Journal of Autonomous Intelligence (2023) Volume 6 Issue 1
doi: 10.32629/jai.v6i1.640

ORIGINAL RESEARCH ARTICLE

Intelligent transmission line fault diagnosis using the Apriori associated rule algorithm under cloud computing environment

Ahmed Hadi Ali AL-Jumaili[1,2*], Ravie Chandren Muniyandi[1*], Mohammad Kamrul Hasan[1], Mandeep Jit Singh[3], Johnny Koh Siaw Paw[4]

[1] *Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Bangi 43600, Malaysia. E-mail: ahmed_hadi@uofallujah.edu.iq; ravie@ukm.edu.my*

[2] *Computer Centre Department, University of Fallujah, Anbar 00964, Iraq.*

[3] *Department of Electrical, Electronic and System Engineering, Faculty of Engineering and Built Environment, Universiti Kebangsaan Malaysia, Bangi 43600, Malaysia.*

[4] *Institute of Sustainable Energy, Universiti Tenaga Nasional (The Energy University), Jalan Ikram-Uniten, Kajang 43000, Selangor, Malaysia.*

ABSTRACT

Electric power production data has the characteristics of massive data scale, high update frequency and fast growth rate. It is significant to process and analyse electric power production data to diagnose a fault. High levels of informationalisation and intellectualization can be achieved in the actual details of developing a Power Plant Fault Diagnosis Management System. Furthermore, cloud computing technology and association rule mining as the core technology based on analysis of domestic and foreign research. In this paper, the optimised Apriori association rule algorithm is used as technical support to realise the function of interlocking fault diagnosis in the intelligent fault diagnosis system module. Hadoop distributed architecture is used to design and implement the power private cloud computing cluster. The functions of private cloud computing clusters for power extensive data management and analysis are realised through MapReduce computing framework and Hbase database. The leakage fault cases verify the algorithm's applicability and complete the correlation diagnosis of water wall leakage fault. Through analysing the functional requirements of the system in the project, using MySQL database and Enhancer platform, the intelligent fault diagnosis management system of cloud computing power plant is designed and developed, which realises the functions of system modules such as system authority management, electronic equipment account, technical supervision, expert database, data centre. The result shows that the proposed method improves the security problem of the system, the message-digest algorithm (MD5) is used to encrypt the user password, and a strict role authorisation system is designed to realise the access and manage the system's security.

Keywords: Cloud Computing; Artificial Intelligent Algorithms; Power Systems; Smart Grid; Big Data

ARTICLE INFO

Received: 17 May, 2023
Accepted: 31 May, 2023
Available online: X July, 2023

1. Introduction

With the continuous introduction of advanced equipment and technology by power generation companies based on the standby operation network, the type and quantity of equipment are increasing. The number of parameters and status information is increasing day by day, the update frequency of data is high, and the growth rate is fast[1,2]. Due to the production plant of the equipment, business and technical standards are not unified, so the production data of electric power has a wide range of data sources on a large scale, and there are also isomerisation and diversification phenomena in data types[3]. In summary, the scale and complexity of power production data are not under the trend of continuous growth. How to achieve efficient and unified management of power data has become the main contradiction[4]. A set of management systems that can manage and analyse big power data and realise highly intelligence has become a new demand for power generation companies[5]. The scale of power monitoring data is vast, but the value density in the data is extremely low. In the data processing of power generation companies, the most widely used is the research in the direction of fault diagnosis based on data mining[6]. In power production, fault diagnosis is vital to ensuring power safety and stable production. When the equipment fails, it may cause immeasurable losses. Determining the fault's location, time, type, and cause is significant. However, due to the different data acquisition systems, many problems such as complex data sources and massive data scales have appeared. However, traditional data mining techniques are in fault diagnosis. It has achieved a wealth of results and technical applications, but in the "big data" processing problem, its data processing capabilities are too hard to meet the new demands of the major power companies[7,8]. The development of cloud computing in recent years provides a platform for data mining techniques to process "big data", enhancing the computing power of traditional data processing technology[9,10]. The development of cloud computing has brought new opportunities to discover the value of electric power in big data. In 2003, Google proposed "cloud computing", which integrates network technology based on traditional computer technology, including distributed computing, parallelisation technology, virtualisation technology and network storage that provide strong scalability, strong fault tolerance, and regulation of large size and low cost[11–13]. It has the property of super large-scale, and it provides distribution in the form of a computer cluster. Distributed parallel processing technology allows users to experience unparalleled computing power.

Cloud computing technology is currently stored in medical data besides good storage results, real-time traffic data, and weather data analysis[14]. The cost of cloud computing is low, and there are no complicated requirements for the servers in the cluster establishment. In power generation enterprises, abundant server resources can be used to support the foundation of cluster construction in cloud computing technology and fully use idle servers in power plants for cluster construction. With the help of cloud computing's large scale and fast computing speed, integration with traditional data mining technology can more effectively realise the management and analysis of power monitoring data. The introduction of cloud computing technology into the power field has necessary research. However, the application in the power industry is still in the exploratory stage, and it needs to be further improved when it is used in power production[15,16].

The processing and analysis of electric power production data are necessary to grasp the production status. The company's commissioned project serves as the basis for the research and development of an intelligent fault detection management system for power plants that uses cloud computing technology. First, build private cloud computing clusters, and then for Apriori's time-consuming and long-term processing of large-scale data sets, the use of cloud computing MapReduce technology is used to parallelise the design. The water wall leaking failure case is also used to evaluate the algorithm's viability. Finally, based on the above, the intelligent fault diagnosis management system for power plants based on cloud computing is developed. Second, cloud computing technologies such as HDFS, MapReduce, and Hbase. System research and development requirements are analysed, and a cloud computing cluster construction plan is designed. Hbase and Hive are implemented in the cluster. The MapReduce technical support and provide cloud computing infrastructure. The optimisation and application of the Apriori association rule algorithm based on cloud computing.

2. Related background

2.1 Association rules mining technology

Data mining refers to processing data in some way, calculating and analysing information hidden behind the data, so the data mining process is essentially the knowledge discovery[17]. The process of knowledge discovery is shown in **Figure 1**, and it is divided into three steps in a broad sense: pre-processing data, converting the data into a unified format that can be analysed; for different purpose, using different data mining algorithms to dig out the knowledge, explaining and evaluating the knowledge; expressing knowledge as a form of understanding.

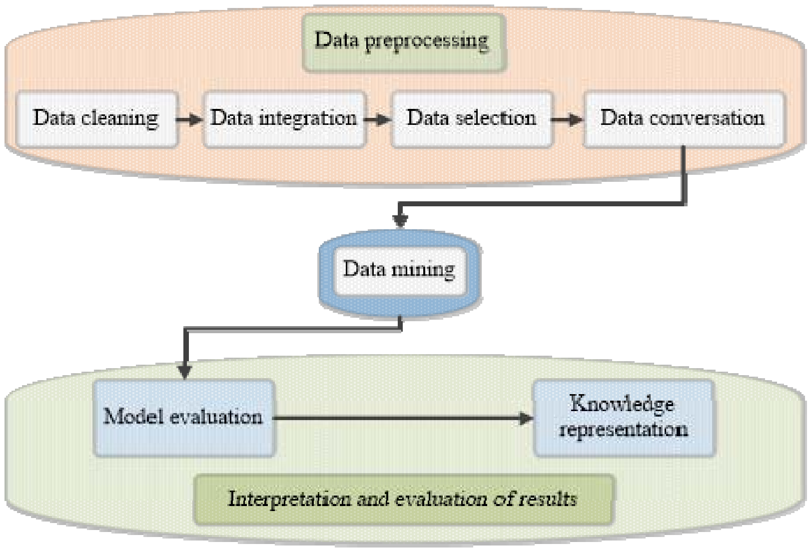


Figure 1. The process of knowledge discovery.

The associated rule mining is one of the branches of data mining technology. It proposes to discover the potential connection between transactions from data. The main implementation method is to find frequent items from the transaction data set and then generate the relationship between transactions between frequent items, which usually refers to this relationship as an associated rule. There are many types of association rules. According to different situations, it can be divided into the Boolean type and numerical type, single layer and multi-layer, single-dimensional and multi-dimensional; commonly associated rules are Apriori algorithms. It evolved from them[18,19]. Implementing the associated rules mining algorithm requires many definitions, theorems, and parameters[20–23]. Details are as follows:

Definition 1: $i = \{i_1, i_2, \ldots, I_M\}$ is the set of transaction sets. I_k ($k = 1, 2, \ldots, n$) is called item (item), and the set of items is called item set. The item set containing the k-item is called the k-item set.

Definition 2: $d = \{d_1, d_2, \ldots, d_m\}$ is a transaction database. D_i ($i = 1, 2, \ldots, m$) is called transaction. Each transaction is generally represented by the unique identifier TID. Let X be the transaction set, and when only $X \subseteq Dd_i$, the transaction D_i contains the items X.

Definition 3: Association rules are indicated as $X \Rightarrow Y$ ($X \subset i$, $Y \subset i$, $x \cap y = \varphi$). X is the preceding paragraph of the rules, and y is the later item of the rule. The excavation process of associated rules refers to the expression of $X \Rightarrow Y$ from the affairs library. If effective restrictions are not set, the transaction database will include an unlimited number of associated rules. Since people only need some of these rules, the evaluation standards that need to set analysis are needed.

The evaluation standards are shown in definition 4.

Definition 4: Support, confidence and improvement.

1) Supporting indicates the frequency of transactions in the concentration of projects. Assuming that S% of S% in D contains both projects set x and project set Y, S% indicates the support of association rules $X \Rightarrow Y$ and records as *Support* ($X \Rightarrow Y$). The calculation as Equation (1).

$$Support\ (X \Rightarrow Y) = \frac{|T(X \cup Y)|}{|T|} \tag{1}$$

where the database contains the number of transactions of $X \cup Y |T(X)|$, $T(X \cup Y)$ is the total number of transactions in the transaction library.

2) The confidence reflects the intensity of the associated rules $X \Rightarrow Y$. For the transaction in D of the project, the existence of C% of the project also includes the project Y. C% indicates the confidence of the associated rule $X \Rightarrow Y$ and records it as confidence ($X \Rightarrow Y$). The calculation as Equation (2).

$$Confidence\ (X \Rightarrow Y) = \frac{|T(X \cup Y)|}{|T|} \tag{2}$$

$|T(X \cup Y)|$ is the number of transactions containing $X \cup Y$ in the transaction database; $|T(X)|$, the number of transactions containing X in the database.

3) Lifting degree indicates the validity of association rule $X \Rightarrow Y$. When it is less than 1, the association rule $X \Rightarrow Y$ is invalid; when it is greater than 1, the association rule $X \Rightarrow Y$ is valid, and the larger the value, the stronger the validity; when it is equal to 1, it means that X and Y are independent of each other. The calculation as Equation (3).

$$Lift = \frac{Confidence(X \Rightarrow Y)}{|T(X)|} \tag{3}$$

where, $|T(Y)|$ is the number of transactions that contain Y in the database.

Definition 5: The support of the itemset X is not less than the set minimum support min-sup, and X is called frequent item sets. If X is a frequent k-item set, then X is called a frequent k-item set. Theminimum support is the minimum statistical standard of itemsets, and the support of frequent items must be greater than or equal to the given minimum support. The minimum confidence min-conf is responsible for measuring the lowest reliability of the association rules and it is responsible for ensuring the accuracy of the association rules.

2.2 The process of the Apriori algorithm

The implementation process of the Apriori algorithm is shown in **Figure 2**. The main steps through the connection operation and the pruning operation: The connection operation is responsible for connecting frequent k-item sets to generate candidate (k + 1)-item sets; in the pruning operation, calculate the support for all candidate item sets; if the calculated support degree is less than the set minimum support threshold or the sub-item of the candidate item set is not a frequent item set, delete this item from the candidate item set[24–26].

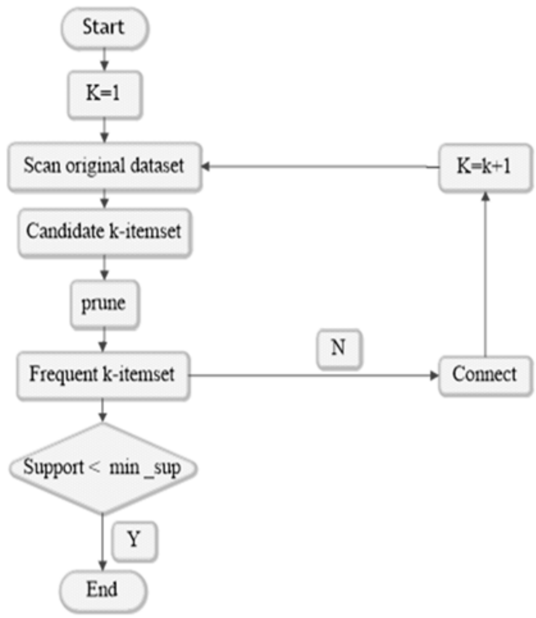


Figure 2. Apriori algorithm flowchart.

There are many challenges in mining meaningful association rules in large datasets. The principle of the Apriori algorithm is simple and easy to implement, but the algorithm implementation requires much computation to identify all frequent item sets. In the process, it needs to traverse the database many times, resulting in an exponential increase in the running time and complexity of the algorithm. It is concluded that the problems existing in the Apriori algorithm are as follows[27]:

(1) Identifying frequent items requires multiple scans of the transaction database, and the database needs to be scanned repeatedly during the connection operation and pruning operation. Either of the above will increase the I/O load, reduce the calculation rate, and require much time to calculate.

(2) The connection operation will generate many candidate item sets, which can be handled calmly when the data is small. However, in the face of massive data, redundancy will occur, increasing a load of calculation and storage to a certain extent and reducing the algorithm efficiency and directly affecting the accuracy of mining results.

(3) The minimum support and minimum confidence thresholds are set based on the user's subjective evaluation. There are significant differences in

the rules generated by different users when generating rules, which requires many experimental verifications and it is difficult to mine scientifically normative, valuable rules.

3. Proposed method

3.1 Apriori algorithm based on MapReduce

In implementing the Apriori algorithm, the minimum support is the minimum statistical standard of the item set. The support of the frequent items in the association rules must be greater than or equal to the given minimum support min-sup, and the minimum confidence min-conf is responsible for measuring the association. The reliability of rules is responsible for ensuring the accuracy of association rules, so in practical applications, the choice of minimum support and minimum confidence will affect the final calculation result[28]. For now, in the process of Apriori algorithm generating rules, the settings of minimum support and minimum confidence are mainly artificially set, and it takes many experimental configurations to calculate valuable rules, and different combinations are required. It will also lead to different results. In order to overcome this difficulty, save time and improve the effectiveness of algorithm mining, this paper uses the weight coefficient to set the threshold function of the minimum support degree and the minimum confidence degree as:

$$\begin{cases} f = \sum_{i=1}^{n} \lambda_i \cdot p_i \\ \sum_{i=1}^{2} \lambda_i = 1 \quad \lambda_i > 0 \end{cases} \tag{4}$$

where, P_1 is the parameter of the association rule algorithm to be solved; λ_i is the weight parameter; P_1 and P_2 represent the minimum support and minimum confidence, λ_1 and λ_2 represent the importance of P_1 and P_2, respectively, and the weight coefficient λ_i is responsible for controlling the importance of the two threshold parameters, and then control the final optimisation result. According to the calculation needs, the value space of f and ip is set, the threshold function is solved by the method of ImCSO algorithm optimisation, and the ideal minimum support and minimum confidence are obtained[29].

3.2 Improved chicken swarm optimisation (CSO) algorithm

Chicken swarm optimisation (CSO) is an algorithm proposed in 2014 to simulate the hierarchical system and group behaviour in chicken flocks[30]. The algorithm divides the flock into roosters, hens and chicks. The hens also include mother chickens. The rooster is the best individual in the group and actively seeks food. The hens follow the rooster to find food, and the chicks follow the mother chickens look for food[31,32]. Suppose NR represents the number of roosters, NH represents the number of female chickens, NC represents the number of chicks, and NM represents the number of mother chickens. There are the following rules:

(1) Individual roosters: Compared with roosters with poor fitness, roosters with better fitness are more likely to obtain food. The following conditions can simulate this situation:

$$x_{i,j}(t+1) = x_{i,j} \cdot (1 + Randn(0, \sigma^2)) \tag{5}$$

$$\sigma^2 = \begin{cases} 1 & if\ f_j < FK \\ exp(\dfrac{f_k - f_j}{|f_j| + \varepsilon}) otherwise \end{cases} \quad K \in [1, N], K \neq i \tag{6}$$

where, $Randn(0, \sigma^2)$ gaussian distribution with mean value 0 and standard deviation σ; ε is used to avoid segmentation errors and is the smallest constant in the calculation; K represents the index of another rooster individual.

(2) Individual hens: The more dominant hens have an advantage in competing for food, and the mathematical method is expressed as follows:

$$x_{i,j}(t+1) = x_{i,j}(t) + S_1 \cdot Rand.(x_{r1,}(at) - x_{i,j}(t)) + S_2 \cdot Rand.(x_{r2,j} - x_{i,j}(t)) \tag{7}$$

$$S_1 = exp \frac{f_i - f_{r1}}{ads(f_i) + \varepsilon} \tag{8}$$

$$S_1 = exp((f_{r2} - f_i)) \tag{9}$$

where, $Rand$ [0,1] uniformly distributed random numbers; $r1$ represents the best individual rooster in the group where the $i–th$ hen belongs; $r2$ represents any individual randomly selected from the flock except chicks and $r_1 \neq r_2$.

(3) Individual chicks: The chicks move around the mother chicken to find food. The rules are as follows:

$$x_{i,j}(t+1) = x_{i,j}(t) + F\cdot(x_{m,j}(t+1) - x_{i,j}(t)) \tag{10}$$

where, F is a random number between 0 and 2 that specifies how far the chicken and its brood should go in quest of food. To a certain extent, the individual chick does not directly follow the position of the individual rooster. The chicks are easily lost and fall into the local optimum, affecting the algorithm's overall efficiency. To optimise this phenomenon and update Equation (10) as follows:

$$x_{i,j}(t+1) = x_{i,j}(t) + \eta_1(x_{m,j}(t) - x_{i,j}(t)) + \eta_2(x_{r,j}(t) - x_{i,j}(t)) \tag{11}$$

$$\eta_1 \begin{cases} \eta_1 \times \dfrac{t_{max} - t_{now}}{t_{max}}, \eta_1 \leq \eta_{min} \\ \eta_{min}, \quad \eta_1 \leq \eta_{min} \end{cases} \tag{12}$$

$$\eta_2 \begin{cases} \eta_2 \times \dfrac{t_{max} - t_{now}}{t_{max}}, \eta_2 \leq \eta_{max} \\ \eta_{max}, \quad \eta_2 \leq \eta_{max} \end{cases} \tag{13}$$

where, m represents that the chicks in the subgroup follow the corresponding individual mother chickens, r represents the individual of the rooster corresponding to the chicks in the subgroup; η_1 represents the degree factor that chicks learn from mother chickens; η_2 represents the degree factor of the chick learning from the rooster; η_{min} the set minimum value of the influence factor; η_{max} the maximum value of the set influence factor; t_{min} the minimum number of iterations set; t_{max} the maximum number of iterations set; t_{now} represents the current iteration number.

3.3 Implementation of ImCSO-Apriori algorithm based on MapReduce

Bare bones minimum of trust. The CSO algorithm is used to solve the minimal support and the minimum support that best match the calculation requirements; the weight coefficients are designed to confine the two thresholds. The minimal support and minimum confidence threshold functions must be designed before the parallel design of the algorithm can begin. This section focuses on using MapReduce to implement a parallel version of the Apriori algorithm[33,34].

The design of the Apriori algorithm based on MapReduce is a combination of cloud computing and association rule technology[35]. In the code section, the "key, value, key-value" pair is summed and counted to derive the final I support. If support is less than min-sup, a label I as a set of frequently occurring items (and output the processing result as a new key-value pair). Label the set of items that do not satisfy the above relationship as "infrequent items"; label the set of infrequent items as "infrequent itemset Linf"; and remove the set of infrequent itemset Linf from the transaction database. In the end, the confidence of the frequent itemset is calculated, and the strong connection rule is established by confidence, with the item defined as a "non-frequent item". In the case where confidence is more than the minimum confidence threshold, the rule is considered strong, and the results are output[36].

CSO is used to restrict the "frequent item sets", which determines the minimum support degree. The Apriori algorithm's Map and Reduce operations are developed, compiled, and installed on the computer cluster. In the first processing phase, the algorithm sorts and calculates the Map function to produce a candidate item set and then constructs the I, sup(I)l > format based on that set. **Figure 3** depicts the algorithm's flowchart.

4. Result and discussion

This paper uses cloud computing to parallelise Apriori for the massive scale of power plant data. In the experiment, a large number of different data sets are considered to verify the performance of the optimised algorithm. The experimental data set includes real data sets and synthetic data sets. The experiment aims to verify whether the Apriori algorithm optimised in this paper can have sufficient capabilities. To deal with large-scale data sets and verify whether the algorithm can improve the mining speed in frequent mining item sets. During the experiment, we

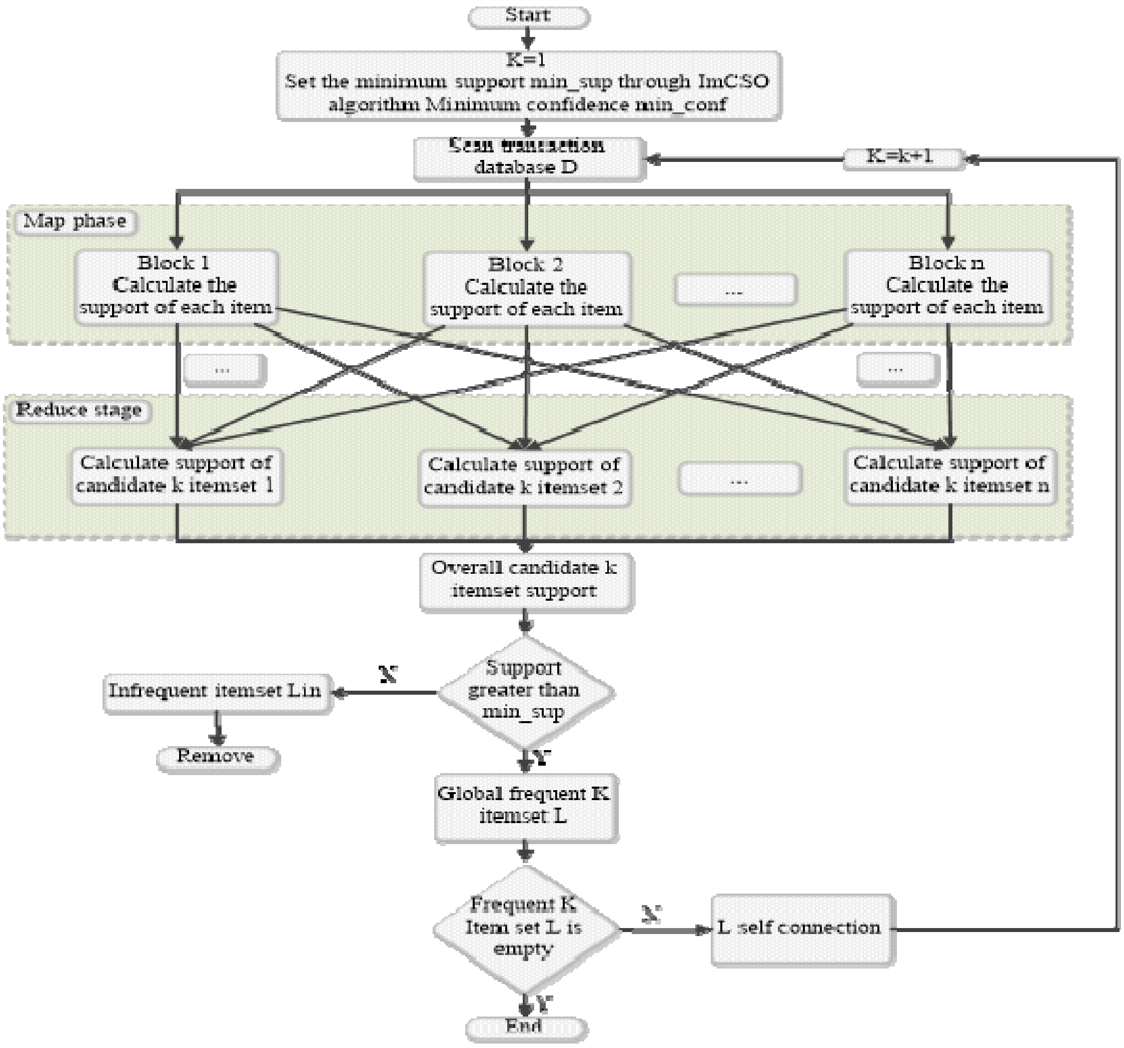


Figure 3. Flowchart of the MapReduce-based ImCSO-Apriori algorithm.

will mine 200,000, 2,000,000, and 20,000,000 relevant transaction records, with 200,000 coming directly from the network and the remaining 200,000 synthesised from self-compiled data.

Case 1: The number of distinct elements in a data collection comprising 200,000 transaction records might range from eight to twenty. The original Apriori technique and the revised algorithm in this study are used to mine frequent item sets in the data set correspondingly. When mining a set of 200,000 transaction records, it was discovered that there were anywhere from eight to twenty unique things. In **Figure 4**, the I-Apriori-MR curve depicts the optimised algorithm, and the Apriori curve represents the original approach. The original Apriori approach for frequent mining item sets has a shorter running time than the optimisation algorithm based on MapReduce when the number of single items is kept constant. The execution times of both algorithms increase exponentially. However, the optimisation method does so more slowly than the original Apriori. Both algorithms become more time-consuming to run as the number of things increases; however, once the number of items reaches 16, both algorithms speed up significantly and become much more efficient.

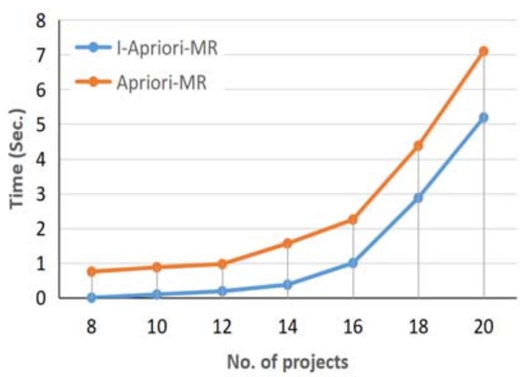
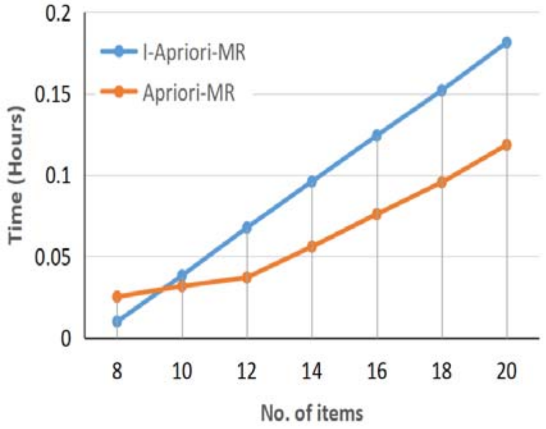
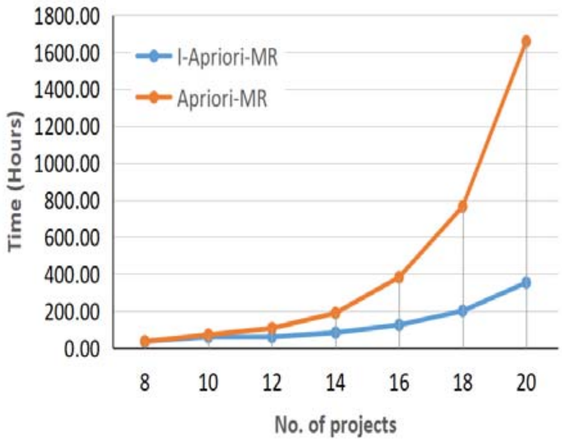


Figure 4. Comparison of mining time for frequent item sets of 5.2 million of data.



a) The y-axis scale is logarithmic.



b) The y-axis scale is linear.

Figure 5. Comparison of mining time of frequent item of 62 million of data.

Case 2: In this experiment, we utilise the original Apriori technique and the revised algorithm to mine frequent item sets in a dataset of 2,000,000 transaction records, where the number of individual items ranges from 0 to 20. In this paper, the Apriori curve represents the original approach, while the I-Apriori-MR curve represents the optimised version (see **Figure 5**). It is clear that the original algorithm's mining performance improves when the number of items in the data set is less than 10 and that it degrades once the number of items in the data set rises above 12. An essential portion of the optimisation algorithm's runtime is devoted to mining frequently occurring item sets. It is faster than the original algorithm's computation time, and the effect is more pronounced as the number of unique things increases; both techniques take a significantly longer time as the number of unique items grows, but the latter takes a much longer time when mining frequent items. There is only a modest but steady increase in the optimisation method's running time, and its speed is far higher than that of the original algorithm. Although the original Apriori approach offers advantages in data processing, when the number of individual items is small, the optimisation algorithm in this study is not considerably different from the original in terms of running time. The optimised algorithm in this research increases stability and calculation speed advantages as the number of individual items increases.

Case 3: It was found in this experiment that the processing time of the standard MapReduce-based Apriori technique and this paper to remove non-frequent items was substantially lower while working with a data set comprising 20 million transaction records than it was when using the original algorithm. The Apriori algorithm uses MapReduce to optimise a set. However, this method is slower. The number of things mined from a batch of 20 million data points ranges from 4 to 18, and two algorithms are employed to do it. Furthermore, an example of a standard MapReduce-optimized algorithm, Apriori-MR, can be seen in **Figure 6**. Compared to Apriori-MR, the algorithm optimised in this study has a lower running time when the number of individual items is constant. When there are fewer projects, the benefit is more noticeable; when more projects are added, the running time cost of the algorithm improved in this work shows an exponentially increasing trend compared with the running times of the frequent item sets, and the gap between the two will progressively reduce.

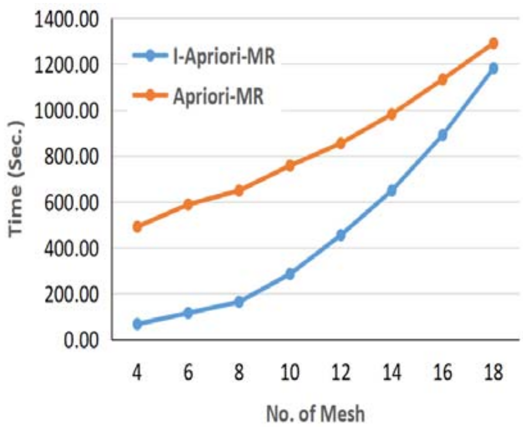


Figure 6. Comparison of mining time for frequent item of 720 million of data.

Case 4: When the transaction data volume is 2 million, and the number of projects is 10, the running time of the ImCSO-Apriori algorithm based on MapReduce parallel optimisation in this paper is compared and analysed under different nodes. **Figure 7** shows that as the number of nodes in the cluster increases, the time required for the optimised Apriori algorithm to calculate in the cluster decreases exponentially, and the more the number of nodes in the cluster, the faster the processing speed of the algorithm. In data processing with a certain amount of data, when the number of nodes increases to a certain level, the increase in computing speed decreases. Therefore, according to different data scales and computing requirements, the number of nodes in the cluster can be increased to improve clustering.

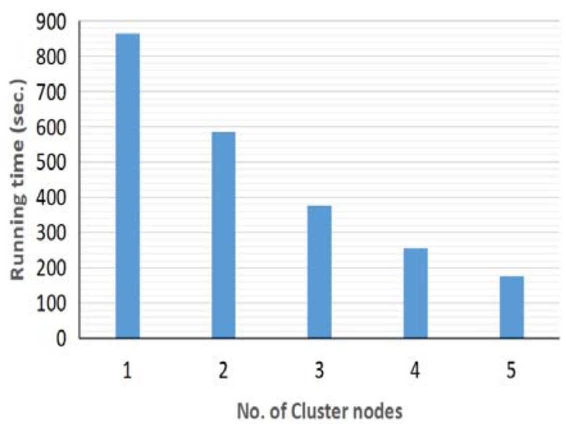


Figure 7. Comparison of mining time of frequent under m different nodes.

5. Conclusion

In this paper, the Apriori algorithm is designed in parallel using MapReduce technology, and a technical means of "determining infrequent items and removing infrequent itemsets" is proposed, reducing redundancy and improving the processing speed of the algorithm. The processing speed and significant data processing capability of the ImCSO-Apriori algorithm are based on MapReduce parallel optimisation. The optimised algorithm is applied to the correlation analysis of transmission line faults, and the correlation diagnosis of transmission line faults is realised. The following conclusions were drawn from the experiments:

(1) Since the cloud computing cluster requires some time during the parallel assignment allocation procedure, the standard Apriori technique offers more significant benefits in the case of low data volume than the algorithm developed in this paper provides less time.

(2) For big datasets, the classical Apriori technique has a little running time when doing frequent item mining for a small number of project individuals, but it is not far behind the optimisation algorithm presented in this study in mining speed. The enhanced algorithm shown in this paper processes data more reliably, takes less time to calculate and completes projects more quickly when dealing with large numbers of project participants.

(3) The classic Apriori technique is unable to execute reliable mining on a huge data set, but the approach of deleting infrequent item sets based on MapReduce has additional advantages, a shorter running time for the algorithm and higher throughput. However, it is possible to cause too many infrequent itemsets when the number of items is too big. Still, there are substantial processing needs when the occurrence-infrequent item sets are excessively massive. This paper's optimised technique is well-suited for power plant fault data analysis mining and analysis due to its ability to process and analyse massive amounts of power plant data effectively.

Conflict of interest

The authors declare no conflict of interest.

References

1. Li X, Liu H, Wang W, *et al*. Big data analysis of the internet of things in the digital twins of smart city based on deep learning. Future Generation Computer Systems 2022; 128: 167–177. doi: 10.1016/j.future.2021.10.006.
2. Liu R. Research on big data acquisition and application of power energy based on big data cloud platform. Proceedings of the SPIE 2022; 12168: 1216834. doi: 10.1117/12.2630992.
3. Paukstadt U, Becker J. Uncovering the business value of the internet of things in the energy domain—A review of smart energy business models. Electronic Markets 2019; 31: 51–66. doi: 10.1007/s12525-019-00381-8.
4. Al-Jumaili AHA, Al Mashhadany YI, Sulaiman R, Alyasseri ZAA. A conceptual and systematics for

intelligent power management system-based cloud computing: Prospects, and challenges. Applied Sciences 2021; 11(21): 9820. doi: 10.3390/APP11219820.
5. Ahmad T, Madonski R, Zhang D, *et al*. Data-driven probabilistic machine learning in sustainable smart energy/smart energy systems: Key developments, challenges, and future research opportunities in the context of smart grid paradigm. Renewable and Sustainable Energy Reviews 2022; 160: 112128. doi: 10.1016/j.rser.2022.112128.
6. Seeger PM, Yahouni Z, Alpan G. Literature review on using data mining in production planning and scheduling within the context of cyber physical systems. Journal of Industrial Information Integration 2022; 28: 100371. doi: 10.1016/j.jii.2022.100371.
7. Ren D, Hao X, Zhao Y, Wang M. Research on power big data analysis technology and application based on cloud computing. Lecture Notes in Electrical Engineering 2022; 827: 1117–1123. doi: 10.1007/978-981-16-8052-6_154.
8. Gui P, Zhang M. Human resource management of energy companies based on big data analysis. Mathematical Problems in Engineering 2022; 2022: 5489369. doi: 10.1155/2022/5489369.
9. Al-Nuaymy AI, Hamed WA, Ali AH. The role of social media in enhancing the learning process of Iraq students. Journal of Engineering and Applied Sciences 2019; 14(19): 7142–7153. doi: 10.36478/JEASCI.2019.7142.7153.
10. Ali AH, Mohammad OKJ. Impacting of the e-platforms on the 4.0th industrial educational revolution. In: Proceedings of ICIST 2019: The 9th International Conference on Information Systems and Technologies; 2019 Mar 24–26; Cairo. New York: Association for Computing Machinery; 2019.
11. Ji B, Wang Y, Song K, *et al*. A survey of computational intelligence for 6G: Key technologies, applications and trends. IEEE Transactions on Industrial Informatics 2021; 17(10): 7145–7154. doi: 10.1109/TII.2021.3052531.
12. Yang C, Huang Q, Li Z, *et al*. Big data and cloud computing: Innovation opportunities and challenges. International Journal of Digital Earth 2017; 10(1): 13–53. doi: 10.1080/17538947.2016.1239771.
13. Qin Y, Yang W, Ai X, Chen L. Fault tolerant storage and data access optimization in data center networks. Journal of Network and Computer Applications 2018; 113: 109–118. doi: 10.1016/j.jnca.2018.04.001.
14. Wang S, Zhong Y, Wang E. An integrated GIS platform architecture for spatiotemporal big data. Future Generation Computer Systems 2019; 94: 160–172. doi: 10.1016/j.future.2018.10.034.
15. Madni SHHH, Latiff MSA, Coulibaly Y, Abdulhamid SM. Recent advancements in resource allocation techniques for cloud computing environment: A systematic review. Cluster Computing 2017; 20(3): 2489–2533. doi: 10.1007/s10586-016-0684-4.
16. Liu Y, Liang S, He C, *et al*. A cloud-computing and big data based wide area monitoring of power grids strategy. IOP Conference Series: Materials Science and Engineering 2019; 677(4): 042055. doi: 10.1088/1757-899X/677/4/042055.
17. Cheng Y, Chen K, Sun H, *et al*. Data and knowledge mining with big data towards smart production. Journal of Industrial Information Integration 2018; 9: 1–13. doi: 10.1016/j.jii.2017.08.001.
18. Zhan F, Zhu X, Zhang L, *et al*. Summary of association rules. IOP Conference Series: Earth and Environmental Science 2019; 252: 032219. doi: 10.1088/1755-1315/252/3/032219.
19. Tan Q. Construction of multidimensional data knowledge base by improved classification association rule mining algorithm. Advances in Computer Science Research 2017; 59: 901–905. doi: 10.2991/emcm-16.2017.171.
20. Ye Y. Research on Apriori algorithm and its application in electronic commerce system. Advances in Social Science, Education and Humanities Research 2016; 63: 161–166. doi: 10.2991/amahs-16.2016.33.
21. Guo Z, Chen T. Improvement of Apriori algorithm based on vector and vertical array. In: Proceedings of 2016 International Conference on Control and Automation (ICCA 2016); 2016 Jun 1–3; Kathmandu. Lancaster: DEStech Transactions on Engineering and Technology Research; 2016.
22. Qiu H. Improvement parallelization in Apriori algorithm. In: Proceedings of 2020 International Conference on Computers, Information Processing and Advanced Education (CIPAE); 2020 Oct 16–18; Ottawa. New York: Association for Computing Machinery; 2020. p. 235–238.
23. Wang L, Meng J, Xu P, Peng K. Mining temporal association rules with frequent itemsets tree. Applied Soft Computing 2018; 62: 817–829. doi: 10.1016/j.asoc.2017.09.013.
24. Du Z. Energy analysis of internet of things data mining algorithm for smart green communication networks. Computer Communications 2020; 152: 223–231. doi: 10.1016/j.comcom.2020.01.046.
25. Yang Q, Fu Q, Wang C, Yang J. A matrix-based Apriori algorithm improvement. In: Proceedings of 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC); 2018 Jun 18–21; Guangzhou. New York: IEEE; 2018. p. 824–828.
26. Wang H, Gao Y. Research on parallelization of Apriori algorithm in association rule mining. Procedia Computer Science 2021; 183: 641–647. doi: 10.1016/j.procs.2021.02.109.
27. Wang C, Zheng X. Application of improved time series Apriori algorithm by frequent itemsets in association rule data mining based on temporal constraint. Evolutionary Intelligence 2020; 13(1): 39–49. doi: 10.1007/s12065-019-00234-5.
28. Feng D, Zhu L, Zhang L. Research on improved Apriori algorithm based on MapReduce and HBase. In: Proceedings of 2016 IEEE Advanced

Information Management, Communicates, Electronic and Automation Control Conference (IMCEC); 2016 Oct 3–5; Xi'an. New York: IEEE; 2017. p. 887–891.
29. Singh S, Garg R, Mishra PK. Review of Apriori based algorithms on map reduce framework. In: Proceedings of International Conference on Communication and Computing (ICC-2014); 2014 Jun 12–14; Bangalore. Amsterdam: Elsevier; 2014. p. 593–604.
30. Wang J, Zhang F, Liu H, *et al*. Interruptible load scheduling model based on an improved chicken swarm optimization algorithm. CSEE Journal of Power and Energy Systems 2021; 7(2): 232–240. doi: 10.17775/CSEEJPES.2020.01150.
31. Wu D, Xu S, Kong F. Convergence analysis and improvement of the chicken swarm optimization algorithm. IEEE Access 2016; 4: 9400–9412. doi: 10.1109/ACCESS.2016.2604738.
32. Deb S, Gao XZ, Tammi K, *et al*. Recent studies on chicken swarm optimization algorithm: A review (2014–2018). Artificial Intelligence Review 2020; 53(3): 1737–1765. doi: 10.1007/s10462-019-09718-3.
33. Sathyaraj R, Ramanathan L, Lavanya K, *et al*. Chicken swarm foraging algorithm for big data classification using the deep belief network classifier. Data Technologies and Applications 2020; 55(3): 332–352. doi: 10.1108/DTA-08-2019-0146.
34. Vinoth R, Ananth JP. Rider chicken optimization algorithm-based recurrent neural network for big data classification in spark architecture. The Computer Journal 2022; 65(8): 2183–2196. doi: 10.1093/comjnl/bxab053.
35. Liang X, Lv H, Guo F, Wang H. A cloud security situational awareness model based on parallel Apriori algorithm. Applied Mechanics and Materials 2014; 556–562: 6294–6297. doi: 10.4028/www.scientific.net/AMM.556-562.6294.
36. Singh S, Garg R, Mishra PK. Performance optimization of MapReduce-based Apriori algorithm on hadoop cluster. Computers & Electrical Engineering 2018; 67: 348–364. doi: 10.1016/j.compeleceng.2017.10.008.