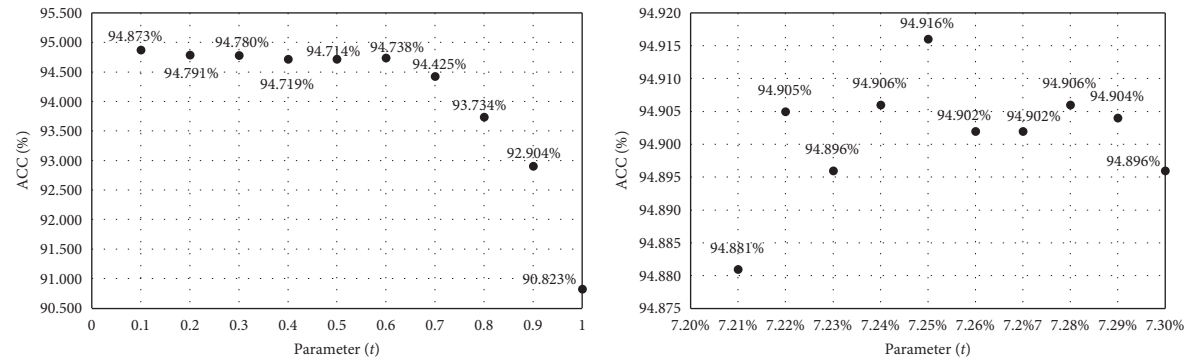
(1) Normalize X
(2) Initialize S,S_n, and S_a, according to X, X_n, and X_a
(3) For each x_i in X
(4) calculate $R(x_i, c)$ by formula (2)
(5) End for
(6) $[R(x_i, c), I] \longleftarrow \text{sortdesc}\{R(x_i, c)\}$
(7) $X \longleftarrow \text{sortIdx}(X, I)$
(8) Obtain S_n and S_a; in other words, according to $R(x_i, c)$, S_n and S_a are sorted in descending order
(9) Select the best t that reaches the best accuracy using 1-NN classifier through cross validation or validation set
(10) Obtain $S_n \longleftarrow S_n * t$ and $S_a \longleftarrow S_a * t$, which select the first $|S_n| * t$ instances in S_n and the first $|S_a| * t$ instances in S_a
(11) Obtain $S \longleftarrow S_n \cup S_a$

Algorithm 1: RBIS.



(a) (b)

Figure 1: The relation of ACC and parameter t on the DoS data set. (a) $t = [0.1, 0.2, \ldots, 1]$; (b) $t = [0.001, 0.002, \ldots, 0.01]$.



(a) (b)

Figure 2: The relation of ACC and parameter t on the DDoS 2016 data set. (a) $t = [0.1, 0.2, \ldots, 1]$; (b) $t = [0.0721, 0.0722, \ldots, 0.0730]$.

representativeness $R(x_i, c)$ and training set X are sorted in descending order (lines 6 and 7). Meanwhile, S_n and S_a are also sorted in descending order. Thirdly, from line 8 to line 11, according to the cross-validation or validation data, 1-NN is used as the classifier; the parameter t with the best balanced accuracy (BA) is selected and the range of parameter t is [0, 1]. In Section 4.3, the selection process of parameter t is shown by Figures 4–6. According to the selected parameter t, select the first $|S_a| * t$ instances and the first $|S_a| * t$ instances from in S_n and S_a, respectively. Finally, according to S_n and S_a, S is determined.

Figure 7 with two dimensions is used to explain the instance selection process of the RBIS-IM algorithm. Figure 7(a) shows two types of original data where the circle

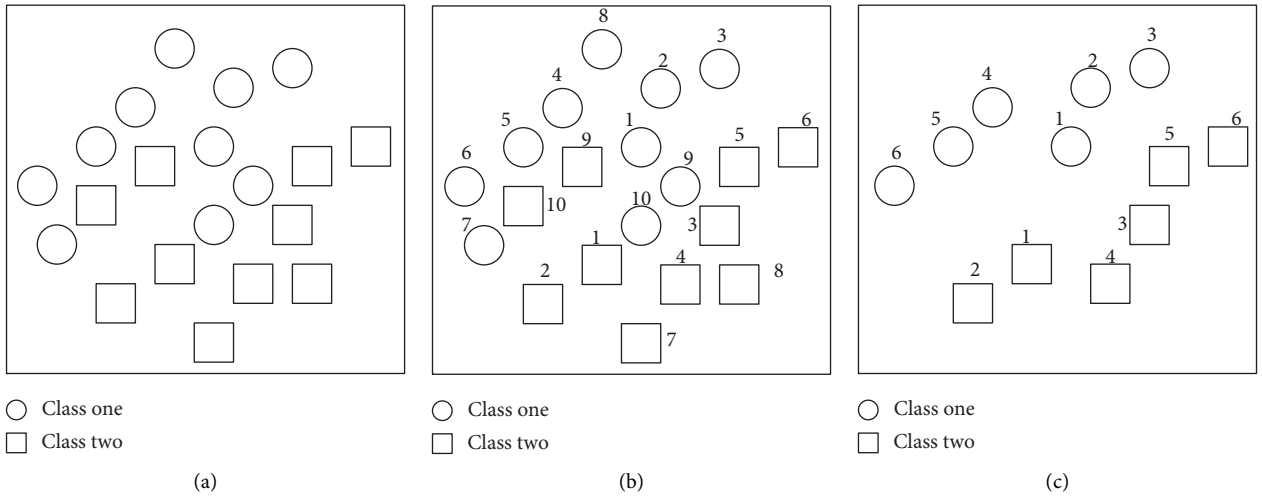


Figure 3: RBIS selects the critical instances of all classes in the balanced data. (a) Original data. (b) Sort the instances according to their representativeness. (c) Select the same proportion of instances according to the parameter (t).

Input: X: Training data set; t: the Ratio of selected instance by cross-validation or validation set; X_n: the Set of normal instances called the majority class; X_a: the Set of attack instances called the minority class.
Output: $S = S_n \cup S_a$; S: Set of selected instances from X; S_n: Set of selected normal instances from X_n; S_a: Set of selected attack instances from X_a

(1) Normalize X
(2) Initialize S,S_n, and S_a, according to X, X_n, and X_a
(3) For each x_i in X
(4) Calculate $R(x_i, c)$ by formula (2)
(5) End for
(6) $[R(x_i, c), I] \longleftarrow \text{sortdesc}\{R(x_i, c)\}$
(7) $X \longleftarrow \text{sortIdx}(X, I)$
(8) Obtain S_n and S_a; In other words, according to $R(x_i, c)$, S_n and S_a are sorted in descending order.
(9) Select the best t that reaches the best balanced accuracy using 1-NN classifier through cross-validation or validation set
(10) Obtain $S_a \longleftarrow S_a * t$ and $S_n \longleftarrow S_a * t$, which select the first $|S_a| * t$ instances in S_n and the first $|S_a| * t$ instances in S_a
(11) Obtain $S \longleftarrow S_n \cup S_a$

Algorithm 2: RBIS-IM.

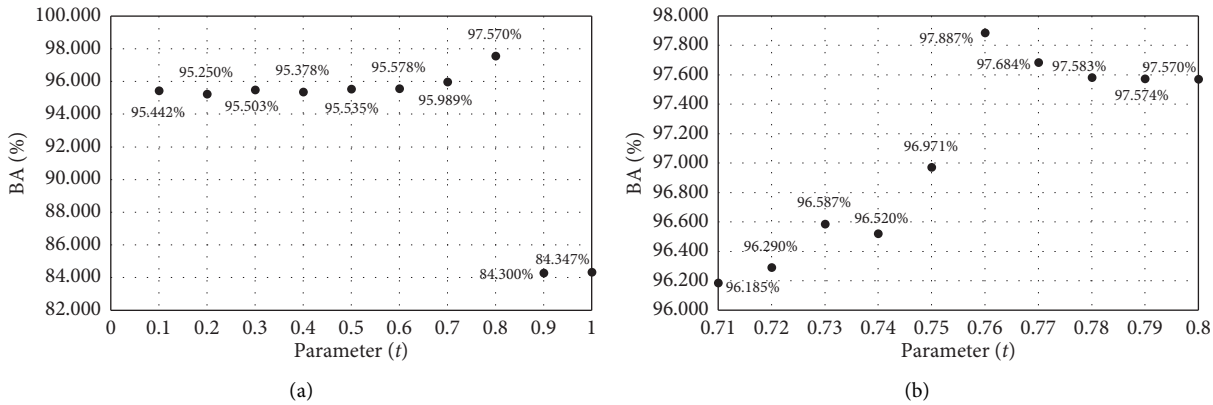


Figure 4: The relation of BA and parameter t on the Probe data set. (a) $t = [0.1, 0.2, \ldots, 1]$; (b) $t = [0.71, 0.72, \ldots, 0.80]$.

expresses majority class and the square expresses minority class. There are 8 instances in majority class, and there are 4 instances in minority class. According to their representativeness, the instances of each class are ranked in Figure 7(b). Similarly, the numbers around the graph indicate the degree of representation of the instance. The

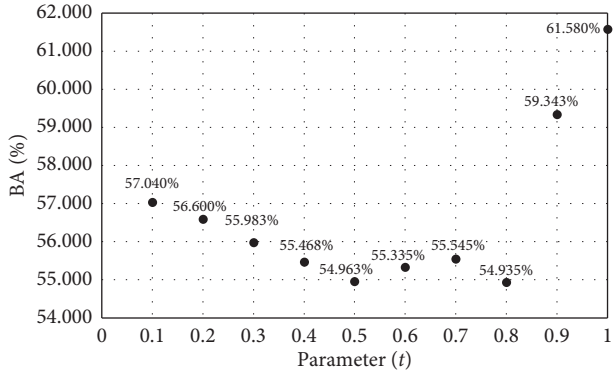


Figure 5: The relation of BA and parameter t on the U2R data set.

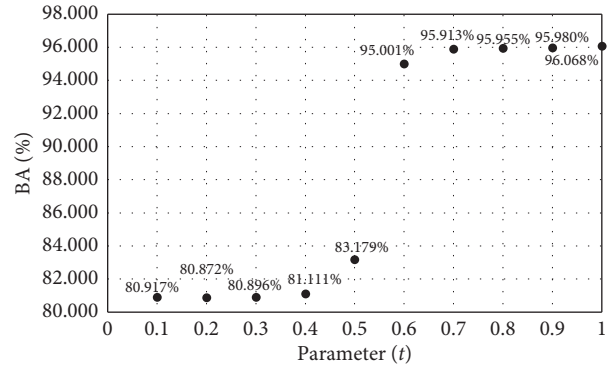


Figure 6: The relation of BA and parameter t on the R2L data set.

smaller the number, the more representative the instance is. In Figure 7(c), when the parameter t is 1, the first four instances of the minority class are selected. Since the number of selected instances of the majority class depends on the number of instances of the minority class and is the same as that selected of the minority class, the first four instances of the majority class are also selected.

Similarly, since the RBIS-IM algorithm is based on the representativeness of instance $R(x_i, c)$, the selected instances can contain all information of original data. And the effectiveness of RBIS-IM algorithm is evaluated by balanced accuracy (BA) and reduction rate (RR). In Section 4.3, compared with other algorithms on the benchmark data sets of intrusion detection, experimental results show that the RBIS-IM algorithm is effective and can achieve a better balance between BA and RR. Since the same number of instances for each class is selected to improve intrusion detection efficiency, RBIS-IM algorithm can deal with the imbalanced data problem. As the time complexity of the RBIS-IM algorithm is the same as the RBIS algorithm, the time complexity of this algorithm is $O(N^2)$.

The difference between RBIS-IM and RBIS algorithms is mainly embodied in three aspects. Firstly, the problems solved by the two algorithms are different. The RBIS-IM algorithm is to solve imbalanced data problem, which refers to the huge difference in the number of normal instances and attack instances; the RBIS algorithm is to deal with balanced data problem, which means that the number of normal instances and attack instances is very close or equal. Secondly, the methods of selected instances of two algorithms are different. In the RBIS-IM algorithm, the selection of instances of majority class is determined by selected instances of minority class. The number of selected instances of two classes is the same. In the RBIS algorithm, the number of instances of each class is close. In the RBIS algorithm, the same proportion of instances are selected for each class. Therefore, the number of selected normal and attack instances is very close. Thirdly, the evaluation criteria of the two algorithms are different, which are shown in Section 4.2. RBIS is evaluated by ACC and RR while RBIS-IM is related to BA and RR.

4. Experiments

In this section, experiments are designed to prove the effectiveness of the proposed algorithms. The section is divided into three subsections. In the first subsection, two experimental data sets are shown. In the second subsection, the evaluation criteria are introduced. In the last subsection, the RBIS and RBIS-IM algorithms are validated on balanced and imbalanced data sets.

4.1. Experimental Data Set. In this article, we use two data sets, which are the Knowledge Discovery and Data Mining

(KDD) Cup 1999 data set and DDoS 2016 data set. Although the KDD 99 data set has some disadvantages, it is still widely used as a benchmark for IDS evaluation [29–31]. In the KDD 99 data set, the 10% KDD training data and the KDD correct data are used as training data and testing data, respectively. The distribution of these data is shown in Table 1. In the KDD Cup 99 data set, the label of data includes the normal class and attack classes, which are divided into four groups: the remote-to-login (R2L), the denial-of-service (DoS), the user-to-root (U2R), and the Probe.

In the KDD Cup 99 data set, every network connection represents a data record that consists of 41 features and a label specifying the status of this record. Each record contains 41 features: 3 nonnumeric features, and 38 numeric features. During data preprocessing, these nonnumeric features, which are the protocol type, service, and flag, must be transformed into numeric data. The protocol type has three kinds of types: tcp, udp, and icmp. According to the different types, the "protocol type" feature is transformed into three features. As the "service" feature has 70 different types and would heavily increase the dimensionality, this single feature is not used in our experiments. The nonnumeric feature conversion is shown in Table 2.

The DDoS 2016 data set was published in 2016, which was created using the network simulator NS2 [32, 33]. There are 2.1 million data records in the data set. Each record contains 28 features: 5 nonnumeric features, and 23 numeric features. These nonnumeric features need to be converted to numerical ones. The data set contains normal data and four types of DDoS attacks, which are UDP flood, smurf, HTTP flood, and SIDDOS. In this section, the data set, which uses normal data and UDP flood, is used to evaluate the performance of the proposed algorithms.

According to balanced and imbalanced domains, the Knowledge Discovery and Data Mining (KDD) Cup 1999 and DDoS 2016 are divided into the balanced data set and the imbalanced data set. The description of data sets is shown in Tables 3 and 4.

4.2. Evaluation Criteria. To evaluate the effectiveness and performance of the proposed algorithms, the confusion matrix is used. The confusion matrix is shown in Table 5. According to the confusion matrix, four performance metrics are applied: the detection rate (DR, also known as the true positive rate), true negative rate (TNR, also known as specificity or selectivity), balanced accuracy (BA), and accuracy (ACC). Meanwhile, the reduction rate (RR) is also applied.

In balanced data, ACC and RR are used to evaluate the performance of the proposed RBIS algorithm. To treat the minority and majority instances equally, BA is selected as the evaluation criterion of the RBIS-IM algorithm in the imbalanced problem.

The DR is the proportion of attack instances that are correctly predicted as attacks in the test data set; it is an important metric reflecting the attack detection model's ability to identify attack instances and is described as

Table 1: The distribution of the KDD 99 data.

Class	The 10% KDD training data	The KDD correct data
Normal	97278	60593
DoS	391458	229853
U2R	52	228
Probe	4107	4166
R2L	1126	16189
Total	494021	311029

Table 2: The nonnumeric feature conversion in the KDD 99 data.

Feature name	Type setting 1	Type setting 2
Protocol type = tcp	tcp = 1	others = 0
Protocol type = udp	udp = 1	others = 0
Protocol type = icmp	icmp = 1	others = 0
Flag	SF = 1	others = 0

Table 3: The balanced data set.

Type	Attribute	Class	Normal/attack in training data	Normal/attack in testing data
Normal and DoS data in the KDD 99 data set	42	2	10000/10000	10000/10000
The DDoS 2016 data set	28	2	10000/10000	10000/10000

Table 4: The imbalanced data set.

In the KDD 99 data set	Attribute	Class	Normal/attack in training data	Normal/attack in testing data
Normal and U2R data	42	2	10000/30	200/20
Normal and probe data	42	2	10000/1550	10000/1000
Normal and R2L data	42	2	10000/1000	10000/1000

Table 5: Confusion matrix.

Class	Predicted negative class	Predicted positive class
Actual negative class	True negative (TN)	False positive (FP)
Actual positive class	False negative (FN)	True positive (TP)

$$\mathrm{DR} = \frac{\mathrm{TP}}{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}. \tag{3}$$

The TNR is the proportion of normal instances that are correctly predicted as normal in the test data set. And, it is an important metric reflecting the detection model's ability to identify normal instances and can be written as

$$\mathrm{TNR} = \frac{\mathrm{TN}}{N} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}. \tag{4}$$

The BA is the average of DR and TNR; it can be a leading metric for imbalanced data sets; it can serve as an overall performance metric for a model.

$$\mathrm{BA} = \frac{\mathrm{DR} + \mathrm{TNR}}{2}. \tag{5}$$

The ACC is the ratio of the number of instances correctly predicted in the test data set to the total number of instances. And, it can reflect the ability of the detection model to distinguish between normal and attack instances and is defined as

$$\mathrm{ACC} = \frac{\mathrm{TN} + \mathrm{TP}}{P + N} = \frac{\mathrm{TN} + \mathrm{TP}}{\mathrm{TN} + \mathrm{TP} + \mathrm{FN} + \mathrm{FP}}. \tag{6}$$

The RR is the ratio of the number of selected instances in the training data set to the total number of instances; it can show the ability of the instance selection model to select optimal instances and can be written as

$$\mathrm{RR} = \frac{|S|}{\|X\|} * 100\%. \tag{7}$$

4.3. Experimental Results and Analysis. In this section, we use the instance subset selected by the proposed instance selection algorithms to verify the effectiveness of instance representation and the algorithms. The experiment is conducted in balanced and imbalanced data sets. All the experimental results are obtained by calculating the average value of 100 experiments.

The RBIS and RBIS-IM algorithms have a parameter t that is used to determine the number of selected instance subsets. In the training phase, the parameter t is determined by grid search on cross validation or verification set. In the RBIS algorithm, the parameter is selected by the best ACC. In the RBIS-IM algorithm, the selected parameter is related to the best BA.

Figures 1 and 2 show the relation of ACC and parameter t on the balanced data sets. Moreover, Figures 1 and 2 reflect the selection process of parameter t in RBIS algorithm on DOS and DDOS 2016 data sets. Figures 1(a) and 2(a) display the change of ACC when the parameter t is in a large interval [0.1, 1]. Figures 1(b) and 2(b) show the change of ACC when the parameter t is in a small interval [0.001, 0.01] and [0.0721, 0.0730]. Figure 1(b) is based on Figure 1(a). Similarly, Figure 2(b) is based on Figure 2(a). From Figure 1(a), the best ACC is achieved when the parameter t takes 0.1 in the interval [0.1, 1]. Therefore, the range of parameter t in Figure 1(b) is in the interval [0, 0.1]. Through experiments, the range of parameter t in Figure 1(b) is in the interval [0.001, 0.01]. In Figure 1(b), according to the best ACC, the parameter t is 0.3%.

Like Figure 1, Figure 2(a) illustrates that the best ACC is obtained when the parameter t takes 0.1 in the interval [0.1, 1].Therefore, the range of parameter t in Figure 2(b) is in the interval [0, 0.1]. Through experiments, the range of parameter t in Figure 2(b) is in the interval [0.0721, 0.0730]. In Figure 2(b), according to the best ACC, the parameter t is 7.25%.

Figures 4-6 display the relation of BA and parameter t on the imbalanced data sets. When the parameter t is in the interval [0.1, 1] and [0.71, 0.80], the change of BA on the Probe data set is shown in Figures 4(a) and 4(b). Figures 5 and 6 show BA changes on the U2R and R2L data sets when the parameter t is in the interval [0.1, 1]. Meanwhile, Figures 4–6 reflect the selection process of parameter t in the RBIS-IM algorithm. Figures 4(a) show the change of BA when the parameter t is in a large interval [0.1, 1]. Figures 4(b) indicate the change of BA when the parameter t is between in a small interval [0.71, 0.80]. Figure 4(b) is based on Figure 4(a). From Figure 4(a), the best BA is obtained when the parameter t takes 0.8 in the interval [0.1, 1].Therefore, the range of parameter t in Figure 4(b) is in the interval [0, 0.8]. Through experiments, the range of parameter t in Figure 4(b) is in the interval [0.71, 0.80]. In Figure 4(b), according to the best BA, the parameter t is 0.76. From Figures 5 and 6, it is obvious that the parameter t is set to 1 under the condition that BA obtains the best on two data sets. Moreover, relevant experiments are conducted in the interval [0.9, 1]. The experimental results show that BA obtains the best when parameter t is 1.

Table 6 shows that on the balanced data set, the three common classifiers, which are 1-NN, SVM, and Adaboost, use the entire training set and instance subset selected to obtain ACC, RR, and average accuracy, respectively. On the DoS data set of KDD cup 99, the accuracy of the three classifiers is greatly improved by using the instance subset selected by the RBIS algorithm. On the DDoS 2016 data set, the three classifiers also achieve good accuracy by using the instance subset. The accuracy of SVM and Adaboost using the instance subset are slightly lower than those of the whole training set, but the RBIS algorithm only uses 7.25% of instances to get good accuracy (i.e. 94.682% or 94.668%). This shows that the RBIS algorithm can reduce RR while maintaining accuracy. On the two balanced data sets, the accuracy of 1-NN using the instance subset is higher than that by the whole training set. This is because the instance subset is selected by the proposed instance selection algorithm and 1-NN. In addition to good ACC, the RR by the three classifiers and instance subsets are very small, which are 0.3% and 7.25%, respectively. This can prove that the RBIS algorithm can achieve a better balance between ACC and RR. On the other hand, from the perspective of average ACC, it is obvious that the average ACC by the instance subset is much higher than that by the whole training set on the DoS data set. Meanwhile, on the DDoS 2016 data set, the average ACC by the instance subset is only slightly higher than that obtained by the whole training set. This indicates that the RBIS algorithm can select optimal instances to improve ACC and reduce RR for IDS.

In Table 6, the experimental results demonstrate that the proposed RBIS algorithm is effective and can deal with balanced data problem. The RBIS algorithm is effective because it is based on the new instance representativeness, which is shown in Section 3.1. Through instance representativeness, the selected instances possess the information of the entire instances and are useful to improve ACC and reduce RR for IDS.

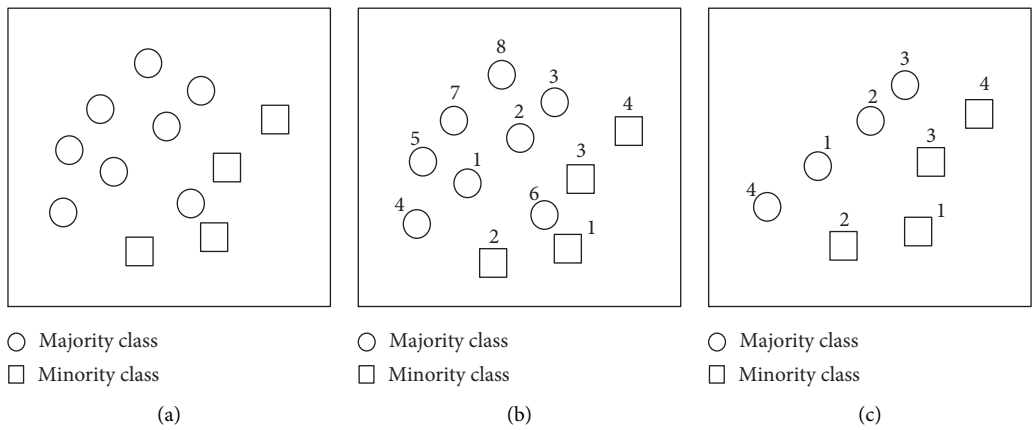


Figure 7: RBIS-IM selects the critical instances of all classes in the imbalanced data. (a) Original data. (b) Sort the instances according to their representativeness. (c) Select the same number of instances according to the parameter t.

Table 6: The efficiency of RBIS algorithm with 1-NN, SVM, and Adaboost is verified on the balanced data set.

Data set	The size of instances	Classifier	ACC (%)	RR	Average ACC (%)
DoS	20000	1-NN	61.000	100	63.692
		SVM	65.044		
		Adaboost	65.033		
	60	1-NN	99.913	0.3	93.362
		SVM	99.910		
		Adaboost	80.263		
DDoS 2016	20000	1-NN	90.823	100	93.653
		SVM	95.059		
		Adaboost	95.077		
	1450	1-NN	94.916	7.25	94.755
		SVM	94.682		
		Adaboost	94.668		

As shown in Table 7, on the imbalanced data sets, three common classifiers, which are 1-NN, SVM, and Adaboost, can obtain BA, RR, and average BA using the whole training set and the instance subset. Three imbalanced data sets are from the KDD Cup 99. On the Probe data set, using the instance subset, the three classifiers get good accuracy. Compared with the whole training set, the BA by the 1-NN classifier using instance subset is slightly lower, while BAby SVM and Adaboost are better. On U2R and R2L data sets, compared to using the whole training set BA of three common classifiers using instance subset is better. The experimental results prove that the RBIS-IM algorithm can achieve a better balance between BA and RR.

Besides, from the perspective of average BA, on the Probe data set, the average BA using the instance subset is slightly higher than that using the whole training set. On the U2R and R2L data sets, compared with the average BA using the whole training set, the average BA using the instance subset is greatly improved. Therefore, the experimental results on imbalanced data sets indicate that the RBIS-IM algorithm is effective and can obtain good RR while improving BA. This is because the RBIS-IM algorithm is also based on the new instance representativeness, which is shown in Section 3.1. Through instance representativeness, the optimal instances are selected to improve BA and reduce RR for IDS. And the experimental results display that the RBIS-IM algorithm can handle imbalanced data problem.

Table 7: The efficiency of RBIS-IM algorithm with 1-NN, SVM, and Adaboost is verified on the imbalanced data set.

Data set	The size of instances	Classifier	BA (%)	RR (%)	Average BA (%)
Probe	11550	1-NN	98.825	100	98.096
		SVM	99.104		
		Adaboost	96.359		
	2356	1-NN	97.887	20.398	98.148
		SVM	99.544		
		Adaboost	97.013		
U2R	10030	1-NN	49.970	100	50.079
		SVM	49.998		
		Adaboost	50.270		
	60	1-NN	61.580	0.598	61.632
		SVM	61.565		
		Adaboost	61.750		
R2L	11000	1-NN	80.465	100	74.860
		SVM	67.665		
		Adaboost	76.449		
	2000	1-NN	96.068	18.182	91.445
		SVM	87.859		
		Adaboost	90.407		

Table 8: Accuracy of ENN, ISAR, BNNT, CNNIR, RIS 1, and RBIS on the balanced data set.

Data set	ENN	ISAR	BNNT	CNNIR	RIS 1	RBIS
DoS	65.173	99.904	65.070	65.142	99.906	99.913
DDoS 2016	84.589	70.533	72.089	73.584	70.520	94.916
Mean	74.881	85.219	68.580	69.633	85.213	97.415

Table 9: Reduction rate of ENN, ISAR, BNNT, CNNIR, RIS 1, and RBIS on the balanced data set.

Data set	ENN	ISAR	BNNT	CNNIR	RIS 1	RBIS (%)
DoS	99.995	50.005	0.065	9.780	49.785	0.300
DDoS 2016	87.255	53.435	9.135	4.820	13.335	7.250
Mean	93.625	51.720	4.600	7.300	31.560	3.775

Tables 8 and 9 display the ACC and RR with the 6 instance selection algorithms on the balanced data sets. The proposed RBIS algorithm is compared with 5 algorithms: edited nearest neighbor (ENN) [22], ISAR [10], BNNT [8], CNNIR [9], and RIS 1 [11]. For ISAR and RIS 1, their instance selection algorithms are only used. On two balanced data sets, compared with the other 5 algorithms, the proposed RBIS algorithm achieves the best experimental results on ACC in Table 8. And the RBIS algorithm achieves the second RR on two balanced data sets in Table 9. In terms of average performance, it is obvious the RBIS algorithm achieves the best experimental results on ACC and RR. This indicates that the RBIS algorithm can achieve a better balance between ACC and RR. And, it can solve balanced data problem. Similarly, it proves that the RBIS algorithm is effective. In other words, the selected instances are optimal and contain the information of the whole instances. This is because four factors in the instance selection process are considered, which are shown in Section 3.1.

Table 10 shows the BA of 6 instance selection algorithms on the imbalanced data set. On the Probe data set, the BA of ENN, ISAR, RIS 1, and RBIS-IM algorithms are very close, and the biggest gap between them is less than 1%. This displays the RBIS-IM algorithm has the ability to distinguish between normal and attack instances. On the U2R and R2L data sets, the BA of the RBIS-IM algorithm is the best. Compared with other algorithms, the minimum gap is at least 10%. From the average BA, the average BA of the ENN, ISAR, and RIS 1 algorithms are very close, while the BA of the RBIS-IM algorithm is the best in Table 10. The experimental results prove that representative instances selected by RBIS-IM algorithm contain the information of the whole instances and the RBIS-IM algorithm can select representative instances to increase the BA for IDS. Moreover, the experimental results demonstrate that RBIS-IM algorithm can deal with imbalanced data problem.

Table 11 presents the RR of 6 instance selection algorithms on the imbalanced data set. On the Probe data set, the RR obtained by ISAR, CNNIR, and RIS 1 algorithms are very close. But, compared with ENN, other algorithms have a big gap with it. On the U2R data set, except for the ENN algorithm, the RR of other algorithms are very close and less than 1%. On R2L data, there is a small difference between the RR of the three algorithms, which are ISAR, CNNIR, and RIS 1 algorithms. From the average RR, the RR of the BNNT algorithm is the best. But, it is obvious that ENN gets poor RR (i.e. 99.879%). Since ENN is based on the nearest neighbor, ENN only removes instances near to the boundary and deletes limited instances of majority class. Moreover, ENN cannot deal with imbalanced data problem. The proposed RBIS-IM algorithm has good RR (i.e. 13.059%). This displays that the RBIS-IM algorithm can select small and representative instances to reduce RR. And the experimental results show that the RBIS-IM algorithm can deal with imbalanced data problem.

Table 10: BA of ENN, ISAR, BNNT, CNNIR, RIS 1, and RBIS-IM on the imbalanced data set.

Data set	ENN	ISAR	BNNT	CNNIR	RIS 1	RBIS-IM
Probe	98.789	98.059	70.510	87.175	98.059	97.887
U2R	49.980	49.592	50.755	50.000	49.642	61.580
R2L	80.434	79.956	53.797	63.592	85.238	96.068
Mean	76.401	75.869	58.354	66.922	77.646	85.178

Table 11: Reduction rate of ENN, ISAR, BNNT, CNNIR, RIS 1, and RBIS-IM on the imbalanced data set.

Data set	ENN (%)	ISAR (%)	BNNT (%)	CNNIR (%)	RIS 1 (%)	RBIS-IM (%)
Probe	99.896	13.680	0.537	10.312	13.680	20.398
U2R	99.950	0.489	0.578	0.680	0.160	0.598
R2L	99.791	9.455	0.945	6.327	9.000	18.182
Mean	99.879	7.875	0.687	5.773	7.613	13.059

The time complexity of 6 instance selection algorithms is present in Table 12. N represents the number of original instances. According to Table 12, the time complexity of the 6 algorithms is divided into two types. One is $O(N\log N)$, which are ENN, BNNT, and CNNIR algorithms. The other is $O(N^2)$, which are ISAR, RIS 1, RBIS, and RBIS-IM algorithms.

Figure 8 shows the relation of average ACC and average RR of 7 algorithms on the balanced data set and is based on Tables 6, 8, and 9. The 1-NN algorithm uses the whole training instances and the other 6 algorithms use the instance subset through their instance selection algorithms. On the balanced data set, the RBIS algorithm achieves the best in ACC and RR. Figure 8 suggests that the RBIS algorithm can select optimal instances to improve ACC and reduce RR for IDS. These optimal instances have the information for the entire instances.

Figure 9, which is based on Tables 7, 10, and 11, shows the relation of average BA and average RR of 7 algorithms on the imbalanced data set. It is obvious that RBIS-IM is the best on average BA. And Figure 9 suggests that the RBIS-IM algorithm can select optimal instances to increase BA and reduce RR for IDS. Although the average RR of the RBIS-IM algorithm is not the minimum, RBIS-IM algorithm can

Table 12: Time complexity of six algorithms.

ID	Algorithms	Time complexity
1	ENN	$O(N\log N)$
2	ISAR	$O(N^2)$
3	BNNT	$O(N\log N)$
4	CNNIR	$O(N\log N)$
5	RIS 1	$O(N^2)$
6	RBIS/RBIS-IM	$O(N^2)$

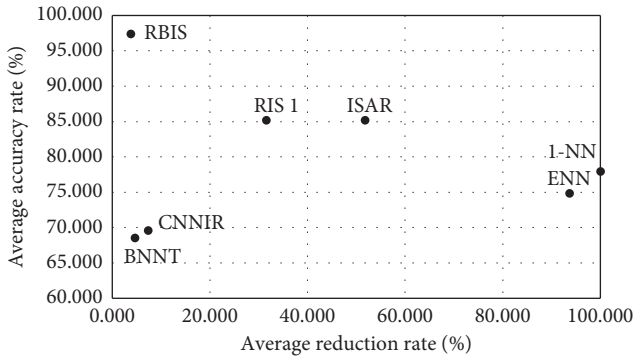


Figure 8: The relation of average ACC and average RR.

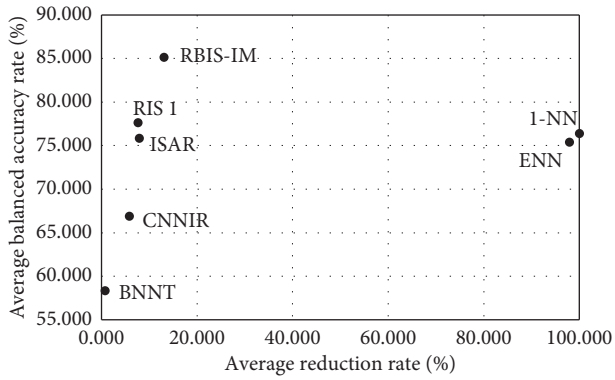


Figure 9: The relation of average BA and average RR.

achieve a good balance between average BA and average RR. Moreover, it is found that the RBIS-IM algorithm can handle imbalanced data problem.

5. Conclusions

In this paper, after analyzing the instance selection algorithm and its defects in intrusion detection, we propose a new representativeness of instance to determine the importance of an instance. Calculating the representativeness of instance, we consider not only the representativeness of instance in its category but also the representativeness of instances in different categories. These two representativenesses are equally important. Moreover, the influence of instances of different classes on selected instance is regarded as an advantage factor. To deal with balanced and imbalanced data problems, we propose the RBIS and RBIS-IM algorithms, respectively. In the process of instance selection, the proposed algorithms need not delete internal instances and noise instances. Compared with other algorithms on the benchmark data sets of intrusion detection, experimental results show that the two algorithms are effective. RBIS algorithm can achieve a better balance between accuracy (ACC) and reduction rate (RR). Similarly, the RBIS-IM algorithm can achieve a better balance between balanced accuracy (BA) and reduction rate (RR). Furthermore, it is also verified that the proposed representativeness of instance is correct and effective.

In future work, we intend to study how to automatically obtain the appropriate parameter t of the proposed approaches, which will reduce the training time of the algorithms. Moreover, obtaining the parameter t automatically can improve and enhance the effectiveness and applicability of the algorithms.

Data Availability

In this paper, two data sets are used for intrusion detection. They are public, which are the Knowledge Discovery and Data Mining (KDD) Cup 1999 data set and DDoS 2016 data set. The corresponding URLs are, respectively, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html and https://www.researchgate.net/publication/292967044_Dataset_Detecting_Distributed_Denial_of_Service_Attacks_Using_Data_Mining_Techniques.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Key R&D Program of China under Grant No. 2017YFB0802300, in part by the Major Scientific and Technological Special Project of Guizhou Province under Grant No. 20183001, in part by the Foundation of Guizhou Provincial Key Laboratory of Public Big Data under Grant Nos. 2018BDKFJJ008 and 2018BDKFJJ020, and in part by the National Statistical Scientific Research Project of China under Grant Nos. 2018LY61 and 2019LY82.

References

[1] T. Phuoc, P. Tsai, T. Jan, and X. Kong, "Network intrusion detection using machine learning techniques," in *Proceedings of the 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, pp. 1–7, Vellore, India, February 2020.

[2] H. Hindy, D. Brosset, E. Bayne et al., "A taxonomy of network threats and the effect of current datasets on intrusion detection systems," *IEEE Access*, vol. 8, pp. 104650–104675, 2020.

[3] O. Adeleke, "Intrusion detection: issues, problems and solutions," in *Proceedings of the 3rd International Conference on Information and Computer Technologies (ICICT)*, pp. 397–402, San Jose, CA, USA, March 2020.

[4] J. Nalepa and M. Kawulok, "Selecting training sets for support vector machines: a review," *Artificial Intelligence Review*, vol. 52, pp. 857–900, 2019.

[5] Z. H. Zhu, Z. Wang, D. D. Li, and W. L. Du, "NearCount: selecting critical instances based on the cited counts of nearest neighbors," *Knowledge-Based Systems*, vol. 190, 2020.

[6] A. D. Haro-Garcia, G. Cerruela-Garcia, and N. Garcia-Pedrajas, "Instance selection based on boosting for instance-based learners," *Pattern Recognition*, vol. 96, 2019.

[7] C. Guo, Y.-J. Zhou, Y. Ping, S.-S. Luo, Y.-P. Lai, and Z.-K. Zhang, "Efficient intrusion detection using representative instances," *Computers & Security*, vol. 39, pp. 255–267, 2013.

[8] J. Li and Y. Wang, "A new fast reduction technique based on binary nearest neighbor tree," *Neurocomputing*, vol. 149, pp. 1647–1657, 2015.

[9] L. Yang, Q. Zhu, J. Huang, Q. Wu, D. Cheng, and X. Hong, "Constraint nearest neighbor for instance reduction," *Soft Computing*, vol. 23, no. 24, pp. 13235–13245, 2019.

[10] C. D. S. Pereira and G. D. C. Cavalcanti, "Instance selection algorithm based on a ranking procedure," in *Proceedings of the 2011 International Joint Conference on Neural Networks*, pp. 2409–2416, San Jose, CA, USA, July 2011.

[11] G. D. C. Cavalcanti and R. J. O. Soares, "Ranking-based instance selection for pattern classification," *Expert Systems with Applications*, vol. 150, 2020.

[12] H. Hmida, S. B. Hamida, A. Borgi, and M. Rukoz, "Hierarchical data topology based selection for large scale learning," in *Proceedings of the 2016 International IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*, pp. 1221–1226, Toulouse, France, July 2016.

[13] J. Hamidzadeh, N. Kashefi, and M. Moradi, "Combined weighted multi-objective optimizer for instance reduction in two-class imbalanced data problem," *Engineering Applications of Artificial Intelligence*, vol. 90, 2020.

[14] L. Li, K. Y. Zhao, R. Z. Sun et al., "Parameter-free extreme learning machine for imbalanced classification," *Neural Processing Letters*, vol. 52, no. 3, pp. 1927–1944, 2020.

[15] H. X. Guo, Y. J. Li, J. Shang et al., "Learning from class-imbalanced data: review of methods and applications," *Expert Systems with Applications*, vol. 73, pp. 220–239, 2017.

[16] C.-H. Chou, B.-H. Kuo, and F. Chang, "The generalized condensed nearest neighbor rule as A data reduction method," in *Proceedings of the 18th International Conference on Pattern Recognition*, vol. 2, pp. 556–559, Hong Kong, China, August 2006.

[17] H. A. Fayed and A. F. Atiya, "A novel template reduction approach for the k-nearest neighbor method," *IEEE Transactions on Neural Networks*, vol. 20, no. 5, pp. 890–896, 2009.

[18] J. Arturo Olvera-López, J. Ariel Carrasco-Ochoa, and J. Francisco Martínez-Trinidad, "A new fast prototype selection method based on clustering," *Pattern Analysis & Applications*, vol. 13, no. 2, pp. 131–141, 2010.

[19] A. A. Akinyelu and A. E. Ezugwu, "Nature inspired instance selection techniques for support vector machine speed optimization," *IEEE Access*, vol. 7, pp. 154581–154599, 2019.

[20] A. Akinyelu and A. O. Adewumi, "On the performance of cuckoo search and bat algorithms based instance selection techniques for SVM speed optimization with application to E-fraud detection," *KSII Transactions on Internet and Information Systems*, vol. 12, no. 3, pp. 1348–1375, 2018.

[21] C. E. Brodley, "Recursive automatic bias selection for classifier construction," *Machine Learning*, vol. 20, pp. 63–94, 1995.

[22] I. Tomek, "An experiment with the edited nearest-neighbor rule," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 6, pp. 448–452, 1976.

[23] N. Jankowski and M. Grochowski, "Comparison of instances selection algorithms I. Algorithms survey," *International Conference on Artificial Intelligence and Soft Computing*, vol. 10, pp. 937–942, 2004.

[24] Q. Y. Wang, X. Q. Ouyang, and J. C. Zhan, "A classification algorithm based on data clustering and data reduction for intrusion detection system over big data," *KSII Transactions on Internet and Information Systems*, vol. 13, pp. 3714–3732, 2019.

[25] P. Ghosh, A. Saha, and S. Phadikar, "Penalty-reward based instance selection method in cloud environment using the concept of nearest neighbor," *Procedia Computer Science*, vol. 89, pp. 82–89, 2016.

[26] L. Yang, Q. Zhu, J. Huang, and D. Cheng, "Adaptive edited natural neighbor algorithm," *Neurocomputing*, vol. 230, pp. 427–433, 2017.

[27] N. García-Pedrajas, J. A. Romero del Castillo, and D. Ortiz-Boyer, "A cooperative coevolutionary algorithm for instance selection for instance-based learning," *Machine Learning*, vol. 78, no. 3, pp. 381–420, 2010.

[28] J. Li, Q. Zhu, and Q. Wu, "A parameter-free hybrid instance selection algorithm based on local Sets with natural neighbors," *Applied Intelligence*, vol. 50, no. 5, pp. 1527–1541, 2020.

[29] B. Jia and Y. Liang, "Anti-D chain: a lightweight DDoS attack detection scheme based on heterogeneous ensemble learning in blockchain," *China Communications*, vol. 17, no. 9, pp. 11–24, 2020.

[30] C. Guo, Y. Ping, N. Liu, and S. S. Luo, "A two-level hybrid approach for intrusion detection," *Neurocomputing*, vol. 214, 2016.

[31] University of California Department of Information and Computer Science, *KDD Cup 99 Intrusion Detection Dataset Task Description*, University of California Department of Information and Computer Science, Berkeley, CA, USA, 1999, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[32] M. Alkasassbeh, G. Al-Naymat, B. A. Ahmad, and M. Almseidin, "Detecting distributed denial of service attacks using data mining techniques," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 1, 2016.

[33] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," *Computers & Security*, vol. 86, pp. 147–167, 2019.