sensors

MDPI

Article

Wireless Sensor Networks Intrusion Detection Based on SMOTE and the Random Forest Algorithm

Xiaopeng Tan, Shaojing Su, Zhiping Huang, Xiaojun Guo *, Zhen Zuo, Xiaoyong Sun and Longqing Li

College of Artificial Intelligence, National University of Defense Technology, Changsha 410073, China; tanxiaopeng14@nudt.edu.cn (X.T.); susj-5@163.com (S.S.); kdhuangzp@163.com (Z.H.); z.zuo@nudt.edu.cn (Z.Z.); sxy15084973192@163.com (X.S.); 15575982374@163.com (L.L.)

* Correspondence: jeanakin@nudt.edu.cn; Tel.: +86-139-7317-7113

Received: 13 December 2018; Accepted: 4 January 2019; Published: 8 January 2019

Abstract: With the wide application of wireless sensor networks in military and environmental monitoring, security issues have become increasingly prominent. Data exchanged over wireless sensor networks is vulnerable to malicious attacks due to the lack of physical defense equipment. Therefore, corresponding schemes of intrusion detection are urgently needed to defend against such attacks. Considering the serious class imbalance of the intrusion dataset, this paper proposes a method of using the synthetic minority oversampling technique (SMOTE) to balance the dataset and then uses the random forest algorithm to train the classifier for intrusion detection. The simulations are conducted on a benchmark intrusion dataset, and the accuracy of the random forest algorithm has reached 92.39%, which is higher than other comparison algorithms. After oversampling the minority samples, the accuracy of the random forest combined with the SMOTE has increased to 92.57%. This shows that the proposed algorithm provides an effective solution to solve the problem of class imbalance and improves the performance of intrusion detection.

Keywords: wireless sensor networks; intrusion detection; class imbalance; SMOTE; random forest

1. Introduction

The wireless sensor network is a distributed intelligent network system. It is composed of a large number of micro sensor nodes deployed in the detection area, which have the ability of wireless communication and computing. It can accomplish the assigned tasks independently according to the changes of environment. With the rapid development of wireless sensor technology, embedded computing technology, wireless communication technology, and distributed information processing technology, wireless sensor networks have very broad application prospects, such as national defense, ecological observation, environmental monitoring, medical security, space exploration, volcano observation, architecture, and city management, etc. [1–4].

Wireless sensor networks can realize real-time monitoring, sensing and collecting information of various environments or monitoring objects through the cooperation of various integrated micro-sensors. Then the information is processed by embedded system. Finally, the perceived information is transmitted to the user terminal by multi-hop relay through random self-organizing wireless communication network. In this process, the sensor nodes are located in a large area without protection or in a harsh environment, which makes them easy to be captured and leak sensitive information. Excessive security mechanisms of wireless sensor networks is not suitable for resource-constrained sensor networks. The characteristics of wireless jump communication make it easier to be eavesdropped, jammed, and attacked. The low cost of sensor nodes also makes it easy

for the nodes to leak the key after being captured, which will lead to the insecurity of the whole network [5–8].

Currently, the security of wireless sensor networks has become a major concern. How to identify various network attacks is a key technology that needs to be solved [9,10]. The security studies of wireless sensor networks mainly focuses on the following aspects: (1) various network attack models and defense strategies; (2) encryption algorithms, key management, and authentication technology; (3) network routing and the security of data fusion; and (4) network intrusion detection systems and response models [11,12]. The security studies of wireless sensor networks can be divided into passive defense and active defense. Compared with the advancement of passive defense studies for wireless sensor networks, there are too few studies on active defense. However, passive defense is the measure taken in response to the characteristics of the attacks after the attacks have occurred, and this is not sufficient for the security of wireless sensor networks. Therefore, it is urgent to study active defense technologies to detect malicious intrusions before attacks occur. As an active defense technology, intrusion detection will play an important role in ensuring the security of wireless sensor networks [13–17]. This paper will focus on introducing intrusion detection technology in wireless sensor networks.

In the past few years, many methods have been proposed to design intrusion detection systems for wireless sensor networks. Lu et al. [18] introduced an evolutionary mechanism to extract intrusion detection rules. In order to extract diverse rules and control the number of rule sets, rules are checked and extracted according to the distance between rules in the same type of rule set and rules in different types of rule sets. Singh et al. [19] proposed an advanced hybrid intrusion detection system (AHIDS) that automatically detects wireless sensor networks attacks. AHIDS utilizes the cluster architecture and enhanced LEACH protocol to reduce the energy consumption of sensor nodes. AHIDS uses anomaly detection and misuse detection based on a fuzzy rule set and a multi-layer perceptron neural network. Due to the advantages of the negative selection algorithm (NSA) in the classification domain, Sun et al. [20] proposed a WSN-NSA intrusion detection model based on the improved V-detector algorithm for wireless sensor networks (WSN). The V-detector algorithm is modified by modifying detector generation rules and optimizing detectors, and principal component analysis is used to reduce detection features. Tajbakhsh et al. [21] proposed an intrusion detection model based on fuzzy association rules, which uses fuzzy association rules to construct classifiers, and uses some matching metrics to evaluate the compatibility of any new samples with different rule sets. And the class corresponding to the best matching rule set is declared as the label of the sample. Xie et al. [22] focus on detecting a special type of anomaly in wireless sensor network (WSN), which appears simultaneously in a collection of neighboring nodes and lasts for a significant period of time. With the proposed distributed segment-based recursive kernel density estimation, a global probability density function can be tracked and its difference between every two periods of time is continuously measured for decision making. Xie et al. [23] focus on a new technique for handling data in a segment-based manner. Considering a collection of neighboring data segments as random variables, they determine those behaving abnormally by exploiting their spatial predictabilities and, motivated by spatial analysis, specifically investigate how to implement a prediction variance detector in a WSN. Haider et al. [24] proposed a metric using a fuzzy logic system based on the Sugeno fuzzy inference model for evaluating the quality of the realism of existing intrusion detection system datasets. Based on the proposed metric results, they designed and generated a synthetically realistic next-generation intrusion detection system dataset, and a preliminary analysis was conducted to assist in the design of future intrusion detection systems.

At present, many intrusion detection systems for wireless sensor networks are rule-based systems whose performance is highly dependent on the rules determined by security experts. Due to the vast amount of network traffic, the process of encoding rules is both expensive and slow. In order to overcome the limitation of rule-based system, data mining technology is used in intrusion detection systems for wireless sensor networks. Data mining is a successful solution for active detection of

network attacks based on features hidden in the data of network behavior. Through the analysis of large datasets, understandable patterns or models are explored, the intrusion patterns for anomaly detection are effectively extracted, and the classifiers used to detect attacks are constructed [25,26]. The intrusion detection system based on data mining is more flexible and easier to deploy. In this paper, an intrusion detection model for wireless sensor networks is proposed. A data mining algorithm called random forest is applied in intrusion detection for wireless sensor networks. The random forest algorithm is an ensemble classification and regression method, and it is one of the most effective data mining technologies [27,28].

One of the challenges of intrusion detection for wireless sensor networks is the imbalance of intrusion, such as Denial of Service (DoS) attacks, which have more connections than probing attacks, user to root (U2R) attacks, and root to local (R2L) attacks. Most data mining algorithms attempt to minimize the overall error rate, but this will increase the error rate for identifying minority intrusions. In the actual wireless sensor network environment, minority attacks are more dangerous than the majority of attacks. Considering the serious class imbalance of the intrusion dataset, synthetic minority oversampling technology (SMOTE) is adopted to balance the dataset, which effectively improves the detection performance of minority intrusions.

Lee et al. [29] proposed a hybrid approach for real-time network intrusion detection systems (NIDS). They adopt the random forest (RF) for feature selection. RF provides the variable importance by numeric values so that the irrelevant features can be eliminated. The experimental results show the proposed approach is faster and more lightweight than the previous approaches while guaranteeing high detection rates so that it is suitable for real-time NIDS. Singh et al. [30] used the parallel processing power of Mahout to build random forest-based decision tree model, which was applied to the problem of peer-to-peer botnet detection in quasi-real-time. The random forest algorithm was chosen because the problem of botnet detection has the requirements of high accuracy of prediction, ability to handle diverse bots, ability to handle data characterized by a very large number and diverse types of descriptors, ease of training, and computational efficiency. Ronao et al. [31] proposed a random forest with weighted voting (WRF) and principal components analysis (PCA) as a feature selection technique for the task of detecting database access anomalies. RF exploits the inherent tree-structure syntax of SQL queries, and its weighted voting scheme further minimizes false alarms. Experiments showed that WRF achieved the best performance, even on very skewed data.

Taft et al. [32] applied SMOTE as an enhanced sampling method in a sparse dataset to generate prediction models to identify adverse drug events (ADE) in women admitted for labor and delivery based on patient risk factors and comorbidities. By creating synthetic cases with the SMOTE algorithm and using a 10-fold cross-validation technique, they demonstrated improved performance of the naïve Bayes and the decision tree algorithms. Sun et al. [33] proposed a new imbalance-oriented SVM method that combines the synthetic minority over-sampling technique (SMOTE) with the Bagging ensemble learning algorithm and uses SVM as the base classifier. It is named as the SMOTE-Bagging-based SVM-ensemble (SB-SVM-ensemble), which is theoretically more effective for financial distress prediction (FDP) modeling based on imbalanced datasets with a limited number of samples. Santos et al. [34] proposed a new cluster-based oversampling approach robust to small and imbalanced datasets, which accounts for the heterogeneity of patients with hepatocellular carcinoma, and a representative dataset was built based on K-means clustering and the SMOTE algorithm, which was used as a training example for different machine learning procedures.

The intrusion detection for wireless sensor networks mainly solves the classification problem of normal data and attack data. To improve the situation of class imbalance of the dataset, this paper proposed a classification method of random forest combined with the SMOTE. A random forest is a combined classifier that uses the method of resampling to extract multiple samples from the original samples and trains the sample set obtained by each sampling to establish a decision tree. Then these decision trees is combined together to form the random forest. The classification results of each decision tree are combined by voting to finally complete the classification prediction. The main content

of the SMOTE is to insert new samples that are generated randomly between minority class samples and their neighbors, which can increase the number of minority class samples. In this paper, SMOTE is used to oversample the dataset, then the training set is reconstructed and the original data of class imbalance is balanced. After that, the random forest algorithm is used to train the new training set and the classifier is generated to realize the intrusion detection for wireless sensor networks.

The remaining sections of this paper are organized as follows. Section 2 describes the principle of SMOTE. Section 3 describes the random forest algorithm. Section 4 describes the intrusion detection technology combined with SMOTE and random forest. Section 5 describes experimental results and analysis, including dataset, evaluation indicators, results and comparison. Finally, Section 6 summarizes the paper.

2. Principle of SMOTE

Synthetic minority oversampling technology (SMOTE) is a heuristic oversampling technique proposed by Chawla et al. to solve the problem of class imbalance [35]. It has significantly improved the situation of over-fitting caused by non-heuristic random oversampling method, so it has been widely used in the field of class imbalance in recent years. The core idea of SMOTE is to insert new samples that generated randomly between minority class samples and their neighbors, which can increase the number of minority class samples and improve the situation of class imbalance [36–39].

Firstly, the K nearest neighbors are searched for each data sample X in the minority class samples. Assuming that the sampling magnification of the dataset is N, N samples are randomly selected from K nearest neighbors (there must be $K > N$), and the N samples are recorded as $y_1, y_2, \cdots, y_N$. The data samples X and y_i are correlated, and the corresponding random interpolation operation is performed by the correlation formula between X and $y_i (i = 1, 2, \cdots, N)$ to obtain the interpolated sample p_i, so that N corresponding minority class samples are constructed for each data sample.

The interpolation formula is as follows:

$$p_i = X + rand(0, 1) \times (y_i - X), i = 1, 2, \cdots, N \tag{1}$$

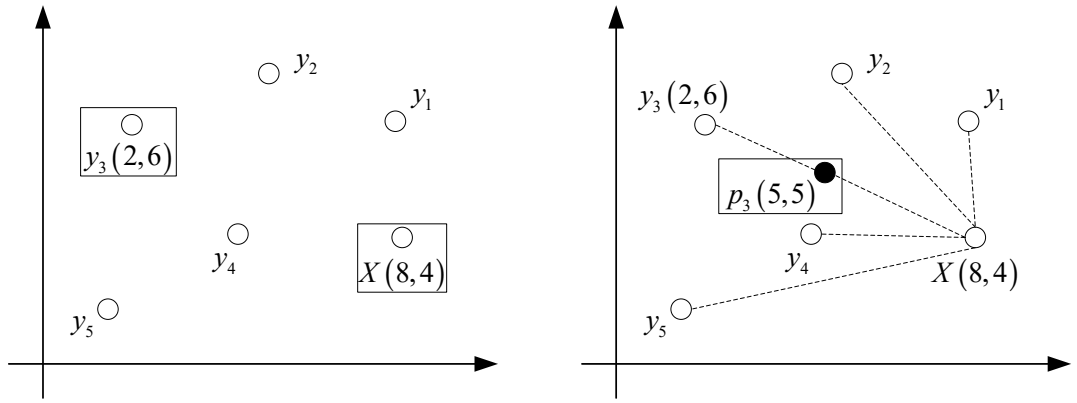
where X represents a data sample in minority class samples, $rand(0, 1)$ represents a random number within the interval (0,1), and y_i represents the ith of the N nearest neighbors of the data sample X.

The sampling magnification N depends on the imbalanced degree of the dataset. The formula for calculating the imbalanced level (IL) between the majority class and the minority class of the dataset is as follows:

$$N = round(IL) \tag{2}$$

where $round(IL)$ represents the value obtained by rounding up the IL. Through the above interpolation operation, the majority class samples and the minority class samples can be effectively balanced, thereby improving the classification accuracy of imbalanced datasets.

In order to represent the interpolation process of SMOTE, it is assumed that there is a two-dimensional dataset, taking one of the data sample points X, whose coordinates are (8,4). The random value of rand (0,1) is set to 0.5, and the coordinates of a nearest sample point of X are set to (2,6). The representations of data sample X and its five nearest neighbors is shown in Figure 1a.



(**a**) (**b**)

Figure 1. (**a**) The data sample X and its five nearest neighbors; and (**b**) interpolation principle of SMOTE.

Figure 1a shows that the data sample $X(8,4)$ has obtained its five nearest neighbors $(y_1, y_2, y_3, y_4, y_5)$, and now the sampling operation between X and the neighbor y_3 is performed.

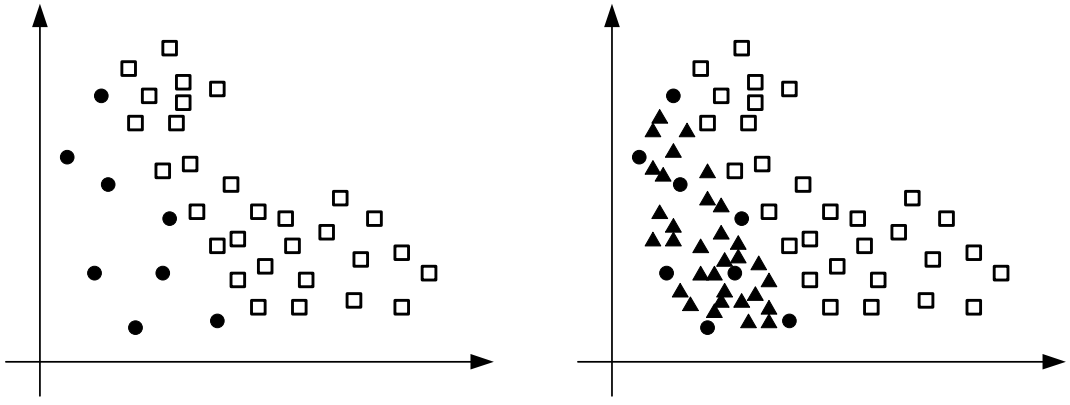
According to Equations (1) and (2), the following results can be obtained:

$$p_3 = X + rand(0,1) \times (y_3 - X) = (8,4) + 0.5 \times ((2,6) - (8,4)) = (5,5) \quad (3)$$

That is, the newly generated interpolation is $p_3(5,5)$.

The whole interpolation process for generating new data is represented on the two-dimensional axis as shown in Figure 1b. It can be seen from the figure that the sampling of SMOTE is a random interpolation operation on the line between the data sample and its nearest neighbor. This method can be regarded as a linear interpolation, but its effect has been greatly improved compared with the simple replication of original data samples.

Now consider a more obvious imbalanced dataset. Suppose that there are 30 samples in the majority class and eight samples in the minority class. The distribution of the dataset is shown in Figure 2a. It can be seen from the figure that the difference between the majority class samples and the minority class samples is large. If data classification is performed in this case, the accuracy of data classification will be seriously reduced. Therefore, SMOTE is used to oversample the imbalanced data. According to the principle of SMOTE and the Equation (1), it can be known that when the sampling magnification is 4, it is possible to make the minority class samples reach the same number as the majority class samples.



(**a**) (**b**)

Figure 2. (**a**) Imbalanced dataset; and (**b**) entirely oversampling.

The effect of oversampling the whole imbalanced dataset with SMOTE is shown in Figure 2b. The circle represents minority class, the square represents majority class, and the triangle represents synthetic data. It can be seen from the figure that for each minority class sample, four minority class samples of its nearest neighbors are selected for interpolation operation, and all interpolations are on a certain line between the original minority class sample and its nearest neighbor.

Through the analysis of SMOTE and the analysis of the imbalanced dataset before oversampling, it can be seen that the oversampling algorithm based on SMOTE has the following advantages. Firstly, SMOTE reduces the limitations and blindness of the oversampling algorithm for imbalanced data in the sampling process. The sampling method before SMOTE is a random upward sampling method, which can balance the dataset, but the sampling effect is not ideal because of the serious lack of principle of random sampling. The basic mathematical theory of linear interpolation is adopted by SMOTE. For data sample X, K samples of its nearest neighbors are selected, and then data are generated purposefully according to certain mathematical rules, which can effectively avoid blindness and limitations. Secondly, SMOTE effectively reduces the phenomenon of over-fitting. The method of replicating data is adopted by the traditional over-sampling technology. Since the decision domain becomes smaller in the sampling process, it leads to over-fitting. SMOTE can effectively avoid this defect.

3. Random Forest Algorithm

Random forest is an ensemble learning model which takes decision tree as a basic classifier. It contains several decision trees trained by the method of Bagging [40]. When a sample to be classified is entered, the final classification result is determined by the vote of the output of a single decision tree. Random forest overcomes the over-fitting problem of decision trees, has good tolerance to noise and anomaly values, and has good scalability and parallelism to the problem of high-dimensional data classification. In addition, random forest is a non-parametric classification method and driven by data. It trains classification rules by learning given samples, and does not require prior knowledge of classification.

The random forest model is based on K decision trees. Each tree votes on which class a given independent variable X belongs to, and only one vote is given to the class it considers most appropriate [41–43]. The description of the K decision trees is as follows:

$$\{h(X, \theta_k), k = 1, 2, \cdots, K\} \tag{4}$$

Among them, K is the number of decision trees contained in random forests. θ_k represents independent and identically distributed random vectors.

The steps to generate a random forest are as follows:

1. The method of random repeated sampling is applied to randomly extract K samples from the original training set as self- service sample set, and then K classification regression trees are generated.
2. Assuming that the original training set has n features, m features are randomly selected at each node of each tree $(m \leq n)$. By calculating the amount of information contained in each feature, a feature with the most classification ability is selected among the m features for node splitting.
3. Every tree grows to its maximum without any cutting.
4. The generated trees are composed of random forest, and the new data is classified by random forest. The classification results are determined by the number of votes of the tree classifiers.

The similarity and correlation of decision trees are important features of random forest to reflect generalization performance, while generalization error reflects generalization ability of the system. Generalization ability is the ability of the system to make correct judgments on new data with the same distribution outside the training sample set. Smaller generalization error can make the system get better performance and stronger generalization ability.

The generalization error is defined as follows:

$$PE^* = P_{X,Y}(mr(X,Y) < 0) \tag{5}$$

where PE^* represents the generalization error, the subscript X, Y indicates the definition space of the probability, and $mr(X,Y)$ is the margin function.

The margin function is defined as follows:

$$mr(X,Y) = avg_k I(h(X,\theta_k) = Y) - \max_{J \neq Y} avg_k I(h(X,\theta_k) = J) \tag{6}$$

where X is the input sample, Y is the correct classification, and J is the incorrect classification. $I(g)$ is an indicative function, $avg_k(g)$ means averaging, and $h(g)$ represents a sequence of classification model. The margin function reflects the extent to which the numbers of votes for the correct classification corresponding to sample X exceeds the maximum number of votes for other incorrect classifications. The larger the value of margin function is, the higher the confidence of the classifier will be.

The convergence expression of generalization error is defined as follows:

$$\lim_{k \to \infty} PE^* = P_{X,Y}(P_\theta(I(h(X,\theta_k) = Y)) - \max_{J \neq Y} P_\theta(I(h(X,\theta_k) = J))) \tag{7}$$

The above formula indicates that the generalization error will tend to an upper bound, and the model will not over-fit with the increase of the number of decision trees.

The upper bound of the generalization error is available, depending on the classification strength of the single tree and the correlation between the trees. The random forest model aims to establish a random forest with low correlation and high classification intensity. Classification intensity S is the mathematical expectation of $mr(X,Y)$ in the whole sample space:

$$S = E_{X,Y} mr(X,Y) \tag{8}$$

θ and θ' are independent and identically distributed vectors, and the correlation coefficients of $mr(\theta, X, Y)$ and $mr(\theta', X, Y)$ is defined as follows:

$$\overline{\rho} = \frac{\mathrm{cov}_{X,Y}(mr(\theta,X,Y), mr(\theta',X,Y))}{sd(\theta) sd(\theta')} \tag{9}$$

Among them, $sd(\theta)$ can be expressed as follows:

$$sd(\theta) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(mr(x_i,\theta) - \frac{1}{N}\sum_{i=1}^{N} mr(x_i,\theta)\right)^2} \tag{10}$$

In Equation (9), the correlation between the trees of $h(X,\theta)$ and $h(X,\theta')$ on the dataset of X and Y can be measured by the $\overline{\rho}$. The larger the $\overline{\rho}$, the larger the correlation coefficient.

According to Chebyshev inequality, the upper bound of generalization error can be derived:

$$P_{X,Y}(mr(X,Y) < 0) \leq \frac{\overline{\rho}(1 - S^2)}{S^2} \tag{11}$$

It can be seen that the bound of generalization error of random forest is negatively correlated with the classification intensity S of a single decision tree and positively correlated with the correlation P between decision trees. That is, the larger the classification intensity S, the smaller the correlation P. The smaller the bound of generalization error is, the higher the classification accuracy of the random forest will be.

4. Intrusion Detection Technology Combined with SMOTE and Random Forest

The intrusion detection for wireless sensor networks can be regarded as a classification problem, and the dataset can be divided into normal data and attack data. To solve the problem of class imbalance between normal data and attack data and improve the classification accuracy, SMOTE is used to oversample the dataset. After oversampling, the training set is reconstructed and the original data of class imbalance is balanced. Then the random forest algorithm is used to train the new training set, which has been balanced. Finally, the classifier is generated to realize the intrusion detection for wireless sensor networks. The architecture of intrusion detection system proposed in this paper is shown in Figure 3.

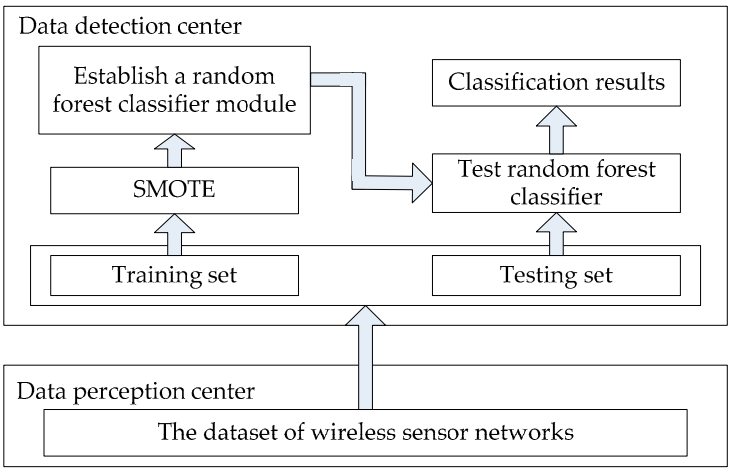


Figure 3. The architecture of intrusion detection system.

The steps of intrusion detection for wireless sensor networks based on SMOTE and random forest algorithm are as follows:

1. Suppose that the sample space of attack data of wireless sensor networks is P and the sample space of normal data is Q. P consists of n samples of attack data, and Y_i represents the features of the ith attack data. Thus, P can be represented as $P = \{Y_1, Y_2, \cdots, Y_n\}$. For each sample, there are f features, recorded as $Y_i = \{F_{i1}, F_{i2}, \cdots, F_{if}\}$.
2. For each sample Y_i in the attack data set, the Euclidean distance is used to calculate the distance from it to all other samples in P, and its K nearest neighbors are obtained.
3. The sampling magnification N is set according to the ratio of the number of attack data samples P to the number of normal data samples Q. N neighbors are randomly selected from the K nearest neighbors of each attack data sample Y_i, recorded as Y_j', where $j = 1, 2, \cdots, N$.
4. Each randomly-selected neighbor sample B constructs a new attack data sample with attack data sample D according to Equation (12). The $rand(0,1)$ represents a random number of the interval [0,1]:

$$Y_{new} = Y_i + rand(0,1) \cdot (Y_j' - Y_j) \tag{12}$$

5. Combine the constructed new samples with the normal data samples Q to generate a new data sample space R.
6. Assuming X_i represents the ith data sample, then $R = \{X_1, X_2, \cdots, X_n\}$. For each sample, there are f features, which are recorded as $X_i = \{F_{i1}, F_{i2}, \cdots, F_{if}\}$. Select the decision tree and use it as the base classifier.
7. A new training set R_j is generated by sampling from the data sample space R using the method of Bootstrap, and a decision tree is constructed by R_j.

8. The $k(k \leq f)$ features are randomly extracted from the nodes of each decision tree. By calculating the amount of information contained in each feature, a feature with the most classification ability is selected among the k features to split the nodes until the tree grows to the maximum.
9. Repeat steps 7 and 8 for m times to train m decision trees.
10. The generated decision trees are composed of random forest, and the new data is classified by the random forest. The classification results are determined by the number of votes of the tree classifiers.

The method of SMOTE + random forest takes attack data as a minority class and generates new attack data through SMOTE, which reduces the difference in the number of attack data and normal data, and reduces the imbalance of the training set. The method can obtain better classification effect and effectively improve the accuracy of intrusion detection for wireless sensor networks.

5. Experimental Results and Analysis

5.1. Dataset and Evaluation Indicators

The KDD Cup 99 dataset [44], which is widely recognized in the field of intrusion detection, is used as training and testing set. The dataset is a network traffic data set created by MIT Lincoln Laboratory by simulating the local area network environment of the U.S. Air Force. There are different probability distributions for testing data and training data, and the testing set contains some types of attacks that do not appear in the training set, which makes the intrusion detection more realistic.

The dataset has 41 different attributes, and it can be divided into five different types, one normal type and four attack types (DoS, Probing, U2R, and R2L). Denial of service (DoS) attacks prevent legitimate requests for network resources by consuming bandwidth or overloading computational resources. Probing attack refers to when an attacker scans the network to collect information about the target system before launching an attack. User to root (U2R) attack refers to that legitimate users obtain the root access right of the system by illegal means. Root to local (R2L) attack refers to the attack method of gaining access to the local host by sending customized network packets. Since the dataset is too large, 49,402 records are randomly selected from the "10% KDD Cup 99 training set" as training data, and 31,102 records are randomly selected from the "KDD Cup 99 corrected labeled test dataset" as testing data. The distribution of various types of data in training set and testing set is shown in Figure 4.

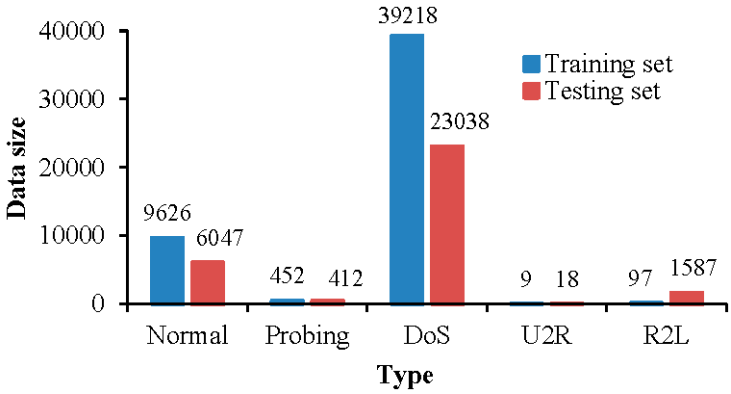


Figure 4. Distribution of various types of data in the dataset.

In intrusion detection systems, accuracy, precision, AUC, etc. are usually used as indicators to evaluate the system [45,46]. Accuracy is the proportion of the records correctly classified, which is defined as follows:

$$accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{13}$$

Among them, *TP* refers to the number of records that attack behavior is recognized as attack behavior, *TN* refers to the number of records that normal behavior is recognized as normal behavior, *FP* refers to the number of records that normal behavior is recognized as attack behavior, *FN* refers to the number of records that attack behavior is recognized as normal behavior.

Precision is the proportion of the records that are actually attacks in the records that are predicted to attacks. Precision is higher, indicating that the false positive rate (FPR) of the system is lower. Precision is defined as follows:

$$precision = \frac{TP}{TP + FP} \tag{14}$$

Area under the curve (AUC) is defined as the area under the ROC curve. Obviously, the value of this area will not be greater than 1. Because ROC curve is generally above the line $y = x$, AUC ranges from 0.5 to 1. The AUC value is used as the evaluation criterion because ROC curve cannot clearly explain which classifier is better in many cases, and AUC as a numerical value can intuitively explain that the classifier with larger AUC has better effect.

5.2. Results and Comparison

The experimental environment of this experiment is mainly based on Weka [47], a famous open source software for machine learning and data mining. All comparison algorithms are also derived from the data packages provided by Weka. The experiment was implemented on 2.6 GHz Intel core i5-3320M processor with 4GB RAM. In this paper, the classical single classifiers and ensemble classifiers in Weka are selected and compared. The single classifiers include J48 [48], NaiveBayes [49], LibSVM [50], and the ensemble classifiers include Bagging [51], AdaBoostM1 [52], and RandomForest [53].

The precision of each classifier is shown in Table 1. It can be seen from the table that the classification results of minority classes, such as probing, U2R, and R2L, are poor, and the problem of class imbalance is obvious. The AUC value of each classifier is shown in Table 2. It can be seen from the table that classifiers such as J48, AdaboostM1 and RandomForest have better classification effect than LibSVM, NaiveBayes and Bagging for the problem of class imbalance.

Table 1. The precision of each classifier.

Type\Classifier	J48	LibSVM	NaiveBayes	Bagging	AdaboostM1	RandomForest
Normal	0.728	0.562	0.730	0.758	0.682	0.728
Probing	0.808	0.697	0.083	0.890	0.000	0.896
DoS	0.998	0.997	0.992	0.982	0.984	0.995
U2R	0.000	0.000	0.049	0.000	0.000	1.000
R2L	0.983	0.000	0.771	0.680	0.000	0.990

Table 2. The AUC value of each classifier.

Type\Classifier	J48	LibSVM	NaiveBayes	Bagging	AdaboostM1	RandomForest
Normal	0.951	0.904	0.970	0.979	0.950	0.974
Probing	0.890	0.686	0.978	0.924	0.919	0.993
DoS	0.980	0.933	0.896	0.995	0.971	0.990
U2R	0.644	0.500	0.519	0.908	0.928	0.935
R2L	0.804	0.500	0.949	0.567	0.888	0.669

The training time and testing time of each classifier are shown in Figure 5a. It can be seen from the figure that the training and testing time of LibSVM classifier is much longer than other classifiers, and the data processing speed is slower. The accuracy of each classifier is shown in Figure 5b. It can be seen from the figure that the accuracy of the J48, Bagging, and RandomForest classifiers is high, especially the classification effect of the RandomForest classifier is the best.

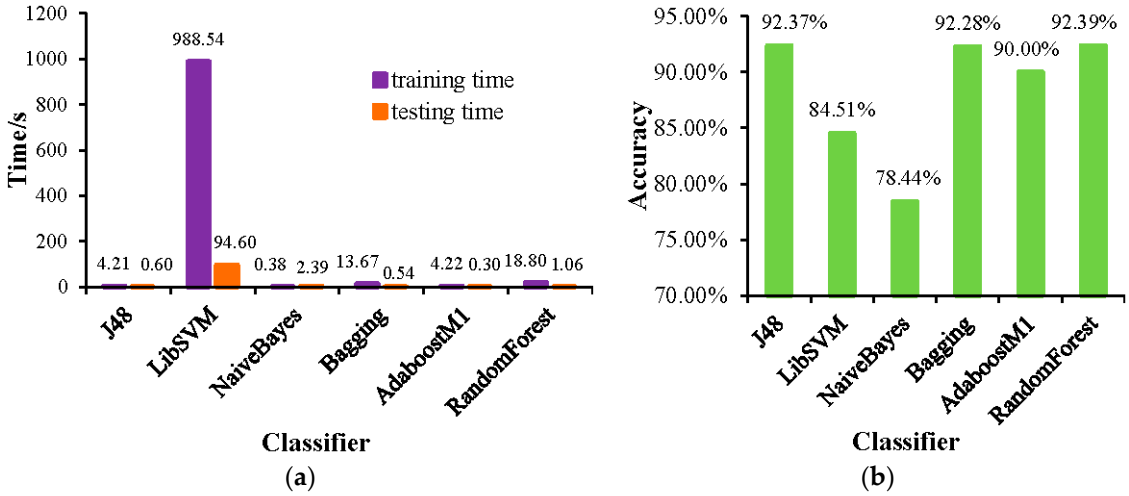


Figure 5. (**a**) The training time and testing time of each classifier; and (**b**) the accuracy of each classifier.

Since the classification effect of minority classes, such as probing, U2R, and R2L, is poor, the method of SMOTE is used to solve the problem of class imbalance. In order to verify the effect of the previous six classifiers combined with the SMOTE, the classifiers are tested with the SMOTE respectively. The precision and AUC value of each classifier combined with the SMOTE are shown in Tables 3 and 4, respectively. It can be seen from the table that values of precision and AUC have been improved.

Table 3. The precision of each classifier combined with the SMOTE.

Type\Classifier	J48	LibSVM	NaiveBayes	Bagging	AdaboostM1	RandomForest
Normal	0.725	0.562	0.809	0.759	0.682	0.728
Probing	0.904	0.697	0.086	0.453	0.000	0.901
DoS	0.998	0.997	0.886	0.994	0.984	0.999
U2R	0.375	0.000	0.044	0.200	0.000	0.333
R2L	0.941	0.000	0.717	0.944	0.000	0.981

Table 4. The AUC value of each classifier combined with the SMOTE.

Type\Classifier	J48	LibSVM	NaiveBayes	Bagging	AdaboostM1	RandomForest
Normal	0.949	0.904	0.970	0.974	0.943	0.976
Probing	0.891	0.686	0.981	0.868	0.774	0.995
DoS	0.982	0.933	0.892	0.987	0.970	0.986
U2R	0.720	0.500	0.542	0.856	0.869	0.995
R2L	0.519	0.500	0.949	0.571	0.921	0.677

The training time and testing time of each classifier combined with the SMOTE are shown in Figure 6a. It can be seen from the figure that the training time and testing time of each classifier are significantly shortened. The accuracy of each classifier combined with the SMOTE is shown in Figure 6b. It can be seen from the figure that the accuracy of Bagging and RandomForest classifiers has been improved after using the method of SMOTE. In this experiment, the accuracy of the method of SMOTE+RandomForest reaches 92.57%, which is the best performance of all methods.

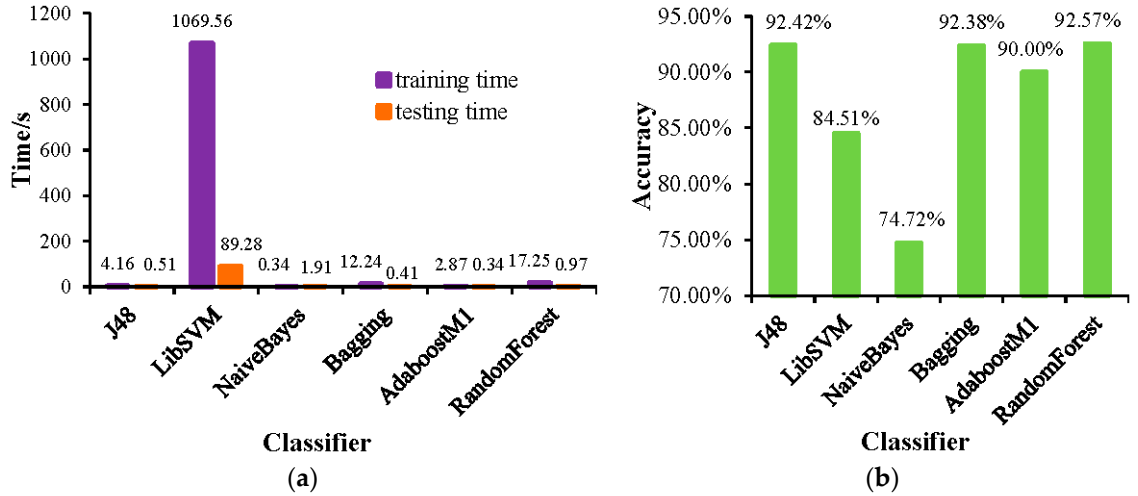


Figure 6. (**a**) The training time and testing time of each classifier combined with the SMOTE; and (**b**) the accuracy of each classifier combined with SMOTE.

Due to the similar accuracy between J48 and RandomForest, the comparative experiments under datasets with different sampling proportions are carried out. Five different datasets are randomly selected from "10% KDD Cup 99 training set" and "KDD Cup 99 corrected labeled test dataset" according to 5%, 7.5%, 10%, 12.5%, and 15% sampling proportions. The amount of training data and testing data in each dataset is shown in Table 5.

Table 5. The amount of training data and testing data in each dataset.

Type\Dataset	Dataset1	Dataset2	Dataset3	Dataset4	Dataset5
Training data	24,701	37,051	49,402	61,752	74,103
testing data	15,551	23,327	31,102	38,878	46,654

The comparison of the performance of J48, RandomForest, S+J48, and S+RandomForest under different proportions of datasets is shown in Figure 7. It can be seen from the figure that the accuracy of RandomForest is better than that of J48, the accuracy of J48 and RandomForest combined with SMOTE is higher than that without SMOTE.

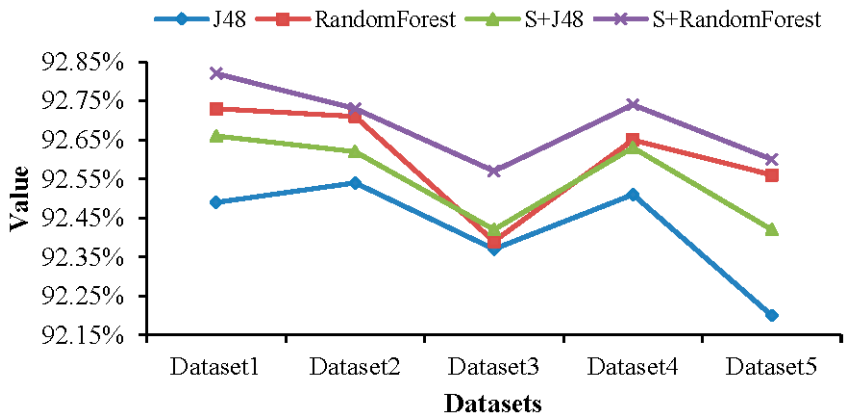


Figure 7. Comparison of the performance of J48, RandomForest, S+J48, and S+RandomForest under different proportions of datasets.

6. Conclusions

The intrusion detection for wireless sensor networks is an important subject in the field of the security of wireless sensor networks. Due to the class imbalance in KDD Cup 99 dataset, this study combines the SMOTE with the random forest algorithm, and proposes an ensemble classifier for imbalanced datasets. Experiments on KDD Cup 99 dataset show that the classification accuracy of random forest algorithm has reached 92.39%, which is higher than other classification methods, such as J48, LibSVM, NaiveBayes, Bagging, and AdaboostM1. After combining with the SMOTE, the classification accuracy of the random forest has increased to 92.57%, which improves the classification effect of minority classes. The random forest method combined with the SMOTE provides an effective solution to solve the problem of class imbalance and improves the classification accuracy of intrusion detection. Moreover, this method is simple to implement and has strong generalization ability. It can be widely used in the field of the security of wireless sensor networks to improve the effect of intrusion detection for wireless sensor networks. In the future, this research will continue to find new classification methods to further improve the recognition effect of the intrusion data of wireless sensor networks.

Author Contributions: Conceptualization: X.T., S.S., and Z.H.; formal analysis: S.S. and Z.H.; investigation: Z.Z.; methodology: X.T., X.G., and Z.Z.; resources: X.G.; software: X.S. and L.L.; supervision: S.S. and Z.H.; validation: X.T., X.S., and L.L.; writing—original draft: X.T., X.G., and Z.Z.; writing—review and editing: S.S. and Z.H.

Funding: This work was supported by the Natural Science Foundation of Hunan Province under grant no. 2018JJ3607, the National Natural Science Foundation of China under grant no. 51575517, the National Technology Foundation Project under grant no. 181GF22006, and the Frontier Science and Technology Innovation Project in the National Key Research and Development Program under grant no. 2016QY11W2003.

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

References

1. Zhang, Z.; Glaser, S.; Bales, R.; Conklin, M.; Rice, R.; Marks, D. Technical report: The design and evaluation of a basin-scale wireless sensor network for mountain hydrology. *Water Resour. Res.* **2017**, *53*, 4487–4498. [CrossRef]
2. Victor, G.F.; Carles, G.; Helena, R.P. A Comparative Study of Anomaly Detection Techniques for Smart City Wireless Sensor Networks. *Sensors* **2016**, *16*, 868–887.
3. Lu, T.; Liu, G.; Chang, S. Energy-efficient data sensing and routing in unreliable energy-harvesting wireless sensor network. *Wirel. Netw.* **2018**, *24*, 611–625. [CrossRef]
4. Fang, X.; Nan, L.; Jiang, Z.; Chen, L. Fingerprint localisation algorithm for noisy wireless sensor network based on multi-objective evolutionary model. *IET Commun.* **2017**, *11*, 1297–1304. [CrossRef]
5. Wang, J.; Jiang, S.; Fapojuwo, A.O. A Protocol Layer Trust-Based Intrusion Detection Scheme for Wireless Sensor Networks. *Sensors* **2017**, *17*, 1227. [CrossRef] [PubMed]
6. Ferng, H.W.; Khoa, N.M. On security of wireless sensor networks: A data authentication protocol using digital signature. *Wirel. Netw.* **2017**, *23*, 1113–1131. [CrossRef]
7. Ismail, B.; In-Ho, R.; Ravi, S. An Intrusion Detection System Based on Multi-Level Clustering for Hierarchical Wireless Sensor Networks. *Sensors* **2015**, *15*, 28960–28978.
8. Li, M.; Lou, W.; Ren, K. Data security and privacy in wireless body area networks. *IEEE Wirel. Commun.* **2016**, *17*, 51–58. [CrossRef]
9. Ren, J.; Zhang, Y.; Zhang, K.; Shen, X. Adaptive and Channel-Aware Detection of Selective Forwarding Attacks in Wireless Sensor Networks. *IEEE Trans. Wirel. Commun.* **2016**, *15*, 3718–3731. [CrossRef]
10. Wang, J.; Wang, F.; Cao, Z.; Lin, F.; Wu, J. Sink location privacy protection under direction attack in wireless sensor networks. *Wirel. Netw.* **2017**, *23*, 579–591. [CrossRef]
11. Xiao, X.; Zhang, R. Study of Immune-Based Intrusion Detection Technology in Wireless Sensor Networks. *Arab. J. Sci. Eng.* **2017**, *42*, 3159–3174. [CrossRef]
12. Yan, J.; Li, X.; Luo, X.; Guan, X. Virtual-Lattice Based Intrusion Detection Algorithm over Actuator-Assisted Underwater Wireless Sensor Networks. *Sensors* **2017**, *17*, 1168. [CrossRef] [PubMed]

13. Kalnoor, G.; Agarkhed, J. Detection of Intruder using KMP Pattern Matching Technique in Wireless Sensor Networks. *Proc. Comput. Sci.* **2018**, *125*, 187–193. [CrossRef]
14. Osanaiye, O.A.; Alfa, A.S.; Hancke, G.P. Denial of Service Defence for Resource Availability in Wireless Sensor Networks. *IEEE Access* **2018**, *6*, 6975–7004. [CrossRef]
15. Ma, T.; Wang, F.; Cheng, J.; Yu, Y.; Chen, X. A Hybrid Spectral Clustering and Deep Neural Network Ensemble Algorithm for Intrusion Detection in Sensor Networks. *Sensors* **2016**, *16*, 1701. [CrossRef] [PubMed]
16. Wazid, M.; Das, A. An Efficient Hybrid Anomaly Detection Scheme Using K-Means Clustering for Wireless Sensor Networks. *Wirel. Pers. Commun.* **2016**, *90*, 1971–2000. [CrossRef]
17. Belavagi, M.C.; Muniyal, B. Performance Evaluation of Supervised Machine Learning Algorithms for Intrusion Detection. *Proc. Comput. Sci.* **2016**, *89*, 117–123. [CrossRef]
18. Lu, N.; Sun, Y.; Liu, H.; Li, S. Intrusion Detection System Based on Evolving Rules for Wireless Sensor Networks. *J. Sens.* **2018**, *2018*, 1–8. [CrossRef]
19. Singh, R.; Singh, J.; Singh, R. Fuzzy Based Advanced Hybrid Intrusion Detection System to Detect Malicious Nodes in Wireless Sensor Networks. *Wirel. Commun. Mob. Comput.* **2017**, *2017*, 1–14. [CrossRef]
20. Sun, Z.; Xu, Y.; Liang, G.; Zhou, Z. An Intrusion Detection Model for Wireless Sensor Networks with an Improved V-Detector Algorithm. *IEEE Sens. J.* **2018**, *18*, 1971–1984. [CrossRef]
21. Tajbakhsh, A.; Rahmati, M.; Mirzaei, A. Intrusion detection using fuzzy association rules. *Appl. Soft. Comput.* **2009**, *9*, 462–469. [CrossRef]
22. Xie, M.; Hu, J.; Guo, S.; Zomaya, A.Y. Distributed Segment-Based Anomaly Detection with Kullback–Leibler Divergence in Wireless Sensor Networks. *IEEE Trans. Inf. Forensic Secur.* **2017**, *12*, 101–110. [CrossRef]
23. Xie, M.; Hu, J.; Guo, S. Segment-based anomaly detection with approximated sample covariance matrix in wireless sensor networks. *IEEE Trans. Parallel Distrib. Syst.* **2015**, *26*, 574–583. [CrossRef]
24. Haider, W.; Hu, J.; Slay, J.; Turnbull, B.P.; Xie, Y. Generating realistic intrusion detection system dataset based on fuzzy qualitative modeling. *J. Netw. Comput. Appl.* **2017**, *87*, 185–192. [CrossRef]
25. Ye, Y.; Li, T.; Adjeroh, D.; Iyengar, S.S. A survey on malware detection using data mining techniques. *ACM Comput. Surv.* **2017**, *50*, 41. [CrossRef]
26. Kumar, M.; Hanumanthappa, M. Intrusion detection system using stream data mining and drift detection method. *Res. Vet. Sci.* **2013**, *93*, 168–171.
27. Khorshidpour, Z.; Hashemi, S.; Hamzeh, A. Evaluation of random forest classifier in security domain. *Appl. Intell.* **2017**, *47*, 558–569. [CrossRef]
28. Paul, A.; Mukherjee, D.P.; Das, P.; Gangopadhyay, A.; Chintha, A.R.; Kundu, S. Improved Random Forest for Classification. *IEEE Trans. Image Process.* **2018**, *27*, 4012–4024. [CrossRef]
29. Lee, S.M.; Kim, D.S.; Park, J.S. A Hybrid Approach for Real-Time Network Intrusion Detection Systems. *IEEE Trans. Veh. Technol.* **2011**, *60*, 457–472.
30. Singh, K.; Guntuku, S.C.; Thakur, A.; Hota, C. Big Data Analytics framework for Peer-to-Peer Botnet detection using Random Forests. *Inf. Sci.* **2014**, *278*, 488–497. [CrossRef]
31. Ronao, C.A.; Cho, S.B. Anomalous query access detection in RBAC-administered databases with random forest and PCA. *Inf. Sci.* **2016**, *369*, 238–250. [CrossRef]
32. Taft, L.M.; Evans, R.S.; Shyu, C.R.; Egger, M.J.; Chawla, N.; Mitchell, J.A.; Thornton, S.N.; Bray, B.; Varner, M. Countering imbalanced datasets to improve adverse drug event predictive models in labor and delivery. *J. Biomed. Inform.* **2009**, *42*, 356–364. [CrossRef] [PubMed]
33. Sun, J.; Shang, Z.; Li, H. Imbalance-oriented SVM methods for financial distress prediction: A comparative study among the new SB-SVM-ensemble method and traditional methods. *J. Oper. Res. Soc.* **2014**, *65*, 1905–1919. [CrossRef]
34. Santos, M.S.; Abreu, P.H.; García-Laencina, P.J.; Simão, A.; Carvalho, A. A new cluster-based oversampling method for improving survival prediction of hepatocellular carcinoma patients. *J. Biomed. Inform.* **2015**, *58*, 49–59. [CrossRef] [PubMed]
35. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
36. Lusa, L. SMOTE for high-dimensional class-imbalanced data. *BMC Bioinform.* **2013**, *14*, 106.
37. Jeatrakul, P.; Wong, K.W.; Fung, C.C. Classification of imbalanced data by combining the complementary neural network and SMOTE algorithm. In Proceedings of the International Conference on Neural Information Processing, Sydney, Australia, 22–25 November 2010; pp. 152–159.

38. Wang, J.; Xu, M.; Wang, H.; Zhang, J. Classification of imbalanced data by using the SMOTE algorithm and locally linear embedding. In Proceedings of the International Conference on Signal Processing, Beijing, China, 16–20 November 2006.
39. Blagus, R.; Lusa, L. Evaluation of smote for high-dimensional class-imbalanced microarray data. In Proceedings of the International Conference on Machine Learning and Applications, Boca Raton, FL, USA, 12–15 December 2012; pp. 89–94.
40. Breiman, L. Bagging predictors. *Mach. Learn.* **1996**, *24*, 123–140. [CrossRef]
41. Hasan, M.A.; Nasser, M.; Ahmad, S.; Molla, M.K. Feature Selection for Intrusion Detection Using Random Forest. *J. Inf. Secur.* **2016**, *7*, 129–140. [CrossRef]
42. Farnaaz, N.; Jabbar, M.A. Random forest modeling for network intrusion detection system. *Proc. Comput. Sci.* **2016**, *89*, 213–217. [CrossRef]
43. Yi, Y.A.; Min, M.M. An analysis of random forest algorithm based network intrusion detection system. In Proceedings of the International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, Kanazawa, Japan, 26–28 June 2017; pp. 127–132.
44. KDD Cup 1999 Data. Available online: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html (accessed on 20 September 2018).
45. Liu, Y.; Xiang, C.; Wang, H. Optimization of feature selection based on mutual information in intrusion detection. *J. Northwest. Univ.* **2017**, *47*, 666–673.
46. Yan, J.H. Optimization Boosting Classification Based on Metrics of Imbalanced Data. *Comput. Eng. Appl.* **2018**, *54*, 1–6.
47. Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; Witten, I.H. The WEKA data mining software: An update. *ACM SIGKDD Explor. Newsl.* **2009**, *11*, 10–18. [CrossRef]
48. Sahu, S.; Mehtre, B.M. Network intrusion detection system using J48 Decision Tree. In Proceedings of the International Conference on Advances in Computing, Communications and Informatics, Kochi, India, 10–13 August 2015.
49. Amor, N.B.; Benferhat, S.; Elouedi, Z. Naive Bayes vs decision trees in intrusion detection systems. In Proceedings of the ACM Symposium on Applied Computing, Nicosia, Cyprus, 14–17 March 2004; pp. 420–424.
50. Chang, C.C.; Lin, C.J. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.* **2011**, *2*, 27. [CrossRef]
51. Gaikwad, D.P.; Thool, R.C. Intrusion Detection System Using Bagging Ensemble Method of Machine Learning. In Proceedings of the International Conference on Computing Communication Control & Automation, Pune, India, 26–27 February 2015.
52. Cortes, E.A.; Martinez, M.G.; Rubio, N.G. Multiclass corporate failure prediction by Adaboost.M1. *Int. Adv. Econ. Res.* **2007**, *13*, 301–312. [CrossRef]
53. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]