The Analysis of Incomplete Business Data

Longao Weng *, Rongxuan Wang

Jinhe Center For Economic Research, Xi'an Jiaotong University, Xi'an, China

*Corresponding Author Email: wlaleo@stu.xjtu.edu.cn

Abstract. Missing values can dramatically reduce the accuracy and availability of missing data, especially when analyzing business data. A common method to deal with the missing data is simply deleting the samples containing missing attributes. However, this will lead to bias and invalid conclusions since some data are too important to be omitted easily. Therefore, we should use certain methods to complete the data set instead of deleting data with missing values. In this paper, we compared several data imputation methods by adopting them to deal with six benchmark business data sets. The result provides us with guidance when dealing with incomplete business data.

Keywords: Data imputation, business, incomplete.

1. Introduction

During data analysis, missing values often exist in some samples of the dataset. The problem of incomplete data is a common issue in numerous real-world datasets, causing troubles for researchers in many fields. On Classification with Incomplete Data [1] indicates that incomplete-data problem exists in engineering subjects like social sciences, computer vision, biological systems, and remote sensing. Classification of Incomplete Data Using Classifier Ensembles [2] points out that in the process of data mining, it is a rare case that the data sets contain known values for each entry. Additionally, an effective and efficient approach to classification with incomplete data [3] illustrates that 45% of datasets in the UCI data set, a commonly used benchmark database for machine learning, contain missing values. And the reasons for the existence of missing values differ. For instance, partial responses from surveys can cause incomplete data in social science research, according to On Classification with Incomplete Data [1]. Also, in the application of remote sensing, the inadequate arrangement of sensors results in incomplete data.

Incomplete data also widely exists in the field of finance and business. Multivariable data imputation for the analysis of incomplete credit data [4] indicates that the data obtained by financial institutions from borrowers to access their ability to repay the loan in time suffers a lot from missing values. The cause is various, including the unwillingness of clients to respond to surveys, data acquisition fraud, and measurement errors.

According to recent studies, the existing methods to deal with incomplete data can be divided into deletion and imputation. The deletion method is one of the simplest ways to handle incomplete data, meaning that we simply delete the sample containing missing values. However, the deletion of certain samples may result in the loss of some useful and important information, since some data containing missing attributes also includes useful information. Generally speaking, deletion is recommendable when only a small portion of samples have missing values, and the biggest advantage of these methods is that they are efficient and time-saving [3]. However, deletion will not always be plausible, especially when a large portion of the samples have missing values according to A Novel Two-Phase Method for the Classification of Incomplete Data [5].

Data imputation means completing the data sets by filling the missing ones with specific values. Data uncertainty is ignored since imputation treats data as known values after filling them in. [1] One of the easiest imputation methods is to fill the missing values with a constant, which we often use as zero. This method is called zero imputation. Other methods include mean imputation, that is, replacing all missing values in a feature with the average value of the same feature. Median imputation works similarly, replacing the missing value with a median. These methods are relatively simple and time-saving.

Data interpolation methods in another field are more complicated, including k-nearest neighbor-based imputation (KNNI) and random forest imputation [6]. These methods are usually based on the correlation between the missing variables and other variables to ensure that the performance of the data after imputation is similar to the real data [3]. These relatively complex data imputation methods are often more effective but not efficient enough [6]. Therefore, it is often hard to select the most proper method. We have to weigh effectiveness and efficiency. Also, the selection can vary from one dataset to another, meaning that the underlying dataset is significant for our choice. Thus, we need to focus on the field of our research.

In this paper, we compare the performance of four different data imputation methods on six benchmark business datasets. Firstly, we artificially delete some attributes of the datasets and then use different methods for data imputation. Then we use the k-nearest-neighbor classifier to classify the filled data. After that, we use two indexes: F1 Score and Accuracy, to test the performance of the classification. The result gives us some suggestions for dealing with incomplete business data.

2. Our Methods

Our paper uses four data imputation methods: zero imputation, mean imputation, k-nearest neighbor-based imputation and random forest imputation. We will introduce their principle of them. After that, k-means clustering is used to evaluate and compare the performance of these data imputation methods. These four data imputation and classification methods are chosen mainly because they are widely used. Each of the data imputation methods has its unique strength and can be suitable for certain datasets.

2.1. Zero Imputation and Mean Imputation

Zero imputation and mean imputation work similarly: they use a specific value to fill in the missing data. Zero imputation uses zero to fill in all missing values without considering the differences between missing data on different attributes. This is the easiest way to deal with incomplete data.

Mean imputation is to use the average value of the attribute which contains the missing value to fill in the missing ones. Mean imputation is the most commonly used data imputation method since it is efficient and easy to understand. When the variables are nominal, we use the mode value imputation.

The common characteristic of these two methods is that they are simple and easy to adopt. Therefore, they are widely used. However, the results from these simple data imputation methods are not accurate enough in many cases. The main limitation of these simple methods is that replacing missing values with the mean or mode or a certain constant will result in a distorted estimate of the distribution function and lower the quality of data mining results [4].

2.2. k-nearest neighbor-based imputation

The k-nearest neighbor-based (KNN) imputation is an imputation method based on the k-nearest neighbor-based (KNN) algorithm. We will introduce the KNN first.

The idea of the KNN algorithm is that a sample belongs to a certain category if most of the k nearest neighbors of this sample in the feature space belong to a certain category. In other words, the method only decides how to classify samples based on the category of the one or more samples that are closest to them. The algorithm flow chart of KNN is shown in Fig. 1:

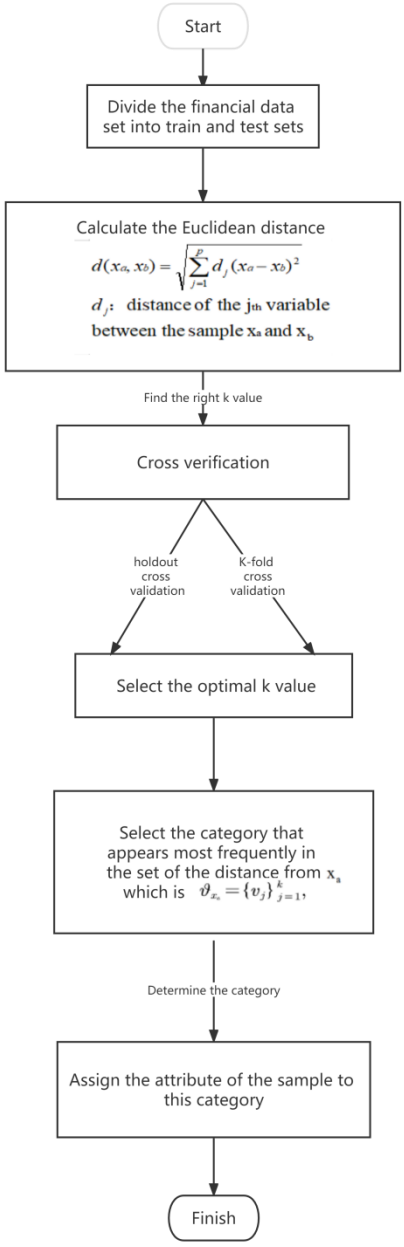

Figure 1. The process of KNN

It is found that the KNN algorithm can achieve relatively accurate classification, which provides researchers with the idea to supplement missing values when dealing with data containing missing values. That is, the classification result of the KNN algorithm is filled as the value of the missing term in this attribute. Therefore, KNNI, based on the KNN algorithm, was created.

KNNI uses k neighbors of the sample containing missing values to generate a value to fill in the missing value. The k neighbors of the sample are the most similar samples from the dataset. The most common value among all neighbors is taken for nominal values, and for numerical values, the average value is used [6]. The algorithm flow chart of KNNI is shown in Fig. 2:

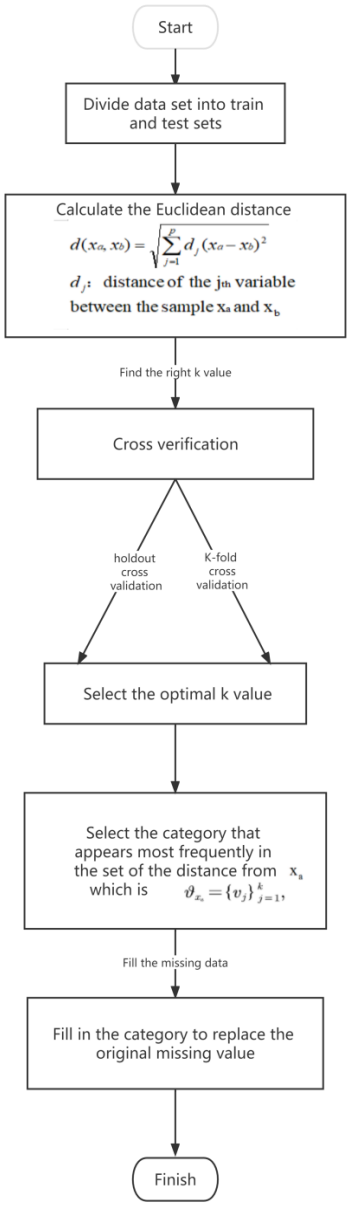

Figure 2. The process of KNNI

The KNNI method is more effective than zero imputation and means imputation. Also, the principle of KNNI is easy to understand. However, KNNI also has its own limitation. Firstly, it can be computationally expensive when the number of samples goes large since, for every sample, it requires searching for k nearest neighbors. Moreover, we use the Euclidean distance to find the most similar neighbor of the sample with missing values in our paper. However, there is still debate in the academic world about which method we should use to generate the k most similar samples [4]. In addition, in a high-dimensional dataset, the difference between the nearest neighbor and the farthest neighbor is very small, so the accuracy of KNN will be reduced.

2.3. Random forest imputation

Ed with those previously mentioned methods, the principle of random forest imputation (RFI) is much more complex. In practical research, RFI is not widely used. But as a complement and contrast to the previous methods, we still use this method for imputation.

The random forest (RF) algorithm is another classification algorithm whose idea is that the probability of a large number of decision trees making mistakes is much lower than the probability of a single tree making mistakes. By creating multiple decision trees and training them without pruning, the classification of a certain attribute of the data is predicted accordingly. The mode of the

prediction result of a large number of decision trees is regarded as the prediction result of the whole. The accuracy of RF classification was found to be very high. The algorithm flow chart of RF is shown in Fig. 3:

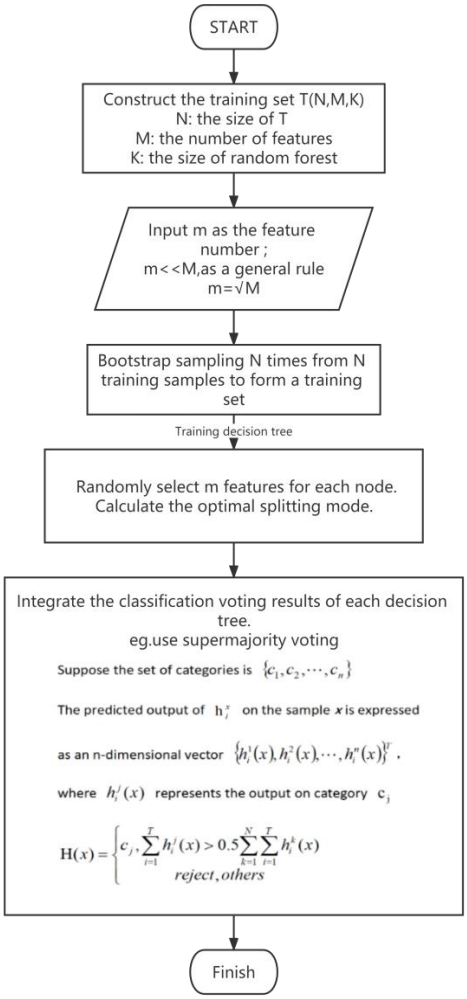

Figure 3. The process of RF classification

RFI is proposed due to the satisfying accuracy of classification that the RF algorithm brings. The classification results of the RF algorithm are added to the data set as fill-in values. In the face of multiple data sets with missing attributes, RFI will select the attributes with the least missing values to start interpolation and apply the interpolation results as new information to fill the subsequent attributes with more missing values. The algorithm flow chart of RFI is shown in Fig. 4:

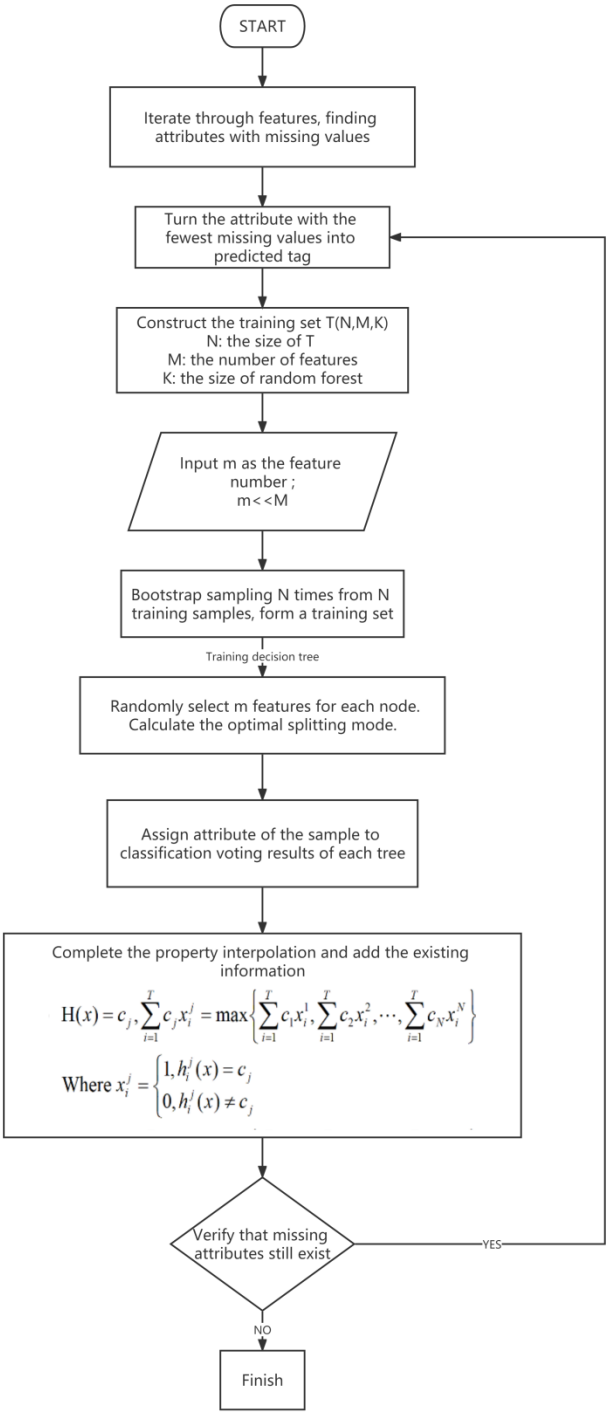

Figure 4. The process of RFI

Just as the high accuracy of its classification performance, RFI is highly effective. However, the disadvantage of this algorithm is that random forest modeling is used repeatedly in the algorithm operation, resulting in a large amount of calculation. Therefore, the running time of the RFI algorithm is always long.

3. The Experiment

3.1. Description of the Dataset

We select six benchmark business datasets, five of which are from the UNI database, a famous database for machine learning, and the dataset for business. And the last dataset is about factors that affect the bankruptcy of corporations. They are South German Credit, Australian Credit Approval,

Bank Marketing, the default of credit card clients, Online Shoppers Purchasing Intention, and Bankrupt. For datasets that contain too many samples, we only select part of them to do the analysis. A detailed description of the datasets is shown in Table I:

Table 1. The Description of Data

dataset	Observations	Features	Classes
South German Credit	1000	21	2
Australian Credit Approval	691	15	2
Bank Marketing	4521	17	2
default of credit card clients	30000	24	2
Online Shoppers Purchasing Intention	6855	18	2
Bankrupt	1359	96	2
Australian Credit Approval	691	15	2
Bank Marketing	4521	17	2

1) The Preprocessing of the data

We use Python for programming and data analysis. We divide each dataset into train and test sets, accounting for eighty percent and twenty percent of the initial dataset separately. In order to compare the performance of different data imputation methods, we randomly delete some features of the sample from the test set of each dataset. To be more specific, for each dataset sample, we randomly choose a positive integer m, which is strictly less than the total number of features of each dataset. Then m features of each sample will be deleted.

2) The imputation and classification of the Missing Values

We use four data imputation methods to fill in the missing values of each of the four test sets mentioned above. In the meantime, we use the train set to train the KNN classifier and then the trained KNN classifier to classify the filled test set. By comparing the real classes of samples and the estimated classes, we can evaluate the accuracy of data classification. A more accurate data classification indicates a more feasible method of data imputation.

3) The Evaluation of Classification

We construct the confusion matrix for the evaluation of the classification. The confusion matrix is defined in Table II:

Table 2. The Confusion Matrix

	Positive	Negative
Positive	TP	FN
Negative	FP	TN

From the confusion matrix, we can construct the index for our valuation. $Accuracy$ is exactly defined as the ratio of the successfully predicted samples to the total samples:

$$Accuracy = (TP + TN)/Total \tag{1}$$

Where $Total = TP + TN + FP + FN$

$Precision$ is defined as:

$$Precision = TP/(TP + FP) \tag{2}$$

$Recall$ is defined as:

$$Recall = TP/(TP + FN) \tag{3}$$

However, $Recall$ will decrease as the $Precision$ increases, and these two indicators are always negative correlated. We can adopt $F1\ Score$, which is a combination of $Precision$ and $Recall$, to evaluate the classification performance.

Then the $F1\ Score$ is defined based on the $Precision$ and $Recall$:

$$F1\ Score = 2 \times Precision \times Recall/(Precision + Recall) \tag{4}$$

Thus, $F1\ Score$ can be regarded as $Precision$ and $Recall$. Using $F1\ Score$ means that these two indicators are seen as equally important. $F1\ Score$ ranges from 0 to 1, and the closer it is to 1, the better the classification.

We choose $Accuracy$ and $F1\ Score$ as the indexes for our classification. From our data, we calculate these two indexes. The result is shown in Fig. 5 and Fig. 6:

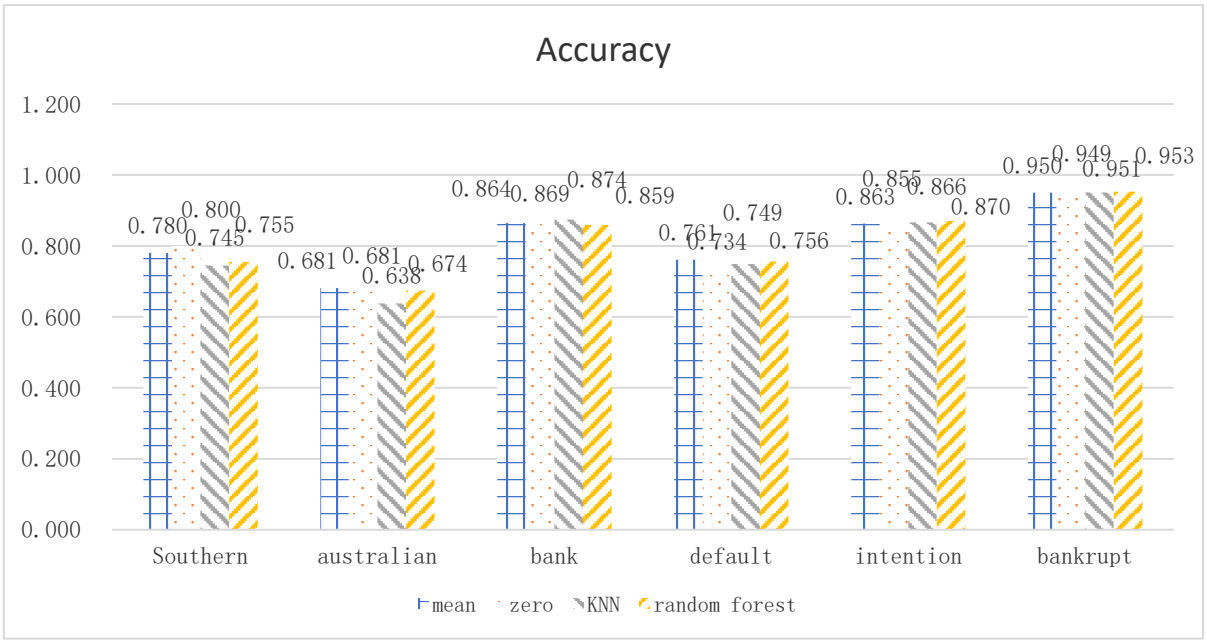

Figure 5. Accuracy of different imputation methods

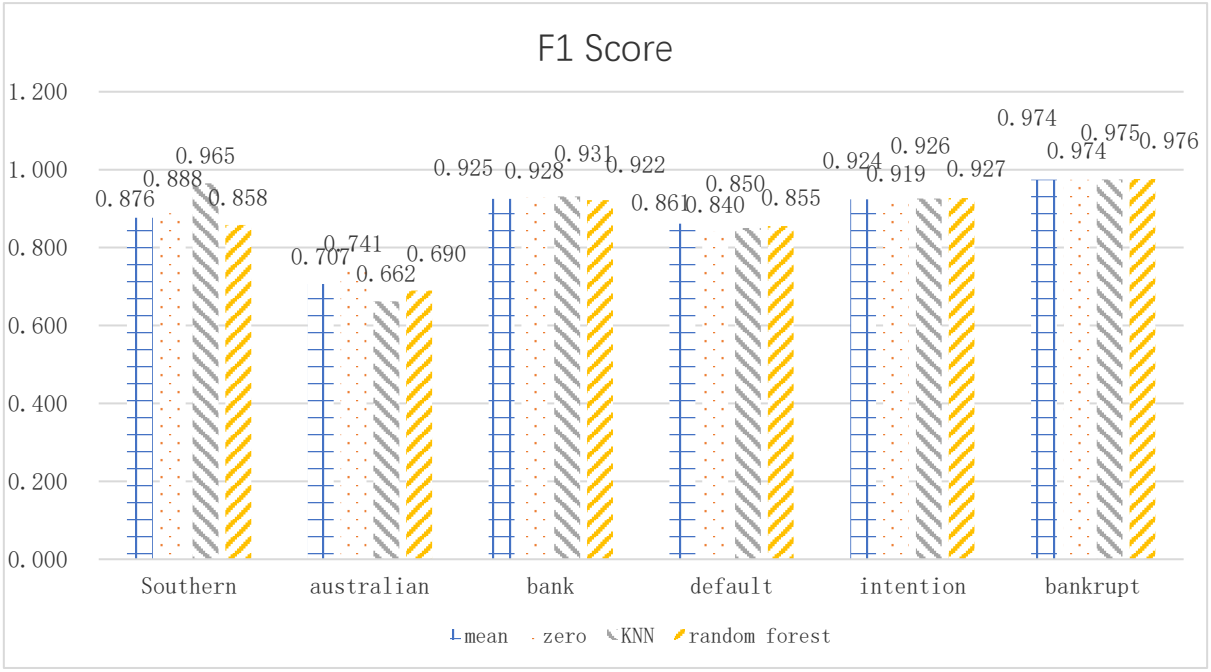

Figure 6. F1 Score of different imputation methods

Since the data of our datasets are all binary, we can easily use F1 score to compare the classification performance, which is much computationally cheaper compared to multiple variable cases. Besides, the classification variables of the datasets except for South German Credit all contain more zero than one, so we set zero as positive instead of one in these datasets.

From the result, we can see that each method's accuracy is similar among the datasets. Specifically, we notice that the accuracy and F1 score of zero and mean imputation, two relatively simple and

easily-adopted methods, are almost the same as that of other relatively complex methods, indicating that there is no reason for us to use computationally expensive methods in most cases.

For the KNNI and RFI algorithm, their Accuracy and F1 Score are close to each other, indicating that the performance of KNNI and RFI on our dataset is also similar. However, RFI is much more time-costing than KNNI. Therefore, we conclude that there is no need to use RFI in most cases due to its high complexity and unintuitive principle. This corresponds to the fact that KNNI is one of the most widely-used imputation methods today.

Also, we emphasize that as the number of features increases, the performance of each imputation method converges. Therefore, when we have a large number of features on our datasets, the performance of each method is not too much different. In this case, we should prioritize the ways that save us time: zero imputation and mean imputation.

4. Conclusion

In this paper, we employ four imputation methods to deal with six artificially missing business datasets to test and compare the performances of different methods. In the process of business data cleansing, the problem of missing data cannot be ignored. In this case, we need to analyze the dataset's characteristics before applying certain methods for data imputation.

References

[1] David Williams, Xuejun Liao, Ya Xue, Lawrence Carin, Balaji Krishnapuram. On Classification with Incomplete Data [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007, 29(): p.427-436.

[2] Fangfang. Research on power load forecasting based on Improved BP neural network [D]. Harbin Institute of Technology, 2021.

[3] Cao Truong Tran, Zhang Mengjie, Andreae Peter, Xue Bing, Lam Thu Bui. An effective and efficient approach to classification with incomplete data[J]. Knowledge-Based Systems, 2018, 154(AUGa15): 1-16.

[4] Lan Qiujun, Xu Xuqing, Ma Haojie, Li Gang. Multivariable data imputation for the analysis of incomplete credit data [J]. Expert Systems with Application, 2020, 141(Mara): 112926.1-112926.12.

[5] Ma Kunlong. Short term distributed load forecasting method based on big data [D]. Changsha: Hunan University, 2018.

[6] Julián Luengo, Salvador García, Francisco Herrera. On the choice of the best imputation methods for missing values considering three groups of classification methods [J]. Knowledge and Information Systems, 2012, 32(1): p.77-108.

[7] Mursalin M, Zhang Y, Chen Y, et al. Automated Epileptic Seizure Detection Using Improved Correlation-based Feature Selection with Random Forest Classifier [J]. Neurocomputing, 2017, 241(JUN.7): 204-214.

[8] Xia Z, Wang X, Sun X, et al. A Secure and Dynamic Multi-Keyword Ranked Search Scheme over Encrypted Cloud Data [J]. IEEE Transactions on Parallel & Distributed Systems, 2016, 27(2): 340-352.

[9] Belgiu, Dragut. Random forest in remote sensing: A review of applications and future directions [J]. ISPRS J PHOTOGRAMM, 2016, 2016, 114(-): 24-31.

[10] Ahmad M W, Mourshed M, Rezgui Y. Trees vs Neurons: Comparison between random forest and ANN for high-resolution prediction of building energy consumption [J]. Energy and Buildings, 2017, 147.

[11] Yang Xi, Nan Xiaoting, Song Bin. D2N4: A Discriminative Deep Nearest Neighbor Neural Network for Few-Shot Space Target Recognition [J]. IEEE Transactions on Geoscience and Remote Sensing, 2020, 58(5): 3667-3676.

[12] Larkey L S, Croft W B. Combining classifiers in text categorization [J]. ACM, 2019.

[13] Paul A, Mukherjee D P, Das P, et al. Improved Random Forest for Classification [J]. IEEE Transactions on Image Processing, 2018: 4012-4024.