Multi-View Feature Construction Using Genetic Programming for Rolling Bearing Fault Diagnosis

Bo Peng, Ying Bi, Bing Xue, Mengjie Zhang, and Shuting Wan

***Abstract*—Rolling bearing fault diagnosis is an important task in mechanical engineering. Existing methods have several limitations, such as requiring domain knowledge and a large number of training samples. To address these limitations, this paper proposes a new diagnosis approach, i.e., multi-view feature construction based on genetic programming with the idea of ensemble learning (MFCGPE), to automatically construct high-level features from multiple views and build an effective ensemble for identifying different fault types using a small number of training samples. The MFCGPE approach uses a new program structure to automatically construct a flexible number of features from every single view. A new fitness function based on accuracy and distance is developed in MFCGPE to improve the discriminability of the constructed features. To further improve the generalization performance, an ensemble of classifiers based on k-nearest neighbor is created by using the constructed features from every single view. Three bearing datasets and 19 competitive methods are used to validate the effectiveness of the new approach. The results show that MFCGPE achieves higher diagnostic accuracy than all the compared methods on the three datasets with a small number of training samples.**

***Index Terms*—Genetic Programming, Feature Construction, Limited Training Data, Fault Diagnosis, Rolling Bearing**

I. Introduction

Rolling bearings are important supporting equipment and have been used in all kinds of rotating machinery, such as electric motors, turbine generators, and high-speed trains [1]. The performance of rolling bearings can be affected by the high temperature, high pressure, and alternating load during the equipment operation [2]. The rolling bearings are often inevitably damaged as the running time increases [2]. The damage of rolling bearings will lead to mechanical equipment failures and the percentage of failures caused by rolling bearing damage reaches nearly 30% [3]. Therefore, it is important to identify the faults of rolling bearing to monitor bearing status, ensure machinery safety, reduce economic losses, and avoid casualties.

In recent years, many methods have been developed for rolling bearing fault diagnosis. Because the vibration signals are easy to collect and often contain information of the running states, the studies on using vibration signals for fault diagnosis have gained much attention. The collected vibration signals often have background noise or information loss. A variety of signal processing methods, such as wavelet transform [4], empirical mode decomposition [5], fast spectral kurtosis [6], maximum correlated kurtosis deconvolution [7], and variational mode decomposition [8], have been used to suppress the interference of noise and harmonics and strengthen signal characteristics. Experts have conducted the spectrum analysis on the processed signals and recognized the fault characteristic frequency for fault diagnosis [9]. However, these methods do not provide satisfactory results and require extensive domain expertise, which is time-consuming and expensive.

To automatically identify faults, methods based on machine learning have been developed for rolling bearing feature analysis and fault diagnosis. These methods extract various types of features from the vibration signals and use traditional classification algorithms to perform fault diagnosis [10]. The commonly used features include simple statistical features of time-domain and frequency-domain, and the non-linear evaluation indicators, such as fractal dimension [11], Lyapunov exponent [12], and entropy-based features [13], [14], [15]. Typically, a large number of features are extracted to describe the vibration signal and these features may contain redundant or irrelevant features, which may reduce the fault diagnosis accuracy. Therefore, feature selection methods, such as max-relevance and min-redundancy [16], ReliefF [17], Laplacian score [18], principal component analysis [19], local discriminant analysis [20], and margin fisher analysis [21], have been employed to select a subset of important features for effective fault diagnosis. To further improve the fault diagnosis performance, different classification methods have been explored for different tasks, including k-nearest neighbor (KNN) [22], artificial neural network (ANN) [23], extreme learning machine (ELM) [24], and support vector machine (SVM) [25]. However, these methods typically contain several steps of signal processing, feature design, feature selection, and classifier learning. These steps need to be appropriately connected to achieve accurate fault diagnosis. In addition, most of these steps require domain expertise, which causes the diagnosis method to be effective only on a specific fault diagnosis task and unable to be generalized to other even similar tasks.

Deep learning is an advanced machine learning approach that has been applied to fault diagnosis [26], [27]. Most deep learning methods are based on neural networks (NNs). Different types of neural networks, such as deep belief networks, sparse autoencoders, and convolutional neural networks, have also been developed for effective fault diagnosis [28], [29], [30]. These methods can automatically learn features from the vibration signals and train classifiers for effective fault diagnosis. However, the NN-based methods often use a large

Bo Peng, North China Electric Power University, CHINA.
Ying Bi, Victoria University of Wellington, NEW ZEALAND.
Bing Xue, Victoria University of Wellington, NEW ZEALAND.
Mengjie Zhang, Victoria University of Wellington, NEW ZEALAND.
Shuting Wan, North China Electric Power University, CHINA.
Corresponding Author: Ying Bi (ying.bi@ecs.vuw.ac.nz)

number of training samples to build models/classifiers for fault diagnosis. In real-world scenarios, the training samples are often difficult to obtain and require extensive manual effort to label. Furthermore, designing an effective architecture for the NN model typically requires rich expertise in the problem and NN domains. Therefore, it is necessary to develop a new intelligent fault diagnosis method that can achieve good performance using a small number of training samples.

Typically, the feature quality is important for effective rolling bearing fault diagnosis. Traditional methods use many ways, such as signal processing, feature selection, and feature learning, to improve the quality of the extracted features [15], [21], [25]. Feature construction is an effective way to generate new informative and high-level features from the original low-level features [31]. As shown in existing work [11], [12], [13], [14], [15], the features of vibration signals can be extracted from multiple views and each view represents different characteristics. However, feature construction, particularly constructing features from multiple views, which can create informative features to improve the fault diagnosis, has not been extensively explored in this field.

Evolutionary computation (EC) methods ordinarily do not use extensive domain knowledge to find solutions and have been successfully applied to many difficult problems [32], [33], [34]. Genetic programming (GP) is an EC technique that has been widely used as one of the most popular feature construction methods [31], [35]. Unlike other EC methods, which use a fixed-length representation, GP has a variable-length tree-based representation, enabling it to automatically construct high-level features from low-level features in a more flexible way [35]. For feature construction, GP typically evolves models that consist of a set of operators (such as $+$, $-$ and $\times$) and the original features. The original features are operated by these operators and new features are then generated. The tree-based solutions of GP potentially have high interpretability to provide insights into what features are important for construction and why the constructed features are effective. Owing to these advantages, many GP methods have been developed to feature construction in various problems and achieved promising results [36], [37], [38].

However, existing GP methods need several improvements to construct effective features for fault diagnosis using a small number of training samples. First, the features extracted from the vibration signals can be multiple views, such as time-domain view and the frequency-domain view [39]. Each view represents different characteristics of the data. Existing methods often simply concatenate all the features from different views, which may not be effective. GP has seldom been developed for constructing high-level features from different views (i.e., multi-view feature construction). Second, existing fitness measures of GP for feature construction often use classification accuracy, which may cause the overfitting issue, particularly when the number of training samples is small. Third, the generalization performance of the features constructed by GP can be further improved by constructing ensembles of classifiers for classification. However, this has seldom been explored. Therefore, this paper develops a new GP approach to address these limitations.

The goal of this paper is to propose a new intelligent approach, i.e., multi-view feature construction based on GP with the idea of ensemble learning (hereafter called MFCGPE), to rolling bearing fault diagnosis using a small number of training samples. The proposed approach is able to construct a flexible number of high-level features from every single view and build an effective ensemble using the constructed multi-view features to achieve high generalization performance when the number of training samples is small. The performance of MFCGPE will be tested on three datasets of varying difficulty and compared with 19 competitive methods. Further analysis will be conducted to show the effectiveness of the constructed features and the ensemble.

The main contributions of this paper are summarized in the following four aspects.

1) A new program structure is developed in MFCGPE to allow it to construct high-level features from multiple views. More importantly, the number of constructed features can be adaptively determined without being pre-defined.
2) A new fitness function based on accuracy and distance is proposed in MFCGPE to enable the constructed features to be accurate and discriminative when the number of training samples is small. The new fitness function is able to maximize the classification performance and the inter-class distances of training samples, and minimize the intra-class distances of training samples.
3) An ensemble is created using the features constructed by MFCGPE from multiple views and using KNN to make predictions for unseen test samples. Using ensemble for classification can further improve the generalization performance, particularly when the number of training samples is small.
4) A new intelligent diagnosis approach, namely MFCGPE, is developed to achieve effective fault diagnosis of rolling bearings with the use of a small number of training samples. MFCGPE can achieve better results than 19 competitive methods on three datasets. Specifically, MFCGPE can achieve a maximal and average accuracy of 100% and above 99% on three fault datasets with only five training samples per class.

The rest of this paper is organized as follows: Section II briefly introduces the GP algorithm and reviews its applications on feature construction and fault diagnosis. Section III describes the MFCGPE approach in detail. Section IV designs the experiments. The experimental results are analyzed and compared in Section V. Section VI further analyses the proposed approach. Section VII presents conclusions and future work.

II. Background and Related Work

A. Genetic Programming (GP)

Unlike other EC techniques such as genetic algorithms (GAs) that have a fixed-length representation, GP has a variable-length representation [40]. An individual of GP is typically represented using a tree-based structure, as shown in Figure 1. This example tree/program consists of internal nodes (the functions or operators selected from the function

set) and leaf nodes (the arguments or constants selected from the terminal set). This example tree can be mathematically expressed as $5\div(y+8)+(2\times x)$, where $+$, $\times$, and $\div$ (protected division, return 0 if the divisor is 0) are the internal nodes, and x, y, 2, 5, and 8 are the leaf nodes. This equation can also be treated as a newly constructed feature, where x and y are two original features.

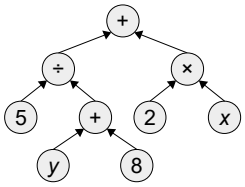


Fig. 1. An example program of GP.

GP can automatically select functions from the function set and terminals from the terminal set to evolve the best trees for solving a task via an evolutionary process. Figure 2 shows the overall evolutionary process of GP. First, a population of computer programs are randomly initialized in the search space. Then, each individual (program) in the population is evaluated using a fitness function and assigned a fitness value. At each generation, a new population of individuals are generated using genetic operators, i.e., *Elitism*, *Crossover*, and *Mutation* operators. The *Elitism* operation is to copy the best individuals from the current generation to the next generation. A number of individuals are selected based on their fitness values via *Tournament selection* to be used as parents to generate new individuals using *Crossover* and *Mutation* operators. The *Crossover* operation is to exchange the subtrees of two parents to generate new offspring. The *Mutation* operation is to randomly delete a subtree of the parent and grow a new subtree from that node. The new population of individuals are evaluated and evolved generation by generation. When a termination criterion is satisfied, the evolutionary process stops, and the best individual is obtained.

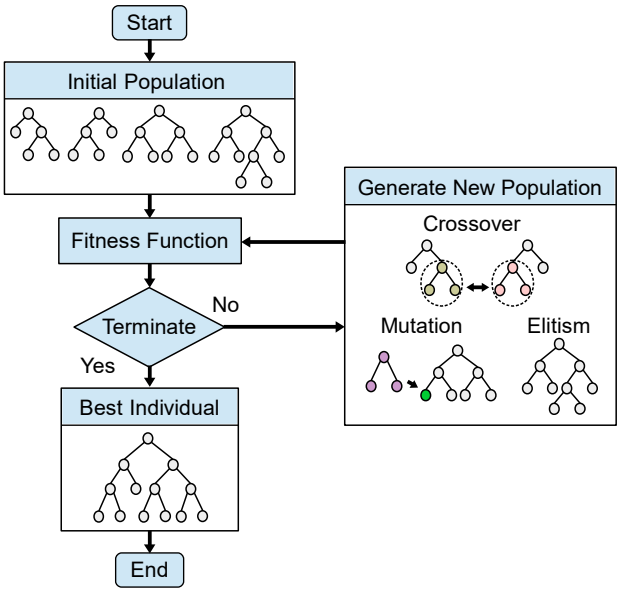


Fig. 2. Evolutionary process of GP.

B. GP for Feature Construction

In recent years, many GP based feature construction methods have been proposed, in which the arithmetic and logical operators are usually used as the function set, and the low-level features are usually used as the terminal set. Otero et al. [41] used information gain ratio as the fitness function of GP to construct features. Muharram et al. [42] comprehensively compared four different fitness functions based on information gain in GP for feature construction. Guo et al. [43], [44] used the Fisher criteria and its improved version as the fitness function of GP. Neshatian et al. [45] proposed a multiple feature construction method for symbolic learning classifiers, where the constructed features are evaluated by a fitness function that maximizes the purity of the class interval. The above methods belong to the filter-based feature construction methods. Guo et al. [36] proposed a method based on GP and KNN to classify EEG signals, which achieved a classification accuracy of 99% on one dataset. Bi et al. [37] proposed a GP method combined with image-related operators and SVM for image classification, which obtained better accuracy than the deep learning methods on some datasets. Aslam et al. [38] combined GP and KNN for automatic modulation classification, and the method achieved better classification performance. The above methods belong to the wrapper-based feature construction methods. Tran et al. [46] developed a fitness function using the classification accuracy and a distance measure to improve the performance of the constructed features on high-dimensional data classification tasks and discussed the impact of the number of constructed features on classification accuracy. Ma et al. [47] designed a hybrid fitness function that combined information gain ratio and the error rate of a classification algorithm, and proposed a feature construction strategy that obtained multiple high-level features using a single GP.

C. GP for Fault Diagnosis

To the best of our knowledge, GP has rarely been applied to fault diagnosis of mechanical equipment. In [48], GP was used as a binary classifier for fault diagnosis of rolling bearing. The classification performance of GP with the use of statistical features, spectral features, and the combination of statistical and spectral features are compared. The results showed that the combined features achieved better fault classification accuracy. Guo et al. [49] proposed a GP-based rolling bearing fault diagnosis method, in which a fitness function based on Fisher criteria was developed to enable GP to construct the high order of moments of vibration signals as informative features. This method used two classification algorithms for multi-class fault classification. Xuan et al. [50] proposed a gear fault diagnosis method that combines GP and SVM, where a distance measure based fitness function was developed and the power spectral features of vibration signals were used to construct high-level features. These two methods only construct a single feature for fault diagnosis, which may not be effective when the machine working conditions become more complicated.

In summary, although these methods successfully show that GP offers possibilities for dealing with fault diagnosis, there are still some problems that need to be addressed. The GP

based feature construction methods in [46], [47], [45] have discussed the effect of the number of constructed features on classification performance. However, these methods set the number of constructed features according to prior knowledge and multiple trials, therefore the adaptability of them is poor. The GP based fault diagnosis method in [50], [48], [49] only used the features of a single view for feature construction. However, since the features of different views have both internal relations and interval differences, using only single view features may ignore the characteristics of the samples. Moreover, these GP based fault diagnosis methods use sufficient training data to construct features and perform classification and do not consider the scenario of a small number of training samples. In general, a small number of training samples can not well represent the class distribution information comprehensively and often leads to poor generalization performance. To address the above issues, this paper proposes a new GP based fault diagnosis approach (i.e., MFCGPE) to adaptively construct a flexible number of informative features from multiple views for representing sample comprehensively, and create an ensemble using these constructed features for effective fault diagnosis with the use of a small number of training samples.

III. The Proposed Approach

In this section, the details of the MFCGPE fault diagnosis approach will be introduced, including the algorithm overview, the program structure, the function set, the terminal set, the fitness function, and ensemble construction for fault diagnosis.

A. Overview of MFCGPE

Figure 3 shows the overall structure of MFCGPE to rolling bearing fault diagnosis with a small number of training samples. First, the collected vibration signals of rolling bearings are transformed into three different-view features by calculating the statistical values of time waveform and frequency spectrum. The 16 time-domain features (TDF), the 13 frequency-domain features (FDF), and the combination of TDF and FDF (named TFDF features) are represented by $View_1$, $View_2$, and $View_3$, respectively. Second, the transformed data set is divided into the training set and the test set, and three independent GPs are utilized to construct high-level features of each single-view feature set, respectively. The program structure and the function set used for feature construction from different views are the same, but the terminal set is different. Only the training set is used for the evolutionary process of GP to construct high-level features that are expected to have a small intra-class distance and a large inter-class distance for effective fault diagnosis. Third, the constructed features are provided as the inputs of classification algorithms for fault diagnosis. We use the features constructed from each view to feed three classifiers/KNNs individually, which are combined via majority voting to form an ensemble to make a good prediction for the unseen test set.

B. Program Structure

To enable MFCGPE to construct a flexible number of high-level features, a new program structure with the input,

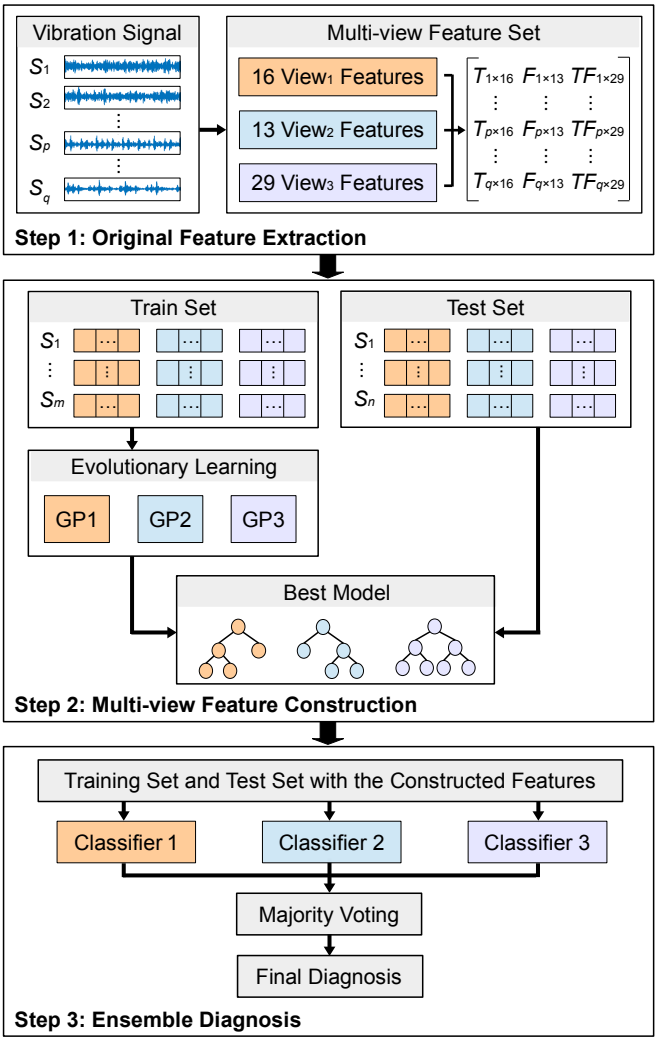


Fig. 3. Overall structure of the MFCGPE fault diagnosis approach.

feature construction, feature combination, and output layers are developed based on strongly-typed GP (STGP) [51]. These layers are connected in a bottom-up manner. Each layer has a specific function and type constraint. The input layer takes the original features as inputs, which are the terminals of GP. The feature construction layer transforms the original features into high-level features. The feature combination layer combines multiple constructed features into a vector to comprehensively describe the sample. The output layer returns the constructed feature vector as outputs for fault diagnosis. The tree depths of the input and output layers are one, and those of the feature construction and feature combination layers are automatically adjusted according to the given task. Figure 4 shows the program structure and an example program of MFCGPE, respectively. This example program represents a feature vector containing three constructed features. The detailed description of the operators and the terminals will be described in the following subsections.

C. Function Set

The function set of MFCGPE is composed of two types of operators, i.e., the feature construction operators and the

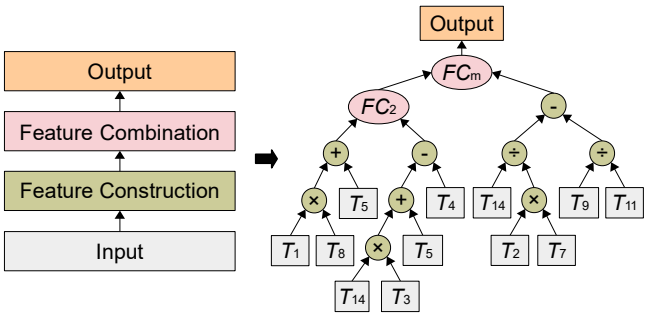


Fig. 4. Program structure of MFCGPE (left) and an example program (right) that can be evolved by MFCPGE.

feature combination operators. Table I lists the detailed information of the function set. Four arithmetic operators including $+$, $-$, $\times$, and $\div$ are for feature construction, where $\div$ is protected by returning 0 if the divisor is 0. The inputs of $+$, $-$, $\times$, and $\div$ operators are two features, and the output of them is a new feature. It should be noted that their outputs/features can be further used for feature construction. To achieve feature combination, two new operators (FC_2 and FC_{m}) are designed to combine multiple features into a vector, where the inputs of FC_2 are two features, and the inputs of FC_{m} can be one feature, one vector, or two vectors. The use of these feature combination operators enables MFCGPE to adaptively construct multiple high-level features without presetting the number of constructed features.

TABLE I
FUNCTION SET OF GP

SYMBOL	INPUT	OUTPUT	DESCRIPTION
$+$	2 FEATURES	1 FEATURE	ADDITION
$-$	2 FEATURES	1 FEATURE	SUBTRACTION
$\times$	2 FEATURES	1 FEATURE	MULTIPLICATION
$\div$	2 FEATURES	1 FEATURE	PROTECTED DIVISION
FC_2	2 FEATURES	1 VECTOR	CONCATENATION
FC_{m}	2 VECTORS OR 1 VECTOR/FEATURE	1 VECTOR	CONCATENATION

D. Terminal Set

The terminal set of MFCGPE is composed of the low-level features extracted from the vibration signals. The terminal sets for feature construction from different views are different. For $View_1$ (only consider the time-domain characteristics of rolling bearing signals), the terminal set contains 16 statistical features $T_1 \sim T_{16}$, which are extracted through performing the simple mathematical calculation on the amplitude of the raw vibration signals [25]. Table II lists the terminal set used under $View_1$, where $s(i)$ represents the time signal, and N represents the number of the signal data points. $T_1 \sim T_{16}$ represent the vibration signal's mean value, standard deviation, square root amplitude, absolute mean value, skewness, kurtosis, variance, maximum value, minimum value, peak-to-peak value, waveform index, peak index, pulse index, margin index, skewness index and kurtosis index, respectively.

TABLE II
TERMINAL SET CORRESPONDING TO $View_1$

SYMBOL	FORMULA	SYMBOL	FORMULA
T_1	$\frac{1}{N}\sum_{i=1}^{N} s(i)$	T_9	$\min\lvert s(i)\rvert$
T_2	$\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}[s(i)-T_1]^2}$	T_{10}	$T_8 - T_9$
T_3	$(\frac{1}{N}\sum_{i=1}^{N}\sqrt{\lvert s(i)\rvert})^2$	T_{11}	$\frac{T_2}{T_4}$
T_4	$\frac{1}{N}\sum_{i=1}^{N}\lvert s(i)\rvert$	T_{12}	$\frac{T_8}{F_2}$
T_5	$\frac{1}{N}\sum_{i=1}^{N}(s(i))^3$	T_{13}	$\frac{T_8}{F_4}$
T_6	$\frac{1}{N}\sum_{i=1}^{N}(s(i))^4$	T_{14}	$\frac{T_8}{T_3}$
T_7	$\frac{1}{N}\sum_{i=1}^{N}(s(i))^2$	T_{15}	$\frac{T_5}{(\sqrt{T_7})^3}$
T_8	$\max\lvert s(i)\rvert$	T_{16}	$\frac{T_6}{(\sqrt{T_7})^2}$

For $View_2$ (only consider the frequency-domain characteristics of rolling bearing signals), the terminal set contains 13 statistical features $F_1 \sim F_{13}$, which are extracted through performing the mathematical calculation on the amplitude of frequency spectrum that was obtained through Fast Fourier Transform for the vibration signals [25]. Table III lists the terminal set used under $View_2$, where $f(j)$ represents the frequency spectrum of time signal $s(i)$, M represents the number of spectrum lines, and A_j represents the frequency value of the jth spectrum line. F_1 represents the energy of the vibration signal in frequency-domain, $F_2 \sim F_5$ represent the position variation of the main frequency band, and $F_6 \sim F_{13}$ represent the concentration and dispersion of the frequency spectrum.

TABLE III
TERMINAL SET CORRESPONDING TO $View_2$

SYMBOL	FORMULA	SYMBOL	FORMULA
F_1	$\frac{\sum_{j=1}^{M} f(j)}{M}$	F_8	$\frac{\sum_{j=1}^{M}[f(j)-F_1]^4}{M(F_6)^2}$
F_2	$\frac{\sum_{j=1}^{M}(A_j f(j))}{\sum_{j=1}^{M} f(j)}$	F_9	$\sqrt{\frac{\sum_{j=1}^{M}[(A_j-F_5)^2 f(j)]}{M}}$
F_3	$\sqrt{\frac{\sum_{j=1}^{M}[A_j^2 f(j)]}{\sum_{j=1}^{M} f(j)}}$	F_{10}	$\frac{F_9}{F_2}$
F_4	$\sqrt{\frac{\sum_{j=1}^{M}[A_j^4 f(j)]}{\sum_{j=1}^{M}[A_j^2 f(j)]}}$	F_{11}	$\frac{\sum_{j=1}^{M}[(A_j-F_2)^3 f(j)]}{M(F_9)^3}$
F_5	$\frac{\sum_{j=1}^{M}[A_j^2 f(j)]}{\sqrt{[\sum_{j=1}^{M}(A_j^4 f(j))][\sum_{j=1}^{M} f(j)]}}$	F_{12}	$\frac{\sum_{j=1}^{M}[(A_j-F_2)^4 f(j)]}{M(F_9)^4}$
F_6	$\frac{\sum_{j=1}^{M}[f(j)-F_1]^2}{M-1}$	F_{13}	$\frac{\sum_{j=1}^{M}[\sqrt{\lvert A_j-F_2\rvert} f(j)]}{M\sqrt{F_9}}$
F_7	$\frac{\sum_{j=1}^{M}[f(j)-F_1]^3}{M(\sqrt{F_6})^3}$	–	–

For $View_3$ (considering both the time-domain and frequency-domain characteristics of rolling bearing signals), the terminal set contains 29 features ($TF_1 \sim TF_{29}$) composed by the features of $View_1$ and $View_2$, where the features $TF_1 \sim TF_{16}$ are equal to the features $T_1 \sim T_{16}$ in $View_1$, and the features $TF_{17} \sim TF_{29}$ are equal to the features $F_1 \sim F_{13}$ in $View_2$. There are two reasons for using TDF features and FDF features to form the multi-view features. One is that the extraction of TDF and FDF features is simple statistical calculations, while the extraction of the time-frequency domain features typically needs complicated signal processing

operations and manual participation. The other is that [48] shows that using the combination of the TDF and FDF features for feature construction is effective for constructing new features. Different from most of the existing works, the proposed method constructs features from these three views, individually, which aims to generate a group of effective and diverse features to comprehensively describe a sample. The effectiveness and diversity of the constructed features can also improve the high generalization performance of the ensemble, which will be described in the later subsection.

E. Fitness Function

An effective fitness function is crucial for guiding GP to construct high-level features for fault diagnosis. When the number of training samples is small, the features constructed by GP can easily achieve good classification performance on the training set, but poor generalization performance on the test set. To address this issue, a new fitness function based on the classification accuracy and the distance measure of the training samples is proposed to guide the search of MFCGPE. To calculate the diagnosis accuracy, KNN is used to perform classification based on the constructed features. The reason for using KNN is that it is a simple classification algorithm, easy to implement, and treats each feature equally without any feature weighting or selection [22], [52]. With the use of KNN, MFCGPE can automatically construct discriminative features and avoid redundant or irrelevant features. The distance measure is to minimize the intra-class distance and maximize the inter-class distance of the training samples based on the constructed features. Using such a distance measure will group the samples in the same class and enlarge the differences of the samples in different classes, which potentially improves the discriminability of the constructed features. This also helps to improve the effectiveness of KNN as it is based on distance. Based on the above analysis, the new fitness function to be maximised is formulated as follows.

$$Fit = (Acc + Dist)/2 \tag{1}$$

where Acc represents the diagnosis accuracy of KNN using the constructed features and $Dist$ represents the distance measure. Only the training set is used in the fitness evaluation step, and the constructed features are transformed into the range of $[0, 1]$ through the min-max normalization method. The Acc is calculated using the 5-fold cross-validation scheme. That is, the new generated features and labels of the training set are divided into 5 folds, evenly. Each time, one fold is used as the sub-test set, and the other two folds are used as the sub-training set. The average value of the three sub-test sets is set as the diagnosis accuracy.

The $Dist$ is calculated according to Equations (2)-(5), which evaluate the distance of the training samples with the constructed features. The calculation of $Dist$ is based on the Euclidean distance. For the given samples X_k=$\{x_{k1}, x_{k2}, ..., x_{kn}\}$ and X_l=$\{x_{l1}, x_{l2}, ..., x_{ln}\}$, the Euclidean distance between them can be calculated as Equation (2), where x_{ko} and x_{lo} represent the features of samples X_k and X_l, respectively, and n is the number of features. The calculation of the intra-class and inter-class distances is based on Equation (3) and Equation (4), where Ω represents the samples in one single class, and N is the number of samples in the Ω class. Equation (5) is the calculation of $Dist$, where the *sigmoid* function is used to transform the difference value of the inter-class and intra-class distances into the range of $[0, 1]$.

$$d(X_k, X_l) = \sqrt{\sum_{o=1}^{n}(x_{ko} - x_{lo})^2} \tag{2}$$

$$D_{intra}(\Omega_i) = \frac{1}{N_i N_i}\sum_{k=1}^{N_i}\sum_{l=1}^{N_i} d(X_k^{(i)}, X_l^{(i)}) \tag{3}$$

$$D_{inter}(\Omega_i, \Omega_j) = \frac{1}{N_i N_j}\sum_{k=1}^{N_i}\sum_{l=1}^{N_j} d(X_k^{(i)}, X_l^{(j)}) \tag{4}$$

$$Dist = \frac{1}{1 + e^{-(\min(\mathrm{D}_{inter}) - \max(\mathrm{D}_{intra}))}} \tag{5}$$

The proposed fitness function optimizes the classification accuracy and the distances of the training samples, simultaneously. When the fitness value approaches one, it indicates that the features constructed by GP have the best classification performance and the training samples have a small intra-class distance and a large inter-class distance.

F. Ensemble for Fault Diagnosis

The MFCGPE system is able to find the three best GP trees/programs that construct features from different views. Since the high-level features are constructed from different views, they are diverse and effective. To effectively use the constructed features, an ensemble is created for fault diagnosis. Consistent with the fitness evaluation process, the ensemble uses three KNNs as the base classifiers, and each KNN uses the features constructed from a single view. The overall fault diagnosis system using ensemble is illustrated in Figure 5.

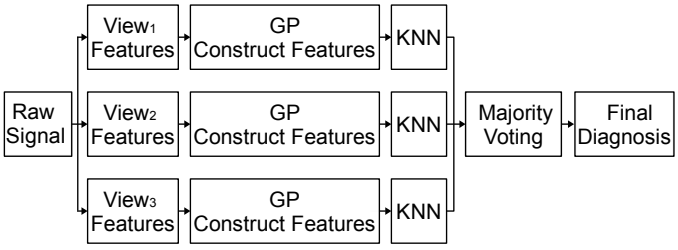


Fig. 5. Architecture of ensemble fault diagnosis system.

The training set (including the transformed feature sets from three different views and the corresponding class labels) are fed into three KNNs, and these three KNNs are used as base classifiers of an ensemble. This ensemble will have high generalization performance since the inputs are the features constructed from different views. This constructed ensemble is applied to predict the class labels for the unseen samples in the test set. In the process, the majority voting algorithm is employed because it is a simple and common method to comprehensively consider the results obtained by the multiple base classifiers [53]. For a test sample, among the class labels/fault types obtained by three KNNs, the fault type with the largest number of votes is chosen as the final fault type.

IV. Experimental Design

A. Dataset Description

The mechanical equipment typically operates normally, and the failure data that can be collected is limited. Collecting a large number of labeled samples requires high cost and a long period of time. Therefore, the goal of this paper is to deal with fault diagnosis using the limited number of training samples. In the experiments, three datasets with a small number of training samples are employed to evaluate the effectiveness of MFCGPE. In each dataset, only five samples of each running condition are randomly selected as the training set, and the remaining samples are used as the test set. The three datasets are described as follows.

1) NCEPU [25]: It is a bearing fault dataset collected by North China Electric Power University (NCEPU) and contains vibration signals of six fault types. Figure 6 shows the used test rig of NCEPU. The six rolling bearing running conditions are the normal state (NOR), inner ring fault (IRF), outer ring fault (ORF), rolling element fault (REF), inner & outer ring compound fault ($IOCF$), and rolling element & outer ring compound fault ($ROCF$), which are simulated by manufacturing defects on normal bearings using electrical discharge machining (EDM) technology. All vibration signals are collected with a sampling frequency of 12,000 Hz. The first 102,400 data points of vibration signals under each running condition are divided into 50 samples on average and there is no overlap between each sample. Table IV lists the detailed information of the NCEPU dataset. Figure 7 shows the time domain waveform of vibration signals under six running conditions in NCEPU.

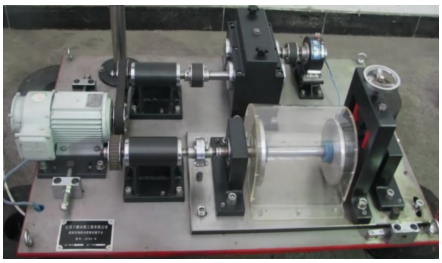


Fig. 6. NCEPU test rig.

TABLE IV
Description of the NCEPU dataset

CLASS LABEL	RUNNING CONDITION	TRAINING SAMPLES	TEST SAMPLES
1	NORMAL	5	45
2	INNER RING FAULT	5	45
3	OUTER RING FAULT	5	45
4	ROLLING ELEMENT FAULT	5	45
5	INNER AND OUTER RING COMPOUND FAULT	5	45
6	ROLLING ELEMENT AND OUTER RING COMPOUND FAULT	5	45

2) CWRU [54]: It is a bearing fault dataset collected by Case Western Reserve University (CWRU) and contains vibration signals of ten fault types and fault degrees. Figure 8 shows the used test rig of CWRU. All faults are generated by manufacturing defects on the inner ring, outer ring, and rolling element surfaces using EDM, and the same fault category with three defect diameters (0.007, 0.014, and 0.021 inches, respectively). The ten different rolling bearing running conditions are the normal state (NOR), three kinds of inner ring fault (*IRF07*, *IRF14*, and *IRF21*), three kinds of outer ring fault (*ORF07*, *ORF14*, and *ORF21*), and three kinds of rolling element fault (*REF07*, *REF14*, and *REF21*), respectively. IRF, ORF, and REF indicate inner ring fault, outer ring fault, and rolling element fault, respectively. 07, 14, and 21 indicate that the defect diameter is 0.007, 0.014, and 0.021 inches, respectively. The vibration signals of driver-end bearing under 1,797 r/min are collected with a sampling frequency of 12,000 Hz. Similar to the NCEPU dataset, the first 102,400 data points of vibration signals under each running condition are divided into 50 samples on average. Five samples of each running condition are randomly selected to form the training set, and the remaining samples are used as the test set. Table V lists the detailed information of the CWRU dataset. Figure 9 shows the time domain waveform of vibration signals under ten running conditions in CWRU.

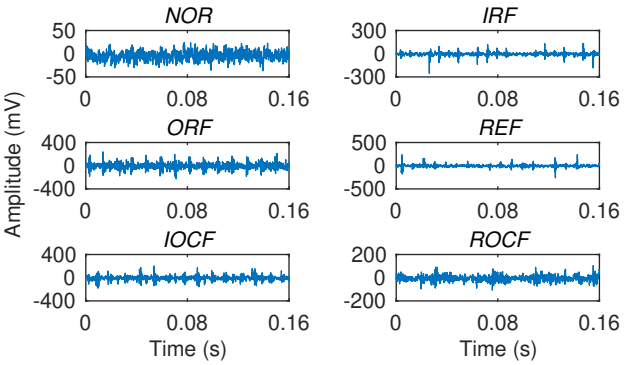


Fig. 7. Time domain waveform of vibration signals under six running conditions in NCEPU.

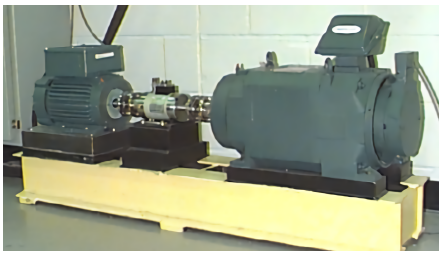


Fig. 8. CWRU test rig.

3) XJTU [55]: It is a run-to-failure bearing fault dataset collected by Xi'an Jiaotong University (XJTU), in which the reason for fault occurrence is the natural damage as the running time increases. Figure 10 shows the used test rig of XJTU. In the experiments, the vibration signals of the test bearings are collected with a sampling frequency of 25,600 Hz until the bearing fails, and the time interval for each sampling is one minute. The vibration signals of the last two recorded data file of the Bearing 2_1, Bearing 2_2, and Bearing 2_3 datasets are used as the analyzed data. The fault types are inner ring fault (IRF), outer ring fault (ORF), and cage fault (CF).

TABLE V
DESCRIPTION OF THE CWRU DATASET

CLASS LABEL	RUNNING CONDITION	DEFECT SIZE (IN)	TRAINING SAMPLES	TEST SAMPLES
1	NORMAL	0	5	45
2	INNER RING FAULT	0.007	5	45
3	INNER RING FAULT	0.014	5	45
4	INNER RING FAULT	0.021	5	45
5	OUTER RING FAULT	0.007	5	45
6	OUTER RING FAULT	0.014	5	45
7	OUTER RING FAULT	0.021	5	45
8	ROLLING ELEMENT FAULT	0.007	5	45
9	ROLLING ELEMENT FAULT	0.014	5	45
10	ROLLING ELEMENT FAULT	0.021	5	45

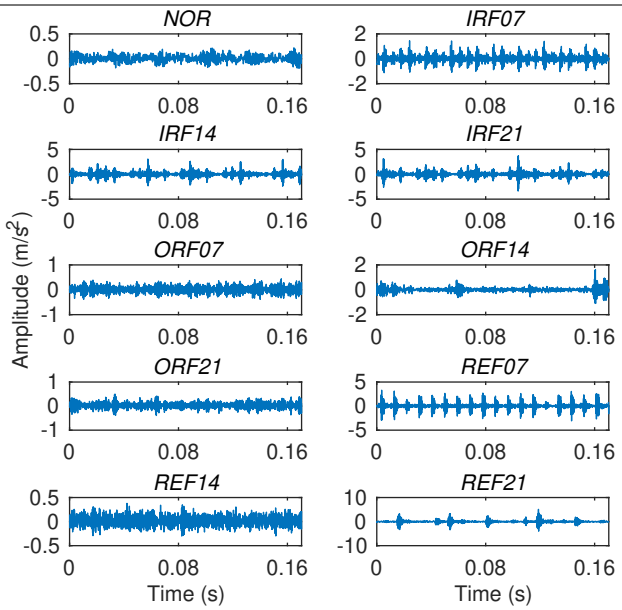


Fig. 9. Time domain waveform of vibration signals under ten running conditions in CWRU.

The signals of the first two recorded data file of the Bearing 2_1 dataset is used as the normal (NOR) vibration signals for analysis. Every vibration signal contains 32,768 data points and is divided into 32 samples on average for conducting the experiments. Each sample contains 2,048 data points. Five samples under each running condition are randomly selected to form the training set, and the remaining samples are used as the test set.Table VI lists the detailed information of the XJTU dataset. Figure 11 shows the time domain waveform of vibration signals under four running conditions in XJTU.

TABLE VI
DESCRIPTION OF THE XJTU DATASET

CLASS LABEL	RUNNING CONDITION	TRAINING SAMPLES	TEST SAMPLES
1	NORMAL	5	27
2	INNER RING FAULT	5	27
3	OUTER RING FAULT	5	27
4	CAGE FAULT	5	27

B. Comparison Methods

To verify the effectiveness of MFCGPE, four categories of competitive methods, i.e., 19 methods, are employed for

Fig. 10. XJTU test rig.

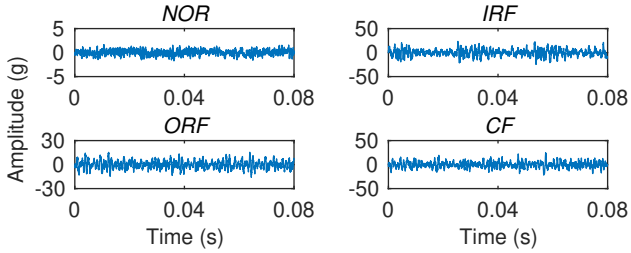


Fig. 11. Time domain waveform of vibration signals under four running conditions in XJTU.

comparisons. The first category includes five different classification algorithms, i.e., KNN, SVM, Naive Bayes (NB), logistic regression (LR), and multilayer perceptron (MLP), which use raw signals amplitude to train the classifiers for fault classification. The second category are KNN using five manually crafted features, i.e., TDF, FDF, multi-domain features (MDF) [25], modified multi-scale symbolic dynamic entropy (MMSDE) [14], and improved multi-scale dispersion entropy (IMDE) [15], for fault classification. The TDF and FDF features have been described in Section III-D. The numbers of features in MDF, MMSDE, and IMDE are 37, 20, and 20, respectively. The comparisons aim to investigate whether the features constructed by MFCGPE are more effective for fault diagnosis than these manually crafted features.

The third category contains six GP based methods using TDF, FDF, and TFDF (described in Section III-D) as the input features to construct high-level features. The constructed features are used as the inputs of the KNN classifier for fault classification. Two types of GP based methods are used, i.e., the GPS methods (using GP to construct a single feature) and the GPM methods (using GP to construct multiple features). The fitness function of these methods is the same as MFCGPE. The comparisons can investigate the effect of the number of constructed features on diagnosis accuracy. The GPS/GPM-TDF, GPS/GPM-FDF, and GPS/GPM-TFDF methods use the TDF, FDF, and TFDF features for feature construction, respectively. According to [46], the number of features constructed by the GPM based methods is the same as the number of the classes in the dataset. For example, GPM-TDF, GPM-FDF, and GPM-TFDF construct 10 new features for classifying the CWRU dataset with 10 classes. The comparisons aim to investigate whether MFCGPE can outperform other GP based

feature construction methods that construct features.

The fourth category of the comparison methods are three ensemble diagnosis methods (i.e., OFE, GPSE, and GPME). These methods use three KNNs as base classifiers and the majority voting algorithm to build an ensemble. In OFE (i.e., ensemble using the original features), the TDF, FDF, and TFDF features are fed into three KNNs, respectively. In GPSE (i.e., ensemble using the features constructed by GPS), the features constructed by GPS-TDF, GPS-FDF, and GPS-TFDF are fed into three KNNs to build an ensemble. In GPME (i.e., ensemble using the features constructed by GPM), the features constructed by GPM-TDF, GPM-FDF, and GPM-TFDF are fed into three KNNs, respectively. The comparisons aim to investigate whether MFCGPE can beat the ensemble diagnosis methods that use the original features and the features constructed by GP.

C. Parameter Settings

In MFCGPE, the GP related parameters are common settings and refer to those in [56]. The population size and the maximal number of generations are set as 100 and 50, respectively. The rates of crossover, mutation, and elitism are set as 0.8, 0.19 and 0.01, respectively. The tree depth is between 2-6. The tree generation method is ramped half-and-half. The tournament selection with size 5 is used to select parents for crossover and mutation operations. The parameters of the GP based comparison methods are the same as MFCGPE.

The GP based methods are implemented using DEAP [57] and the classification algorithms are implemented using scikit-learn [58]. The number of the neighbors in KNN is set as 3 according to [22]. The parameters in the classification algorithms are the default values of the scikit-learn package. The parameter values of the methods using manually extracted features are set according to [14], [15], [25]. Each method has been executed 30 independent runs on each dataset with different random seeds. The evolutionary process of GP and the classifier training step only use the training set. The fault diagnosis results of the test sets are reported.

V. Results and Discussions

This section discusses and analyses the fault diagnosis results obtained by MFCGPE and the 19 comparison methods on the three different datasets with a small number of training samples. Table VII lists the classification accuracy of the 20 methods, including the maximum (Max) value, average (Avg) value and standard deviation (Std) of the accuracy of the 30 runs, where the best results of each dataset are highlighted in bold. Wilcoxon rank-sum test with a 5% significance level is employed to evaluate the significant difference in performance improvement of MFCGPE compared to a method. In Table VII, the "+" symbol indicates that the performance of MFCGPE is significantly better than the comparison method. The summary of the significance test results on each dataset is listed in the last row of Table VII.

Rows 1-5 of Table VII list the classification results of the five traditional classifiers using raw signals amplitude (RSA). It can be seen that the diagnosis accuracy of these methods is very low. On the NCEPU and CWRU datasets, MLP achieves better results than KNN, LR, SVM, and NB. Specifically, MLP achieves an average accuracy of 22.01% on NCEPU and of 21.92% on CWRU. On the XJTU dataset, NB achieves a maximal and average accuracy of 52.78%, which is better than other classifiers. Compared with these five methods, MFCGPE achieves much higher accuracy, i.e., over 99%, on the three datasets. The results show that constructing high-level features is very important for fault diagnosis of rolling bearings.

Rows 6 to 10 of Table VII list the classification results of the KNN classifier using five different types of manually crafted features (i.e., TDF, FDF, MDF, MMSDE, and IMDE). It can be seen that the accuracy of these five methods is higher than the KNN, LR, SVM, NB, and MLP methods using raw signals amplitude. Among these five methods using the manually crafted features, the IMDE method achieves the best results (the maximal and average diagnosis accuracy are 91.11%) on NCEPU, the MMSDE method achieves the best results (the maximal and average diagnosis accuracy are 92.67%) on CWRU, and the FDF method achieves the best results (the maximal and average diagnosis accuracy are 97.22%) on XJTU. The results show that it is necessary to extract a set of effective features according to the datasets to perform fault diagnosis because the performance of the manually extracted features varies with datasets. Compared with these methods, MFCGPE achieves much better performance by automatically constructed multi-view high-level features and building an ensemble for fault diagnosis. The adaptability of MFCGPE is much higher than these five methods as it achieves the best results on all the three datasets.

Rows 11 to 13 of Table VII are the classification results of the GP based comparison methods for constructing a single feature, i.e., GPS-TDF, GPS-FDF, and GPS-TFDF. Among these methods, GPS-TFDF achieves the best results on all three datasets by obtaining an average accuracy of 90.73% on NCEPU, 90.17% on CWRU and 96.67% on XJTU. Compared with these three methods, MFCGPE achieves better and stable classification performance. The results show that only constructing one high-level feature may not be effective for multi-class fault diagnosis. Compared with these methods, MFCGPE is able to construct a flexible number of effective features from every single view for fault diagnosis.

Rows 14 to 16 of Table VII are the classification results of the GP based comparison methods that construct multiple features for fault diagnosis, i.e., GPM-TDF, GPM-FDF, and GPM-TFDF. Among these three methods, GPM-TFDF achieves better results on all three datasets. Specifically, GPM-TFDF achieves an average diagnosis accuracy of 94.56%, 96.16%, and 98.21% on NCEPU, CWRU and XJTU, respectively. Compared with the GPS-based methods, the GPM-based methods achieve higher average accuracy and smaller standard deviation values. These results indicate that using multiple constructed high-level features is more effective than using a single constructed high-level feature to improve diagnosis performance. Compared with these three methods, MFCGPE achieves better performance on the three datasets. MFCGPE builds an effective ensemble using the features

TABLE VII
DIAGNOSIS ACCURACY (%) OF MFCGPE AND THE COMPARISON METHODS ON THE NCEPU, CWRU, AND XJTU DATASETS

ROW	METHOD	NCEPU		CWRU		XJTU	
		MAX	AVG±STD	MAX	AVG±STD	MAX	AVG±STD
1	RSA+KNN	16.67	16.67±0.00+	10.00	10.00±0.00+	25.00	25.00±0.00+
2	RSA+LR	17.04	17.04±0.00+	21.33	21.33±0.00+	28.70	28.70±0.00+
3	RSA+SVM	16.67	16.67±0.00+	16.44	16.44±0.00+	28.70	28.70±0.00+
4	RSA+NB	17.41	17.41±0.00+	23.33	23.33±0.00+	52.78	52.78±0.00+
5	RSA+MLP	31.85	22.01±4.24+	32.22	21.92±1.10+	37.96	31.45±3.26+
6	TDF	79.26	79.26±0.00+	76.67	76.67±0.00+	87.96	87.96±0.00+
7	FDF	82.96	82.96±0.00+	83.56	83.56±0.00+	97.22	97.22±0.00+
8	MDF	88.15	88.15±0.00+	90.22	90.22±0.00+	75.93	75.93±0.00+
9	MMSDE	87.78	87.78±0.00+	92.67	92.67±0.00+	83.33	83.33±0.00+
10	IMDE	91.11	91.11±0.00+	85.78	85.78±0.00+	82.41	82.41±0.00+
11	GPS-TDF	90.74	87.83±2.76+	88.89	84.62±2.56+	91.67	88.22±3.71+
12	GPS-FDF	92.59	89.42±3.78+	91.78	87.52±3.08+	99.07	96.45±2.36+
13	GPS-TFDF	94.81	90.73±3.53+	93.78	90.17±2.91+	100.0	96.67±3.37+
14	GPM-TDF	94.44	92.37±1.94+	96.22	93.37±2.42+	95.37	93.09±2.98+
15	GPM-FDF	95.92	93.30±2.17+	97.11	95.34±1.58+	100.0	97.87±2.36+
16	GPM-TFDF	97.03	94.56±2.35+	97.56	96.16±1.76+	100.0	98.21±1.97+
17	OFE	89.25	89.25±0.00+	90.44	90.44±0.00+	95.37	95.37±0.00+
18	GPSE	95.55	93.84±1.56+	94.46	92.81±1.45+	100.0	98.37±1.81+
19	GPME	98.51	97.47±1.12+	98.45	97.65±0.85+	100.0	99.02±1.09+
20	MFCGPE	**100.0**	**99.56±0.30**	**100.0**	**99.23±0.67**	**100.0**	**99.61±0.26**
21	**OVERALL**		**19+**		**19+**		**19+**

constructed from multiple views, which allows it to obtain higher generalization performance than these methods using a single classifier.

Rows 17 to 19 of Table VII are the classification results of the ensemble diagnosis methods, i.e., OFE, GSE, and GPME. It can be found that on the NCEPU, CWRU, and XJTU datasets, the diagnostic performance of OFE, GPSE, and GPME is ranked the third, the second, and the first, respectively. Compared with these three methods, MFCGPE achieves higher average accuracy and smaller standard deviation values. The reason is that MFCGPE has a new program structure, function set, terminal set, and fitness function, which allow it to construct more discriminative features and build a more effective ensemble to obtain better diagnosis accuracy.

Row 20 and Row 21 of Table VII lists the classification results and the significance test results of the MFCGPE approach, respectively. On the NCEPU, CWRU, and XJTU dataset, MFCGPE achieves the maximal accuracy of 100% and the average accuracy of above 99%. The results of the significance tests show that the diagnosis performance of MFCGPE is significantly better than the 19 comparison methods on the three datasets.

In summary, the results show that MFCGPE is able to achieve excellent fault diagnosis performance on three datasets with a small number of training samples. MFCGPE can adaptively construct a flexible number of features for fault diagnosis, which can not only achieve a comprehensive description of the raw signal but also avoid redundancy and information interference. The fitness function based on accuracy and distance enables MFCGPE to construct new effective features that achieve a small intra-class distance and a large inter-class distance of the training samples, which can improve the generalization performance on the unseen test set. MFCGPE constructs multiple high-level features from different views, individually, and constructs an effective ensemble based on these constructed features through the majority voting, which further improves the effectiveness of the diagnostic approach and avoids the overfitting issue caused by using a small number of training samples. Thus, MFCGPE is a practical and promising approach to engineering applications.

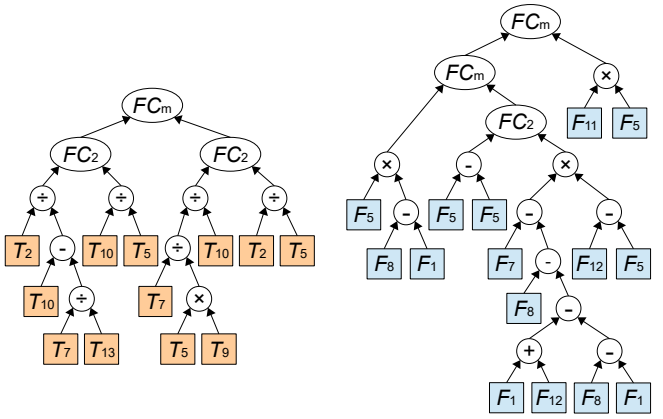
VI. FURTHER ANALYSIS

This section further analyses the evolved GP tree/models, the features constructed by GP from different views, and the constructed ensemble to provide insights into why the new approach is effective.

A. Example Trees/Models

Figure 12 shows the best trees/models evolved by MFCGPE from the three different views on the NCEPU dataset, where the white oval nodes are the feature combination operators, the white circle nodes are the feature construction operators, and the orange, blue, and pink rectangle nodes are the features of $View_1$, $View_2$, and $View_3$, respectively.

As it can be seen from Figure 12, for $View_1$, six distinguished features among the input 16 features are selected to construct four high-level features. For $View_2$, six distinguished features among the 13 input features are selected to construct four high-level features. For $View_3$, five distinguished features among the 29 input features are selected to construct three high-level features. This indicates that MFCGPE can automatically select the useful features and determine the number of constructed features, which avoid information interference and feature redundancy. The depths of the trees evolved from $View_1$, $View_2$, and $View_3$ are 5, 7, and 3, respectively, which indicates that MFCGPE can adaptively evolve the optimal variable-length models with the use of the time-domain, frequency-domain, time&frequency-domain characteristics of vibration signals. The above analysis



(a) Best tree obtained under $View_1$ (b) Best tree obtained under $View_2$

(c) Best tree obtained under $View_3$

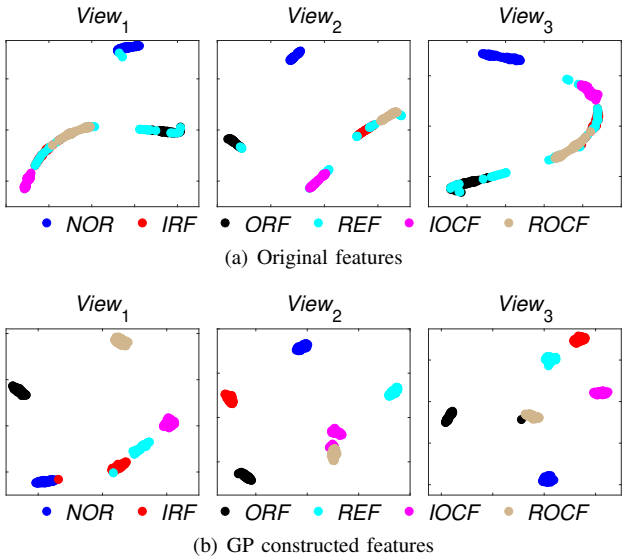
Fig. 12. Example trees evolved by MFCGPE on the NCEPU dataset.

shows that MFCGPE has the ability to learn effective and diverse features from multi-view input features to capture different patterns of the vibration signal, which is beneficial in describing data characteristics comprehensively.

B. Visualization of the GP-Constructed Features

To better demonstrate that the evolved models/trees shown in Figure 12 are effective for fault diagnosis, the original features and the newly constructed features are visualized for comparisons. The features of the samples in the test set are visualized by using t-distributed stochastic neighbor embedding (t-SNE) [59] method. Figure 13 shows the visualization results, where six colored points represent six running conditions of the rolling bearing, i.e., normal (NOR), inner ring fault (IRF), outer ring fault (ORF), rolling element fault (REF), inner & outer ring compound fault ($IOCF$), and rolling element & outer ring compound fault ($ROCF$).

As shown in Figure 13(a), using the original features of $View_1$ for visualization, the points in different classes are all overlapping; when using the original features of $View_2$ and $View_3$ for visualization, the points in the NOR class are clustered separately, but the points in other classes are also overlapping. These visualization results indicate that the original features of vibration signals in different running conditions cannot effectively separate the fault types into different classes to achieve a high diagnostic accuracy. As shown in Figure 13(b), using the features constructed from $View_1$ for visualization, few points in the NOR, IRF, and REF classes are overlapping, and the other colored points are gathered together and do not overlap. Using the features constructed from $View_2$ for visualization, only a few points in the $IOCF$ and $ROCF$ classes are overlapping. Using



(a) Original features

(b) GP constructed features

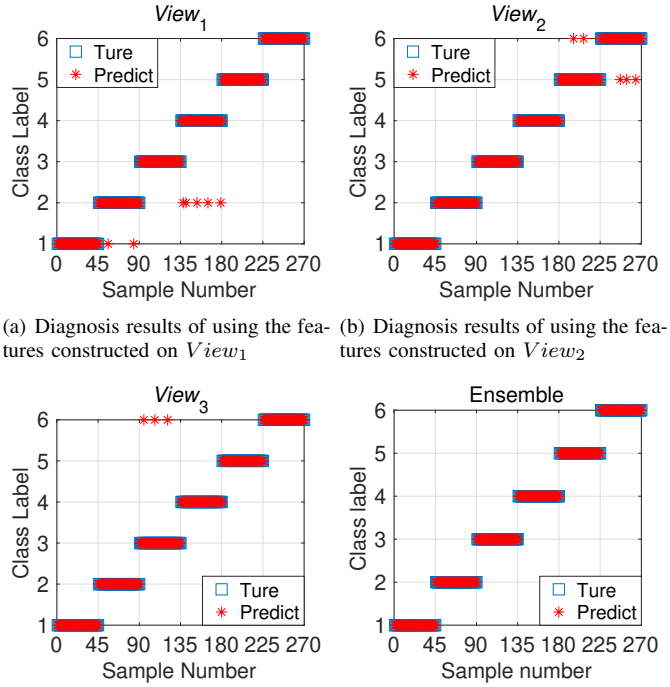
Fig. 13. Feature visualization on the NCEPU dataset using t-SNE.

the features constructed from $View_3$ for visualization, only a few points in the ORF and $ROCF$ classes are overlapping. Compared with the original features, these newly constructed features have a good similarity among the same class and have a big difference between different classes. These visualization results indicate that the constructed features can effectively represent the vibration signals in different running conditions, which makes fault diagnosis simpler and more accurate.

C. Diagnosis Results Using the GP-Constructed Features and the Constructed Ensemble

To analyze the effectiveness of the constructed features of Figure 13(b) and the built ensemble, Figure 14 shows their diagnosis results and the results of the built ensemble, where the blue box represents the true label of the sample, the red asterisks represent the predicted label of the sample, and the class labels 1, 2, 3, 4, 5, and 6 represent the NOR, IRF, ORF, REF, $IOCF$, and $ROCF$ running conditions, respectively. The overlap of boxes and asterisks indicates that the diagnosis is correct. No-overlapping of boxes and asterisks indicates that the diagnosis is wrong.

It can be seen from Figure 14(a) that using the features constructed from $View_1$ for fault diagnosis will cause seven samples to be misclassified, where two samples in the IRF class are identified as the NOR class and five samples in the REF class are identified as the IRF class. Figure 14(b) shows the diagnosis results of using the features constructed from $View_2$. It shows that two types of compound faults are misclassified, i.e., two samples in the $IOCF$ class are classified into the $ROCF$ class and three samples in the $ROCF$ class are identified as the $IOCF$ class. Figure 14(c) shows the diagnosis results of using the features constructed from $View_3$, where three samples in the ORF class are identified as the $ROCF$ class. The diagnosis accuracy of using the TDF, FDF, and TFDF based constructed features are 97.40%, 98.14% and 98.88%, respectively. The ensemble diagnosis accuracy is obtained by integrating the three diagnosis results via the majority voting method and shown in Figure 14(d), in which



(a) Diagnosis results of using the features constructed on $View_1$

(b) Diagnosis results of using the features constructed on $View_2$

(c) Diagnosis results of using the features constructed on $View_3$

(d) Diagnosis results of using the constructed ensemble

Fig. 14. Diagnosis results on the NCEPU dataset using the constructed features of different views and the constructed ensemble.

no samples are misclassified. Obviously, the ensemble strategy improves the diagnosis accuracy.

By analyzing the best models evolved by MFCGPE, the feature construction process is clearly displayed. In addition, the visualization results and the diagnosis results of the newly constructed features are demonstrated for interpretability. Owing to the program structure and the new fitness function, MFCGPE constructs a flexible number of diverse and effective high-level features from different views. The constructed features can make the classification of different fault types easier and more accurate. Owing to the ensemble diagnosis strategy, MFCGPE constructs multi-view features and gains a higher diagnosis accuracy by using an ensemble built from the features constructed from different views.

VII. Conclusions

The goal of this paper was to develop a new GP-based approach to achieving effective fault types diagnosis of rolling bearings using a small number of training samples. This goal has been successfully achieved by developing the MFCGPE approach. A new GP program structure, a function set, and a terminal set were developed to enable MFECPE to construct a flexible number of high-level features from three different views of features. A new fitness function based on the measures of accuracy and distance was developed to enable these newly constructed features to be accurate and discriminative. To improve the diagnosis performance, an ensemble classifier was created by using constructed features from multiple views and using KNN. With these designs, MFCGPE can not only automatically select and construct informative and discriminative features from different views but also build an effective ensemble to achieve a high generalization performance.

The effectiveness of MFCGPE was evaluated on three rolling bearing fault datasets and compared with the 19 competitive methods. The results showed that MFCGPE achieved the best diagnosis accuracy on the three datasets among all the methods. The highlight of MFCGPE was that multiple discriminative features were automatically constructed using MFCGPE, and the ensemble diagnosis can address the issue of poor generalization of the diagnosis model caused by using a small number of training samples.

This paper shows that the proposed MFCGPE approach is effective for fault diagnosis of rolling bearings. In addition to fault diagnosis, remaining life prediction can also help to analyze rolling bearing degradation. In the future, we will investigate how GP is used to construct the health index to predict the remaining life of the rolling bearing.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under grant 51777075, the Natural Science Foundation of Hebei Province under grant E2019502064, the Marsden Fund of New Zealand Government under Contracts VUW1509 and VUW1615, the Science for Technological Innovation Challenge (SfTI) fund under grant E3603/2903, the University Research Fund at Victoria University of Wellington grant number 223805/3986, MBIE Data Science SSIF Fund under the contract RTVU1914, and National Natural Science Foundation of China (NSFC) under Grant 61876169.

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, 2018.

[2] H. Li, T. Liu, X. Wu, and Q. Chen, "Enhanced frequency band entropy method for fault feature extraction of rolling element bearings," *IEEE Trans. Ind. Inform.*, vol. 16, no. 9, pp. 5780–5791, 2020.

[3] Y. Zhang and R. Randall, "Rolling element bearing fault diagnosis based on the combination of genetic algorithms and fast kurtogram," *Mech. Syst. Signal Process.*, vol. 23, no. 5, pp. 1509–1517, 2009.

[4] R. Yan, R. X. Gao, and X. Chen, "Wavelets for fault diagnosis of rotary machines: A review with applications," *Signal Process.*, vol. 96, pp. 1–15, 2014.

[5] Y. Lei, J. Lin, Z. He, and M. Zuo, "A review on empirical mode decomposition in fault diagnosis of rotating machinery," *Mech. Syst. Signal Process.*, vol. 35, no. 1-2, pp. 108–126, 2013.

[6] J. Antoni, "The spectral kurtosis: a useful tool for characterising non-stationary signals," *Mech. Syst. Signal Process.*, vol. 20, no. 2, pp. 282–307, 2006.

[7] G. L. McDonald, Q. Zhao, and M. J. Zuo, "Maximum correlated kurtosis deconvolution and application on gear tooth chip fault detection," *Mech. Syst. Signal Process.*, vol. 33, pp. 237–255, 2012.

[8] K. Dragomiretskiy and D. Zosso, "Variational mode decomposition," *IEEE Trans. Signal Process.*, vol. 62, no. 3, pp. 531–544, 2013.

[9] R. B. Randall, J. Antoni, and S. Chobsaard, "The relationship between spectral correlation and envelope analysis in the diagnostics of bearing faults and other cyclostationary machine signals," *Mech. Syst. Signal Process.*, vol. 15, no. 5, pp. 945–962, 2001.

[10] S. Maurya, V. Singh, N. K. Verma, and C. K. Mechefske, "Condition-based monitoring in variable machine running conditions using low-level knowledge transfer with dnn," *IEEE Trans. Autom. Sci. and Eng.*, DOI 10.1109/TASE.2020.3028151, 2020.

[11] J. Yang, Y. Zhang, and Y. Zhu, "Intelligent fault diagnosis of rolling element bearing based on svms and fractal dimension," *Mech. Syst. Signal Process.*, vol. 21, no. 5, pp. 2012–2024, 2007.

[12] W. Caesarendra, B. Kosasih, A. K. Tieu, and C. A. Moodie, "Application of the largest lyapunov exponent algorithm for feature extraction in low speed slew bearing condition monitoring," *Mech. Syst. Signal Process.*, vol. 50, pp. 116–138, 2015.

[13] J. Zheng, H. Pan, and J. Cheng, "Rolling bearing fault detection and diagnosis based on composite multiscale fuzzy entropy and ensemble support vector machines," *Mech. Syst. Signal Process.*, vol. 85, pp. 746–759, 2017.

[14] Y. Li, Y. Yang, G. Li, M. Xu, and W. Huang, "A fault diagnosis scheme for planetary gearboxes using modified multi-scale symbolic dynamic entropy and mrmr feature selection," *Mech. Syst. Signal Process.*, vol. 91, pp. 295–312, 2017.

[15] X. Yan and M. Jia, "Intelligent fault diagnosis of rotating machinery using improved multiscale dispersion entropy and mrmr feature selection," *Knowl. Based Syst.*, vol. 163, pp. 450–471, 2019.

[16] V. Singh and N. K. Verma, "Intelligent condition based monitoring techniques for bearing fault diagnosis," *IEEE Sens. J.*, DOI 10.1109/JSEN.2020.3021918, 2020.

[17] X. Zhang, Q. Zhang, M. Chen, Y. Sun, X. Qin, and H. Li, "A two-stage feature selection and intelligent fault diagnosis method for rotating machinery using hybrid filter and wrapper method," *Neurocomputing*, vol. 275, pp. 2426–2439, 2018.

[18] Z. Huo, Y. Zhang, L. Shu, and M. Gallimore, "A new bearing fault diagnosis method based on fine-to-coarse multiscale permutation entropy, laplacian score and svm," *IEEE Access*, vol. 7, pp. 17 050–17 066, 2019.

[19] Y. Gu, X. Zhou, D. Yu, and Y. Shen, "Fault diagnosis method of rolling bearing using principal component analysis and support vector machine," *J. Mech. Sci. Technol.*, vol. 32, no. 11, pp. 5079–5088, 2018.

[20] J. Yu, "Machinery fault diagnosis using joint global and local/nonlocal discriminant analysis with selective ensemble learning," *J. Sound Vib.*, vol. 382, pp. 340–356, 2016.

[21] X. Zhao and M. Jia, "Fault diagnosis of rolling bearing based on feature reduction with global-local margin fisher analysis," *Neurocomputing*, vol. 315, pp. 447–464, 2018.

[22] J. Tian, C. Morillo, M. H. Azarian, and M. Pecht, "Motor bearing fault detection using spectral kurtosis-based feature extraction coupled with k-nearest neighbor distance analysis," *IEEE Trans. Ind. Electron.*, vol. 63, no. 3, pp. 1793–1803, 2016.

[23] M. Unal, M. Onat, M. Demetgul, and H. Kucuk, "Fault diagnosis of rolling bearings using a genetic algorithm optimized neural network," *Measurement*, vol. 58, pp. 187–196, 2014.

[24] M. Luo, C. Li, X. Zhang, R. Li, and X. An, "Compound feature selection and parameter optimization of elm for fault diagnosis of rolling element bearings," *ISA Trans.*, vol. 65, pp. 556–566, 2016.

[25] 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, 2018.

[26] M. He and D. He, "Deep learning based approach for bearing fault diagnosis," *IEEE Trans. Ind. Appl.*, vol. 53, no. 3, pp. 3057–3065, 2017.

[27] S. Zhang, S. Zhang, B. Wang, and T. G. Habetler, "Deep learning algorithms for bearing fault diagnostics—a comprehensive review," *IEEE Access*, vol. 8, pp. 29 857–29 881, 2020.

[28] W. Deng, H. Liu, J. Xu, H. Zhao, and Y. Song, "An improved quantum-inspired differential evolution algorithm for deep belief network," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 10, pp. 7319 –7327, 2020.

[29] S. Maurya, V. Singh, and N. K. Verma, "Condition monitoring of machines using fused features from emd-based local energy with dnn," *IEEE Sens. J.*, vol. 20, no. 15, pp. 8316–8327, 2019.

[30] C. Cheng, B. Zhou, G. Ma, D. Wu, and Y. Yuan, "Wasserstein distance based deep adversarial transfer learning for intelligent fault diagnosis with unlabeled or insufficient labeled data," *Neurocomputing*, vol. 409, pp. 35–45, 2020.

[31] H. Al-Sahaf, Y. Bi, Q. Chen, A. Lensen *et al.*, "A survey on evolutionary machine learning," *J. Roy. Soc. N. Z.*, vol. 49, no. 2, pp. 205–228, 2019.

[32] W. Deng, J. Xu, X.-Z. Gao, and H. Zhao, "An enhanced msiqde algorithm with novel multiple strategies for global optimization problems," *IEEE Trans. Syst. Man. C.*, DOI 10.1109/TSMC.2020.3030792, 2020.

[33] W. Deng, J. Xu, Y. Song, and H. Zhao, "Differential evolution algorithm with wavelet basis function and optimal mutation strategy for complex optimization problem," *Appl. Soft Comput.*, DOI 10.1016/j.asoc.2020.106724, 2020.

[34] Y. Song, D. Wu, W. Deng, X.-Z. Gao, T. Li, B. Zhang, and Y. Li, "Mppcede: multi-population parallel co-evolutionary differential evolution for parameter optimization," *Energ. Convers. Manage.*, vol. 228, DOI 10.1016/j.enconman.2020.113661, 2021.

[35] K. Krawiec, "Genetic programming-based construction of features for machine learning and knowledge discovery tasks," *GPEM*, vol. 3, no. 4, pp. 329–343, 2002.

[36] L. Guo, D. Rivero, J. Dorado, C. R. Munteanu, and A. Pazos, "Automatic feature extraction using genetic programming: An application to epileptic eeg classification," *Expert Syst. Appl.*, vol. 38, no. 8, pp. 10 425–10 436, 2011.

[37] Y. Bi, B. Xue, and M. Zhang, "Genetic programming with image-related operators and a flexible program structure for feature learning in image classification," *IEEE Trans. Evol. Comput.*, vol. 25, no. 1, pp. 87–101, 2021.

[38] M. W. Aslam, Z. Zhu, and A. K. Nandi, "Automatic modulation classification using combination of genetic programming and knn," *IEEE Trans. Wirel. Commun.*, vol. 11, no. 8, pp. 2742–2750, 2012.

[39] Z. Gao, C. Cecati, and S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques—part i: Fault diagnosis with model-based and signal-based approaches," *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3757–3767, 2015.

[40] J. R. Koza, *Genetic Programming: on the Programming of Computers by Means of Natural Selection*. MIT press, 1992.

[41] F. E. Otero, M. M. Silva, A. A. Freitas, and J. C. Nievola, "Genetic programming for attribute construction in data mining," in *Proc. EuroGP*, pp. 384–393. Springer, 2003.

[42] M. Muharram and G. D. Smith, "Evolutionary constructive induction," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 11, pp. 1518–1528, 2005.

[43] H. Guo and A. K. Nandi, "Breast cancer diagnosis using genetic programming generated feature," *Pattern Recognit.*, vol. 39, no. 5, pp. 980–987, 2006.

[44] H. Guo, Q. Zhang, and A. K. Nandi, "Feature extraction and dimensionality reduction by genetic programming based on the fisher criterion," *Expert Syst.*, vol. 25, no. 5, pp. 444–459, 2008.

[45] K. Neshatian, M. Zhang, and P. Andreae, "A filter approach to multiple feature construction for symbolic learning classifiers using genetic programming," *IEEE Trans. Evol. Comput.*, vol. 16, no. 5, pp. 645–661, 2012.

[46] B. Tran, B. Xue, and M. Zhang, "Genetic programming for multiple-feature construction on high-dimensional classification," *Pattern Recognit.*, vol. 93, pp. 404–417, 2019.

[47] J. Ma and G. Teng, "A hybrid multiple feature construction approach for classification using genetic programming," *Appl. Soft Comput.*, vol. 80, pp. 687–699, 2019.

[48] L. Zhang, L. B. Jack, and A. K. Nandi, "Fault detection using genetic programming," *Mech. Syst. Signal Process.*, vol. 19, no. 2, pp. 271–289, 2005.

[49] H. Guo, L. B. Jack, and A. K. Nandi, "Feature generation using genetic programming with application to fault classification," *IEEE Trans. Syst., Man, Cybern. C, Cybern.*, vol. 35, no. 1, pp. 89–99, 2005.

[50] J. Xuan, H. Jiang, T. Shi, and G. Liao, "Gear fault classification using genetic programming and support vector machines," *Int. J. Inf. Technol.*, vol. 11, no. 9, pp. 18–27, 2005.

[51] D. J. Montana, "Strongly typed genetic programming," *Evol. Comput.*, vol. 3, no. 2, pp. 199–230, 1995.

[52] Y. Lei, B. Yang, X. Jiang, F. Jia, N. Li, and A. K. Nandi, "Applications of machine learning to machine fault diagnosis: A review and roadmap," *Mech. Syst. Signal Process.*, vol. 138, DOI 10.1016/j.ymssp.2019.106587, 2020.

[53] X. Dong, Z. Yu, W. Cao, Y. Shi, and Q. Ma, "A survey on ensemble learning," *Front. Comput. Sci.*, pp. 1–18, 2020.

[54] "Case western reserve university bearing data center. [online]," http://csegroups.case.edu/bearingdatacenter/home/, accessed April, 2018.

[55] B. Wang, Y. Lei, N. Li, and N. Li, "A hybrid prognostics approach for estimating remaining useful life of rolling element bearings," *IEEE Trans. Reliab.*, vol. 69, no. 1, pp. 401 – 412, 2020.

[56] Y. Bi, B. Xue, and M. Zhang, "An effective feature learning approach using genetic programming with image descriptors for image classification [research frontier]," *IEEE Comput. Intell. Mag.*, vol. 15, no. 2, pp. 65–77, 2020.

[57] F.-A. Fortin, F.-M. De Rainville, and et al., "DEAP: Evolutionary algorithms made easy," *J. Mach. Learn. Res.*, vol. 13, pp. 2171–2175, 2012.

[58] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[59] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *J. Mach. Learn. Res.*, vol. 9, no. Nov, pp. 2579–2605, 2008.