A Fraudulent Transactions Simulation Method Based on Genetic Algorithm

To cite this article: Xiaoguo Wang et al 2019 J. Phys.: Conf. Ser. 1302 022090

View the article online for updates and enhancements.
A Fraudulent Transactions Simulation Method Based on Genetic Algorithm

Xiaoguo Wang, Yuanxiu Li and Ran Zhao

College of Electronics and Information Engineering, Tongji University, Shanghai, China
Email: liyuanxiu@tongji.edu.cn

Abstract. When using machine learning or other methods to construct the fraud detection models, the banking industry faces such problems: the number of fraud transactions data is too small, which affect the training of anti-fraud model and the detection effect of fraud transaction. This paper proposed a data simulation algorithm based on genetic algorithm (GA-DS). By studying the feature of real fraudulent transactions, we designed the crossover mutation and other genetic operators, explored the suitable fitness function that can evaluate the quality of simulated data, and generated simulated data satisfying the characteristics of the original transaction. The experiment result shows that mixing the simulated data and the original data into the training can improve the detection ability of anti-fraud model.

1. Introduction

With the rapid development of Internet technology, online financial service platforms such as third-party payment have risen rapidly. While bringing convenience to people's life, fraud risk comes along with it. When establishing anti-fraud systems, Banks are faced with such problems: the number of fraudulent transactions is significantly less than normal transactions, which leads to insufficient analysis results of the model and affects the accuracy of fraud sample identification. How to solve the problem of data imbalance has become an urgent need in modelling. Constructing a simulated data model is an effective method to solve this problem. In this paper, we will study the simulation method of fraudulent data in third-party transactions by combining the genetic algorithm with the data similarity theory.

2. Design of Simulated Data Generation

In the transaction records of third-party payment, the category distribution of datasets is extremely uneven, the scale of normal transactions is huge, and fraudulent transactions are seriously insufficient. As a result, it is difficult for the anti-fraud model to extract meaningful information from small sample during the modelling process, and the results are more likely to be biased towards normal transactions during fraud detection. Effective simulated data generation method will improve this problem. The generated simulated data should meet the requirements of consistency and diversity. Consistency means that each attribute of simulated data should fit the original data as much as possible. Diversity refers to the richness of simulated data, which can avoid the over-fitting problem caused by insufficient generalization of the model. According to these principles, this paper presents a data simulation algorithm based on genetic algorithm(GA-DS), and verify the effect by anti-fraud model.
3. Data simulation method based on genetic algorithm (GA-DS)

3.1. Algorithm Framework
Genetic algorithm is a highly parallel combinatorial optimization algorithm with stochastic search. Based on GA, this paper solves the problem of unbalanced third-party fraudulent transactions. GA-DS will encode the bank transaction data, use the adaptive crossover mutation operator to generate new transaction data, and according to fitness, select the optimal simulated data from potential population function generation after generation. The algorithm flow is shown in figure 1. GA-DS algorithm has the following advantages in data simulation: Genetic algorithm inherits the characteristics of data and can inherit the prior knowledge and rules hidden in the real fraudulent data to the simulated data set. Using fitness function to guide the generation process can ensure the diversity of data and the final simulated data set and the original data set have the same attributes, similar interval distribution of attribute values, consistent attribute correlation and so on.

![Algorithm Flow Chart](image)

3.2. Initial Population
Transaction records contains event code, order submission date, order submission time, amount, and other fields, it includes some irrelevant and redundant characteristics that should be screened. Based on the principle and technology of feature selection, we select coding fields from several aspects, such as the dimension of RFM, the associated attribute of fraud rules and the prior knowledge of experts, including transaction time, transaction amount, transaction account and transaction merchant.

Once fields are identified, they need to be effectively encoded. Coding method [1] not only affects the operation mode of genetic operators such as crossover and mutation, but also determines the efficiency of genetic evolution. There are two kinds of encoding methods commonly used in genetic algorithm, one is binary encoding, the other is real-number encoding. By analysis the characteristics of transaction data, we found that real-number encoding has the following advantages:

(1) Real-number coding can directly reflect the inherent structural characteristics and essence of the problem to be solved, and can directly carry out genetic operation on the phenotype of the solution.
(2) There is no mapping error when continuous function is discretized in real number coding, which is applicable to bank transaction data with wide range of number field and high numerical accuracy, improving the computational complexity of genetic algorithm and operational efficiency.

(3) Real-number coding is easy to be mixed with other optimization methods such as simulated annealing, which can improve the convergence speed of the algorithm.

Therefore, GA-DS uses real-number coding to establish a direct mapping between the actual representation of the target problem and the chromosome structure of the genetic algorithm. Set the size of initial population $C$ to $m$, $C = (X_1, X_2, X_3, ... X_m)$. After field filtering, the number of variables in any chromosome $X_i$ is $n$, $X_i = (x_1, x_2, x_3, ..., x_n)$. Fill and complete the transaction amount and other fields, so as to maintain the same attribute length and facilitate subsequent crossover and mutation operations.

3.3. Crossover Operator

Crossover [2] refers to the operation of replacing and recombining the partial structure of two parent individuals to generate new individuals under a certain probability. Gene recombination can produce genes that fluctuate more than the parent generation, which mainly controls the speed of new individuals in the population and has a very important influence on the process of genetic algorithm. Commonly used crossover operators are:

(1) Arithmetic Crossover. The arithmetic crossover produces new individuals by a linear combination of two parents. The operation process is as follows: with a certain probability of $P_c$, by means of random multi-point crossover, the corresponding gene positions of two parent chromosomes are selected and crossed according to the formula to form two new offspring. $X_i$ and $Y_i$ were used to represent the $i$th locus of any two parent chromosomes, $X'_i$ and $Y'_i$ were used to represent the $i$th locus of progeny chromosomes, the formula is:

$$X'_i = aX_i + (1 - a)Y_i$$  \hfill (1)

$$Y'_i = aY_i + (1 - a)X_i$$  \hfill (2)

Among them, the recombination parameter $a$ controls the degree of crossover. When the value is constant, it is uniform crossover; when the value is random number between $[0, 1]$, it is non-uniform crossover.

(2) Partially Mapped Crossover. Partial mapping crossover was proposed by Goldberg, the operation process was as follows: with a certain probability of $P_c$, two segmentation points $c$ and $d$ were randomly selected in the chromosome, mapping fragments were generated between these two segmentation points, and the mapping fragments between the two parents were exchanged to generate new offspring.

Arithmetic crossover and partially mapped crossover and other traditional crossover operations can well realize the parent restructuring and generate new individuals, but the randomness of their recombination makes the probability of producing an individual child superior to its parent uncertain. In addition, the crossover probability of the traditional crossover operator is fixed or varies linearly with the fitness value, these factors will cause the phenomenon of genetic algorithm premature convergence and slow convergence speed. To solve this problem, we combine the characteristics of different fields of bank transaction data, consider the crossover probability, crossover position and crossover mode. Based on the adaptive crossover operator [3], the improvement and application are carried out to ensure that the crossover probability, position and mode of individuals can be dynamically adjusted along with the evolutionary generations, the degree of superiority and inferiority of the current evolution results and the diversity of individuals within the current population, so as to ensure the comprehensiveness and accuracy of the algorithm search. At the beginning of evolution, large crossover probability was selected for rough search to maintain the diversity of the population. In the later stage of evolution, the crossover probability is continuously reduced and meticulous search is conducted to approach the optimal solution step by step. In addition, for the two parents to be crossed, we need to set selection conditions: if any chromosome is selected, another parent chromosome is
selected based on the nearest principle. This method can reduce the problem of generating more sample noise due to random selection of parents and improve the effectiveness of synthetic samples.

Set the crossover probability $P_c'$ as follows:

$$P_c' = (P_{c,\text{max}} - P_{c,\text{min}}) \left( e^{\frac{\text{stable}-j}{\text{generation}}} + e^{\frac{F}{\text{fitness}(C_i)}} \right)^{-1} + P_{c,\text{min}}$$

(3)

$P_{c,\text{max}}$ is the maximum crossover probability, $P_{c,\text{min}}$ is the minimum crossover probability, $j$ represents the current evolutionary generation, stable is the number of evolutionary generations in which fitness values remain constant, Genetation is the threshold of evolutionary generations, fitness($C_i$) is the current fitness value, $F$ is the threshold of the fitness value, they collectively affect the value of the current crossover probability $P_c'$. In general, in the early evolution, the optimum population is difficult to remain stable, fitness value and the threshold value is large, so the values of stable * $j$ and $F/f$ are smaller, the crossover probability $P_c'$ is big, ensure the global search ability of the algorithm. When entering the later stage of evolution, the optimal population gradually stabilized and the fitness value gradually approached the threshold value, the crossover probability $P_c'$ is small. In addition, fitness($C_i$) contains the function that can judge the diversity of the individuals in the population, the larger the individual coincidence rate is, the smaller the value of $F/f$ is, and the higher the crossover probability is.

For the numerical and structural characteristics of different fields in chromosome, we use different crossover methods. For the attributes with strong restriction such as transaction time, we adopt the adaptive partial mapped crossover method and set the positions of the slices for crossover can be changed dynamically with the effect of evolution and the number of evolutionary generations. For attributes such as transaction amount, we adopt the adaptive non-uniform crossover method and set the random value range of the recombination parameter can be changed dynamically with the effect of evolution and the number of evolutionary generations.

3.4. Mutation Operator
Mutation [4] refers to the operation of generating new individuals by changing a certain site of the parent chromosome under a certain probability. The mutation probability mainly controls gene perturbation, and the mutation step controls the fine tuning of the offspring in the search space. Mutation operators maintain the diversity of solutions to a certain extent. With the increase of evolutionary algebra, they can accelerate the convergence of results to the optimal solution, which plays a very important role in genetic algorithms. The success ratio of the traditional uniform mutation operator decreases monotonically with the increase of the evolutionary generations. When the population tends to converge, its efficiency is very low. Therefore, the mutation probability, position and mode of individuals should be dynamically adjusted along with the evolutionary generations, the degree of superiority and inferiority of the current evolution results and the diversity of individuals within the current population. In this paper, we combine the characteristics of different fields of bank transaction data, based on the adaptive mutation operator [3], the improvement and application are carried out to ensure that the algorithm can automatically change the search scope according to the advantages and disadvantages of the current results. When the quality of the results is good, the scope is reduced, and when the quality of the results is bad, the scope is expanded, which further strengthens the local search ability of the algorithm.

Set the mutation probability $P_m'$ as follows:

$$P_m' = (P_{m,\text{max}} - P_{m,\text{min}}) \left( e^{-\frac{\text{stable}-j}{\text{generation}}} + e^{\frac{F}{\text{fitness}(C_i)}} \right)^{-1} + P_{m,\text{min}}$$

(4)

$P_{m,\text{max}}$ is the maximum mutation probability, $P_{m,\text{min}}$ is the minimum mutation probability. The other factors have the same meaning as the crossover probability formula, they collectively affect the value of the current mutation probability $P_m'$. When falling into the local optimum, the probability of mutation increases, and new individual genes of the population are added to jump out of the local optimum.

Set the mutation step function [5] as:
Set the field of the mutated offspring $X_k$ is $x_i'$:

$$x_i' = \begin{cases} 
    x_i + h(t, x_{i,\text{max}} - x_i) & \text{if } a = 0 \\
    x_i - h(t, x_i - x_{i,\text{min}}) & \text{if } a = 1 
\end{cases}$$

where $\{x_{i,\text{min}}, x_{i,\text{max}}\}$ is the value range of this field in the real transaction data, $\alpha$ is the random number of value 0 or 1, $r$ is the random number within the interval of $[0,1]$, $F$ is the threshold of the fitness value, $\text{fitness}(C_i)$ is the current fitness value, $p$ is a parameter, they collectively affect the value of the mutation step. In the early stage of evolution, the fitness value of the population is greatly different from the threshold value, and the variation step value is large, which can accelerate the process of evolution. When the fitness value approximates the threshold value gradually, the variation step decreases gradually and promotes the evolutionary convergence.

3.5. Selection

Crossover and mutation are the fundamental reasons why offspring are different from their fathers, while selection makes offspring have the tendency to be superior to their fathers. We divide the community that extended by crossover and mutation into several subgroups which are the same size as the original population to form the offspring population, calculate the fitness value of each offspring population, select the optimal population and keep it. Set $C_{i,\text{best}}$ as the optimal subpopulation, $C_{\text{best}}$ as the historical optimal population, $\text{fitness}(C_i)$ is the fitness value of population $C_i$. Then the parent population of the next generation is:

$$C_{\text{father}} = \begin{cases} 
    C_{i,\text{best}} & \text{if } \text{fitness}(C_{i,\text{best}}) > \text{fitness}(C_{\text{best}}) \\
    C_{\text{best}} & \text{if } \text{fitness}(C_{i,\text{best}}) < \text{fitness}(C_{\text{best}}) 
\end{cases}$$

Compare the contemporary optimal population with the historical optimal population. If the contemporary population is better, it will be regarded as a new parent population and the historical optimal population will be updated. This operation can control the evolution direction and ensure that the offspring is definitely better than the father.

3.6. Fitness Function

Fitness function can evaluate the ability of individuals to adapt to the environment, directly determines the goal and direction of optimization, and fundamentally affects the performance of genetic algorithm. Under the background of the practical application of bank transaction and based on the previous data similarity theory, this paper attempts to propose an appropriate fitness function to reflect the quality of the population, so that the population with low fitness is eliminated and the population with high fitness is retained, and finally generates the simulated data that matches the characteristics of the original data.

Take the original population that consisted by actual fraudulent transactions as the reference object. Compare the similarity between the current population $C_i$ and the original population $C_{\text{ori}}$ from the aspects of transaction behavior pattern, data distribution, correlation between attributes, and individual diversity within the population, so as to obtain the fitness value of the population. Set the original population:

$$C_{\text{ori}} = (X_1, X_2, ..., X_{i,\text{max}}, ..., X_m)$$

any one of these chromosomes $X_k = (x_{k,1}, x_{k,2}, ..., x_{k,i}, ..., x_{k,n})$. Set the current population: $C_i = (Y_1, Y_2, ..., Y_{i,\text{max}}, ..., Y_m)$, any one of these chromosomes $Y_k = (y_{k,1}, y_{k,2}, ..., y_{k,i}, ..., y_{k,n})$.

1) Similarity function based on transaction behaviour mode

For an account with fraudulent transactions, we calculated the historical normal behaviour pattern $M_{\text{trans}}$ for that account, including time-based derived variables and merchant based derived variables. Among them, the time-based derivative variables include the average transaction amount and the maximum transaction amount under a certain time window. The derivative variable based on the merchant is the maximum transaction amount of the account at a merchant, the average amount, etc.
Suppose the real fraud transaction of this account is $X_{\text{real\_fraud}}$, and the simulated fraud transaction is $Y_{\text{simu\_fraud}}$, the difference between the actual fraudulent transaction and the historical behavior pattern of the account is $d(X_{\text{real\_fraud}}, M_{\text{trans}})$, the difference between the simulated fraudulent transaction and the historical behavior pattern of the account is $d(Y_{\text{simu\_fraud}}, M_{\text{trans}})$. $f_{\text{mode}}$ is used to represent the similarity of historical transaction behavior mode differences between $C_i$ and $C_{\text{ori}}$. The smaller the value of $f_{\text{mode}}$, the more similar $C_i$ and $C_{\text{ori}}$ are.

$$f_{\text{mode}}(C_i, C_{\text{ori}}) = \sum_{k,j} |d(X_{k,\text{simu\_fraud}}, M_{\text{trans}}) - d(Y_{j,\text{simu\_fraud}}, M_{\text{trans}})|$$

(2) Similarity function based on information entropy

Using information entropy [6] $\text{ent}(x_i)$ to measure the average distribution of all individuals in a population on a certain attribute $x_{k,i}$. $f_{\text{ent}}$ is used to calculate the difference of information entropy between $C_i$ and $C_{\text{ori}}$ for each attribute. The smaller the value of $f_{\text{ent}}$, the more similar $C_i$ and $C_{\text{ori}}$ are.

$$\text{ent}(x_i) = -\sum k x_{k,i} \ln x_{k,i}$$

$$f_{\text{ent}}(C_i, C_{\text{ori}}) = \sum_l |\text{ent}(x_i) - \text{ent}(y_j)|$$

(3) Similarity function based on mutual information

Each attribute of the transaction data contains some correlation relations, such as the transaction time and amount of a user, and the simulation data should inherit these implicit relationships as much as possible. Using mutual information [7] $\text{mutual\_info}(x_i)$ measures the correlation between two attributes $x_i, x_j$ in a population, $f_{\text{mutual\_info}}$ is used to calculate the similarity of attribute association between $C_i$ and $C_{\text{ori}}$. The smaller the value of $f_{\text{mutual\_info}}$, the more similar $C_i$ and $C_{\text{ori}}$ are.

$$\text{mutual\_info}(x_i, x_j) = \text{ent}(x_i) + \text{ent}(x_j) - \text{ent}(x_i, x_j)$$

$$f_{\text{mutual\_info}}(C_i, C_{\text{ori}}) = \sum_{l,j} |\text{mutual\_info}(x_i, x_j) - \text{mutual\_info}(y_i, y_j)|$$

(4) Similarity function based on Kullback-Leibler divergence

KL divergence [8] can measure the difference in the distribution of the same attribute $x_i, y_i$ between two populations. Using $f_{\text{kl}}$ calculate the similarity of the global distribution between $C_i$ and $C_{\text{ori}}$. The smaller the value of $f_{\text{mutual\_info}}$, the more similar $C_i$ and $C_{\text{ori}}$ are.

$$kl(x_i, y_i) = \sum k x_{k,i} \ln \frac{x_{k,i}}{y_{k,i}}$$

$$f_{\text{kl}}(C_i, C_{\text{ori}}) = \sum_l |kl(x_i, y_i)|$$

(5) A function that measures the diversity of individuals

The distance between individuals in the population is the key to maintain diversity. Using $f_{\text{diversity}}$ evaluate the degree of overlap between individuals in the population. The smaller the value of $f_{\text{diversity}}$, the lower the coincidence rate of $C_i$.

In this paper, we define the fitness $\text{fitness}(C_i)$ based on the above five indicators.

$$\text{fitness}(C_i) = f_{\text{mode}} + f_{\text{ent}} + f_{\text{mutual\_info}} + f_{\text{kl}} + f_{\text{diversity}}$$

In conclusion, the smaller the fitness value is, the better the simulated data quality of the population will be.

3.7. Design of Convergence

Due to the random initialization of genetic algorithm, the searching space of individual population is uncertain. At the same time, crossover and mutation have great randomness, it is blind to determine the probability of the next generation only according to the efficiency of contemporary operators. Considering the convergence of the algorithm, this paper establishes a constraint mechanism from the crossover mutation and selection operator, the crossover and mutation probability can be changed adaptively with evolutionary generations and evolutionary effect, by keeping the historical optimal
population, the offspring must be better than the parent generation, and the population gradually gets closer to the global optimum.

4. Experiment and Verification

4.1. Experiment

This paper takes the real fraudulent transaction data of Banks as the original data and input it into model. The model parameters are shown in table 1, the termination conditions of evolution were considered from the aspects of fitness threshold and maximum evolutionary generations. When the maximum evolutionary generation is 200, 500 and 1000, the change curve of fitness value with the evolutionary generation is shown in figure 2, which shows that the model can converge gradually.

| Table 1. Configuration of Model Parameters |
|-------------------------------------------|
| Parameter      | Value |
|----------------|-------|
| P_{c,\text{max}} | 0.6   |
| P_{c,\text{min}} | 0.1   |
| P_{m,\text{max}} | 0.2   |
| P_{m,\text{min}} | 0.001 |
| p              | 3     |
| Generation     | 200,500,1000 |
| F              | 0.2   |

Figure 2. Change Curve of Fitness Value with the Evolutionary Generation

4.2. Verification

The goal of this paper is to improve the problem of unbalanced data in training of the fraud detection model. In order to verify the validity of the GA - DS algorithm, we will train the fraud detection model with the original data and mixed data Respectively, where, mixed data is mixed by original data and simulated data. Then classify and detect the same batch of transaction data, and the effect of the simulated data is verified by comparing the evaluation indicators.

4.2.1. Evaluation indicator. The proportion of the bank’s normal transactions reached 99.9%, far exceeding the fraudulent transactions. Even if all the data were judged as normal transactions, the accuracy was as high as 99.9%. However, once minority class samples are misclassified, the cost will be much higher than that of majority class [9]. Therefore, the banking industry tends to pay more attention to the classification results of fraudulent transactions, so there is little sense in measuring the effect of the model only by Overall Accuracy. Therefore, based on the confusion matrix in table 2, this paper selects indicators such as fraud sample identification accuracy rate, non-fraud sample identification accuracy rate, g-mean, F1 Score to evaluate the performance of the model.
Table 2. Confusion Matrix

|                              | judge as fraud | judge as normal |
|------------------------------|----------------|-----------------|
| the true value is fraud      | TP             | FN              |
| the true value is normal     | FP             | TN              |

4.2.2. Construction of dataset. This experiment builds the dataset based on the historical transaction data of cooperative bank from 2015 to 2017, and divides the training set and test set with January 1, 2017 as the boundary. Dataset 1: the test set was composed of 1 million pieces of normal transaction data and all fraudulent transactions in 2017; the training set was composed of 200,000 pieces of normal transaction data and all fraudulent transactions in these two years. Dataset 2: the training set was composed of the data in training set 1 and simulated data generated by GA-DS algorithm, the test set remains unchanged. The details of data are shown in table 3:

Table 3. Information about Datasets

| Dataset                        | Amount      | the number of normal transactions | the number of fraud transactions |
|--------------------------------|-------------|----------------------------------|---------------------------------|
| Training set of dataset 1      | 202156      | 202128                           | 28                              |
| Test set of dataset 1          | 994666      | 994654                           | 12                              |
| Training set of dataset 2      | 202180      | 202128                           | 52                              |
| Test set of dataset 2          | 994666      | 994654                           | 12                              |

4.2.3. Results and analysis. In this paper, we adopt the fraud detection model based on the random forest classification algorithm to evaluate the effect of GA-DS algorithm. Except for the differences in training set data, the characteristics and parameters of the detection model are consistent. The performance of the classifier was evaluated with OA, Sensitivity, g-mean, Recall, F1 Score and other indicators, and the results are shown in table 4.

Table 4. Results

| indicator         | detection model | GA-DS + detection model |
|-------------------|-----------------|-------------------------|
| OA                | 0.99723         | 0.99536                 |
| Sensitivity       | 0.08333         | 0.16667                 |
| G – mean          | 0.28828         | 0.40730                 |
| Recall            | 0.08333         | 0.16667                 |
| Precision         | 0.00036         | 0.00043                 |
| F1 Score          | 0.00073         | 0.00087                 |

According to the experiment results, the indicators as Sensitivity, G-mean, Recall, Precision, F1-Score have all improved, it can be concluded that the performance of the fraud detection model is improved after putting the mixed-data into training. The main reason is: the fraudulent data in the training set is seriously insufficient, the class distribution of datasets is extremely unbalanced, and isolated samples may be drowned by noise. Under the guidance of fitness function, GA-DS algorithm uses the information of the minority samples to generate simulated data near the real data, which effectively improves the unbalanced degree of sample distribution, enriches the potential fraud characteristics of the data, and improves the detection ability of the model.

5. Conclusion
For the problem of imbalanced data, this paper proposes GA-DS algorithm. The algorithm expands the original population through adaptive crossover and mutation operators, and sets up a comprehensive fitness function to make the simulated dataset converge towards the optimal direction. In the research
of this paper, the real fraud transactions are taken as the reference sample to generate the simulated data, and then mix them up and put into the anti-fraud model for the effect verification. The experiment result shows that the fraud detection ability of the model with data amplification was improved. In the next stage, we will continue our research on the fitness function in GA-DS algorithm.

6. Reference

[1] Zhao Y, Cai Y, Cheng D. A novel local exploitation scheme for conditionally breeding real-coded genetic algorithm[J]. Multimedia Tools and Applications, 2017, 76(17): 17955-17969.

[2] Chuang Y C, Chen C T, Hwang C. A real-coded genetic algorithm with a direction-based crossover operator[J]. Information Sciences, 2015, 305: 320-348.

[3] Kuang Y H, Jin J, Su Y. Improvement of Cross-mutation Operator in Adaptive Genetic Algorithms [J]. CEA, 2006(12):93-96+99.

[4] Evolutionary computation 1: Basic algorithms and operators[M]. CRC press, 2018.

[5] Shi Y. Research on Improving the Efficiency of Real-Number Genetic Algorithms [D]. Nanjing University of Aeronautics and Astronautics, 2002.

[6] Meng F, Chen X. Entropy and similarity measure for Atanassov’s interval-valued intutionistic fuzzy sets and their application[J]. Fuzzy Optimization and decision making, 2016, 15(1): 75-101.

[7] WEI Wei-jie, ZHANG Bin, WANG Bo. A Method Generating Data Sets to Test Data Mining Algorithms [J]. Journal of Northeastern University Natural Science, 2008(03):328-331

[8] Liu S, Lu M, Liu G, et al. A novel distance metric: generalized relative entropy[J]. Entropy, 2017, 19(6): 269.

[9] Menardi G, Torelli N. Training and assessing classification rules with imbalanced data[J]. Data Mining and Knowledge Discovery, 2014, 28(1): 92-122.