Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2016, Article ID 8752181, 9 pages
http://dx.doi.org/10.1155/2016/8752181

Research Article

An Efficient Cost-Sensitive Feature Selection Using Chaos Genetic Algorithm for Class Imbalance Problem

Jing Bian,[1,2] Xin-guang Peng,[1] Ying Wang,[1] and Hai Zhang[3]

[1]*College of Computer Science and Technology, Taiyuan University of Technology, Yingze Street 79, Taiyuan 030024, China*
[2]*The Center of Information and Network, Shanxi Medical College of Continuing Education, Shuangtasi Street 22, Taiyuan 030012, China*
[3]*The Technology and Product Management, Shanxi Branch of Agricultural Bank of China, Nanneihuan Street 33, Taiyuan 030024, China*

Correspondence should be addressed to Xin-guang Peng; sxgrant@126.com

Received 1 February 2016; Revised 17 May 2016; Accepted 19 May 2016

Academic Editor: Muhammad N. Akram

In the era of big data, feature selection is an essential process in machine learning. Although the class imbalance problem has recently attracted a great deal of attention, little effort has been undertaken to develop feature selection techniques. In addition, most applications involving feature selection focus on classification accuracy but not cost, although costs are important. To cope with imbalance problems, we developed a cost-sensitive feature selection algorithm that adds the cost-based evaluation function of a filter feature selection using a chaos genetic algorithm, referred to as CSFSG. The evaluation function considers both feature-acquiring costs (test costs) and misclassification costs in the field of network security, thereby weakening the influence of many instances from the majority of classes in large-scale datasets. The CSFSG algorithm reduces the total cost of feature selection and trades off both factors. The behavior of the CSFSG algorithm is tested on a large-scale dataset of network security, using two kinds of classifiers: C4.5 and k-nearest neighbor (KNN). The results of the experimental research show that the approach is efficient and able to effectively improve classification accuracy and to decrease classification time. In addition, the results of our method are more promising than the results of other cost-sensitive feature selection algorithms.

1. Introduction

The class imbalance problem is found in various scientific and social arenas, such as fraud/intrusion detection, spam detection, risk management, technical diagnostics/monitoring, financial engineering, and medical diagnostics [1–4]. In most applications, it is more important to correctly classify the minority class compared to the majority class although the minority class is much smaller in number than the majority class.

There are essentially two methods to address the class imbalance problem: sampling methods and cost-sensitive learning methods [1]. The objective of sampling methods and synthetic data generation is to provide a relatively balanced distribution from oversampling and/or undersampling techniques [5]. A very popular oversampling approach is the Synthetic Minority Oversampling Technique (SMOTE), which produces synthetic minority class samples, as opposed to oversampling with replacement [6]. For high-dimensional data, Blagus and Lusa showed that SMOTE does not change the class-specific mean values, and it decreases data variability, introducing correlation between samples [7]. Cost-sensitive learning methods introduce a cost matrix to minimize total costs while maximizing accuracy [8]. When learning from imbalanced data, most classifiers are overwhelmed by most class samples, so the false negative rate is always high [9]. Researchers have introduced many methods to address these problems, including combining sampling techniques with cost-sensitive learning, setting the cost ratio by inverting prior class distributions, and collecting the cost of features before classification [5, 8, 9].

Most data mining techniques are not designed to cope with large numbers of features, and such is the case with feature selection. Currently, the class imbalance problem

is severe when data dimensionality is high. Of the many methods that exploit feature selection, the most common are those that address only relevant features, and these methods are also the most efficient and effective, which is widely known as the curse of dimensionality [10]. Many studies have realized the importance of feature selection and addressed the subject from various perspectives; this approach has been used increasingly in class imbalance problems.

In this paper, we investigate cost-sensitive feature selection issues in an imbalanced scenario. Specifically, before briefly introducing cost-sensitive learning and its application to feature selection, we illustrate the imbalanced problem, which is the most relevant topic of study in the current research. Then, we propose a new method for feature selection whose goal is to develop an efficient approach in the field of network security, an arena in which large numbers of imbalanced datasets are typical. Thus, rather than improving on previous methods, our purpose is to match the performance of previous cost-sensitive feature selection approaches using a method that addresses very large datasets with imbalance problems.

2. Related Work

2.1. Cost-Sensitive Learning. Different costs are associated with different misclassification errors in real world applications [11]. Cost-sensitive learning takes into account the variable cost of misclassifying different classes [12, 13]. In most cases, cost-sensitive learning algorithms are designed to minimize total costs while introducing multiple costs. In 2000, Turney presented the main types of costs involved in inductive concept learning [14].

Cost-sensitive learning has two major types of costs: misclassification and test costs [11]. Misclassification costs can be viewed as the penalties that result from incorrectly classifying an object within a certain class. Traditional machine learning methods are in large part aimed at minimizing the error rate and are dedicated to uniform error costs. They assume equal misclassification costs and relatively balanced class distributions. Typically, the cost of misclassifying an abnormal incident as a normal incident is much higher than the cost of misclassifying a normal incident as an abnormal incident. Thus, misclassification costs must be minimized rather than misclassification errors. There are two types of misclassification costs: example-dependent costs and class-dependent costs [11].

Test costs typically refer to money, time, computing, or other resources that are expended to obtain data items related to an object [8]. There are numerous types of measurement methods with different test costs; higher test costs are required to obtain data characterized by smaller measurement error. An appropriate measurement should be selected, and the total test cost should be minimized.

Some studies focus on misclassification costs but fail to consider the cost of the test [15]. Others consider test costs but not misclassification costs [16]. However, because Turney first considered both test and misclassification costs, his approach has gradually become one of the foremost trends in the research.

2.2. Cost-Sensitive Feature Selection. In general, classification time increases with the number of features based on the computational complexity of the classification algorithm. However, it is possible to alleviate the curse of dimensionality by reducing the number of features, although this may weaken discriminating power.

A classifier can be understood as a specific function that maps a feature vector onto a class label [17]. An algorithm that can guide the active acquisition of information and balance the costs is often termed a cost-sensitive classifier [18]. The acquisition cost for selected features is an important issue in some applications, and more researchers have taken the feature acquisition cost into account in the feature selection criterion [19]. Ji and Carin introduced many cost-sensitive feature selection criteria, while traditional methods select all the useful features simultaneously by setting the weights on the redundant features to zero [17].

Several works have addressed cost-sensitive feature selection in recent years. For example, Bosin et al. [20] present a cost-sensitive approach feature selection that focuses on the quality of features and the cost of computation. In spite of the complexity, this method is able to increase classifier accuracy and judges the goodness of a subset of features with a particular classification model by defining a feature selection criterion.

Mejía-Lavalle [21] proposes a feature selection heuristic that takes into account a cost-sensitive classification. Unlike most feature selection studies that evaluate only accuracy and processing time, this heuristic evaluates different feature selection-ranking methods over large datasets. In addition, they can separate relevant and irrelevant features by stressing the issue around the boundary.

Wang et al. [16] address the issue of data overfitting by designing three simple and efficient strategies—feature selection, smoothing, and threshold pruning—against the test cost-sensitive decision tree method. Before applying the test cost-sensitive decision tree algorithm, they use a feature selection that considers test costs and misclassification costs to preprocess the dataset.

In study by Lee et al. [22], a spam detection model is proposed, the first to take into account the importance of feature variables and the optimal number of features. The optimal number of selected features is decided using two methods: the use of one parameter optimization during the overall feature selection and parameter optimization in every feature elimination phase.

Chang et al. [23] propose an efficient hierarchical classification scheme and cost-based group-feature selection criterion to improve feature calculation and classification accuracy. This approach adopts computational cost-sensitive group-feature selection criterion with the Sequential Forward Selection (SFS) to obtain the class-specific quasioptimal feature set.

Zhao et al. [24] define a new cost-sensitive feature selection problem on a covering-based rough set model with normal distribution measurement errors. Unlike existing models, it proposes backtracking and heuristic algorithms mainly on the error boundaries with test costs and misclassification costs.

Liu et al. [25] propose a new cost-sensitive learning method for software defect prediction, which is divided into two stages: utilizing cost information in the classification stage as well as the feature selection stage.

However, few studies have focused on the class imbalance problem in view of cost-sensitive feature selection. To the best of our knowledge, no study addresses cost-sensitive feature selection in the security network field because of the significant domain differences and dependences.

3. Cost-Sensitive Feature Selection Model

3.1. Problem Formulation. Here, we present the common notations and an intrusion detection event that was taken from the KDD CUP'99 dataset [26] to illustrate them.

Let the original feature set be $F = \{f_1, f_2, \ldots, f_d\}$, where d is the feature dimension count. The feature selection problem is to find a subset F' such that $F' \subset F$ should maximize some scoring function; simultaneously, F' is an optimal subset that gives the best possible classification accuracy.

Assume that, in the instance space, we have a set of samples $X = \{x_1, x_2, \ldots, x_{ND}\}$, $(i = 1, 2, \ldots, N;\ j = 1, 2, \ldots, D)$, where N is the number of samples. Let x_{ij} denote the jth feature of sample x_i. The labeled instance space, which is also called universal instance space L, is defined as a Cartesian product of the input instance space and the target feature domain; that is, $U = X \times Y$. The training is denoted as $T = (\langle x_1, y_1\rangle, \ldots, \langle x_N, y_M\rangle)$, where T contains M classes, $x_i \in X$, and $y_i \in Y$. The notation $fT : X \to Y$ represents a classifier that was trained using inducer f on the training dataset T.

The cost-sensitive feature selection problem is also called the feature selection with minimal average total cost problem [24]. In this paper, we focus on cost-sensitive feature selection based on both misclassification costs and test costs. Unlike the generic algorithm of feature selection, we use it to achieve accuracy or to reduce measurement error. Another purpose of feature selection in our study is to minimize average cost by considering the trade-off between test costs and misclassification costs [8]. In other words, our optimization objective is to minimize average total cost.

Let MC be the misclassification cost matrix and let TC be the test cost matrix. The average total cost should be the following:

$$\begin{aligned}&\mathrm{AvgCost}\left(F'\right)\\ &\quad = \arg\min_{G\in T}\left\{\sum_{F'\subseteq F}\left(\mathrm{MC}\left(F'\right) + \mathrm{TC}\left(F'\right)\right)\right\}.\end{aligned} \tag{1}$$

3.2. Cost Information. In the real world, there are many types of costs associated with a potential instance, such as the cost of additional tests, the cost associated with expert analysis, and intervention costs [11]. Different applications are usually associated with various costs involving misclassification and test costs [19].

Without loss of generality, let $\mathrm{MC}(y_i, y_j)$ be the cost of predicting an instance of class y_i as class y_j. When addressing imbalance problems, misclassification costs can be categorized into four types: (1) false positive (FP), notation C(+, −), is the cost of misclassifying an instance of a positive class as a negative class; (2) false negative (FN), notation C(−, +), is the cost of the opposite case; (3) the misclassification costs of true positive (TP) are equal to true negative (TN), that is, zero. Typically, it is more important to recognize positive rather than negative instances (C(+, −) > C(−, +)).

The cost-sensitive classification problem can be constructed as a decision theory problem using Bayesian Decision Theory [27, 28]. We assume that the probability $p(y \mid x)$ is defined by a subset of d features in instance x, while the remaining features are irrelevant or redundant. The optimal prediction, for example, x, is the class y_j that minimizes the expected loss [27]:

$$R\left(y_i \mid x\right) = \sum_{y\in Y} \mathrm{C}\left(y_i, y_j\right) p\left(y_j \mid x\right). \tag{2}$$

Although decision tree building does not need to be cost-sensitive to measure feature selection, an algorithm requires the cost-sensitive property to rank or weight features based on their importance [28]. Feature selection could confirm or expand domain knowledge by using such ranking.

Borrowing ideas from credit card and cellular phone fraud detection in related fields, the Lee research group identifies three major cost factors related to intrusion detection: damage cost (DCost), response cost (RCost), and operational cost (OpCost) [29]. The misclassification cost can be identified by the following formula:

$$\mathrm{MC}\left(y_i, y_j\right) = \mathrm{DCost}\left(y_i\right) + \varepsilon \mathrm{RCost}\left(y_j\right), \tag{3}$$

where $\varepsilon \in [0, 1]$ is the function of the progress and effect of the attack. For example, from Table 1 we can see back → ipsweep: misclassification cost (DOS → PROBE) = RCost(PROBE) + DCost(DOS).

The expected misclassification cost for a new example drawn at random from $p(x, y)$ distribution is as follows:

$$\mathrm{AvgMC}\left(y_i \mid x\right) = \sum_{\forall(x,y)\in S} \mathrm{MC}\left(y_i, y_j\right) p\left(y_j \mid x\right). \tag{4}$$

Based on study by Lee et al. [29], which extracts and constructs predictive features from network audit data, this approach divides features into four relative levels based on their computational costs; see Table 2. While the OpCost is the cost of time and computing resources spent on extracting or analyzing features for processing the stream of events [29], we assume that the acquisition cost of features can be associated with each feature level.

We assume that both misclassification and test costs are given in the same cost scale. Therefore, summing together the two costs to obtain the average total cost is feasible.

3.3. Cost-Sensitive Fitness Function. Unlike the traditional feature selection algorithm, whose purpose is to improve classification accuracy or reduce measurement error, this paper attempts to minimize total costs and make trade-offs between costs and classification accuracy. The final objective

Table 1: Cost metrics of intrusion categories.

Main category (by results)	Description	DCost	RCost
U2R	Illegal root access is obtained	DCost = 100	RCost = 60
R2L	Illegal user access is obtained from outside	DCost = 50	RCost = 40
DOS	Denial-of-Service of target is accomplished	DCost = 30	RCost = 15
PROBE	Information about the target is gathered	DCost = 2	RCost = 7
Normal	Normal events	DCost = 0	RCost = 0

Table 2: Operation cost metrics.

Feature level	Lev. 1 features	Lev. 2 features	Lev. 3 features	Lev. 4 features
Relative magnitudes	1	5	10	100

of the feature selection problem is to select a feature subset with minimum size, total average costs, and classification accuracy.

Let feature subset F' have k features. Then, we calculate the average total cost of the dataset AvgCost(F') and reconstruct, test, and train datasets to obtain recognition rate R by the nearest neighbor method for each group of subsets selected. Given the number of features of candidate k, construct the feature fitness function using nearest neighbor for each feature:

$$J = R^{(1+k^n)} - \frac{\lambda \text{AvgCost}\left(F'\right)}{k}, \tag{5}$$

where $n \in [0, 1]$ is a parameter for balancing the number of features and the weight of recognition rate R. λ is a parameter introduced to weight the influence of the cost in the fitness function. Here, we see that the fewer the feature candidates that are selected, the greater the recognition rate R and the fitness function J.

3.4. Chaos Optimization and Genetic Algorithm. Genetic algorithm (GA) is a popular parallelized method because of its powerful quality of global search, which is widely used in feature selection [30]. Weiss et al. [8] proposed a cost-sensitive feature selection using histograms based on genetic search. Chen et al. [30] combined chaos optimization with a GA to choose a subset of available features by eliminating unnecessary features from the classification task. Shen and Gao [31] proposed feature selection based on a chaos search to improve the classification accuracy of the weld detect. According to the rules of this approach, chaotic movement can cover all states in a certain range without repetition; the chaotic optimization algorithm (COA) vastly improved the low search efficiency that characterizes the late evolving period of GA [32, 33].

Most studies introduce a logistics map [31]: $w_{i+1} = \eta w_i(1 - w_i)$, $i = 1, 2, \ldots, t$, where η is a control parameter smaller than 4, w_i is the ith chaotic variable, and t denotes the number of iterations. When $\eta = 4$ and w_i is distributed in the range $[0, 1]$, it can be a deterministic dynamic system that is in a complete state of chaos. However, the numbers of sequence distribution boundaries generated by logistic map are more than can be satisfied with the unknown distribution problem, which requires a high level of evenness of individual distribution to avoid the asymmetry of the initial population in the GA.

By comparing ten one-dimensional chaotic maps in terms of the convergence rate, algorithm speed, and accuracy, Tavazoei and Haeri found that no single map has the best global optimization ability [34]. In this paper, Tent map was applied as it has the maximum convergence rate, and the mathematical expression can be defined by [35]

$$w_{n+1} = \alpha - 1 - \alpha \left| w_n \right|, \quad \alpha \in [1, 2]. \tag{6}$$

When $\alpha = 2$, the result is the well-known Center Tent map, and the expression is shown by

$$w_{k+1} = \begin{cases} 2w_k & w_k \in [0, 0.5] \\ 2\left(1 - w_k\right) & w_k \in (0.5, 1] \end{cases} \quad k = 0, 1, 2, \ldots. \tag{7}$$

With the Bernoulli shift transformation, the mathematical expression can be written as

$$w_{k+1} = \begin{cases} 2w_k & w_k \in [0, 0.5] \\ 2w_k - 1 & w_k \in (0.5, 1] \end{cases} \quad k = 0, 1, 2, \ldots. \tag{8}$$

As the Tent map requires the maximum computational time, which seriously affects the algorithm speed, we improved it by deploying the random equation based on [35]. The chaos expression is shown by

$$w_{k+1} = \begin{cases} 2\left(w_k + 0.1 \times \text{rand}\,(0, 1)\right) & w_k \in [0, 0.5] \\ 2\left(1 - \left(w_k + 0.1 \times \text{rand}\,(0, 1)\right)\right) & w_k \in (0.5, 1] \end{cases} \quad k = 0, 1, 2, \ldots. \tag{9}$$

As a consequence, the Tent map was able to achieve global chaos optimization more efficiently by reaching into the chaotic state at a small cycle point.

3.5. Cost-Sensitive Feature Selection Model Using Chaos Genetic Algorithm. In this section, we propose a cost-sensitive feature selection model that uses a new cost-sensitive fitness function and chaos GA to solve the class imbalance

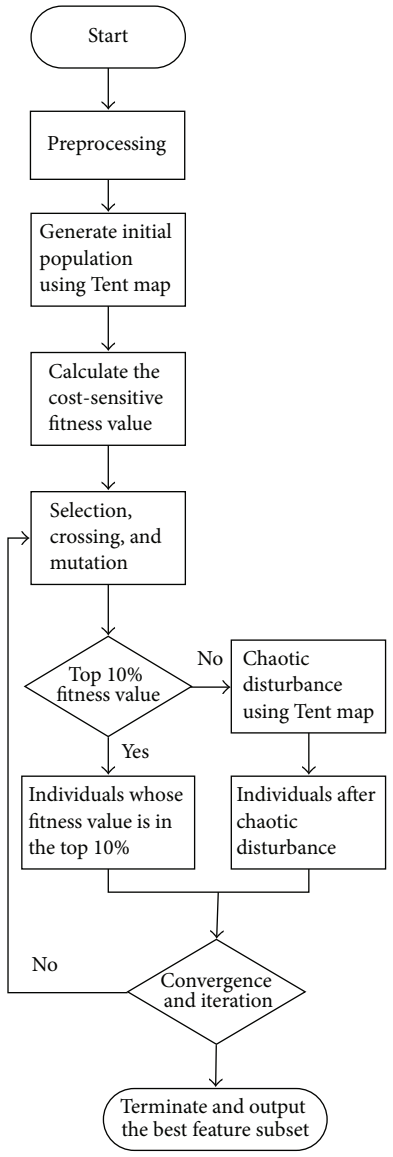

Figure 1: Flowchart for the CSFSG algorithm.

problem. The algorithm follows the filter approach, which is not associated with a particular classifier. Finding a minimal optimal cost feature subset is NP-hard, particularly in those situations with an imbalanced dataset. However, it is important to combine the feature selection procedure with the learning model [8]. Therefore, the proposed algorithm, the CSFSG algorithm, employs a chaos GA as a search method to address this problem.

Our algorithm consists of four main steps (see Figure 1).

Step 1 (preprocessing). Convert the discrete numeric attribute and normalization values to the range $[0, 1]$ in accordance with the cost-sensitive heuristic rule and calculate the misclassification cost matrix and test cost.

Step 2. Generate the initial population using the Tent map and encode. Provide the probability of crossover p_c, probability of mutation p_m, the population size, and number of generations.

Step 3. Calculate the fitness value of each individual and select the optimum population based on the cost-sensitive fitness function.

Step 4. Apply GA to the population, search the candidate feature subset by using the chaos optimize algorithm, and update the current population.

4. Results of the Experimental Investigation

We conducted a series of experiments to compare the overall performance of our approach with some existing algorithms. In this section, we introduce the implementation setting and the evaluation measurements used in our experiments. Then, we describe the relative comparison experiments. Finally, we discuss the results.

4.1. Datasets and Implementation Setting. Our experiments were implemented in the Weka framework with Java, which is available at http://www.cs.waikato.ac.nz/ml/weka/index.html. The datasets are from the public KDD CUP'99 datasets, which have significant imbalances between class 41 features and large-scale instances [36]. Because our research target is imbalanced datasets, we preprocess the data from the dataset by volatility differentiating numerical attributes and normalization values within the range $[0, 1]$.

The raw dataset comprises approximately 4 GB and 5 million instances, which are divided into two classes—labeled and unlabeled—of three types: continuous, discrete, and string. In this study, we use the ten percent version, which consists of 494,021 connections and 24 types of attacks with 5 classes (Normal, DOS, U2R, R2L, and PROBE). The four main types of attack are DOS (Denial of Service Attack, e.g., land attack), U2R (User to Root Attack, e.g., rootkit attack), R2L (Remote to Local Attack, e.g., guess password attack), and probing (information about the target is gathered, e.g., nmap attack). The detailed characteristics of these datasets are shown in Table 3.

Our feature selection, which is a filter approach, does not depend on classifiers. We choose two types of popular classifiers for the classification assignment: k-nearest neighbors and decision trees. Their corresponding implementations are KNN and C4.5. The latter is also used as a base classifier with Laplace smoothing, as it can be transformed into cost-sensitive decision tree classifiers in many related studies.

For the sake of comparison, we designed two group experiments (feature selection and classification), one that uses our cost-sensitive feature selection method and one that does not. In the feature selection stage, the average total cost is applied (the sum of the average test and misclassification costs) to validate the effectiveness of our proposed method. In the classification stage, weighted accuracy is applied, which substitutes accuracy and is more suitable for imbalanced datasets. Moreover, we use tenfold cross-validation to evaluate the performance of the classifier as well as comparing the execution time of each classifier with and without feature selection.

The algorithm we proposed in this paper focused on the cost-sensitive fitness function but not parameter optimization of the GA. Thus the parameter of GA itself is not discussed in this paper. Based on the majority of configurations of the GAs, the parameters were configured to be nearly the same as those in Weiss et al. [8]. Details are as follows: the population size is 20, the number of generations is 20, the probability of crossover is 0.6, the probability of mutation is 0.033, and report frequency is 20.

Bolón-Canedo et al. had studied and evaluated the behavior of the methods under the influence of parameter λ [37]. Increasing λ means that the correlation between features has given more weight to cost; that is to say, the smaller λ the higher the total cost and the lower the error. Also the errors can be obtained by using a Kruskal-Wallis statistical test which can help us to choose the value of the λ parameter [37]. The value of λ is investigated from 0 to 0.5 with step 0.1 by using the total cost for evaluating. Here we chose $\lambda = 0.3$. The common classifiers exhibit bias evaluation and result in majority classes; however, people pay great attention to the classification accuracy of minority classes as well as the whole. The confusion matrix shown in Table 4 is used to represent the contingency table for imbalance problems [8].

We chose precision, recall, and F-measure and ROC area to evaluate the classifiers as follows:

$$\begin{aligned}
\text{Precision}_i &= \frac{\text{C}(y_i, y_i)}{\sum_{k=1}^{M} \text{C}(y_k, y_i)}, \\
\text{Recall}_i &= \frac{\text{C}(y_i, y_i)}{\sum_{k=1}^{M} \text{C}(y_i, y_k)}, \\
F\text{-Measure}_i &= \frac{\left(1 + \beta^2\right) \times \text{Precision}_i \times \text{Recall}_i}{\beta^2 \times \text{Precision}_i + \text{Recall}_i},
\end{aligned} \tag{10}$$

where β (usually $\beta = 1$) is coefficient, used to adjust the relative importance of precision versus the recall [38, 39]. Among them, recall and F-measure are the main criteria aimed at the minority class. F-measure is an effective measurement for an imbalance problem that is a combination of recall and precision. The value of F-measure will be higher only when both the values of recall and precision are higher [39].

4.2. Experimental Design and Results. For the sake of comparison, we designed two stages of experiments. In the first feature selection stage, we compared the effect of our cost-sensitive feature selection method with traditional ones on an imbalanced dataset. In the second classification stage, we compared the precision of two types of classifiers (KNN, C4.5) using CSFSG via 10-fold cross-validation in the imbalanced environment.

Because there were multiple datasets with deficient class numbers and evaluation metrics, we designed many combinations of comparison experiments for our proposed feature selection method using two feature selection algorithms: correlation-based feature selection (CFS) [40] and CASH [8]. The CFS used here was not sensitive and was a multivariate CFS that yields better results than the wrapper CFS for small datasets, while CASH is sensitive to both test and misclassification costs using histograms. Also, the CASH has been proved to be superior to several other cost-sensitive algorithms [8].

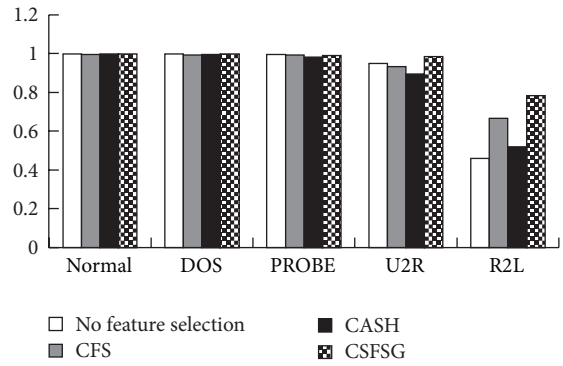

Figure 2: Precision values of KNN with feature selection.

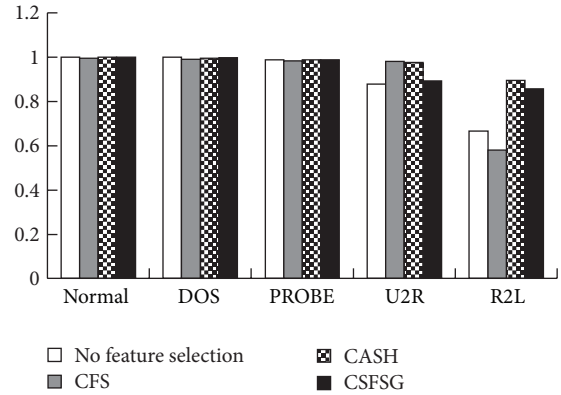

Figure 3: Precision values of C4.5 with feature selection.

Table 5 presents the dataset, which used cost-sensitive feature selection methods with more features but did not take the most time to build the classification models.

As seen in Figures 2 and 3, cost-sensitive feature selection methods exhibited high performance with regard to the evaluation measures, but our method was superior to the CASH, especially with respect to classifying the minority class (the class of R2L and U2R). CSFSG helped to reduce the number of features used to distinguish some types of attack and increase the efficiency of the classifiers. The feature selection stage is important and effective because it can save considerable time with an increasing number of instances, particularly for large-scale imbalance problems.

In the classification stage, 10-fold cross-validation is applied on the datasets. The results of classification with feature selection are shown in Table 6. Here we use F-measure, recall, and ROC area to compare the classification model. In the table, we could get a general idea that feature selection methods help to find out the anomaly attacks from the known attacks with classification model, despite the few outliers.

Although the result in the first two columns (normal and DOS) does not show obvious differences, the result derived from the cost-sensitive feature selection in the last three columns presents higher values of F-measure, recall,

Table 3: Class distribution in the raw and experimental datasets of the KDD Cup'99 dataset.

Type	DOS	U2R	R2L	PROBE	Normal
Number	391458	52	1126	4107	97278
Percentage	79.23%	0.01%	0.23%	0.83%	19.69%

Table 4: Confusion matrix.

		Predicted class				
		y_1	$\cdots$	y_j	$\cdots$	y_M
Actual class	y_1	$C(y_1, y_1)$	$\cdots$	$C(y_1, y_j)$	$\cdots$	$C(y_1, y_M)$
	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
	y_i	$C(y_i, y_1)$	$\cdots$	$C(y_i, y_j)$	$\cdots$	$C(y_i, y_M)$
	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
	y_M	$C(y_M, y_1)$	$\cdots$	$C(y_M, y_j)$	$\cdots$	$C(y_M, y_M)$

Table 5: Comparing the feature number and execution time (in seconds).

Feature selection algorithms	Feature number	Execution time (with KNN)	Execution time (with C4.5)
—	41	0.16	17.12
CFS	11	0.04	2.78
CASH	23	0.21	33.14
CSFSG	17	0.05	7.19

Table 6: Classification evaluation results for the *K*NN and C4.5 algorithm with feature selection (NON, CFS, CASH).

Algorithm				*F*-measure		
Feature selection	Classifier	Normal	DOS	PROBE	R2L	U2R
—	*K*NN	0.996	0.998	0.990	0.936	0.876
CFS	*K*NN	0.997	0.998	0.990	0.938	0.899
CASH	*K*NN	**0.999**	**0.998**	**0.990**	0.936	0.866
CSFSG	*K*NN	0.997	0.997	0.989	**0.942**	**0.916**
—	C4.5	0.997	0.998	0.988	0.963	0.966
CFS	C4.5	0.997	0.998	0.988	0.967	0.966
CASH	C4.5	0.997	0.989	**0.969**	0.942	0.963
CSFSG	C4.5	**0.997**	**0.998**	0.953	**0.968**	**0.966**
Algorithm				Recall		
Feature selection	Classifier	Normal	DOS	PROBE	R2L	U2R
—	*K*NN	0.996	0.998	0.986	0.779	0.6
CFS	*K*NN	0.997	0.998	0.78	0.724	0.8
CASH	*K*NN	**0.999**	0.998	**0.971**	0.75	0.4
CSFSG	*K*NN	0.997	**0.998**	0.964	**0.794**	**0.6**
—	C4.5	0.997	0.998	0.971	0.735	0.4
CFS	C4.5	0.997	0.998	0.788	0.75	0.2
CASH	C4.5	0.997	0.989	0.967	**0.75**	0.4
CSFSG	C4.5	**0.997**	**0.998**	**0.969**	0.738	**0.6**
Algorithm				ROC area		
Feature selection	Classifier	Normal	DOS	PROBE	R2L	U2R
—	*K*NN	0.996	0.998	0.990	0.983	0.899
CFS	*K*NN	0.999	0.998	0.996	0.977	0.99
CASH	*K*NN	**0.999**	0.998	0.988	0.938	**0.899**
CSFSG	*K*NN	0.998	**0.999**	**0.993**	**0.947**	0.988
—	C4.5	0.999	0.998	0.992	0.937	0.579
CFS	C4.5	0.997	0.998	0.963	0.886	0.733
CASH	C4.5	0.997	0.989	**0.998**	0.887	0.721
CSFSG	C4.5	**0.997**	**0.998**	0.996	**0.935**	**0.733**

and ROC area. Comparing with other cost-sensitive feature selection algorithms, our proposed method has better performance, especially when applied to minority class.

In addition, we can obtain better classification performance with CSFSG than with CASH in the time given, which is one of the most important factors in network security.

In conclusion, our cost-sensitive feature selection method does not save a considerable amount of time; however, it does facilitate the detection of the minority and CSFSG class, particularly under a class imbalance environment.

5. Conclusions

In response to the rapid growth of big data, this study presents a novel cost-sensitive feature selection method using a chaotic genetic search for imbalanced datasets. We introduce cost-sensitive learning into the feature selection method, considering both the misclassification cost and test cost with respect to the field of network security.

It can be seen from the experimental results that cost-sensitive feature selection using chaotic genetic search efficiently reduces complexity in the feature selection stage. Meanwhile, it can improve classification accuracy and decrease classification time.

Several future works will address problems with large numbers of features. Furthermore, future research will focus on the application of the proposed method to other fields, such as medicine or biology.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This study is supported by the project supported by the National Science Foundation for Young Scientists of China (Grant no. 61401300), the Outstanding Graduate Student Innovation Projects of Shanxi Province (no. 20123030), and the Scientific Research Project of Shanxi Provincial Health Department (no. 201301006).

References

[1] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[2] A. Anand, G. Pugalenthi, G. B. Fogel, and P. N. Suganthan, "An approach for classification of highly imbalanced data using weighting and undersampling," *Amino Acids*, vol. 39, no. 5, pp. 1385–1391, 2010.

[3] L. Abdi and S. Hashemi, "To combat multi-class imbalanced problems by means of over-sampling techniques," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 1, pp. 238–251, 2016.

[4] J. F. Díez-Pastor, J. J. Rodríguez, C. I. García-Osorio, and L. I. Kuncheva, "Diversity techniques improve the performance of the best imbalance learning ensembles," *Information Sciences*, vol. 325, pp. 98–117, 2015.

[5] P. Thanathamathee and C. Lursinsap, "Handling imbalanced data sets with synthetic boundary data generation using bootstrap re-sampling and AdaBoost techniques," *Pattern Recognition Letters*, vol. 34, no. 12, pp. 1339–1347, 2013.

[6] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[7] R. Blagus and L. Lusa, "SMOTE for high-dimensional class-imbalanced data," *BMC Bioinformatics*, vol. 14, article 106, 2013.

[8] Y. Weiss, Y. Elovici, and L. Rokach, "The CASH algorithm-cost-sensitive attribute selection using histograms," *Information Sciences*, vol. 222, pp. 247–268, 2013.

[9] N. Thai-Nghe, Z. Gantner, and L. Schmidt-Thieme, "Cost-sensitive learning methods for imbalanced data," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '10)*, pp. 1–8, IEEE, Barcelona, Spain, July 2010.

[10] L. Yin, Y. Ge, K. Xiao, X. Wang, and X. Quan, "Feature selection for high-dimensional imbalanced data," *Neurocomputing*, vol. 105, pp. 3–11, 2013.

[11] R. H. Thomas, *Living in an imbalanced world [Ph.D. thesis]*, Department of Computer Science and Engineering, University of Notre Dame for the Degree of Doctor of Philosophy, Notre Dame, Ind, USA, 2012.

[12] P. Domingos, "Metacost: a general method for making classifiers cost-sensitive," in *Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining (KDD '99)*, pp. 155–164, San Diego, Calif, USA, August 1999.

[13] H. Lu, E. Zheng, Y. Lu, X. Ma, and J. Liu, "ELM-based gene expression classification with misclassification cost," *Neural Computing and Applications*, vol. 25, no. 3-4, pp. 525–531, 2014.

[14] P. Turney, "Types of cost in inductive concept learning," in *Proceedings of the Workshop on Cost-Sensitive Learning at the Seventeenth International Conference on Machine Learning*, pp. 15–21, 2000.

[15] K. Iswandy and A. Koenig, "Feature selection with acquisition cost for optimizing sensor system design," *Advances in Radio Science*, vol. 4, pp. 135–141, 2006.

[16] T. Wang, Z. Qin, Z. Jin, and S. Zhang, "Handling over-fitting in test cost-sensitive decision tree learning by feature selection, smoothing and pruning," *Journal of Systems and Software*, vol. 83, no. 7, pp. 1137–1147, 2010.

[17] S. Ji and L. Carin, "Cost-sensitive feature acquisition and classification," *Pattern Recognition*, vol. 40, no. 5, pp. 1474–1485, 2007.

[18] R. Greiner, A. Grove, and D. Roth, "Learning active classifiers," in *Proceedings of the 13th International Conference on Machine Learning*, Bari, Italy, July 1996.

[19] P. D. Turney, "Cost-sensitive classification: empirical evaluation of a hybrid genetic decision tree induction algorithm," *Journal of Artificial Intelligence Research*, vol. 2, pp. 369–409, 1995.

[20] A. Bosin, N. Dessi, and B. Pes, "A cost-sensitive approach to feature selection in micro-array data classification," in *Proceedings of the 7th International Workshop on Fuzzy Logic and Applications*, pp. 7–10, Camogli, Italy, July 2007.

[21] M. Mejía-Lavalle, "Applying cost sensitive feature selection in an electric database," in *Foundations of Intelligent Systems*, A. An, S. Matwin, Z. W. Raś, and D. Ślęzak, Eds., vol. 4994 of *Lecture Notes in Computer Science*, pp. 644–649, Springer, New York, NY, USA, 2008.

[22] S. M. Lee, D. S. Kim, and J. S. Park, "Cost-sensitive spam detection using parameters optimization and feature selection," *Journal of Universal Computer Science*, vol. 17, no. 6, pp. 944–960, 2011.

[23] Y. Chang, N. Kim, Y. Lee, J. Lim, J. B. Seo, and Y. K. Lee, "Fast and efficient lung disease classification using hierarchical one-against-all support vector machine and cost-sensitive feature selection," *Computers in Biology and Medicine*, vol. 42, pp. 1157–1164, 2012.

[24] H. Zhao, F. Min, and W. Zhu, "Cost-sensitive feature selection of numeric data with measurement errors," *Journal of Applied Mathematics*, vol. 2013, Article ID 754698, 13 pages, 2013.

[25] M. Liu, L. Miao, and D. Zhang, "Two-stage cost-sensitive learning for software defect prediction," *IEEE Transactions on Reliability*, vol. 63, no. 2, pp. 676–686, 2014.

[26] http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[27] C. P. Adam, *Feature selection via joint likelihood [Ph.D. thesis]*, Faculty of Engineering and Physical Sciences, Manchester University, 2012.

[28] M. Robnik-Šikonja, "Experiments with cost-sensitive feature evaluation," in *Machine Learning: ECML 2003*, vol. 2837 of *Lecture Notes in Computer Science*, pp. 325–336, Springer, Berlin, Germany, 2003.

[29] W. Lee, W. Fan, M. Miller, S. J. Stolfo, and E. Zadok, "Toward cost-sensitive modeling for intrusion detection and response," *Journal of Computer Security*, vol. 10, no. 1-2, pp. 5–22, 2002.

[30] H. Chen, W. Jiang, C. Li, and R. Li, "A heuristic feature selection approach for text categorization by using chaos optimization and genetic algorithm," *Mathematical Problems in Engineering*, vol. 2013, Article ID 524017, 6 pages, 2013.

[31] Q. Shen and J. Gao, "Improving the classification accuracy of the weld defect by chaos-search-based feature selection," *Insight: Non-Destructive Testing and Condition Monitoring*, vol. 52, no. 10, pp. 530–539, 2010.

[32] M. E. ElAlami, "A filter model for feature subset selection based on genetic algorithm," *Knowledge-Based Systems*, vol. 22, no. 5, pp. 356–362, 2009.

[33] H. Gu, Z. Wu, X. Huang, and J. Song, "Zoning modulus inversion method for concrete dams based on chaos genetic optimization algorithm," *Mathematical Problems in Engineering*, vol. 2015, Article ID 817241, 9 pages, 2015.

[34] M. S. Tavazoei and M. Haeri, "Comparison of different one-dimensional maps as chaotic search pattern in chaos optimization algorithms," *Applied Mathematics and Computation*, vol. 187, no. 2, pp. 1076–1085, 2007.

[35] X. Fu, X. Chen, Q. Hou, Z. Wang, and Y. Yin, "An improved chaos genetic algorithm for T-shaped MIMO radar antenna array optimization," *International Journal of Antennas and Propagation*, vol. 2014, Article ID 631820, 6 pages, 2014.

[36] M. DeGroot and M. Schervish, *Probability and Statistics*, Addison-Wesley, New York, NY, USA, 3rd edition, 2001.

[37] V. Bolón-Canedo, I. Porto-Díaz, N. Sánchez-Maroño, and A. Alonso-Betanzos, "A framework for cost-based feature selection," *Pattern Recognition*, vol. 47, no. 7, pp. 2481–2489, 2014.

[38] X. Deng, Q. Liu, Y. Deng, and S. Mahadevan, "An improved method to construct basic probability assignment based on the confusion matrix for classification problem," *Information Sciences*, vol. 340-341, pp. 250–261, 2016.

[39] J. Yang, Z. Qu, and Z. Liu, "Improved feature-selection method considering the imbalance problem in text categorization," *The Scientific World Journal*, vol. 2014, Article ID 625342, 17 pages, 2014.

[40] M. A. Hall, *Correlation-based feature selection for machine learning [Ph.D. thesis]*, Department of Computer Science, The University of Waikato, Hamilton, New Zealand, 1999.