Eng Int Syst (2023) 3: 229–237

Engineering
Intelligent Systems

Deep Mining Method of Distributed Data Association Based on Decision Tree Algorithm

Jingjing Cai[1,*] **and Yongsheng Ding**[2]

[1] *College of Modern Information Technology, Henan Polytechnic, Zhengzhou 450046, China*
[2] *Zhengdong New Area Wenyuan School, Zhengzhou 450046, China*

In this era of big data, the enormous increase in the amount of data being generated makes it problematic to determine the association between data. Hence, this paper is concerned with the deep mining of distributed data association based on the decision tree algorithm. The top-down recursive method is adopted to compare the attribute values of the internal nodes of the decision tree, determine the downward branches from the node according to different attribute values, and generate a decision tree through the probability estimation of single tree and multiple trees. The gain ratio algorithm is used to optimize the information gain algorithm to obtain the heuristic information of the decision tree and select the most appropriate test attributes. At the same time, the pruning strategy applied to a decision tree is optimized by setting multiple thresholds. The optimized decision tree is used to deeply mine the association between horizontal and vertical data distribution. The results show that the decision tree constructed by this method can accurately and deeply mine different attributes, the mining process has good stability, and the mining results can meet the needs of practical application.

Keywords: Decision tree algorithm; Distributed data; Association; Deep mining; Probability estimation; Pruning strategy.

1. INTRODUCTION

Big data brings challenges to machine learning, and efficiency has become the key issue of large-scale machine learning. With the development of mobile Internet, people are producing more and more data. According to the new Moore's law, the scale of data increases 10 times every five years [1]. In addition to the increase in the volume of data, the dimension of data is becoming increasingly greater. This creates more valuable opportunities for machine learning, but it also poses more severe challenges: the growth of data volume and dimension makes the computing overhead of machine learning increase sharply, and the learning efficiency becomes less and less. The reason is that the computational complexity of classical machine learning algorithms mostly increases in a super linear manner with the growth of data volume or dimension [2]. The problem posed by the increase in the scale of data and the corresponding reduction of learning efficiency has become more and more acute.

In order to solve the problem of the rapidly increasing computing overhead associated with machine learning, research on distributed machine learning algorithms and platforms has become a hot topic in the field of machine learning [3]. However, the distributed research on the existing algorithms only solves the problem of computing power expansion and how to reduce the I/O, network and synchronization overhead in the expansion process, but the association between the growth of data volume and the decreased efficiency of the algorithm has not been solved. The continuous iteration of super linear time complexity involved in computing makes it difficult for them to be deployed in parallel on distributed clusters [4]. Therefore, the research on learning algorithms that can be deployed in parallel, efficiently and accurately on distributed platforms has become the focus of studies in related fields.

*Email of corresponding author: 12008@hnzj.edu.cn

Li et al. [5] studied the distributed edge computing unloading algorithm based on deep reinforcement learning. Based on the real-time state of the network and the attributes of the task, they used the deep deterministic policy gradient (DDPG) of actor criticism and policy gradient to optimize the computing load. This method is too simple for large-scale association mining. Yuan et al. [6] researched the overall traffic pattern prediction method of large-scale data based on the Vomm method and the AR mining algorithm. This method is based on the variable order Markov model theory and the probability suffix tree. Because association rules are extracted from historical traffic data and describe the relationship between the state of traffic in different regions, association rules are used to improve the prediction performance. This method is mainly used for the association mining of fuzzy data, but the mining results are poor. Cheng and Yang [7] studied a fast and efficient algorithm for mining minimum functional dependencies from large-scale distributed data using spark. This method is used for fuzzy association mining and, again, the mining results are poor. Zhu [8] proposed a spatiotemporal feature mining algorithm for the Internet of Things based on multiple minimum support of pattern growth, modeled the location sequence, and added the time information to the model. Then, the mining algorithm for asynchronous periodic sequence pattern was adopted. The algorithm was based on multiple minimum supports of pattern growth. According to multiple minimum supports, the asynchronous periodic sequence pattern was deeply and recursively mined. However, this method is unable to conduct nonlinear mapping, and there are some errors in the sequence mining results.

The stochastic decision tree algorithm is a simple and efficient learning algorithm, which performs well in terms of accuracy and efficiency. However, due to the complexity of a tree structure, it is difficult to calculate in parallel, and it cannot deal efficiently with big data problems. Inspired by the random decision tree theory, in this paper, a deep mining method for distributed data association is proposed based on the decision tree algorithm, several problems existing in the random decision tree algorithm are addressed, the deep mining of distributed data association is achieved on this basis, and the practical application performance of this method is verified through experiments.

2. MATERIALS AND METHODS

2.1 Decision Tree Construction

The classification algorithm is used mainly to analyze the input data and find an accurate description or model for each class through the characteristics of the data in the training set [9]. The generated class description is used to predict and classify the future test data. The decision tree algorithm is an important algorithm used for the training set. The internal node of a decision tree is a set of attributes, and the leaf node is the class to be learned and divided. When a decision tree is generated after a batch of training instance sets has been trained, it can classify an unknown instance set according to the value of attributes [10]. When using the decision tree to deep mine distributed data association, the value of the attribute of the object is gradually tested from the root of the tree, and goes down the branch until it reaches a leaf node. The class represented by this leaf node is the class of the object.

The decision tree learning adopts the top-down recursive method, compares the attribute values at the internal nodes of the decision tree, judges the downward branches from the node according to different attribute values, and obtains the conclusion at the leaf nodes. Therefore, the path from root to leaf node corresponds to a conjunction rule, and the whole decision tree corresponds to a set of disjunctive expression rules. The generation of a decision tree involves two steps [11]: one is the generation of the tree where, at the beginning, all data is at the root node, and then the data is divided recursively; second is tree pruning, which is done to remove any data that may be noise or abnormal. Segmentation by the decision tree terminates when the data on a node belongs to the same category, and no attributes can be used to segment data. Figure 1 depicts the steps for the generation of the decision tree.

2.1.1 Probability Estimation of Single Tree

When using a decision tree to deeply mine the association of distributed data, x is taken as the instance of distributed data and y as the category. For binary classification problems, if a posteriori distribution can be found, the corresponding prediction results can be given. However, the actual probability distribution function is always unknown, so it can estimate only the function $H^*(y = +|x|)$. For simplicity, $H_+(x)$ and $H_+^*(x)$ are used to represent $H(y = +|x|)$ and $H^*(y = +|x|)$ respectively to study the error values of $H_+(x)$ and $H_+^* - (x)$ defined by random decision tree. The equation is as follows:

$$\begin{aligned} MSE(H_+^*(x)) &= E\left\{\left[H_+^*(x) - H_+(x)\right]^2\right\} \times \phi \\ &= \int \left[H_+^*(x) - H_+(x)\right]^2 dx \times \phi \end{aligned} \quad (1)$$

where ϕ represents error correction.

In the case of univariate probability estimation, the random decision tree algorithm divides the data space into multiple bins (longitudinal stripes). Each bin contains several data instances. In addition to the uneven width of bins, a random decision tree is more similar to a histogram and is a non-parametric estimation method. In statistics, nonparametric estimation is more robust than parametric estimation for different data distributions. For a random decision tree, a single tree can be regarded as a random histogram. If there is one dimension, the following equation can be obtained:

$$H_+^*(x) = v_{+j}^* \times \frac{1}{v_j^*}, x \epsilon Z_j \quad (2)$$

where Z_j is a bin in the histogram, k_j is the width of Z_j, and v_{+j}^* is the number of instances in Z_j. Then, the estimated probabilities of several trees are averaged to obtain the final estimation of the positive category probability of a given data instance. Then, the results of mean square analysis of variance can be obtained:

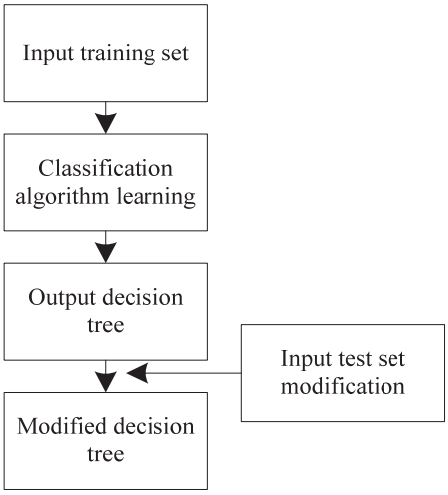

Figure 1 Generation process of decision tree.

The error value MSE between the estimated probability and the true probability depends on the width k_j of bin and the number of instances v^* in bin. When k_j becomes narrow, the deviation αk^4 will decrease. However, the variance $\frac{\epsilon}{v}$ decreases with the increase of v^* value. However, the decrease of k_j and the increase of v^* cannot occur at the same time; that is, the larger the width of bin, more instances can fall into bin. Therefore, k_j or v^* can reduce the MSE of the model [12]. In practice, it is more difficult to control k_j multidimensional problems, especially when there are many discrete and classification features in the data set. Therefore, v^* is used as the control parameter of the model. The MSE results of Equation (4) are as follows:

$$MSE \approx \left(\frac{\varepsilon}{v}\right) + \alpha k^4 \tag{4}$$

where α and β are constants.

2.1.2 Probability Estimation of Multiple Trees

After studying the error in a bin, it will continue to prove that the estimation probability of average multiple trees has less error than that of a single tree [13]. Given x and the number of trees m, $H_+(x)$ can be obtained as:

$$\begin{aligned} H_+^*(x) &= \frac{\sum_{i=1}^{m} h(k_i)(x)}{m} \\ &\approx \frac{\left[\sum_{i=1}^{m} H_+(t_i) + \sum_{i=1}^{m} \frac{H_+''(t_i)k_i^2}{24}\right]}{m} \end{aligned} \tag{5}$$

where k and t_i represent the width and midpoint of the ith bin respectively. When $m \to \infty$, $t_i \epsilon \left[\frac{x-a}{2}, \frac{x+a}{2}\right]$. Let $k = \max(k_1, \cdots, k_m)$, then:

$$H_+^*(x) \approx \frac{1}{\alpha}\left[\int_{x-\frac{a}{2}}^{x+\frac{a}{2}} H_+(t)dt + \frac{k^2}{24}\int_{x-\frac{a}{2}}^{x+\frac{a}{2}} H_+''(t)dt\right] \tag{6}$$

It can be seen from $H_+(x) \approx H_+(0) + xH_+'(0) + \frac{x}{2}H_+''(0)$ that:

$$\begin{aligned} H_+^*(x) \approx H_+(x) + \frac{k^2 H_+''(x)}{24} + \frac{k^2}{24} \\ \times \left[H_+'\left(x + \frac{\alpha}{2}\right) - H_+'\left(x - \frac{\alpha}{2}\right)\right] \end{aligned} \tag{7}$$

This means that as m increases, $H_+^*(x)$ will approach $H_+(x)$. This shows that the more the number of trees is, the smaller the estimation error of random decision tree is.

2.2 Improvement of Decision Tree Algorithm

2.2.1 Improvement of Test Attribute Selection Algorithm

A key step in decision tree algorithm is to select test attributes. General decision tree algorithm uses information gain based on entropy measurement as heuristic information to select the most appropriate test attributes [14].

(1) Information gain algorithm

Let X be a collection of x distributed data samples. Assuming that the class label attribute has m different values, m different classes $C_i (i = 1, 2, \cdots, m)$ are defined. Let x_i be the number of samples in class C_i. The expected information required for the classification of a given sample is given by the following equation:

$$I(x_1, x_2, \cdots, x_m) = \sum_{i=1}^{m} h_i \log_2(h_i) \tag{8}$$

where h_i is the probability that any sample belongs to C_i and is estimated by $\frac{x_i}{X}$. Let attribute A have v different values $\{a_1, a_2, \ldots, a_v\}$. X can be divided into v subset $\{X_1, X_2, \ldots, X_v\}$ with attribute A. Where X_j contains a sample of X having a value a_j on A. If A is selected as the test attribute, these subsets correspond to the branches growing from the nodes containing set X [15]. Let X_{ij} be the number of samples of class C_i in subset X_j. According to the entropy or expected information divided into subsets by A, it is given by the following equation:

$$E(A) = \sum_{j=1}^{v} \frac{(X_{1j} + \cdots + X_{mj}) I(X_{1j} + \cdots + X_{mj})}{X} \quad (9)$$

Item $\frac{X_{1j}+\cdots+X_{mj}}{X}$ acts as the weight of the j-th subset and is equal to the number of samples in the subset divided by the total number of samples in X. The smaller the entropy, the greater is the purity of the subset division. For a given subset:

$$I(X_{1j} + \cdots + X_{mj}) = -\sum_{i=I}^{m} {h_{ij}}^2 \quad (10)$$

where $\frac{X_{ij}}{|X_j|}$ is the probability that the sample in X_j belongs to class C_i.

The coding information obtained by branching on A is:

$$Gain(X, A) = I(x_1, x_2, \ldots, x_m) - E(A) \quad (11)$$

$Gain(A)$ is the expected compression of entropy caused by knowing the value of attribute A.

The algorithm calculates the information gain of each attribute. The attribute with the highest information gain is selected as the test attribute of a given set, S. It can create a node and mark it with the attribute, create branches for each value of the attribute, and divide samples accordingly.

(2) Improvement of gain algorithm-gain ratio algorithm

In a practical application, the information gain algorithm prefers those attributes with a large number of values, resulting in many small and pure subsets. In order to avoid this problem, an information gain ratio algorithm is proposed [16]. Firstly, the amount of information of attributes irrelevant to classification is calculated, that is, the entropy of S about the values of attribute A:

$$Split\ Info(X, A) = \sum_{m=I}^{v} \frac{X_m \left(\log_2 \left(\frac{X_m}{X}\right)\right)}{X} \quad (12)$$

where X_m is the number of samples in attribute A and x is the total number of samples. The more average the samples of an attribute divided by value is, the larger the $SplitInfo$ is.

The gain ratio uses $SplitInfo$ to avoid selecting these attributes:

$$Gain\ Ratio(X, A) = \frac{Gain(X, A)}{SplitInfo(X, A)} \quad (13)$$

where $GainRatio$ is the gain ratio, $Gain$ is the information gain of the attribute, and $SplitInfo$ is the entropy of the attribute.

2.2.2 Improvement of Decision Tree Pruning Strategy

During decision tree learning, if the decision tree is too complex, the cost of storage will be greater. If the number of nodes is too large, the smaller the number of instances contained in each node, the smaller is the number of instances that support the assumption of each leaf node, and the probability of error will increase after learning. Moreover, this is difficult for users to understand, which makes the construction of a classifier unfeasible to a large extent. Time shows that simple assumptions can better reflect the relationship between things. Therefore, the decision tree should be simplified for decision tree learning [17].

There are two common simplification methods: pre-pruning and post-pruning. When decision trees are built, many of the branches may reflect noise or outliers in the training data which leads to over fitting the data. Pre-pruning attempts to identify and remove such branches, and stops the growing of the tree earlier before it classifies the training set. The termination of the decision tree growth is given by the algorithm. This may occur when the decision tree reaches a certain height, or it sets a threshold based on the gain ratio of parent node and child node, or sets a threshold for the maximum sample ratio, etc. Post-pruning is used to cut off the branches of the fully grown tree, thereby removing the nodes on these branches.

In regard to the decision tree pre-pruning and post-pruning algorithms, pre-pruning is simple to implement and can greatly improve the system performance. However, it also has the disadvantage that the pruning threshold is manually defined, which can easily result in pruning that is either too coarse or too fine. Therefore, multiple constructions are used to determine the optimal threshold [18], usually done by establishing multiple thresholds. When the algorithm used for constructing the decision tree is applied to split the current node, it can calculate the ratio of the gain ratio of the parent node of the current node to the gain ratio of this node. If the ratio is less than the given threshold, this node is considered to be pure, so the node splitting is terminated, the thresholds of trimmed decision trees are compared, and the best threshold is chosen.

2.3 Deep Mining of Distributed Data Association

In the distributed data environment, data usually has two distribution modes [19]: horizontal data distribution and vertical data distribution. The following subsections discuss the association deep mining under these two data distribution modes.

2.3.1 Distributed Mining of Decision Tree Based on Horizontal Data Distribution

When the data is horizontally distributed, the data records are scattered among different stations. If the data set contained in each station is $X_1, X_2, \ldots X_n$ and the corresponding complete data set is defined as X, in the case of horizontal segmentation, these data sets must meet the following requirements: $X_1 \bigcup X_2 \cdots \bigcup X_n = X$.

According to the research ideas presented in section 1, in the case of data horizontal segmentation, when applying the decision tree method to deeply mine the association of distributed data, the following steps are taken:

(1) The query constraint $constr(h)$ is transmitted to each data site by controlling the site. For the root node,

the query constraint is empty; For non-leaf nodes, the constraint $constr(h)$ is passed.

(2) At each data site, execute the query and statistics operation I_d including constraint $constr(h)$ to obtain the local class distribution table of each site.

(3) The local class distribution map of each station is transmitted to the control station.

(4) At the control site, the local class distribution tables are combined into the global class distribution table through the synthesis operation CL [20]. Based on the global class distribution table, the $gini$ coefficient or information entropy of each segmentation is calculated to obtain the global optimal segmentation point.

(5) At the control site, based on the global optimal segmentation point, the decision tree is further grown through the H operation.

In the steps above, I_d is carried out at each data station, and CL and H are carried out at the control station. The communication information between each data station and control station is the local category distribution map.

2.3.2 Decision Tree Distributed Mining Based on Vertical Data Distribution

In a vertically distributed data set, each data record is divided into several sub records, and each sub-record of each data record shares the same index value. Let $A_1, A_2, \ldots, A_n$ represent the attribute set, the values of these attributes are stored in the data site $1, 2, \ldots, S$ respectively, and A is all the attributes of the complete data set, then in the case of vertical segmentation of data, $A_1 \bigcup A_2, \cdots, A_s \bigcup A$. $X_1, X_2 \cdots X_s$ is set as the data stored in site $1, 2, \ldots, S$ and X is the complete data set. ${t_x}^i_j \cdot index$ represents the ith record in the sub dataset X_j, ${t_x}^i_j \cdot index$ is defined as the index value of the record $t^i_{x_j}$, and R represents the connection ($join$) operation. Vertically segmented data should have these properties:

(1) $X_1 \times X_2 \times \ldots \times X_s = X$

(2) $\forall D_j, D_k, t^i_{D_j} \cdot index = t^i_{D_k} \cdot index$

For this kind of distributed data, the deep mining of distributed data association based on the decision tree method involves these steps:

(1) Using the property of $t^i_{X_j} \cdot index = t^i_{X_k} \cdot index$, the index set matching constraint $constr(h)$ is obtained and transmitted from the control site to each data site.

(2) At each data site, the index set is used to perform the query and statistics operation I_d to obtain the global class distribution table of candidate attributes contained in each site.

(3) The global class distribution table of candidate attributes contained in each site is transmitted to the control site. Through operation of CL, the $gini$ coefficient or information entropy of each segmentation is calculated, and then the global optimal segmentation point is obtained by comparison.

(4) At the control site, based on the global optimal segmentation point, the decision tree is further grown through the H operation.

In the steps above, I_d is carried out at each data station, and CL and H are carried out at the control station. The information communicated between each data station and the control station includes the index set of records that meet the constraint $constr(h)$ and the statistical information of all candidate attributes stored in each station.

3. EXPERIMENTAL RESULTS

The experiment was conducted to determine the effectiveness of the proposed deep mining method for distributed data association based on the decision tree algorithm. Taking a city as the target area, the distributed power inspection information was collected from the target area to generate a data set. Taking the data set as the experimental object, this method is used to mine and analyze its association. The results are presented below.

3.1 Results of Decision Tree Construction

The method proposed in this paper is used to construct a decision tree for the information in the experimental object. Taking the inspection line, power equipment and power fault as examples, the decision tree is constructed. The results are shown in Figure 2.

As shown in Figure 2, when calculating the information gain and determining the information attributes, the proposed method focuses too much on dividing the attributes with more samples, such as patrol lines, power equipment and power faults in the experimental object, which is convenient for the division of sample attributes, but also has some impact on the rationality of the mining process. The decision tree constructed by the proposed method can carry out accurate deep mining for different attributes, and the mined information is accurate, which shows that the decision tree constructed by this method is more accurate.

3.2 ROC Curve

As an analysis tool of coordinate schema, the ROC curve can describe the classification performance of different algorithms, and the area of its range can reveal the correctness of an algorithm. Figure 3 shows the ROC curve comparison results of the method in this paper, spark-based method and multiple minimum support method of pattern growth for deep mining of the research object.

As shown in Figure 3, the ROC curve of the proposed method has a broader range than the two methods used for comparison, indicating that the decision tree constructed by this method achieves greater accuracy for information deep mining in the experimental object than the two other methods.

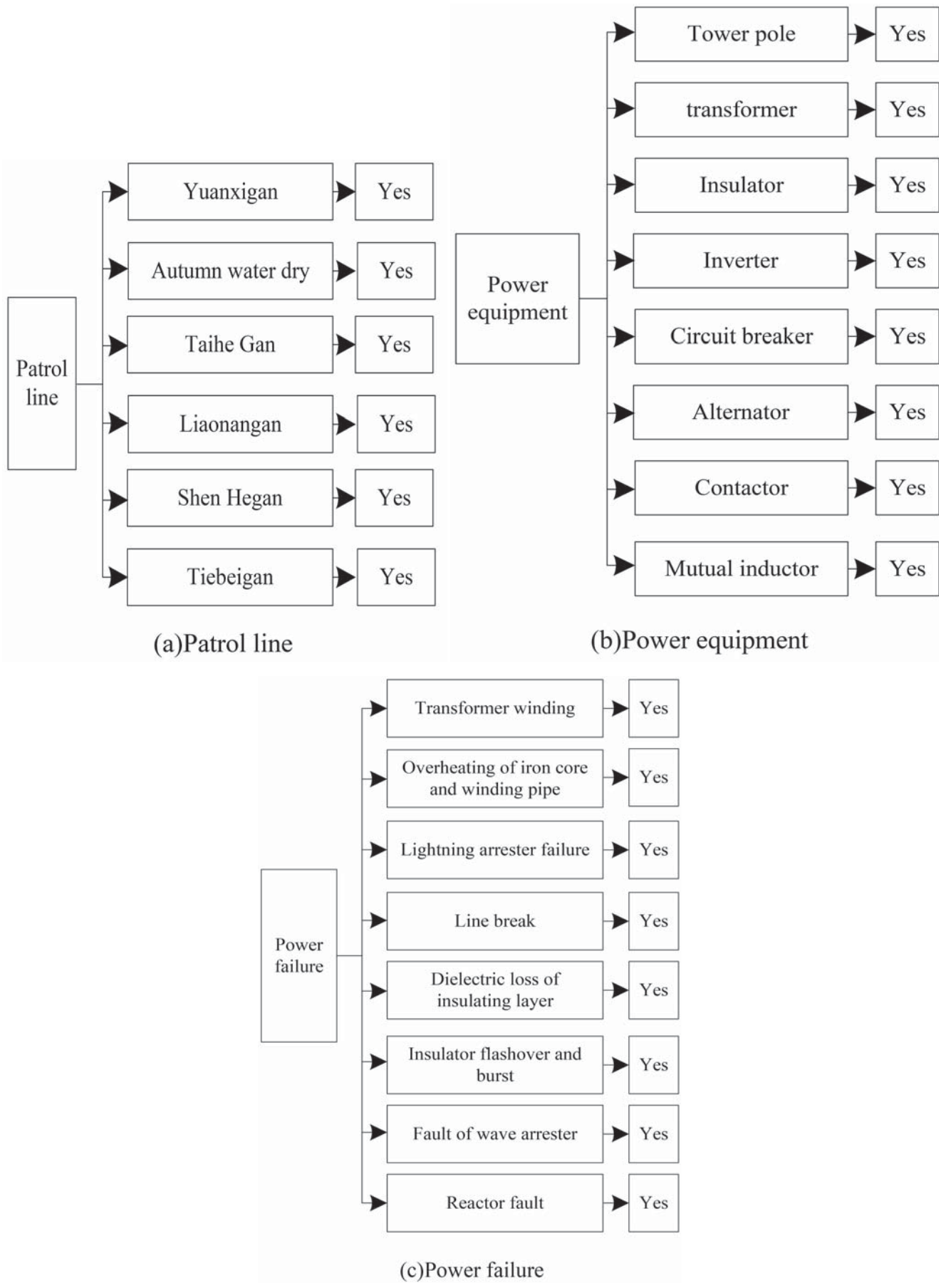

Figure 2 Decision tree construction results.

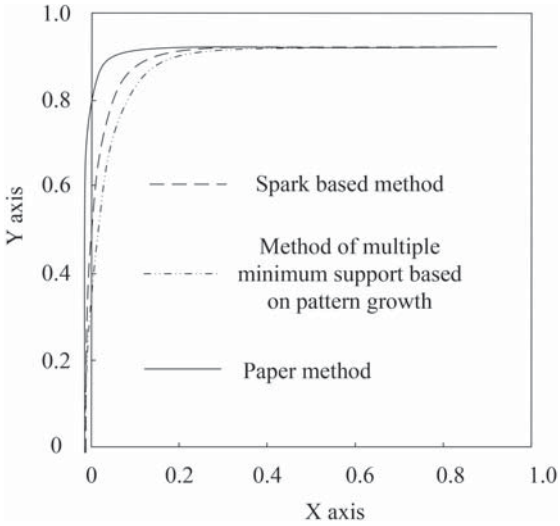

Figure 3 ROC curve.

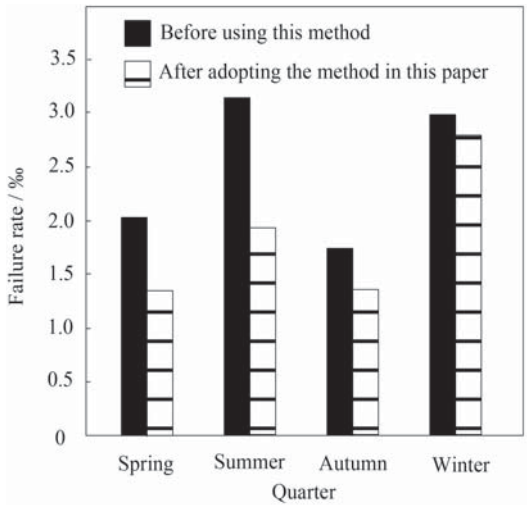

Figure 4 Change of failure rate.

3.3 Stability Test

In order to verify the stability of the proposed method in the application of distributed data deep mining, this method was used to carry out association data deep mining on different amounts of distributed data information, and the results of the mining accuracy and recall of this method were recorded. The results are given in Table 1.

The results in Table 1 show that when applied to different amounts of data, fluctuation range of the accuracy of data association deep mining of the proposed method is 96.97%–98.30%; the fluctuation range of recall rate is 95.33%–97.02%; the predicted fluctuation range of excavation level is 98.40%–98.92%. These results are significantly higher than those of the association data mining standards, which shows that the proposed method has good stability.

3.4 Fault Cause and Diagnosis Results

The main purpose of using the method in this paper to deeply mine the power inspection information in the target area is to identify the causes of faults in the power system of the target area and ensure the safety and stability of its power system. Therefore, 15 nodes were randomly selected in the target area power system. Table 2 shows the results of fault-cause identification in the power system of the target area, obtained by the proposed method.

According to Table 2, the results of fault-cause identification obtained by the proposed method are basically consistent with the actual fault causes, which shows that this method performs well when implemented in a practical context.

The algorithm in this paper is used to mine the multi-scale perception information of power inspection in the target area, and control and maintain the power system equipment and operation in the target area according to the mining results. Figure 4 shows the changes in the overall failure rate of power system equipment in the target area (taking the quarter as the unit and taking the average value in the quarter) after the deep mining of experimental object information by using the method proposed in this paper.

Figure 4 indicates that, after mining the multi-scale perception information of power patrol inspection in the target area by using the proposed method, the failure rate of power

Table 1 The stability test results of this method.

Distributed data information volume/Pb	Accuracy of data association deep mining/%	Data association deep mining recall/%	Distributed data mining level prediction
1	97.58	95.33	98.40
10	96.97	95.96	98.43
100	97.54	96.81	98.61
1000	98.09	97.02	98.75
10000	98.30	96.39	98.92

Table 2 Results of fault-cause identification in power system of target area.

Power system node	Actual fault cause	Based on the deep mining results of this method, the fault causes are obtained
1	Generator set failure	Generator set failure
2	Abnormal network link of protection device merging unit	Abnormal network link of protection device merging unit
3	Transformer fault	Transformer fault
4	No fault	No fault
5	Transmission line fault	Transmission line fault
6	Capacitor failure	Capacitor failure
7	No fault	No fault
8	Protect goose link disconnection	Protect goose link disconnection
9	Abnormal current voltage link	Abnormal current voltage link
10	DC system grounding fault	DC system grounding fault
11	Protect goose link disconnection	Protect goose link disconnection
12	Abnormal current voltage link	Abnormal current voltage link
13	Abnormal network link of protection device merging unit	Abnormal network link of protection device merging unit
14	DC system grounding fault	DC system grounding fault
15	No fault	No fault

system equipment in the target area shows a significant decline in each quarter. This result shows that the control and maintenance of power system in the target area according to the information mining results of this method can effectively reduce the failure rate and ensure the operational safety of the power system in the target area.

4. CONCLUSION

Given the problem of distributed data mining resulting from big data, this paper examines the deep mining method of distributed data association based on a decision tree algorithm, optimizes the decision tree algorithm, and mines the data distributed both horizontally and vertically. The experimental results show that this method performs well in a practical application. Due to time constraints, the proposed method has not been tested thoroughly. In the subsequent optimization process, more comprehensive testing is required as to improve the effectiveness of this method.

REFERENCES

1. Liu, J., Liang, X., Ruan, W., Zhang, B. (2021). High-performance medical data processing technology based on distributed parallel machine learning algorithm. The Journal of Supercomputing, 78(4), 5933–5956.
2. Modalavalasa, S., Sahoo, U.K., Sahoo, A.K. (2020). Sparse distributed learning based on diffusion minimum generalised rank norm. IET Signal Processing, 14(9), 683–692.
3. Qin, J., Wan, Y., Yu, X., Kang, Y. (2019). A newton method-based distributed algorithm for multi-area economic dispatch. IEEE Transactions on Power Systems, 35(2), 986–996.
4. Song, Q., Zhou, P., Peng, H., Hu, Y., Jia, B. (2020). Improved localization algorithm for distributed fiber-optic sensor based on merged Michelson-Sagnac interferometer. Optics Express, 28(5), 7207–7220.
5. Li, Y., Qi, F., Wang, Z., Yu, X., Shao, S. (2020). Distributed edge computing offloading algorithm based on deep reinforcement learning. IEEE Access, 8, 85204–85215.
6. Yuan, C., Yu, X., Li, D., Xi, Y. (2019). Overall traffic mode prediction by VOMM approach and AR mining algorithm with large-scale data. IEEE Transactions on Intelligent Transportation Systems, 20(4), 1508–1516.
7. Cheng, F., Yang, Z. (2019). Fastmfds: A fast, efficient algorithm for mining minimal functional dependencies from large-scale distributed data with spark. The Journal of Supercomputing, 75(5), 2497–2517.
8. Zhu, A. (2020). Spatiotemporal feature mining algorithm based on multiple minimum supports of pattern growth in internet of things. The Journal of Supercomputing, 76(12), 9755–9771.
9. Beigy, H., Meybodi, M.R. (2020). An iterative stochastic algorithm based on distributed learning automata for finding the stochastic shortest path in stochastic graphs. The Journal of Supercomputing, 76, 5540–5562.
10. Liu, W., Li, Z. (2019). An efficient parallel algorithm of n-hop neighborhoods on graphs in distributed environment. Frontiers of Computer Science, 13(6), 1309–1325.

11. Kraemer, K.H., Gelbrecht, M., Pavithran, I., Sujith, R.I., Marwan, N. (2022). Optimal state space reconstruction via monte Carlo decision tree search. Nonlinear Dynamics, 108(2), 1525–1545.
12. Fu, Y., Zhou, W. (2022). A heterogeneous parallel implementation of the Markov clustering algorithm for large-scale biological networks on distributed CPU-GPU clusters. The Journal of Supercomputing, 78(7), 9017–9037.
13. Banik, A., Majumder, M., Biswal, S.K., Bandyopadhyay, T.K. (2021). Polynomial neural network-based group method of data handling algorithm coupled with modified particle swarm optimization to predict permeate flux (%) of rectangular sheet-shaped membrane. Chemical Papers, 76(1), 79–97.
14. Zhang, Z., Yang, B., Liu, M., Li, Z., Guo, X. (2019). A quaternary-encoding-based channel hopping algorithm for blind rendezvous in distributed IoTs. IEEE Transactions on Communications, 67(10), 7316–7330.
15. Ma, J. (2021). Intelligent decision system of higher educational resource data under artificial intelligence technology. International Journal of Emerging Technologies in Learning (iJET), 16(5), 130.
16. Sharma, R., Nitin, N., Alshehri, M., Dahiya, D. (2021). Priority-based joint EDF-RM scheduling algorithm for individual real-time task on distributed systems. The Journal of Supercomputing, 77(1), 890–908.
17. Fajar, R., Jupri, P. (2020). P1835 prediction of chronic of kidney failure based on artificial intelligence system using the fuzzy decision tree algorithm. Nephrology Dialysis Transplantation, 35(Supplement_3), 1835.
18. Wu, Q. (2019). MOOC learning behavior analysis and teaching intelligent decision support method based on improved decision tree c4.5 algorithm. International Journal of Emerging Technologies in Learning (iJET), 14(12), 29–41.
19. Zhu, M., Chen, Q. (2020). Big data image classification based on distributed deep representation learning model. IEEE Access, 8, 133890–133904.
20. Fan, X., Huang, C., Zhu, J., Fu, B. (2019). R-DRA: A replication-based distributed randomized algorithm for data dissemination in connected vehicular networks. Wireless Networks, 25(7), 7–9.