IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received January 31, 2021, accepted February 9, 2021, date of publication February 12, 2021, date of current version February 23, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3059221

A Novel Bearing Fault Diagnosis Method Using Spark-Based Parallel ACO-K-Means Clustering Algorithm

LANJUN WAN[1,2], GEN ZHANG[1,2], HONGYANG LI[1,2], AND CHANGYUN LI[2]

[1] School of Computer Science, Hunan University of Technology, Zhuzhou 412007, China
[2] Hunan Key Laboratory of Intelligent Information Perception and Processing Technology, Hunan University of Technology, Zhuzhou 412007, China

Corresponding author: Lanjun Wan (wanlanjun@hut.edu.cn)

This work was supported in part by the National Natural Science Foundation for Young Scientists of China under Grant 61702177, in part by the Natural Science Foundation of Hunan Province, China under Grant 2019JJ60048, in part by the National Key Research and Development Project under Grant 2018YFB1700204 and Grant 2018YFB1003401, and in part by the Key Research and Development Project of Hunan Province under Grant 2019GK2133.

ABSTRACT K-Means clustering algorithm is a typical unsupervised learning method, and it has been widely used in the field of fault diagnosis. However, the traditional serial K-Means clustering algorithm is difficult to efficiently and accurately perform clustering analysis on the massive running-state monitoring data of rolling bearing. Therefore, a novel fault diagnosis method of rolling bearing using Spark-based parallel ant colony optimization (ACO)-K-Means clustering algorithm is proposed. Firstly, a Spark-based three-layer wavelet packet decomposition approach is developed to efficiently preprocess the running-state monitoring data to obtain eigenvectors, which are stored in Hadoop Distributed File System (HDFS) and served as the input of ACO-K-Means clustering algorithm. Secondly, ACO-K-Means clustering algorithm suitable for rolling bearing fault diagnosis is proposed to improve the diagnosis accuracy. ACO algorithm is adopted to obtain the global optimal initial clustering centers of K-Means from all eigenvectors, and the K-Means clustering algorithm based on weighted Euclidean distance is used to perform clustering analysis on all eigenvectors to obtain a rolling bearing fault diagnosis model. Thirdly, the efficient parallelization of ACO-K-Means clustering algorithm is implemented on a Spark platform, which can make full use of the computing resources of a cluster to efficiently process large-scale rolling bearing datasets in parallel. Extensive experiments are conducted to verify the effectiveness of the proposed fault diagnosis method. Experimental results show that the proposed method can not only achieve good fault diagnosis accuracy but also provide high model training efficiency and fault diagnosis efficiency in a big data environment.

INDEX TERMS Ant colony optimization, fault diagnosis, K-Means clustering, rolling bearing, spark, wavelet packet decomposition.

I. INTRODUCTION

Rolling bearing is one of the most commonly used and easily damaged components of rotating machinery equipment, and rolling bearing fault diagnosis is very important to ensure the normal running of rotating machinery equipment [1]. In the field of rolling bearing fault diagnosis, the common sensors are only used to collect vibration signals, and they don't have the ability to do rolling bearing fault diagnosis. Recently, some intelligent sensors with fault detection function have emerged, they can employ some simple signal processing technologies to detect the most simple and obvious fault signals of rolling bearing, but they don't have the ability to identify complex and illegible fault signals yet. Typically, the working conditions of rolling bearing are complex, the vibration signals collected by sensors are often non-stationary, non-linear, and multi-component, thus the signal processing technologies are difficult to be used to carry out complex fault diagnosis of rolling bearing effectively and accurately.

The data-driven intelligent fault diagnosis methods based on machine learning or deep learning can fully dig the underlying fault feature information from the massive and complex vibration signals of rolling bearing, thus they are

The associate editor coordinating the review of this manuscript and approving it for publication was Massimo Cafaro.

suitable for complex fault diagnosis of rolling bearing. In recent years, more and more researches have focused on the data-driven rolling bearing fault diagnosis, such as random forest (RF) [2], k-nearest neighbor [3], support vector machine [4], back propagation neural network [5], improved LeNet-5 network [6], deep convolutional neural network [7]–[9], deep recurrent neural network [10], deep residual learning [11], deep auto-encoder [12], and stacked sparse auto-encoder [13].

Most of existing data-driven fault diagnosis methods can get a satisfactory diagnosis accuracy with sufficient supervised training samples, but the process of labeling large-scale training samples is time-consuming and labor-intensive in practical industrial production. Compared with the traditional machine learning methods, the deep learning methods can obtain a higher fault diagnosis accuracy, but its complex network structure will have a greater impact on the training speed in the training of massive samples. Clustering algorithms are typical unsupervised learning methods which do not require labeling the training samples and have lower computational complexity, thus they are suitable for rolling bearing fault diagnosis in the big data environment.

Recently, fuzzy C-means (FCM) clustering algorithm and K-Means clustering algorithm have been widely used in fault diagnosis. Bai *et al.* [14] proposed a fault diagnosis method based on empirical wavelet transform and FCM clustering algorithm. In [15], [16], the empirical mode decomposition and FCM clustering algorithm are combined and applied to fault diagnosis. Ramos *et al.* [17] designed a fault diagnosis system of steam generator using FCM clustering algorithm. Liu *et al.* [18] proposed a fault diagnosis method based on Gaussian kernel FCM clustering algorithm. Shi *et al.* [19] developed a fault diagnosis method based on local mean decomposition (LMD) and K-Means clustering algorithm, LMD is used to decompose the vibration signals of rolling bearing, the probability density function is utilized to optimize the selection of initial clustering centers of K-Means, and the optimized K-Means clustering algorithm is adopted to effectively diagnose rolling bearing faults. Zhang *et al.* [20] improved the choice method of initial clustering centers of K-Means, and the results show that the fault diagnosis accuracy of rolling bearing obtained using the modified K-Means clustering algorithm is increased by 7.5% than that obtained using the traditional K-Means clustering algorithm. Mjahed *et al.* [21] proposed an engine fault diagnosis method based on genetic algorithm (GA) and K-Means clustering algorithm, and a novel engine fault diagnosis method based on particle swarm optimization (PSO) algorithm and K-Means clustering algorithm was devised in [22], which can effectively identify different kinds of faults in engines. In [21], [22], both GA and PSO algorithm are exploited to improve the random initialization of K-Means clustering algorithm for engine fault diagnosis. Compared with FCM clustering algorithm, K-Means clustering algorithm has lower computational complexity and faster convergence speed, thus it is more suitable for the clustering analysis of big data.

With the increase of the complexity of mechanical equipment and the expansion of industrial production scale, and multiple sensors are used to monitor the running states of mechanical equipment in real time, the running-state monitoring data generated by a large number of mechanical equipment are growing rapidly. It is a new challenge of the field of fault diagnosis that how to diagnose mechanical equipment faults accurately and quickly according to the massive running-state monitoring data [23]. In the last few years, the researches and applications of fault diagnosis based on big data technology are increasing gradually. For example, Miao *et al.* [24] built a fault diagnosis model of SF6 electrical equipment using back propagation neural network based on MapReduce. Imani *et al.* [25] adopted RF based on Spark to rapidly diagnose wind turbine gearbox faults. Yu *et al.* [26] built a fault diagnosis platform of industrial equipment using MapR-DB, Hive, MapReduce, Spark, principal component analysis, and other technologies. Most of the existing researches on fault diagnosis based on big data technology apply the parallel machine learning algorithms based on MapReduce or Spark to fault diagnosis, which improve the performance of fault diagnosis. Compared with MapReduce, Spark has faster processing speed and is more suitable for machine learning algorithms that require a large number of iterative computations.

In view of the obvious advantages of Spark and K-Means clustering algorithm in industrial big data analysis, this paper proposes a novel fault diagnosis method of rolling bearing using Spark-based parallel ACO-K-Means clustering algorithm, which can effectively diagnose rolling bearing faults through rapid and accurate mining of fault information from the massive running-state monitoring data of rolling bearing.

The main contributions of the proposed approach are as follows.

- A Spark-based three-layer wavelet packet decomposition approach is developed, which can efficiently preprocess the massive running-state monitoring data of rolling bearing to obtain eigenvectors as the input of ACO-K-Means clustering algorithm.
- ACO-K-Means clustering algorithm suitable for rolling bearing fault diagnosis is proposed to improve the diagnosis accuracy. ACO algorithm is used to get the global optimal initial clustering centers of K-Means from all eigenvectors, and the K-Means clustering algorithm based on weighted Euclidean distance is used to perform clustering analysis on all eigenvectors to obtain a rolling bearing fault diagnosis model.
- The parallelization of ACO-K-Means clustering algorithm for rolling bearing fault diagnosis is implemented on a Spark platform, which can efficiently and accurately perform clustering analysis on the massive running-state monitoring data of rolling bearing.

- The effectiveness of the proposed fault diagnosis method is verified and analyzed through a series of experiments, the results show that it cannot only obtain a satisfactory fault diagnosis accuracy, but also offer a higher model training efficiency and fault diagnosis efficiency for large-scale rolling bearing datasets.

The rest of the paper is organized as follows. Section II outlines ACO algorithm, K-Means clustering algorithm, and Spark computing framework. Section III describes the proposed fault diagnosis method of rolling bearing using Spark-based parallel ACO-K-Means clustering algorithm. Section IV presents experimental results and analysis. Section V gives the conclusion.

II. BACKGROUND

A. OVERVIEW OF ACO ALGORITHM

The inspiration of ACO algorithm [27] comes from the observation of foraging behavior of real ant colonies. During the round trip from the food source to the ant nest, each ant secretes a chemical substance called pheromone on its path, and it can perceive the existence and intensity of pheromones and move along the direction with higher intensity pheromones. Thus the entire ant colony can find the shortest path between the food source and the ant nest after a period of time.

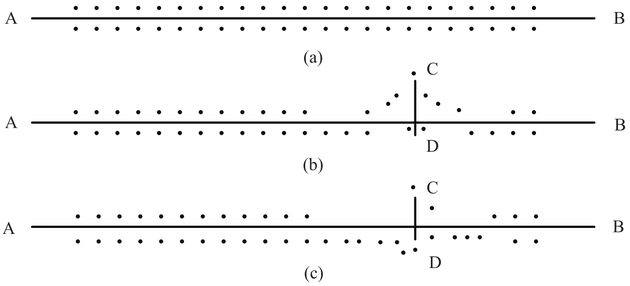

FIGURE 1. An example of ants finding the shortest path.

Suppose that an ant wants to move from point A to point B, if there are no obstacles, it will move along path AB, as shown in Fig. 1(a). If there are obstacles, it will randomly choose a path between paths ACB and ADB, as shown in Fig. 1(b). Since path ADB is shorter than path ACB and there are more ants passing through point D between points A and B, resulting in the intensity of pheromones deposited on path ADB is greater than that of pheromones deposited on path ACB, and more ants choose path ADB, as shown in Fig. 1(c).

In the ACO algorithm, m ants can cooperate to perform the foraging task, and the transition probability P_{ij}^k from point i to point j for the k-th ant can be calculated by the pheromone intensity $\tau_{ij}(t)$ and visibility $\eta_{ij}(t)$ at time t, as in

$$P_{ij}^k(t) = \frac{(\tau_{ij}(t))^\alpha(\eta_{ij}(t))^\beta}{\sum_{s\in allowed_k}(\tau_{is}(t))^\alpha(\eta_{is}(t))^\beta}, \tag{1}$$

where α is the pheromone heuristic factor, β is the expected heuristic factor, and $allowed_k$ is the next arrival point that the k-th ant can choose. The visibility η_{ij} can be calculated by

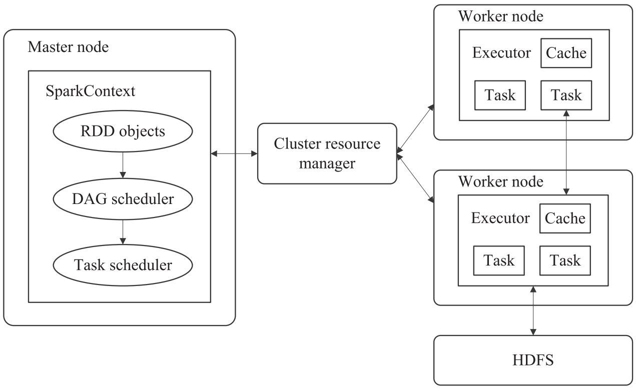

FIGURE 2. Spark computing framework.

$$\eta_{ij} = \frac{1}{\varepsilon_{ij}}, \tag{2}$$

where ε_{ij} represents the Euclidean distance between point i and point j. The pheromone intensity τ_{ij} at time $t+n$ can be calculated by

$$\tau_{ij}(t+n) = (1-\rho)\tau_{ij}(t) + \sum_{k=1}^{m}\Delta\tau_{ij}^k, \tag{3}$$

where ρ is the pheromone volatilization factor ($0<\rho<1$), and $\Delta\tau_{ij}$ represents the pheromone increment. $\Delta\tau_{ij}^k$ denotes the pheromone laid on path (i,j) by the k-th ant, which can be calculated by

$$\Delta\tau_{ij}^k = \begin{cases} \frac{1}{l_k}, & \text{if the } k\text{-th ant passes the path } (i, j), \\ 0, & \text{otherwise,} \end{cases} \tag{4}$$

where l_k represents the length of the path the k-th ant passes between time t and time $t+n$.

B. OVERVIEW OF K-MEANS CLUSTERING ALGORITHM

K-Means clustering algorithm [28] is one of the most classic clustering algorithms, which uses distance as the evaluation index of similarity, that is, the closer the distance between two objects is, the greater the similarity is. K-Means clustering algorithm is an iterative clustering analysis method, the distance between any two samples is calculated for a given sample set, and the sample set is divided into k clusters according to the distances between samples, mainly including the following steps.

Step 1: Randomly select k samples from a given sample set $X = \{x_1, x_2, \ldots, x_n\}$ as the initial clustering centers $M = \{\mu_1, \mu_2, \ldots, \mu_k\}$.

Step 2: Calculate the Euclidean distance between each sample and each clustering center by

$$d_{ij} = \left\|x_i - \mu_j\right\|_2, \tag{5}$$

and each sample is classified into the nearest cluster.

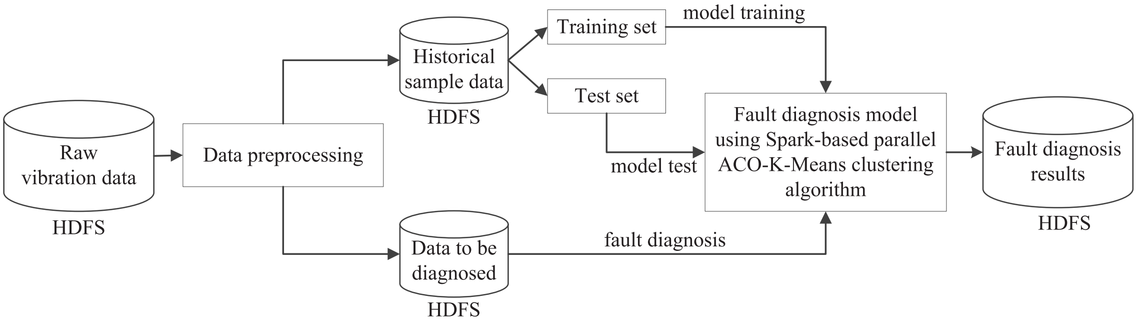

FIGURE 3. Flowchart of the proposed fault diagnosis method of rolling bearing using Spark-based parallel ACO-K-Means clustering algorithm.

Step 3: Recalculate the centroids of k clusters by

$$\mu_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i, \tag{6}$$

where C_j represents the j-th cluster.

Step 4: Calculate the total mean square error between all samples and their corresponding clustering centers by

$$MSE = \frac{1}{n} \sum_{j=1}^{k} \sum_{x_i \in C_j} \left\| x_i - \mu_j \right\|_2^2, \tag{7}$$

if it reaches the convergence threshold or the maximum number of iterations has reached, go to Step 5; otherwise, go to Step 2.

Step 5: Output the clustering results.

C. OVERVIEW OF SPARK COMPUTING FRAMEWORK

Spark is one of the most commonly used big data computing platforms, it is a parallel computing framework for big data based on memory computing, which can be used to build faster and more efficient big data analysis applications [29]. Fig. 2 depicts the Spark computing framework, which mainly includes a cluster resource manager, a master node and multiple worker nodes that perform tasks. The cluster resource manager can be the Spark's own standalone cluster manager, YARN [30], or other resource management frameworks.

When a Spark application is submitted, a SparkContext object will be created on the master node, it reads data from HDFS to create RDD objects, and it applies for resources from the cluster resource manager. The cluster resource manager allocates resources to one or more executor processes of each worker node, and each worker node reports resource usage and running states of tasks to the cluster resource manager using a heartbeat mechanism. The SparkContext object builds a directed acyclic graph (DAG) according to the dependencies among multiple RDDs, and the DAG is submitted to the DAG scheduler. The DAG scheduler parses the DAG into multiple task sets, which are submitted to the task scheduler. The task scheduler assigns tasks to executor processes, after an executor process receives a task, a thread will be taken from the thread pool of the executor process to perform the task.

III. PROPOSED FAULT DIAGNOSIS METHOD OF ROLLING BEARING

A. FAULT DIAGNOSIS PROCESS OF ROLLING BEARING

The proposed fault diagnosis method of rolling bearing using Spark-based parallel ACO-K-Means clustering algorithm is depicted in Fig. 3. Firstly, the raw vibration data of rolling bearing collected by sensors in real time (i.e., the running-state monitoring data of rolling bearing) are stored in HDFS. Secondly, Spark-based three-layer wavelet packet decomposition is adopted to preprocess the running-state monitoring data to obtain eigenvectors, which are stored in HDFS and served as the input of ACO-K-Means clustering algorithm. Thirdly, the historical sample data composed of eigenvectors are randomly divided into training set and test set, and Spark-based parallel ACO-K-Means clustering algorithm is performed to train and test the fault diagnosis model of rolling bearing. Finally, the trained fault diagnosis model is used in actual fault diagnosis, the clustering analysis is carried out on the data to be diagnosed which are composed of eigenvectors stored in HDFS, and the fault diagnosis results are output.

B. PROPOSED SPARK-BASED THREE-LAYER WAVELET PACKET DECOMPOSITION APPROACH

In the preprocessing of the raw vibration data of rolling bearing, the wavelet packet decomposition [31] is often used to extract eigenvectors of rolling bearing. In order to efficiently preprocess the massive vibration data of rolling bearing, a Spark-based three-layer wavelet packet decomposition approach is proposed. Fig. 4 presents the flowchart of the proposed Spark-based three-layer wavelet packet decomposition approach, which mainly includes the following steps.

Step 1: Read the raw vibration data of rolling bearing to create an RDD. w pieces of raw vibration data of rolling bearing are read from HDFS to create an RDD *rawRDD* containing n partitions, and each RDD partition contains w/n pieces of vibration data of rolling bearing.

Step 2: Divide samples. For the s-th RDD partition $= \{R_{(s-1)w/n+1}, R_{(s-1)w/n+2}, \ldots, R_{s*w/n}\}$ of *rawRDD*, every l pieces of continuous vibration data of rolling bearing are divided into a sample, where $1 \leq s \leq n$. The sample size l should be more than the number of vibration signals collected

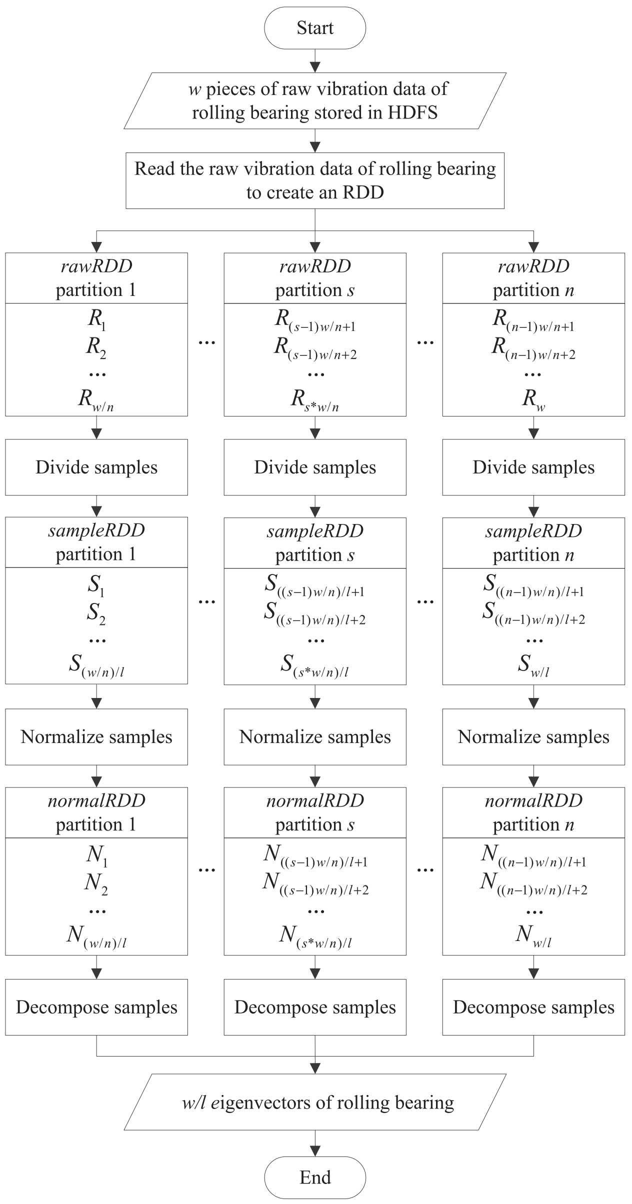

FIGURE 4. Flowchart of the proposed Spark-based three-layer wavelet packet decomposition approach.

in one rotation cycle of rolling bearing, but not more than that collected in two rotation cycles of rolling bearing. A new RDD *sampleRDD* containing w/l samples is obtained after finishing the sample division.

Step 3: Normalize samples. For the s-th RDD partition = $\{S_{((s-1)w/n)/l+1}, S_{((s-1)w/n)/l+2}, \ldots, S_{(s*w/n)/l}\}$ of *sampleRDD*, each piece of vibration data contained in the sample $S_{((s-1)w/n)/l+j}$ is normalized by

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}}, \tag{8}$$

where $1 \leq s \leq n$, $1 \leq j \leq (w/n)/l$, x is one piece of vibration data contained in the sample, $x_{\max}$ is the maximum value, and $x_{\min}$ is the minimum value. A new RDD *normalRDD* is obtained after all samples are normalized.

Step 4: Decompose samples. For the s-th RDD partition = $\{N_{((s-1)w/n)/l+1}, N_{((s-1)w/n)/l+2}, \ldots, N_{(s*w/n)/l}\}$ of *normalRDD*, the sample $N_{((s-1)w/n)/l+j}$ is decomposed by three-layer wavelet packet decomposition to obtain wavelet packet decomposition coefficients of eight frequency bands, where $1 \leq s \leq n$ and $1 \leq j \leq (w/n)/l$. The energy proportion E_k of the k-th frequency band can be calculated by

$$E_k = \frac{(W_k)^T W_k}{\sum_{k=1}^{8} (W_k)^T W_k}, \tag{9}$$

where W_k is the wavelet packet decomposition coefficient of the k-th frequency band, for all $1 \leq k \leq 8$. Since the energy proportion of each frequency band obtained from the decomposition of vibration data corresponding to different running states of rolling bearing is different, each eigenvector of rolling bearing can be constructed from energy proportions of eight different frequency bands. Finally, w/l eigenvectors are obtained after all samples are decomposed.

TABLE 1. Examples of eigenvectors of different running states of rolling bearing.

Frequency Band	Energy Proportions of Different Frequency Bands			
	Normal State	Ball Fault	Inner Race Fault	Outer Race Fault
FB 1	0.3027	0.0323	0.0387	0.1034
FB 2	0.5843	0.0283	0.1042	0.0622
FB 3	0.0104	0.3257	0.3324	0.1764
FB 4	0.0961	0.0144	0.0828	0.0407
FB 5	0.0002	0.0007	0.0016	0.0158
FB 6	0.0013	0.0031	0.0046	0.0342
FB 7	0.0016	0.5772	0.3531	0.4032
FB 8	0.0034	0.0183	0.0826	0.1641

C. PROPOSED ACO-K-MEANS CLUSTERING ALGORITHM

1) K-MEANS CLUSTERING ALGORITHM BASED ON WEIGHTED EUCLIDEAN DISTANCE

As described in Section III-B, each eigenvector of rolling bearing is composed of energy proportions of eight different frequency bands, and the examples of eigenvectors of four different running states of rolling bearing are presented in Table 1. As can be seen in Table 1, the energy proportions of different frequency bands in eigenvectors of different running states of rolling bearing are different. For example, the energy distributions of the ball fault of rolling bearing are mainly concentrated in the third and seventh frequency bands. If the Euclidean distance measure shown in (5) is adopted to calculate the distance between each eigenvector and each clustering center, the differences of the energy distributions of different running states of rolling bearing are neglected, which may affect the fault diagnosis accuracy. Therefore, the K-Means clustering algorithm based on weighted Euclidean distance is put forward.

For any two samples $x_i = \{x_{i1}, x_{i2}, \ldots, x_{ip}\}$ and $x_j = \{x_{j1}, x_{j2}, \ldots, x_{jp}\}$, the weighted Euclidean distance between

x_i and x_j can be calculated by

$$d_{ij} = \sqrt{\sum_{k=1}^{p} \left(\frac{x_{ik} - x_{jk}}{\delta_k} \right)^2}, \tag{10}$$

where δ_k is the standard deviation of $\{x_{ik}, x_{jk}\}$, for all $1 \leq k \leq p$.

To compare the fault diagnosis accuracy of the K-Means clustering algorithm using weighted Euclidean distance with that of the K-Means clustering algorithm using Euclidean distance, the raw vibration data of rolling bearing [32] provided by Case Western Reserve University (CWRU) are adopted, and the three-layer wavelet packet decomposition is performed to decompose the raw vibration data to obtain eigenvectors as the input of K-Means clustering algorithm. Table 2 presents the fault diagnosis accuracies of K-Means clustering algorithms using different distance measure methods. It can be seen from Table 2 that the fault diagnosis accuracy of the K-Means clustering algorithm using weighted Euclidean distance is 0.81% higher than that of the K-Means clustering algorithm using Euclidean distance. Therefore, the weighted Euclidean distance measure is adopted to calculate the distance between each eigenvector and each clustering center, which can improve the fault diagnosis accuracy of K-Means clustering algorithm to a certain extent.

TABLE 2. Fault diagnosis accuracies of K-Means clustering algorithms using different distance measure methods.

Distance Measure Method	Fault Diagnosis Accuracy
Euclidean distance	91.12%
Weighted Euclidean distance	91.93%

2) ACO-K-MEANS CLUSTERING ALGORITHM

The traditional K-Means clustering algorithm does not guarantee that the global optimal solution can be obtained, and the clustering effect depends on the selection of initial clustering centers. Therefore, many studies [19]–[22], [33], [34] have focused on optimizing the selection of initial clustering centers of K-Means. In this paper, ACO algorithm is used to get the global optimal initial clustering centers of K-Means. The proposed ACO-K-Means clustering algorithm for rolling bearing fault diagnosis is described in Algorithm 1, mainly including the following steps.

Algorithm 1 The ACO-K-Means Clustering Algorithm for Rolling Bearing Fault Diagnosis

```
Input: $m$ eigenvectors of rolling bearing, the number of clusters $k$, the pheromone heuristic factor $\alpha$, the expected heuristic factor $\beta$, the pheromone volatilization factor $\rho$, the maximum number of iterations $maxNumIter$, the convergence thresholds $\lambda$ and $\varphi$
Output: a rolling bearing fault diagnosis model
Randomly select $k$ initial clustering centers $M = \{\mu_1, \mu_2, \ldots, \mu_k\}$;
do
    $t \leftarrow t + 1$;
    for $i \leftarrow 1$ to $m$ do
        for $j \leftarrow 1$ to $k$ do
            Calculate the visibility $\eta_{ij}$ by (2);
            if $t = 1$ then
                The pheromone intensity $\tau_{ij} \leftarrow 1$;
            else
                Calculate the pheromone intensity $\tau_{ij}$ by (3);
            end if
            Calculate the transition probability $P_{ij}^i$ by (1);
        end for
        Assign the $i$-th eigenvector to a cluster according to the maximum transition probability $\max(P_{i1}^i, P_{i2}^i, \ldots, P_{ik}^i)$;
    end for
    Update $k$ initial clustering centers by (6);
    Calculate the mean square error $MSE$ by (7);
while $MSE \leq \lambda$
Output $k$ global optimal initial clustering centers;
do
    for $i \leftarrow 1$ to $m$ do
        for $j \leftarrow 1$ to $k$ do
            Calculate the weighted Euclidean distance $d_{ij}$ by (10);
        end for
        Classify the $i$-th eigenvector into the nearest cluster;
    end for
    Update $k$ clustering centers by (6);
    Calculate the mean square error $MSE$ by (7);
while $MSE \leq \varphi$ or $++numIter = maxNumIter$
return a rolling bearing fault diagnosis model.
```

Step 1: Randomly select the initial clustering centers. k eigenvectors are randomly selected from all m eigenvectors of rolling bearing and served as the initial clustering centers $M = \{\mu_1, \mu_2, \ldots, \mu_k\}$.

Step 2: Calculate the transition probability. The visibility η_{ij} between the i-th eigenvector and the j-th initial clustering center is calculated by (2), the pheromone intensity τ_{ij} between the i-th eigenvector and the j-th initial clustering center is calculated by (3), and the transition probability P_{ij}^i from the i-th eigenvector to the j-th initial clustering center for the i-th ant is calculated by (1), where $1 \leq i \leq m$ and $1 \leq j \leq k$.

Step 3: Assign m eigenvectors to k clusters. The i-th eigenvector is assigned to the j-th cluster according to the maximum transition probability $P_{ij}^i = \max(P_{i1}^i, P_{i2}^i, \ldots, P_{ik}^i)$, where $1 \leq i \leq m$ and $1 \leq j \leq k$.

Step 4: Update the initial clustering centers. k initial clustering centers are recalculated by (6), which are regarded as the new initial clustering centers.

Step 5: Determine whether the initial clustering centers have converged. At first, the total mean square error MSE

between all eigenvectors and their corresponding initial clustering centers is calculated by (7). Then, determine whether MSE is less than or equal to the convergence threshold, if yes, the global optimal initial clustering centers are output and go to Step 6; if not, go to Step 2.

Step 6: Classify each eigenvector into the nearest cluster. The weighted Euclidean distance d_{ij} between the i-th eigenvector and the j-th clustering center is calculated by (10), and the i-th eigenvector is classified into the nearest cluster, where $1 \leq i \leq m$ and $1 \leq j \leq k$.

Step 7: Update the clustering centers. k clustering centers are recalculated by (6), which are served as the new clustering centers.

Step 8: Determine whether the termination conditions have been met. At first, the total mean square error MSE between all eigenvectors and their corresponding clustering centers is calculated by (7). Then, determine whether MSE is less than or equal to the convergence threshold or the maximum number of iterations has reached, if yes, a rolling bearing fault diagnosis model is obtained; if not, go to Step 6.

D. PROPOSED SPARK-BASED PARALLEL ACO-K-MEANS CLUSTERING ALGORITHM

In order to efficiently and accurately perform clustering analysis on the massive eigenvectors of rolling bearing, a Spark-based parallel ACO-K-Means clustering algorithm for rolling bearing fault diagnosis is proposed. Fig. 5 presents the flowchart of the proposed Spark-based parallel ACO-K-Means clustering algorithm, which mainly includes the following steps.

Step 1: Read the eigenvectors of rolling bearing to create an RDD. The training set with m eigenvectors is read from HDFS to create an RDD *eigenvectorRDD* containing n partitions, each RDD partition contains m/n eigenvectors, and each worker node will handle multiple RDD partitions.

Step 2: Randomly select the initial clustering centers. k eigenvectors are randomly selected from *eigenvectorRDD* as the initial clustering centers $M = \{\mu_1, \mu_2, \ldots, \mu_k\}$, which are broadcasted from the master node to each worker node.

Step 3: Calculate the transition probability in parallel. For the s-th RDD partition $= \{E_{(s-1)m/n+1}, E_{(s-1)m/n+2}, \ldots, E_{s*m/n}\}$ of *eigenvectorRDD*, the transition probability P_{ij}^{i} from the i-th eigenvector $E_{(s-1)m/n+i}$ to the j-th initial clustering center μ_j for the i-th ant is calculated by (1), and the i-th ant selects the t-th initial clustering center as an access point $V_{\forall t \in [1,k]}$ according to the maximum transition probability $P_{it}^{i} = \max(P_{i1}^{i}, P_{i2}^{i}, \ldots, P_{ik}^{i})$, where $1 \leq s \leq n$, $1 \leq i \leq m/n$, $1 \leq j \leq k$, and $1 \leq t \leq k$. An eigenvector and its corresponding access point are regarded as a value and a key respectively, and a key-value pair RDD *pathRDD* is obtained.

Step 4: Update the initial clustering centers in parallel. For the s-th RDD partition $= \{< V_{\forall t \in [1,k]}, E_{(s-1)m/n+1} >, < V_{\forall t \in [1,k]}, E_{(s-1)m/n+2} >, \ldots, < V_{\forall t \in [1,k]}, E_{s*m/n} >\}$ of *pathRDD*, all eigenvectors of the s-th RDD partition are divided into k groups according to different access points, and the average value $\bar{E}_{sj}$ of all eigenvectors in the j-th group is calculated by (6), where $1 \leq s \leq n$ and $1 \leq j \leq k$. k average values of each RDD partition of *pathRDD* are gathered from each worker node to the master node, and $\mu_j = \frac{1}{n}\sum_{s=1}^{n} \bar{E}_{sj} (1 \leq j \leq k)$ is used as the j-th new initial clustering center, and k updated initial clustering centers are broadcasted from the master node to each worker node.

Step 5: Determine whether the initial clustering centers have converged. Firstly, the mean square error MSE_s between all eigenvectors of the s-th RDD partition of *pathRDD* and their corresponding initial clustering centers is calculated by (7), where $1 \leq s \leq n$. Secondly, the mean square error obtained from each RDD partition is gathered from each worker node to the master node, and the total mean square error $MSE = \frac{1}{n}\sum_{s=1}^{n} MSE_s$ is obtained. Finally, determine whether MSE is less than or equal to the convergence threshold, if yes, the global optimal initial clustering centers are output and go to Step 6; if not, go to Step 3.

Step 6: Classify each eigenvector into the nearest cluster in parallel. For the s-th RDD partition of *eigenvectorRDD*, the weighted Euclidean distance between each eigenvector and each clustering center is calculated by (10), and each eigenvector is classified into the nearest cluster, where $1 \leq s \leq n$. An eigenvector and the centroid of its corresponding nearest cluster are regarded as a value and a key respectively, and a key-value pair RDD *clusterRDD* is obtained.

Step 7: Update the clustering centers in parallel. For the s-th RDD partition $= \{< \mu_{\forall j \in [1,k]}, E_{(s-1)m/n+1} >, < \mu_{\forall j \in [1,k]}, E_{(s-1)m/n+2} >, \ldots, < \mu_{\forall j \in [1,k]}, E_{s*m/n} >\}$ of *clusterRDD*, the centroids $\{\mu_{s1}, \mu_{s2}, \ldots, \mu_{sk}\}$ of k clusters of the s-th RDD partition are recalculated by (6), where $1 \leq s \leq n$. The centroids of k clusters of each RDD partition of *clusterRDD* are gathered from each worker node to the master node, $\mu_j = \frac{1}{n}\sum_{s=1}^{n} \mu_{sj} (1 \leq j \leq k)$ is used as the j-th new clustering center, and k updated clustering centers are broadcasted from the master node to each worker node.

Step 8: Determine whether the clustering centers have converged or the maximum number of iterations has reached. Firstly, the mean square error MSE_s between all eigenvectors of the s-th RDD partition of *clusterRDD* and their corresponding clustering centers is calculated by (7). Secondly, the mean square error obtained from each RDD partition is gathered from each worker node to the master node, and the total mean square error $MSE = \frac{1}{n}\sum_{s=1}^{n} MSE_s$ is obtained. Finally, determine whether MSE is less than or equal to the convergence threshold or the maximum number of iterations has reached, if yes, the fault diagnosis model of rolling bearing is output; if not, go to Step 6.

After a well-trained fault diagnosis model of rolling bearing is obtained, the vibration data of rolling bearing can be diagnosed practically. As shown in Fig. 3, firstly, the vibration data of rolling bearing to be diagnosed are preprocessed by Spark-based three-layer wavelet packet decomposition to obtain eigenvectors, which are stored in HDFS. Secondly, all eigenvectors are read from HDFS to create an RDD.

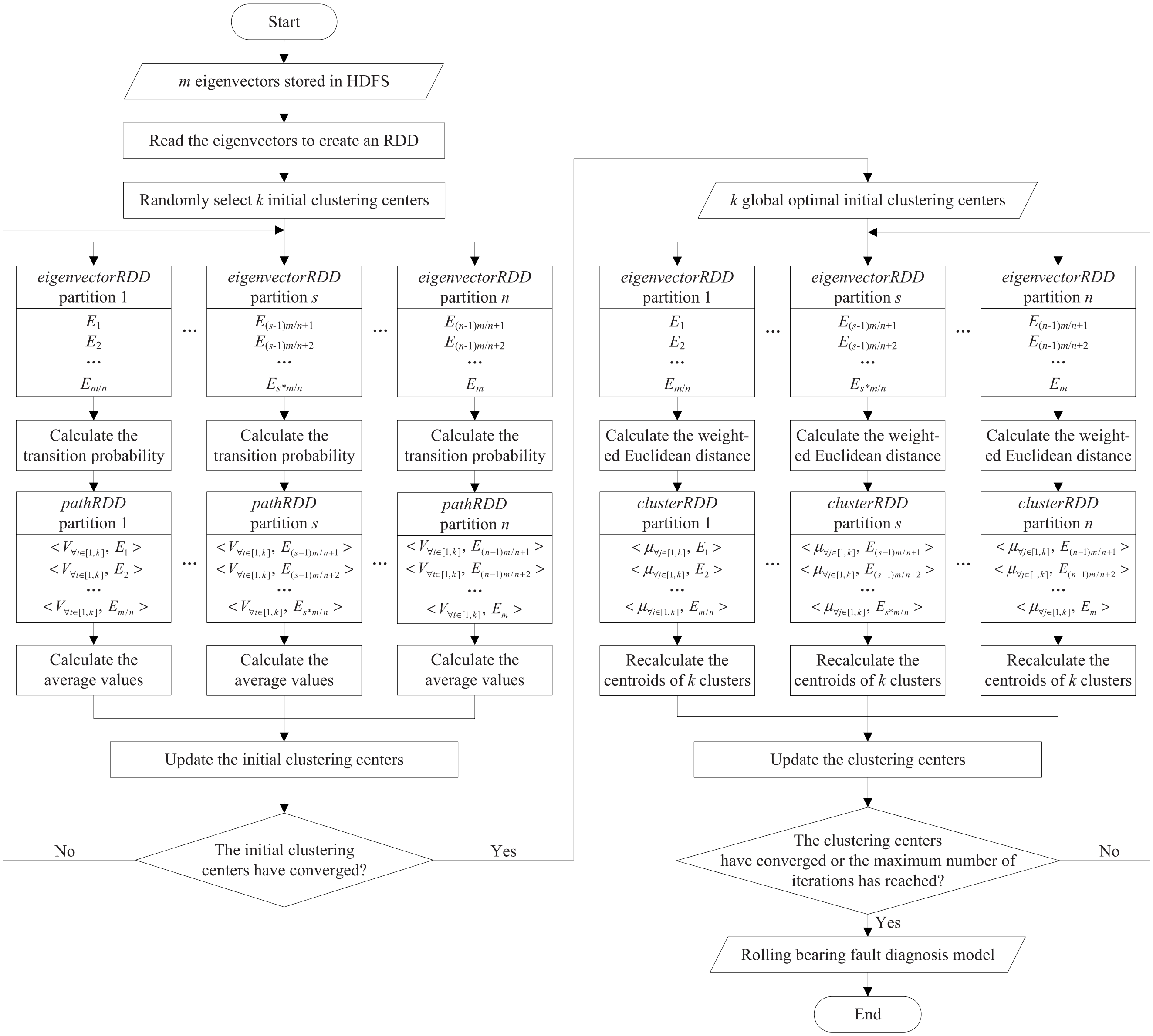
Start
m eigenvectors stored in HDFS
Read the eigenvectors to create an RDD
Randomly select k initial clustering centers
eigenvectorRDD partition 1
E_1
E_2
…
E_{m/n}
eigenvectorRDD partition s
E_{(s-1)m/n+1}
E_{(s-1)m/n+2}
…
E_{s*m/n}
eigenvectorRDD partition n
E_{(n-1)m/n+1}
E_{(n-1)m/n+2}
…
E_m
Calculate the transition probability
pathRDD partition 1
< V_{∀t∈[1,k]}, E_1 >
< V_{∀t∈[1,k]}, E_2 >
< V_{∀t∈[1,k]}, E_{m/n} >
pathRDD partition s
< V_{∀t∈[1,k]}, E_{(s−1)m/n+1} >
< V_{∀t∈[1,k]}, E_{(s−1)m/n+2} >
< V_{∀t∈[1,k]}, E_{s*m/n} >
pathRDD partition n
< V_{∀t∈[1,k]}, E_{(n−1)m/n+1} >
< V_{∀t∈[1,k]}, E_{(n−1)m/n+2} >
< V_{∀t∈[1,k]}, E_m >
Calculate the average values
Update the initial clustering centers
The initial clustering centers have converged?
No
Yes
k global optimal initial clustering centers
Calculate the weighted Euclidean distance
clusterRDD partition 1
< μ_{∀j∈[1,k]}, E_1 >
< μ_{∀j∈[1,k]}, E_2 >
< μ_{∀j∈[1,k]}, E_{m/n} >
clusterRDD partition s
< μ_{∀j∈[1,k]}, E_{(s−1)m/n+1} >
< μ_{∀j∈[1,k]}, E_{(s−1)m/n+2} >
< μ_{∀j∈[1,k]}, E_{s*m/n} >
clusterRDD partition n
< μ_{∀j∈[1,k]}, E_{(n−1)m/n+1} >
< μ_{∀j∈[1,k]}, E_{(n−1)m/n+2} >
< μ_{∀j∈[1,k]}, E_m >
Recalculate the centroids of k clusters
Update the clustering centers
The clustering centers have converged or the maximum number of iterations has reached?
No
Yes
Rolling bearing fault diagnosis model
End

FIGURE 5. Flowchart of the proposed Spark-based parallel ACO-K-Means clustering algorithm.

Thirdly, the weighted Euclidean distance between each eigenvector of the RDD and each clustering center provided by the fault diagnosis model is calculated by (10) in parallel. Finally, each eigenvector is classified into the nearest cluster, and the fault diagnosis results are output.

IV. EXPERIMENT

A. EXPERIMENTAL SETUP

The rolling bearing dataset [32] provided by CWRU is used to verify the effectiveness of the proposed fault diagnosis method of rolling bearing using Spark-based parallel ACO-K-Means clustering algorithm. This dataset contains plenty of raw vibration data collected under different working conditions, but the size of the eigenvectors obtained from the three-layer wavelet packet decomposition for these data is only 31.73 MB. It is difficult to make an effective evaluation of the proposed fault diagnosis method for too little data, thus the sliding window method [35] is adopted to enhance the original vibration data, where the size and the offset of sliding window are set to 4000 and 1 respectively. Three different size of datasets (i.e., DataSet A, DataSet B, and DataSet C) are obtained from the three-layer wavelet packet decomposition for the enhanced vibration data of rolling bearing, as shown in Table 3. Each dataset contains normal state data, ball fault data, inner race fault data, and outer race fault data.

The experimental platform is a Spark cluster, which consists of one master node and eight worker nodes, and whose cluster resource manager is Spark's own standalone cluster manager. The hardware environment and software environment of this experimental platform are presented in Table 4

TABLE 3. Description of the rolling bearing dataset.

Dataset	Vibration Data Size (GB)	Eigenvector Size (GB)	Data Source
DataSet A	1630.26	33.27	Enhanced vibration data of rolling bearing
DataSet B	3421.44	69.89	Enhanced vibration data of rolling bearing
DataSet C	5867.52	119.80	Enhanced vibration data of rolling bearing

TABLE 4. Hardware environment of the experimental platform.

Node Type	Number of Nodes	CPU Model	CPU Cores Per Node	Memory Size (GB)	Port Speed (Mbps)
Master node	1	Intel Xeon E3-1225 v5	8	64	1000
Worker node	8	Intel Core i7-9700K	8	64	1000

TABLE 5. Software environment of the experimental platform.

Software Name	Software Version
Linux operating system	CentOS 7.1
Java runtime environment	jdk1.8.0_65
Hadoop	hadoop-2.7.3
Spark	spark-3.0.0
Scala	scala-2.11.8

TABLE 6. Parameter settings of the Spark cluster.

Parameter Name	Parameter Value
Number of executors per worker node	2
Number of cores per executor	4
Memory size per executor	30 GB
Memory size of Spark driver	32 GB

and Table 5 respectively, and the parameter settings of the Spark cluster are listed in Table 6.

In the training of the proposed fault diagnosis model of rolling bearing, the parameter settings of ACO-K-Means clustering algorithm are listed in Table 7. Since the vibration data of rolling bearing include normal state data, ball fault data, inner race fault data, and outer race fault data, the number of clusters is set to 4. If the number of running states of rolling bearing contained in the dataset is known, the number of clusters is determined by the number of running states of rolling bearing. If the number of running states of rolling bearing contained in the dataset is unknown, the number of clusters can be dynamically determined by elbow method [36] or silhouette coefficient method [37]. The pheromone heuristic factor α indicates the relative importance of pheromone intensity, if the value of α is too large, the random search ability of the algorithm is easily weakened. If the value of α is too small, it is easy to fall into local optimum. The expected heuristic factor β indicates the relative importance of visibility, if the value of β is too large, it is also easy to fall into local optimum. If the value of β is too small, it is easy to fall into pure random search, which makes it difficult to find the global optimal solution. The pheromone volatilization factor ρ indicates the disappearance level of pheromone, if the value of ρ is too large, it is easy to affect the randomness of search and the global optimality of solution. If the value of ρ is too small, the convergence speed will be decreased.

TABLE 7. Parameter settings of ACO-K-Means clustering algorithm.

Paramenter Name	Parameter Value
Number of clusters	4
Pheromone heuristic factor	$\alpha = 2$
Expected heuristic factor	$\beta = 4$
Pheromone volatilization factor	$\rho = 0.3$
Convergence threshold of ACO	0.025
Convergence threshold of K-Means	0.021
Maximum number of iterations	600

The parameter tuning process of ACO-K-Means clustering algorithm is as follows. Firstly, the value ranges of three key parameters are determined, i.e., $\alpha \in \{1, 2, 3, 4, 5\}$, $\beta \in \{2, 3, 4, 5, 6\}$, and $\rho \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$. Secondly, the fault diagnosis model is trained and tested according to different combinations of the three parameters, and the model training time and fault diagnosis accuracy are observed. Finally, the combination of parameters with the highest fault diagnosis accuracy and the shortest model training time is regarded as the best combination of parameters. The experimental results show that the best combination of parameters is $\alpha = 2$, $\beta = 4$, and $\rho = 0.3$.

In this paper, in order to accurately measure the diagnosis accuracy, training time, and fault diagnosis time of a fault diagnosis model, each experiment is repeated 30 times, and the measurement results are averaged.

B. EVALUATION OF SPARK-BASED THREE-LAYER WAVELET PACKET DECOMPOSITION APPROACH

To better evaluate the efficiency of using the proposed Spark-based three-layer wavelet packet decomposition approach to preprocess the massive vibration data of rolling bearing, three different size of vibration data are preprocessed by the proposed approach on two different Spark clusters, i.e., the Spark cluster with a single worker node and the Spark cluster with 8 worker nodes.

Fig. 6 presents the data preprocessing time obtained with Spark-based three-layer wavelet packet decomposition for three different size of vibration data on two different Spark clusters. Compared with the Spark cluster with a single worker node, the data preprocessing efficiency obtained by the proposed approach on the Spark cluster with 8 worker nodes is improved by 85.92%, 86.49%, and 86.84% for 1.59 TB, 3.34 TB, and 5.73 TB of vibration data, respectively. The improvement of efficiency is mainly because more worker nodes are used to perform the data preprocessing task in parallel, and there is no communication overhead between

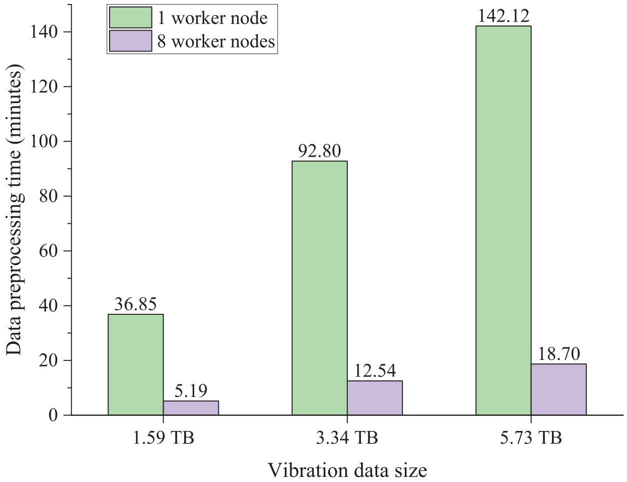

FIGURE 6. Data preprocessing time obtained with Spark-based three-layer wavelet packet decomposition for different size of vibration data.

worker nodes in data preprocessing. The results demonstrate that the proposed Spark-based three-layer wavelet packet decomposition approach can fully utilize the computing resources of multiple worker nodes to efficiently preprocess the massive vibration data of rolling bearing.

C. ANALYSIS OF FAULT DIAGNOSIS ACCURACY

To better analyze the diagnosis accuracy of the proposed fault diagnosis method of rolling bearing, the Spark-based parallel K-Means clustering algorithm provided by Spark MLlib [38] (Spark-K-Means) and the proposed Spark-based parallel ACO-K-Means clustering algorithm (Spark-ACO-K-Means) are used to train and test the fault diagnosis model of rolling bearing on the Spark cluster with 8 worker nodes. Moreover, in order to analyze the impact of the size of dataset on the fault diagnosis accuracy, three different size of datasets listed in Table 3 are used for the training and testing of fault diagnosis model, and all eigenvectors contained in each dataset are randomly divided into training set and test set according to the ratio of 7:3.

Fig. 7 presents the fault diagnosis accuracies obtained with two different fault diagnosis methods for three different size of datasets on the Spark cluster with 8 worker nodes. It can be seen from Fig. 7 that the proposed Spark-ACO-K-Means achieves a satisfactory diagnosis accuracy, and the fault diagnosis accuracies reach up to 97.73%, 97.87%, and 97.99% for DataSet A, DataSet B, and DataSet C, respectively. Compared with Spark-K-Means, Spark-ACO-K-Means achieves better fault diagnosis results, and the fault diagnosis accuracy is increased by 4.92% on average.

To compare the fault diagnosis effect of Spark-K-Means and that of Spark-ACO-K-Means more intuitively, the principal component analysis (PCA) [39] is used to reduce the dimensions of each clustering center and each eigenvector contained in the fault diagnosis results from 8 to 2, and the fault diagnosis results are visualized in 2-dimensional space.

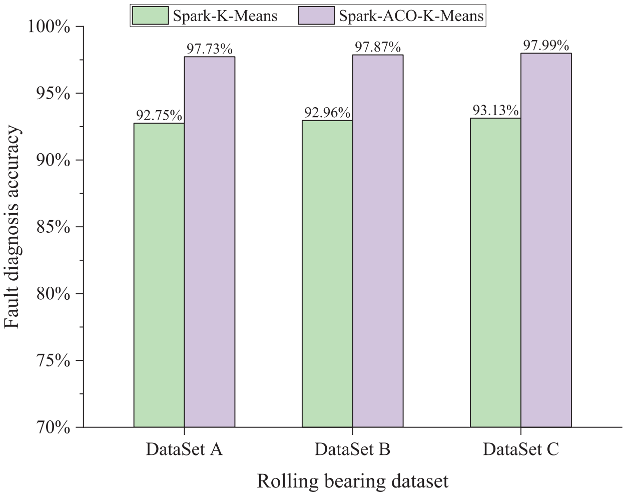

FIGURE 7. Fault diagnosis accuracies obtained with different fault diagnosis methods for different size of datasets.

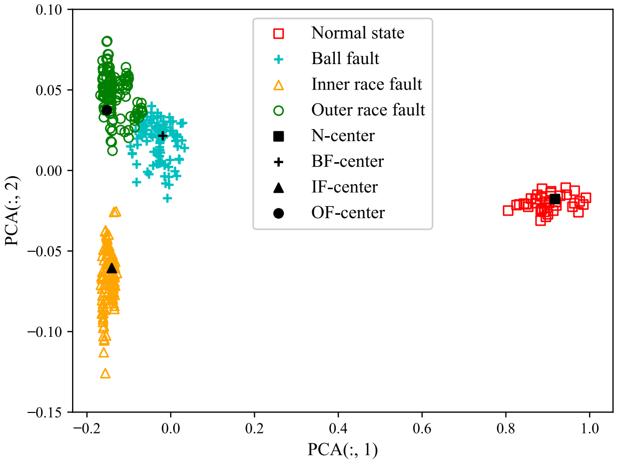

FIGURE 8. Clustering effect of Spark-K-Means.

Figs. 8 and 9 demonstrate the clustering effects of Spark-K-Means and Spark-ACO-K-Means, respectively. As shown in Fig. 8, the clustering effects of normal state and inner race fault of rolling bearing are obvious, whereas the clustering effects of ball fault and outer race fault are not satisfactory. This is because the eigenvectors of ball fault and that of outer race fault are similar, and Spark-K-Means is easy to fall into local optimum, which results in the ball fault and outer race fault are easy to be misdiagnosed. As can be seen from Fig. 9, compared with Spark-K-Means, Spark-ACO-K-Means achieves better clustering effect, and especially the clustering effects of ball fault and outer race fault are changed obviously. This is because Spark-ACO-K-Means obtains the global optimal initial clustering centers, and the weighted Euclidean distance measure is utilized to improve the calculation of the distance between each eigenvector and each clustering center to enhance the clustering ability of K-Means clustering algorithm to a certain extent. Thus, the proposed Spark-ACO-K-Means can obtain more stable and higher fault diagnosis accuracy.

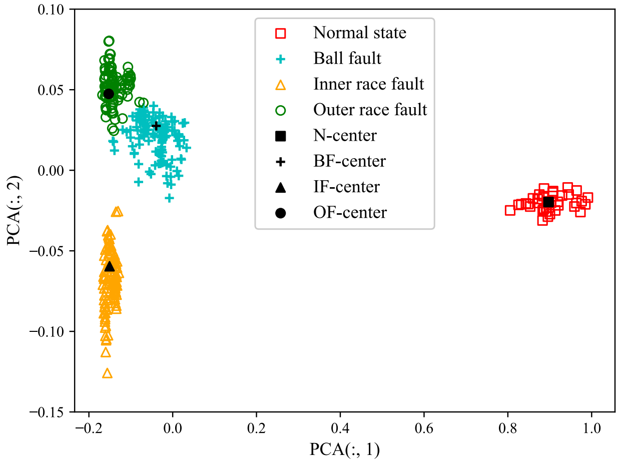

FIGURE 9. Clustering effect of Spark-ACO-K-Means.

As shown in Fig. 7, the size of dataset can affect the fault diagnosis accuracy of rolling bearing, as the size of dataset increases, the fault diagnosis accuracies obtained using Spark-K-Means and Spark-ACO-K-Means are gradually increased respectively. For example, for Spark-ACO-K-Means, the fault diagnosis accuracy obtained with DataSet C is 0.26% and 0.12% higher than that obtained with DataSet A and DataSet B, respectively. Generally speaking, the larger the rolling bearing dataset, the more the monitoring data of various running states of rolling bearing contained in the dataset, and the increase of the number and diversity of training samples is helpful to train a better fault diagnosis model of rolling bearing, which can improve the fault diagnosis accuracy of rolling bearing.

D. ANALYSIS OF TRAINING EFFICIENCY AND DIAGNOSIS EFFICIENCY OF FAULT DIAGNOSIS MODEL

To effectively analyze the training efficiency and diagnosis efficiency of the fault diagnosis model built in this paper, for DataSet A, DataSet B, and DataSet C, the proposed Spark-ACO-K-Means is used to train the fault diagnosis model of rolling bearing on the Spark clusters with different number of worker nodes, and the well-trained model is used for fault diagnosis, where 70% and 100% data of each dataset are used for model training and fault diagnosis respectively. In the fault diagnosis, the reason to use all the data in each dataset is to better evaluate the diagnosis efficiency of fault diagnosis model for a large-scale dataset.

Tables 8 and 9 present the model training time and fault diagnosis time obtained with different size of datasets and different number of worker nodes respectively, where the model training time refers to the running time of the Spark application for the training of fault diagnosis model, and the fault diagnosis time refers to the running time of the Spark application for fault diagnosis. As can be seen from Tables 8 and 9, for three different size of datasets, with the increase of the number of worker nodes in a Spark cluster, both the model training time and fault diagnosis time gradually decrease. For DataSet A, DataSet B, and DataSet C, the training time of fault diagnosis model obtained with 8 worker nodes is 82.51%, 84.64%, and 85.87% less than that obtained with a single worker node respectively, and the fault diagnosis time obtained with 8 worker nodes is 86.67%, 86.83%, and 87.18% less than that obtained with a single worker node respectively. The results demonstrate that with the increase of the size of dataset, both the model training efficiency and fault diagnosis efficiency are gradually improved. Therefore, the proposed fault diagnosis method is more suitable for processing large-scale rolling bearing datasets.

TABLE 8. Model training time obtained with different size of datasets and different number of worker nodes.

Dataset	Training Time of Fault Diagnosis Model (minutes)				
	1 Worker Node	2 Worker Nodes	4 Worker Nodes	6 Worker Nodes	8 Worker Nodes
DataSet A	168.78	98.46	51.72	36.78	29.52
DataSet B	344.04	197.58	100.38	67.32	52.86
DataSet C	592.02	313.86	159.12	109.38	83.64

TABLE 9. Fault diagnosis time obtained with different size of datasets and different number of worker nodes.

Dataset	Fault Diagnosis Time (minutes)				
	1 Worker Node	2 Worker Nodes	4 Worker Nodes	6 Worker Nodes	8 Worker Nodes
DataSet A	2.25	1.19	0.66	0.41	0.30
DataSet B	4.86	2.53	1.36	0.86	0.64
DataSet C	8.58	4.51	2.28	1.43	1.10

As shown in Tables 8 and 9, compared with a single worker node, the model training time of three datasets obtained with 2, 4, 6, and 8 worker nodes are reduced by 44.80%, 71.83%, 80.68%, and 84.97% on average respectively, the fault diagnosis time of three datasets obtained with 2, 4, 6, and 8 worker nodes are reduced by 47.55%, 72.59%, 82.79%, and 87.00% on average respectively. The results show that with the expansion of the scale of Spark cluster, both the model training efficiency and fault diagnosis efficiency are also gradually improved. It is not difficult to see from Tables 8 and 9 that with the increase of the number of worker nodes, the reduction trends of both model training time and fault diagnosis time gradually tend to be flat. Therefore, when the proposed fault diagnosis method is applied, the performance of both model training and fault diagnosis can be improved by appropriately enlarging the scale of Spark cluster.

Seeing that the speedup and parallel efficiency are two important indexes to evaluate the performance of parallel processing, the speedup and parallel efficiency are used to evaluate the performance of fault diagnosis model training in addition to the model training time in this paper. The speedup can be calculated by

$$S_n = \frac{T}{T_n}, \tag{11}$$

where S_n denotes the relative speedup obtained in the training of fault diagnosis model by Spark-ACO-K-Means on the Spark cluster with n worker nodes, T represents the model training time obtained with a single worker node, and T_n refers to the model training time obtained with n worker nodes. Note that the speedup is obtained by comparing the model training time when using a single worker node containing 8 CPU cores and 8 worker nodes containing 64 CPU cores both reach the same accuracy. The parallel efficiency can be calculated by

$$E_n = \frac{S_n}{n} \times 100\%, \tag{12}$$

where E_n represents the parallel efficiency obtained in the training of fault diagnosis model by Spark-ACO-K-Means on the Spark cluster with n worker nodes, which can reflect the effective utilization degree of computing resources of all worker nodes that participate in the training of fault diagnosis model on the Spark cluster.

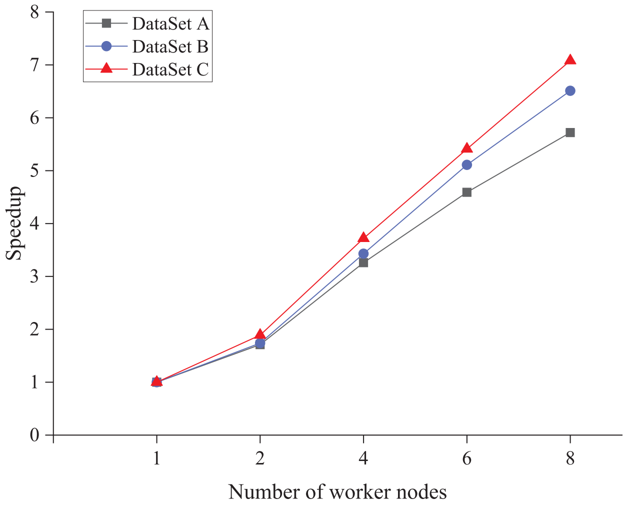

FIGURE 10. Speedups of fault diagnosis model training.

Fig. 10 presents the speedups obtained in the training of fault diagnosis model with different size of datasets and different number of worker nodes. As shown in Fig. 10, as the number of worker nodes increases, the obtained speedup is gradually increased. When the number of worker nodes is increased from 1 to 8, the average speedup obtained in the training of fault diagnosis model with three different size of datasets increases from 1.00× to 6.43×, which shows that Spark-ACO-K-Means has good parallelism. As can be seen from Fig. 10, the speedup approaches linear growth, but it does not reach its theoretical value, because the communication cost and task scheduling cost caused by the increase of the number of worker nodes reduce the performance of model training to a certain extent. It can also be seen from Fig. 10 that the speedups of 5.72×, 6.5×, and 7.08× are respectively obtained in the training of fault diagnosis model with DataSet A, DataSet B, and DataSet C when the number of worker nodes is 8. The results show that a higher speedup can be obtained in the training of fault diagnosis model with a larger dataset when the number of worker nodes is fixed. Therefore, a larger dataset is more helpful to play the advantage of parallel computing of the proposed fault diagnosis method.

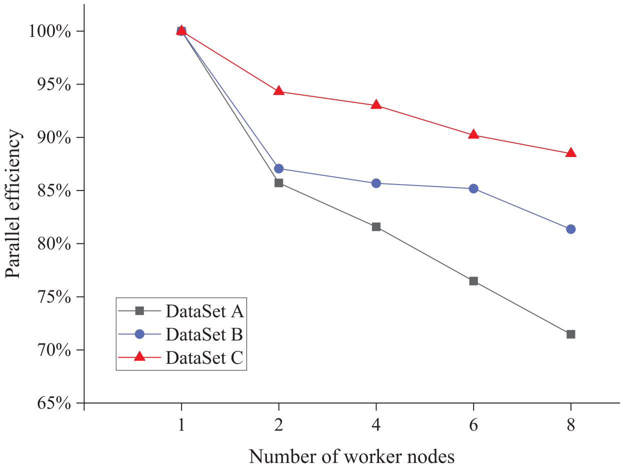

FIGURE 11. Parallel efficiency of fault diagnosis model training.

Fig. 11 shows the parallel efficiency obtained in the training of fault diagnosis model with different size of datasets and different number of worker nodes. As shown in Fig. 11, when the numbers of worker nodes that participate in model training are 2, 4, 6, and 8 respectively, the average parallel efficiency obtained in the training of fault diagnosis model with three different size of datasets reaches up to 89.03%, 86.76%, 83.96%, and 80.43% respectively, which shows that Spark-ACO-K-Means has good parallel efficiency, that is, the computing resources of the Spark cluster are effectively utilized. However, the increase of the number of worker nodes will affect the parallel efficiency to a certain extent, because the additional overhead brought by the expansion of the scale of Spark cluster partly offsets the improvement of computing performance brought by it. Also can be seen from Fig. 11, when the number of worker nodes is fixed, as the size of the dataset becomes larger, the obtained parallel efficiency is gradually increased. For example, when the number of worker nodes is 8, the parallel efficiency obtained with DataSet C is 17.01% and 7.12% higher than that obtained with DataSet A and DataSet B respectively. Therefore, the proposed fault diagnosis method can make full use of the computing resources of a Spark cluster to efficiently process large-scale rolling bearing datasets in parallel.

To better evaluate the performance of model training and fault diagnosis obtained using Spark-ACO-K-Means, the serial ACO-K-Means clustering algorithm (Serial-ACO-K-Means) is also used to train and test the fault diagnosis model of rolling bearing for three different size of datasets. Serial-ACO-K-Means is performed on one CPU core of a single node, and Spark-ACO-K-Means is carried out on the Spark cluster with 8 worker nodes. Fig. 12 shows the speedups of Spark-ACO-K-Means over

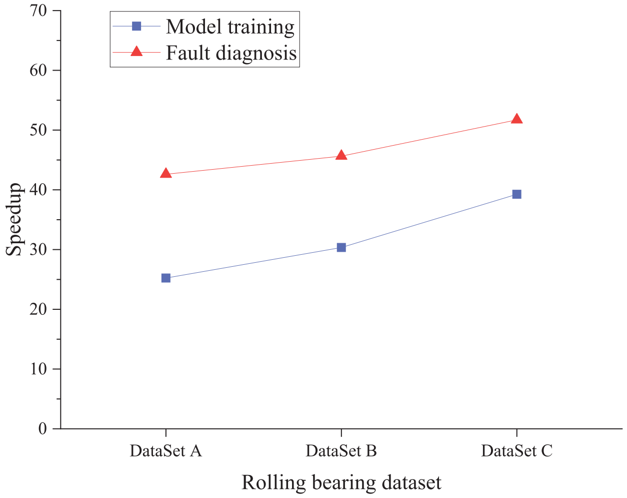

FIGURE 12. Speedups of Spark-ACO-K-Means over Serial-ACO-K-Means.

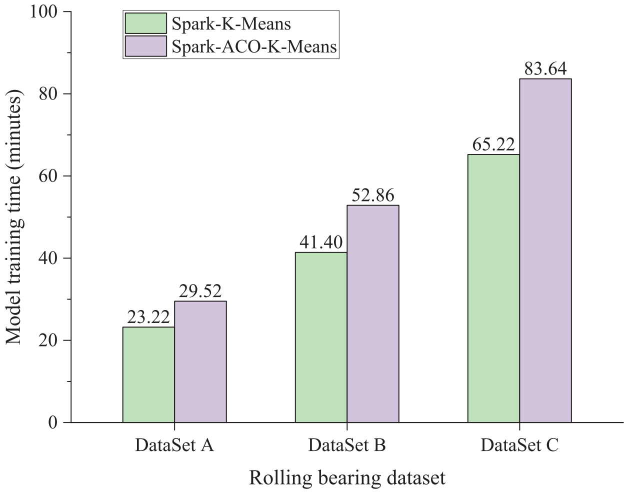

FIGURE 13. Model training time of Spark-K-Means and Spark-ACO-K-Means.

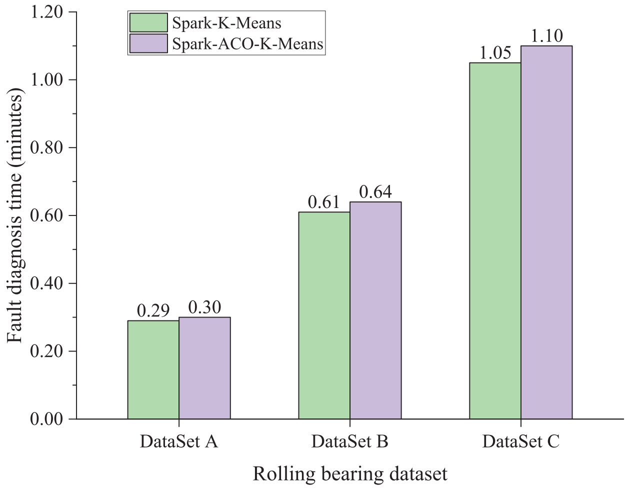

FIGURE 14. Fault diagnosis time of Spark-K-Means and Spark-ACO-K-Means.

Serial-ACO-K-Means for model training and fault diagnosis on three different size of datasets, where the training speedup is obtained by comparing the model training time when Serial-ACO-K-Means and Spark-ACO-K-Means both reach the same accuracy. As shown in Fig. 12, the performance of Spark-ACO-K-Means is greatly improved than that of Serial-ACO-K-Means. For example, compared with Serial-ACO-K-Means, Spark-ACO-K-Means obtains the speedups of 39.25× and 51.71× in model training and fault diagnosis for DataSet C, respectively. This is mainly because Spark-ACO-K-Means can efficiently utilize many CPU cores of multiple worker nodes to perform model training and fault diagnosis in parallel on the Spark cluster for large-scale rolling bearing datasets. Moreover, when Serial-ACO-K-Means is used to perform model training and fault diagnosis for large-scale rolling bearing datasets, due to the limited memory space of a single node, a part of data will be spilled onto disk, which will greatly affect the performance of model training and fault diagnosis.

E. ANALYSIS OF THE IMPACT OF ACO AND WEIGHTED EUCLIDEAN DISTANCE MEASURE ON PERFORMANCE

To analyze the impact of ACO and weighted Euclidean distance measure adopted in Spark-ACO-K-Means on the model training efficiency and fault diagnosis efficiency, Spark-K-Means and Spark-ACO-K-Means are used to train the fault diagnosis model of rolling bearing on the Spark cluster with 8 worker nodes for three different size of datasets, and the well-trained model is used for fault diagnosis.

Fig. 13 presents the model training time of Spark-K-Means and Spark-ACO-K-Means for three different size of datasets. The model training time of Spark-ACO-K-Means is increased by 27.69% on average than that of Spark-K-Means for three different size of datasets. The initial clustering centers are randomly selected in Spark-K-Means, whereas the global optimal initial clustering centers are selected by ACO algorithm in Spark-ACO-K-Means, which is the main reason for the increase of model training time of Spark-ACO-K-Means. Although a lot of time is spent on optimizing the selection of initial clustering centers in Spark-ACO-K-Means, after getting the global optimal initial clustering centers, the clustering centers of Spark-ACO-K-Means can converge in a shorter time compared with Spark-K-Means. In addition, compared with the Euclidean distance measure adopted in Spark-K-Means, the weighted Euclidean distance measure adopted in Spark-ACO-K-Means also increases the model training time to a certain extent.

Fig. 14 presents the fault diagnosis time of Spark-K-Means and Spark-ACO-K-Means for three different size of datasets. The fault diagnosis time of Spark-ACO-K-Means is increased by 4.62% on average than that of Spark-K-Means for three different size of datasets. In the fault diagnosis of rolling bearing, the distance between each eigenvector and each clustering center only needs to be calculated once, and all the clustering centers are provided by the trained fault diagnosis model of rolling bearing. Therefore, the reason for the slight increase of fault diagnosis time of Spark-ACO-K-Means is that the computational cost of weighted Euclidean distance measure is slightly higher than that of Euclidean distance measure. In addition, ACO algorithm only participates in the training of fault diagnosis model, thus it is not related to the fault diagnosis efficiency of Spark-ACO-K-Means.

In a word, ACO algorithm only affects the model training time, and the weighted Euclidean distance measure has a little impact on the model training time and fault diagnosis time, but they can improve the fault diagnosis accuracy (see Section IV-C).

F. COMPARISON WITH OTHER SWARM INTELLIGENCE OPTIMIZATION ALGORITHMS

To better evaluate the effectiveness of the proposed ACO-K-Means clustering algorithm, the other two different swarm intelligence optimization algorithms including GA [21] and PSO algorithm [22] are also used for optimizing the selection of initial clustering centers of K-Means. Similar to Spark-ACO-K-Means, Spark-based parallel GA-K-Means clustering algorithm (Spark-GA-K-Means) and Spark-based parallel PSO-K-Means clustering algorithm (Spark-PSO-K-Means) are implemented, and the weighted Euclidean distance measure is also used in Spark-GA-K-Means and Spark-PSO-K-Means. In this experiment, Spark-GA-K-Means, Spark-PSO-K-Means, and Spark-ACO-K-Means are used to train and test the fault diagnosis model of rolling bearing on the Spark cluster with 8 worker nodes for DataSet C. The parameter settings of GA and PSO algorithm are as follows.

- GA: The population size is set to 100, the crossover probability is set to 0.5, the mutation probability is set to 0.025, and the maximum number of iterations is set to 600.
- PSO: The swarm size is set to 100, the inertia weight is set to 0.6, all the acceleration constants are set to 2.0, and the maximum number of iterations is set to 600.

Table 10 gives the fault diagnosis accuracies, model training time, and fault diagnosis time of fault diagnosis methods optimized by different swarm intelligence optimization algorithms. As can be seen from Table 10, the fault diagnosis accuracy of Spark-ACO-K-Means is 0.11% lower than that of Spark-GA-K-Means and is 0.32% higher than that of Spark-PSO-K-Means. For Spark-GA-K-Means, the method of searching for the optimal solution based on probability is adopted in GA, which can search the slightly better initial clustering centers compared with ACO algorithm. For Spark-PSO-K-Means, with the increase of the number of iterations of PSO algorithm, the velocities of a very few particles may become low or zero before searching for the global optimal initial clustering centers, which results in these particles have not enough power to jump out of the local optimum.

It can be seen from Table 10 that the model training time of Spark-ACO-K-Means is 28.88% and 12.08% lower than that of Spark-GA-K-Means and Spark-PSO-K-Means respectively. The reasons for the increases of model training time of Spark-GA-K-Means and Spark-PSO-K-Means are that GA and PSO algorithm respectively need to spend more time to obtain the global optimal initial clustering centers than ACO algorithm. It can also be seen from Table 10 that the fault diagnosis time of the three fault diagnosis methods are almost the same. This is because the difference of fault diagnosis time among Spark-GA-K-Means, Spark-PSO-K-Means, and Spark-ACO-K-Means depends on the calculation method of the distance between the eigenvector and the clustering center, and it is not related to GA, PSO algorithm, and ACO algorithm used in the training of fault diagnosis model. The reason why the fault diagnosis time of Spark-GA-K-Means, Spark-PSO-K-Means, and Spark-ACO-K-Means are almost the same is that they all use the weighted Euclidean distance measure in fault diagnosis.

TABLE 10. Comparison of fault diagnosis methods optimized by different swarm intelligence optimization algorithms.

Fault Diagnosis Method	Fault Diagnosis Accuracy	Model Training Time (minutes)	Fault Diagnosis Time (minutes)
Spark-GA-K-Means	98.10%	117.60	1.098
Spark-PSO-K-Means	97.67%	95.13	1.102
Spark-ACO-K-Means	97.99%	83.64	1.099

G. COMPARISON WITH OTHER FAULT DIAGNOSIS METHODS

To better evaluate the effectiveness of the proposed fault diagnosis method of rolling bearing, the other widely used classification algorithms including RF [31], AlexNet [40], and ResNet [41] are also used to build the fault diagnosis model of rolling bearing. Similar to Spark-ACO-K-Means, Spark-based parallel AlexNet (Spark-AlexNet) and Spark-based parallel ResNet (Spark-ResNet) are implemented, whereas Spark-based parallel RF (Spark-RF) is provided by Spark MLlib [38]. In this experiment, Spark-RF, Spark-AlexNet, Spark-ResNet, and Spark-ACO-K-Means are used to train and test the fault diagnosis model of rolling bearing on the Spark cluster with 8 worker nodes. For Spark-RF and Spark-ACO-K-Means, all 119.80 GB of eigenvectors contained in DataSet C are randomly divided into training set and test set according to the ratio of 7:3. For Spark-AlexNet and Spark-ResNet, the 119.80 GB of enhanced vibration data of rolling bearing are transformed into 2-D gray images of 64 pixels × 64 pixels, and the dataset composed of gray images is also randomly divided into the training set and test set according to the ratio of 7:3. In the training of fault diagnosis model, the number of sub-trees of Spark-RF is set to 100, and the settings of network structure and hyper-parameters of Spark-AlexNet and that of Spark-ResNet can be found in [40] and [41] respectively, where the batch size is set to 1024 and the model training is terminated after 30 epochs. Considering the model training efficiency and fault diagnosis efficiency of Spark-ResNet, ResNet-18 which has fewer network layers than ResNet-50 is adopted in Spark-ResNet.

TABLE 11. Comparison of different fault diagnosis methods.

Fault Diagnosis Method	Fault Diagnosis Accuracy	Model Training Time (minutes)	Fault Diagnosis Time (minutes)
Spark-RF	98.32%	382.86	34.73
Spark-AlexNet	99.85%	1548.20	56.73
Spark-ResNet	99.93%	4093.42	91.63
Spark-ACO-K-Means	97.99%	83.64	1.10

Table 11 shows the diagnosis accuracies, training time and diagnosis time of fault diagnosis models obtained using four different fault diagnosis methods, where the fault diagnosis time is the time it takes to diagnose all the data in the dataset using the trained model. As shown in Table 11, the fault diagnosis accuracy of Spark-ACO-K-Means is 0.33%, 1.86%, and 1.94% lower than that of Spark-RF, Spark-AlexNet, and Spark-ResNet, respectively. However, the model training speed of Spark-ACO-K-Means is 3.58×, 17.51×, and 47.94× faster than that of Spark-RF, Spark-AlexNet, and Spark-ResNet respectively, and the fault diagnosis speed of Spark-ACO-K-Means is 30.57×, 50.57×, and 82.30× faster than that of Spark-RF, Spark-AlexNet, and Spark-ResNet respectively. Compared with Spark-RF, Spark-AlexNet, and Spark-ResNet, the proposed Spark-ACO-K-Means has lower computational complexity, and therefore it can get higher model training efficiency and fault diagnosis efficiency. Besides, before the fault diagnosis model is to be trained, Spark-RF, Spark-AlexNet, and Spark-ResNet all need considerable time to label the dataset, whereas Spark-ACO-K-Means does not require labeling the dataset. Thus, the proposed fault diagnosis method can not only efficiently process large-scale rolling bearing datasets but also achieve a satisfactory fault diagnosis accuracy.

V. CONCLUSION

Facing the massive running-state monitoring data of rolling bearing, a fault diagnosis method of rolling bearing using Spark-based parallel ACO-K-Means clustering algorithm is proposed to achieve efficient and accurate fault diagnosis of rolling bearing. Spark-based three-layer wavelet packet decomposition can efficiently extract eigenvectors from the massive running-state monitoring data of rolling bearing. ACO-K-Means clustering algorithm can not only obtain the global optimal initial clustering centers of K-Means from all eigenvectors, but also optimize the calculation method of distance between the eigenvector and the clustering center using the weighted Euclidean distance measure, which improves the fault diagnosis accuracy. By parallelizing ACO-K-Means clustering algorithm on the Spark platform, the large-scale eigenvectors of rolling bearing can be processed in parallel with multiple worker nodes, which effectively improves the training efficiency and fault diagnosis efficiency of fault diagnosis model of rolling bearing in the big data environment. On the Spark clusters with different number of worker nodes, different size of datasets are used to verify the diagnosis accuracy, model training efficiency, and fault diagnosis efficiency of the proposed method. The results show that the proposed method can fully utilize the computing resources of a Spark cluster to achieve efficient and accurate fault diagnosis for a large-scale rolling bearing dataset. Facing a rolling bearing dataset containing 119.80 GB of eigenvectors, the model training time and fault diagnosis time obtained using the proposed method on the Spark cluster with 8 worker nodes are 85.87% and 87.18% less than that obtained using the proposed method on the Spark cluster with a single worker node respectively, and the fault diagnosis accuracy obtained using the proposed method on the Spark cluster with 8 worker nodes reaches up to 97.99%.

In the practical production, the running-state monitoring data of rolling bearing become more and more and contain a variety of fault information. In the next step, an improved clustering algorithm will be developed to further improve the fault diagnosis accuracy, model training efficiency and fault diagnosis efficiency on the GPU-accelerated Spark platform.

REFERENCES

[1] R. Liu, B. Yang, E. Zio, and X. Chen, "Artificial intelligence for fault diagnosis of rotating machinery: A review," *Mech. Syst. Signal Process.*, vol. 108, pp. 33–47, Aug. 2018.

[2] Y. Chen, T. Zhang, W. Zhao, Z. Luo, and K. Sun, "Fault diagnosis of rolling bearing using multiscale amplitude-aware permutation entropy and random forest," *Algorithms*, vol. 12, no. 9, p. 184, Sep. 2019.

[3] D. K. Appana, M. R. Islam, and J.-M. Kim, "Reliable fault diagnosis of bearings using distance and density similarity on an enhanced k-NN," in *Proc. 3rd Australas. Conf. Artif. Life Comput. Intell. (ACALCI)*, Jan. 2017, pp. 193–203.

[4] X. Yan and M. Jia, "A novel optimized SVM classification algorithm with multi-domain feature and its application to fault diagnosis of rolling bearing," *Neurocomputing*, vol. 313, pp. 47–64, Nov. 2018.

[5] L. Wan, H. Li, Y. Chen, and C. Li, "Rolling bearing fault prediction method based on QPSO-BP neural network and Dempster–Shafer evidence theory," *Energies*, vol. 13, no. 5, p. 1094, Mar. 2020.

[6] L. Wan, Y. Chen, H. Li, and C. Li, "Rolling-element bearing fault diagnosis using improved LeNet-5 network," *Sensors*, vol. 20, no. 6, p. 1693, Mar. 2020.

[7] L. Wen, X. Li, L. Gao, and Y. Zhang, "A new convolutional neural network-based data-driven fault diagnosis method," *IEEE Trans. Ind. Electron.*, vol. 65, no. 7, pp. 5990–5998, Jul. 2018.

[8] J. Wang, Z. Mo, H. Zhang, and Q. Miao, "A deep learning method for bearing fault diagnosis based on time-frequency image," *IEEE Access*, vol. 7, pp. 42373–42383, 2019.

[9] L. Wen, X. Li, and L. Gao, "A transfer convolutional neural network for fault diagnosis based on ResNet-50," *Neural Comput. Appl.*, vol. 32, no. 10, pp. 6111–6124, May 2020.

[10] H. Jiang, X. Li, H. Shao, and K. Zhao, "Intelligent fault diagnosis of rolling bearings using an improved deep recurrent neural network," *Meas. Sci. Technol.*, vol. 29, no. 6, Jun. 2018, Art. no. 065107.

[11] W. Zhang, X. Li, and Q. Ding, "Deep residual learning-based fault diagnosis method for rotating machinery," *ISA Trans.*, vol. 95, pp. 295–305, Dec. 2019.

[12] H. Shao, H. Jiang, H. Zhao, and F. Wang, "A novel deep autoencoder feature learning method for rotating machinery fault diagnosis," *Mech. Syst. Signal Process.*, vol. 95, pp. 187–204, Oct. 2017.

[13] J. Sun, C. Yan, and J. Wen, "Intelligent bearing fault diagnosis method combining compressed data acquisition and deep learning," *IEEE Trans. Instrum. Meas.*, vol. 67, no. 1, pp. 185–195, Jan. 2018.

[14] L. Bai, C. Zhu, Z. Ye, and M. Hui, "Rolling bearings fault diagnosis method based on EWT approximate entropy and FCM clustering," in *Proc. 4th Int. Conf. Electr. Inf. Technol. Rail Trans. (EITRT)*, Oct. 2019, pp. 67–78.

[15] R. Zeng, S. Zhang, R. Zeng, H. Shen, and L. Zhang, "A method of fault detection on diesel engine based on EMD-fractal dimension and fuzzy C-mean clustering algorithm," in *Proc. 29th Chin. Control Decis. Conf. (CCDC)*, May 2017, pp. 7679–7683.

[16] Y. Hu, S. Zhang, A. Jiang, L. Zhang, W. Jiang, and J. Li, "A new method of wind turbine bearing fault diagnosis based on multi-masking empirical mode decomposition and fuzzy C-means clustering," *Chin. J. Mech. Eng.*, vol. 32, no. 1, p. 46, Dec. 2019.

[17] A. R. Ramos, R. D. García, J. L. V. Galdeano, and O. L. Santiago, "Fault diagnosis in a steam generator applying fuzzy clustering techniques," in *Soft Computing for Sustainability Science*. Cham, Switzerland: Springer, 2017, pp. 217–234.

[18] S. Liu, L. Dong, X. Liao, X. Cao, and X. Wang, "Photovoltaic array fault diagnosis based on Gaussian kernel fuzzy C-means clustering algorithm," *Sensors*, vol. 19, no. 7, p. 1520, Mar. 2019.

[19] Z. Shi, W. Song, and S. Taheri, "Improved LMD, permutation entropy and optimized K-means to fault diagnosis for roller bearings," *Entropy*, vol. 18, no. 3, p. 70, Feb. 2016.

[20] X. Zhang, X. Ni, J. Zhao, F. Sun, and Z. Du, "Rolling bearing fault diagnosis using modified K-means cluster analysis," *Vibroeng. Procedia*, vol. 10, pp. 155–160, Dec. 2016.

[21] S. Mjahed, S. El Hadaj, K. Bouzaachane, and S. Raghay, "Engine fault signals diagnosis using genetic algorithm and K-means based clustering," in *Proc. Int. Conf. Learn. Optim. Algorithms, Theory Appl. (LOPAL)*, May 2018, pp. 1–6.

[22] S. Mjahed, S. El Hadaj, K. Bouzaachane, and S. Raghay, "Improved PSO based K-means clustering applied to fault signals diagnosis," in *Proc. Int. Conf. Control, Automat. Diagnosis (ICCAD)*, Mar. 2018, pp. 1–6.

[23] Y. Xu, Y. Sun, J. Wan, X. Liu, and Z. Song, "Industrial big data for fault diagnosis: Taxonomy, review, and applications," *IEEE Access*, vol. 5, pp. 17368–17380, 2017.

[24] H. Miao, H. Zhang, M. Chen, B. Qi, and J. Li, "Two-level fault diagnosis of SF6 electrical equipment based on big data analysis," *Big Data Cognit. Comput.*, vol. 3, no. 1, p. 4, Jan. 2019.

[25] M. B. Imani, M. Heydarzadeh, L. Khan, and M. Nourani, "A scalable spark-based fault diagnosis platform for gearbox fault diagnosis in wind farms," in *Proc. IEEE Int. Conf. Inf. Reuse Integr. (IRI)*, Aug. 2017, pp. 100–107.

[26] W. Yu, T. Dillon, F. Mostafa, W. Rahayu, and Y. Liu, "A global manufacturing big data ecosystem for fault detection in predictive maintenance," *IEEE Trans. Ind. Informat.*, vol. 16, no. 1, pp. 183–192, Jan. 2020.

[27] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Comput. Intell. Mag.*, vol. 1, no. 4, pp. 28–39, Nov. 2006.

[28] S. Na, L. Xumin, and G. Yong, "Research on k-means clustering algorithm: An improved k-means clustering algorithm," in *Proc. 3rd Int. Symp. Intell. Inf. Technol. Secur. Informat.*, Apr. 2010, pp. 63–67.

[29] S. Tang, B. He, C. Yu, Y. Li, and K. Li, "A survey on spark ecosystem: Big data processing infrastructure, machine learning, and applications," *IEEE Trans. Knowl. Data Eng.*, early access, Feb. 24, 2020, doi: 10.1109/tkde.2020.2975652.

[30] V. K. Vavilapalli, A. C. Murthy, C. Douglas, S. Agarwal, M. Konar, R. Evans, T. Graves, J. Lowe, H. Shah, S. Seth, B. Saha, C. Curino, O. O'Malley, S. Radia, B. Reed, and E. Baldeschwieler, "Apache Hadoop YARN: Yet another resource negotiator," in *Proc. 4th Annu. Symp. Cloud Comput.*, Oct. 2013, pp. 1–16.

[31] Z. Wang, Q. Zhang, J. Xiong, M. Xiao, G. Sun, and J. He, "Fault diagnosis of a rolling bearing using wavelet packet denoising and random forests," *IEEE Sensors J.*, vol. 17, no. 17, pp. 5581–5588, Sep. 2017.

[32] CWRU Bearing Data Center. *CWRU Rolling Bearing Dataset*. Accessed: Jan. 10, 2020. [Online]. Available: http://csegroups.case.edu/bearingdatacenter/home

[33] T. Wu, X. Chen, L. Xie, and Z. Qiu, "An optimized K-means clustering algorithm based on BC-QPSO for remote sensing image," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2017, pp. 4766–4769.

[34] G. Zhang, C. Zhang, and H. Zhang, "Improved K-means algorithm based on density canopy," *Knowl.-Based Syst.*, vol. 145, pp. 289–297, Apr. 2018.

[35] U. Yun, G. Lee, and E. Yoon, "Advanced approach of sliding window based erasable pattern mining with list structure of industrial fields," *Inf. Sci.*, vol. 494, pp. 37–59, Aug. 2019.

[36] M. Syakur, B. Khotimah, E. Rochman, and B. Satoto, "Integration K-means clustering method and elbow method for identification of the best customer profile cluster," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 336, pp. 12–17, Apr. 2018.

[37] D.-T. Dinh, T. Fujinami, and V.-N. Huynh, "Estimating the optimal number of clusters in categorical data clustering by silhouette coefficient," in *Proc. Int. Symp. Knowl. Syst. Sci. (KSS)*, Nov. 2019, pp. 1–17.

[38] X. Meng, J. Bradley, B. Yavuz, E. Sparks, S. Venkataraman, D. Liu, J. Freeman, T. Tsai, M. Amde, S. Owen, D. Xin, R. Xin, M. J. Franklin, R. Zadeh, M. Zaharia, and A. Talwalkar, "MLlib: Machine learning in Apache spark," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 1235–1241, 2016.

[39] S. Mishra, U. Sarkar, S. Taraphder, S. Datta, D. Swain, R. Saikhom, S. Panda, and M. Laishram, "Multivariate statistical data analysis-principal component analysis (PCA)," *Int. J. Livestock Res.*, vol. 7, no. 5, pp. 60–78, 2017.

[40] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

LANJUN WAN was born in Hunan, China, in 1982. He received the B.S. and M.S. degrees in computer science and technology from the Hunan University of Technology, Zhuzhou, China, in 2005 and 2009 respectively, and the Ph.D. degree in circuits and systems from Hunan University, Changsha, China, in 2016. He is currently an Assistant Professor with the School of Computer Science, Hunan University of Technology. He has published many research articles in international conferences and journals, such as *JPDC*, *CCPE*, *Parallel Computing*, and *Sensors*. His research interests include industrial big data analysis, industrial equipment fault diagnosis, high-performance computing, and parallel computing. He serves as a Reviewer for the *JPDC* and *CCPE*.

GEN ZHANG was born in Anhui, China, in 1995. He received the B.S. degree in network engineering from West Anhui University, Luan, China, in 2019. He is currently pursuing the M.S. degree in computer science and technology with the Hunan University of Technology, Zhuzhou, China. His research interests include industrial big data analysis and industrial equipment fault diagnosis.

HONGYANG LI was born in Heilongjiang, China, in 1995. He received the B.S. degree in electronic science and technology from the Tianjin University of Technology, Tianjin, China, in 2017. He is currently pursuing the M.S. degree in computer science and technology with the Hunan University of Technology, Zhuzhou, China. His research interests include industrial big data analysis and industrial equipment fault diagnosis.

CHANGYUN LI was born in Hunan, China, in 1972. He received the Ph.D. degree in computer science and technology from Zhejiang University, Hangzhou, China, in 2007. He is currently a Full Professor of computer science and the Dean of the Graduate School, Hunan University of Technology, Zhuzhou, China. He has published many research articles in international conferences and journals, such as *JICT*, *JSW*, and *JCP*. His major research interests include industrial big data analysis, industrial equipment fault diagnosis, intelligent information perception and processing technology, the Internet of Things, and software methodology.