Application of Improved Genetic Programming for Feature Extraction in the Evaluation of Bearing Performance Degradation

HAO WANG, GUANGMING DONG, AND JIN CHEN
State Key Laboratory of Mechanical System and Vibration, Shanghai Jiao Tong University, Shanghai 200000, China
Corresponding author: Guangming Dong (gmdong@sjtu.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFB2004600, and in part by the National Natural Science Foundation of China 22 under Grant 51575339 and Grant 51775330.

ABSTRACT In the evaluation of bearing performance degradation, discovering a good HI (Health Indicator) is one of the most crucial parts, because it determines whether a precise result can be obtained in the prediction of remaining useful life. In this paper, GP (Genetic Programming), which is a heuristic iterative search algorithm inspired by the theory of biological evolution, is improved in genetic operation and fitness function, and a feature weighted matrix is used in GP innovatively. The improved GP is applied to discover a HI by fusing multiple features, which is very close to linearity. Furthermore, by optimizing the discovered HIs, an optimization HI is obtained, which has a higher fitness and can get a more precise result in the prediction of RUL. The proposed approach is verified in the experimental data for the entire life of the bearing provided by 2012 IEEE PHM challenge, and a total of three bearings are used in the verification.

INDEX TERMS Bearing performance degradation, feature extraction, genetic programming, health indicator, prediction, remaining useful life.

I. INTRODUCTION

Bearing is an essential part of rotating machinery, and it is widely used. The performance of bearing determines whether the rotating machinery can work in healthy state, thus, it is vital to evaluate the bearing performance degradation. Generally, the evaluation is divided into four parts [1]. (1) The first part is data acquisition. (2) Then, the second part is HI (Health Indicator) construction. (3) After that, the third part is named health stage division. (4) Lastly, the fourth part is the prediction of RUL (Remaining Useful Life). The chief part is HI construction, because a good HI can simplify the diagnostic model and obtain a precise diagnostic result.

HI is mainly divided into two types: Physical HI and Virtual HI [2]. Physical HI is generally acquired by using statistical learning or signal processing, while virtual HI is the combination of multiple Physical HIs or multi-sensor signals. At present, numerous HIs are applied in the evaluation of bearing performance degradation. Kurtosis is used to select FPT (First Predicting Time), and then RMS (Root Mean Square) is utilized to predict bearing life [3], [4]; Kurtosis is extracted by the band-pass filter signal to predict the bearing life [5]; The average amplitude of defect frequency and harmonic is calculated and used as HI of bearing [6]; What mentioned above are the approaches of constructing physical HI based on signal processing. Furthermore, the approaches based on statistical learning are also applied to constructing physical HI. The vibration signals generated by two different time series are collected and the correlation coefficient between them is applied as HI [7]. The statistical complexity of envelope entropy is utilized as HI to predict the bearing life [8]. Li et al. [9] calculate the energy ratio between the adaptive filtered signal and the original signal and use it as HI. The logarithmic transformation of the cumulative spectrum is taken as HI of sliding average wear degradation [10]. Researches of Virtual HI also develop rapidly. Widodo et al. use PCA (Principal Component Analysis) to reduce the dimension of the feature set, afterwards, the deviation between the unknown state and the health state is calculated as HI [11]. By calculating the Mahalanobis distance between the unknown state and the health state, multiple time domain features are combined as HI of bearing.
The hidden Markov model is applied to calculate the probability of health state and multiple features are fused as HI of bearing [12].

One of machine learning approaches, GP is employed to extract features from the raw vibration data taken from a rotating machine with several different conditions, and created features are input to artificial neural network for the identification of different bearing conditions [13]. Genetic programming system, which comprises genetic programming, weighted genetic programming, and soft-computing polynomials, is developed for determining the ultimate bearing capacity of shallow foundations [14]. Liao proposes to use GP to integrate multiple features and obtains HI with monotonic trend to predict bearing life [15]. However, the HI obtained is not always close to linearity. In this paper, an improved GP, which is more suitable for fusing features to output a HI with more linearity, is put forward. Besides, a feature weighted matrix is proposed, and the dimension of HI improves. By optimizing the discovered HIs, an optimization HI is proposed. The superiority of optimization HI is proved in the prediction of the bearing life.

The rest of the paper is organized as follows: Section II introduces the process of GP and some improvements; Section III describes the performance of the HI and optimization HI output by GP, and the results of RUL prediction. A conclusion is drawn in section IV.

II. BACKGROUND
A. GENETIC PROGRAMMING
Genetic programming, which began in 1996, is an evolutionary programming algorithm inspired by the theory of biological evolution [16]. GP is essentially a heuristic iterative search algorithm, and the primary differences between GP and GA (Genetic Algorithm) are the presentation of program and the corresponding fitness evaluation [17]. In the GP, firstly, a program population is generated randomly, then genetic operation is applied in each program according to fitness and a new program population is produced. Afterwards, calculation of fitness and generation of new program population by genetic operation are repeated. At last, the program with the highest fitness is output. A tree structure is utilized to the presentation of program. On the other hand, the fitness function is set on the basis of specific task.

Genetic operation mainly includes: crossover operation, mutation operation, and replication operation, which occurs according to the corresponding fitness of program. The crossover operation involves swapping parts of selected pairs to produce new and different offspring that become part of the new generation of programs. The mutation operation involves substitution of some part of a program with some other part of a program. The replication operation is reserving the program with the highest fitness in current program population. The newly generated programs from the crossover and mutation operation and the retained program from the replication operation replace the old programs to form a new program population and enter the next iteration.

In GP, fitness is the main indicator to describe program performance. The programs are updated by the corresponding values of fitness, so GP is driven by the fitness. Fitness is applied to estimate the difference between the current program and the optimal program, namely, the bigger fitness is, the less difference between them is. The goal of GP is to find the program with the highest fitness, meanwhile, the fitness is calculated by the fitness function. Therefore, the fitness function determines the output of GP, and that means setting fitness function is the most critical part of GP.

The termination conditions of GP have two forms. One is reaching the highest fitness established in advance, and the other one is reaching the maximum number of iterations. According to the termination condition mentioned in [15], this paper takes reaching the maximum number of iterations as the termination condition. Considering the goal of fusing multiple features for discovering a HI close to linearity, the process of GP was roughly shown as FIGURE 1.

![FIGURE 1. The flowchart of GP.](image)

B. THE REPRESENTATION OF PROGRAM
The reason of choosing GP to fuse multiple features instead of GA is that the representation of program in GP is a tree structure. Hence, multiple features can be fused into a tree structure as a program, and the output is a program, namely, the outputting program is the virtual HI mentioned before. An example tree structure is shown in FIGURE 2.
where \( f1, f2, f3 \) and \( f4 \) represent different features. ‘+’ and ‘-’ are arithmetic operators between features, and the expression of this program is shown in (1). Seven arithmetic operators are employed in this paper including: plus, minus, multiplication, division operation, sqrt (square root), square and logarithm.

\[
p = (f1 + f2) ÷ f3 − f4
\] (1)

C. IMPROVED CROSSOVER OPERATION

In the process of feature fusion, the trend of feature itself has a great effect on the linearity. In order to preserve the key features which have greater effects, the programs are sorted from high to low in fitness. First half of the programs are selected for crossover operation, and the improvement is that crossover operation is only carried out on the arithmetic nodes in the tree structure, while the feature node does not carry out crossover operation. An example of crossover operation is shown in FIGURE 3. The improved crossover operation is more advantageous to generate better arithmetic nodes between the nodes of key features at the next iteration.

D. MUTATION OPERATION AND REPLICATION OPERATION

As mentioned in the previous section, the programs in the population are sorted from high to low in terms of fitness. The first half of the programs are selected for crossover operation, while the remaining half of the programs with lower fitness are selected for mutation operation. Fitness is mainly affected by the trend of features in the program. If the fitness is low, the features in program are not likely to be the key features. Based on this assumption, the mutation operation becomes that randomly generate a new node to replace the old node for each node in the program, so the programs in the remaining half will be brand new after mutation operation.

GP is an iterative algorithm with some randomness. In replication operation, the program with the highest fitness in each iteration is retained. And the program population of the next generation is formed with the program retained and the new programs generated by crossover operation and mutation operation. Therefore, it ensures that the highest fitness of the program population in current generation is not less than that in the previous generation.

E. FEATURE WEIGHTED MATRIX

At present, many proposed methods assign corresponding weights to features for processing data, and it is applied in GP.

According to the genetic operation described above, it can be found that the key features will appear more in the program population with the iteration conducted. In order to assign the key feature a larger weight in the program, a feature weighted matrix is proposed. The matrix is shown in following equations.

\[
W = \begin{bmatrix}
    w_{11} & \cdots & w_{1n} \\
    \vdots & \ddots & \vdots \\
    w_{m1} & \cdots & w_{mn}
\end{bmatrix}
\] (2)

where

\[
w_{ij} = \frac{\text{times}_{ij}}{\text{times}_{\text{mean}}} \tag{3}
\]

\[
\text{times}_{\text{mean}} = \frac{n \times m}{k} \tag{4}
\]

\( W \) is feature weighted matrix, and \( w_{ij} \) represents the weight of the \( j \text{th} \) feature of the \( i \text{th} \) program in the program population. \( m \) represents the number of programs in the program population, and \( n \) represents the number of features in a program, and \( k \) represents the total number of features in the feature set. \( \text{times}_{ij} \) is the number of times corresponding to the \( j \text{th} \) feature of the \( i \text{th} \) program appears in the current program population. \( \text{times}_{\text{mean}} \) is a constant that represents the average number of times that each feature should appear. The number of each feature varies with the iterations, so the weighted matrix also gets updated through iterations. The feature weighted matrix is employed to allocate the weight of all features in the
current population. The specific calculation is shown in (5).
\[ f_{wij} = W (i, j) \times f_{ij} = w_{ij} \times f_{ij} \]  
(5)
where \( w_{ij} \) is the \( i^{th} \) row and \( j^{th} \) column element of the matrix \( W \), \( f_{ij} \) is the \( j^{th} \) feature of the \( i^{th} \) program. \( f_{wij} \) is the \( f_{ij} \) with the weight. Therefore, the tree structure in section 2.2 is improved, as shown in FIGURE 4, and the expression is shown in (6).
\[ p = (w_{i1} \times f_{i1} + w_{i2} \times f_{i2}) \div (w_{i3} \times f_{i3} - w_{i4} \times f_{i4}) \]  
(6)

**FIGURE 4.** The tree structure with weight assignment.

F. FEATURE WEIGHTED MATRIX
The purpose of GP is to obtain a feature combination as HI, the trend of which is close to linearity. So the fitness function should reflect the linearity of the program as much as possible. Firstly, the definition of \( \text{vec} \) function is given, as shown in (7).
\[ y = \text{vec} (A > x) \]  
(7)
where \( A \) is a column vector. \( x \) is a constant, and \( y \) represents the number of elements, which are larger than \( x \) in \( A \). The fitness function is defined as follows.
\[ \text{fitness} = 100 \times (\text{moncity} + \text{trend} + \text{deviation}) / 3 \]  
(8)
where \( \text{fitness} \) is the fitness of program. \( \text{moncity} \) is monotonic quantity, and \( \text{trend} \) is trend quantity. \( \text{deviation} \) is deviation quantity. The following is the definition of these three quantities.
(1) Monotonic quantity
Monotonic quantity represents the degree of monotonic trend of the program. Its definition is as follows.
\[ \text{moncity} = \frac{\begin{vmatrix} \text{vec}\left(\frac{d^2(p_{pre})}{dt^2} > 0\right) - \text{vec}\left(\frac{d^2(p_{pre})}{dt^2} < 0\right) \\ \text{vec}\left(\frac{d^2(p_{later})}{dt^2} > 0\right) - \text{vec}\left(\frac{d^2(p_{later})}{dt^2} < 0\right) \end{vmatrix}}{\begin{vmatrix} \text{vec}\left(\frac{d^2(p_{pre})}{dt^2} > 0\right) + \text{vec}\left(\frac{d^2(p_{pre})}{dt^2} < 0\right) \\ \text{vec}\left(\frac{d^2(p_{later})}{dt^2} > 0\right) + \text{vec}\left(\frac{d^2(p_{later})}{dt^2} < 0\right) \end{vmatrix}} \]  
(9)
where \( t \) represents the time, and \( p \) is a program. \( d(p)/dt \) is the first derivative of the program, if it is greater than zero, the program increases at the time \( t \), on the contrary, if it is less than zero, the program decreases at the time \( t \). \( \text{moncity} = 1 \) means that the program is monotone increasing or decreasing in time series.
(2) Trend quantity
Trend quantity describes the increase and decrease balance level of program in time series. Its definition is as follows:
\[ \text{trend} = \begin{cases} 0 & \text{if trend}_{pre} > z \text{ or trend}_{later} > z \\ 1 - \frac{\begin{vmatrix} \text{vec}\left(\frac{d^2(p_{pre})}{dt^2} > 0\right) - \text{vec}\left(\frac{d^2(p_{pre})}{dt^2} < 0\right) \\ \text{vec}\left(\frac{d^2(p_{later})}{dt^2} > 0\right) - \text{vec}\left(\frac{d^2(p_{later})}{dt^2} < 0\right) \end{vmatrix}}{\begin{vmatrix} \text{vec}\left(\frac{d^2(p_{pre})}{dt^2} > 0\right) + \text{vec}\left(\frac{d^2(p_{pre})}{dt^2} < 0\right) \\ \text{vec}\left(\frac{d^2(p_{later})}{dt^2} > 0\right) + \text{vec}\left(\frac{d^2(p_{later})}{dt^2} < 0\right) \end{vmatrix}} & \text{else} \end{cases} \]  
(10)
where
\[ \text{trend}_{pre} = \begin{vmatrix} \text{vec}\left(\frac{d^2(p_{pre})}{dt^2} > 0\right) - \text{vec}\left(\frac{d^2(p_{pre})}{dt^2} < 0\right) \\ \text{vec}\left(\frac{d^2(p_{later})}{dt^2} > 0\right) - \text{vec}\left(\frac{d^2(p_{later})}{dt^2} < 0\right) \end{vmatrix} \]  
(11)
\[ \text{trend}_{later} = \begin{vmatrix} \text{vec}\left(\frac{d^2(p_{pre})}{dt^2} > 0\right) + \text{vec}\left(\frac{d^2(p_{pre})}{dt^2} < 0\right) \\ \text{vec}\left(\frac{d^2(p_{later})}{dt^2} > 0\right) + \text{vec}\left(\frac{d^2(p_{later})}{dt^2} < 0\right) \end{vmatrix} \]  
(12)
\( p_{pre} \) represents the program in a short period of time after the bearing starts working, and \( p_{later} \) represents the program in a short period of time before bearing failure. And \( z \) is a constant.
(3) Deviation quantity
Deviation quantity represents the degree of the program deviates from linearity in time series. Its definition is as follows.
\[ \text{deviation} = 1 - \frac{\sum_{t=1}^{end} p(t) - \hat{p}(t)}{\sum_{t=1}^{end} p(t)} \]  
(13)
where \( end \) is the cycle when the bearing is broken. \( p(t) \) is the output of program when the cycle equals \( t \). \( \hat{p}(t) \) is the output of program after least square fitting when the cycle equals \( t \).

III. EXPERIMENT AND VERIFICATION
Original features are extracted from the vibration signal of bearing, and the evaluation of bearing performance degradation based on GP is shown in the FIGURE 5. Eventually, the reliability of the GP is verified in the prediction of RUL.

A. TEST AND DATA ACQUISITION
The bearing data provided by 2012 IEEE PHM challenge is applied in this paper [18]. The test bed can accelerate the degradation of bearing under the invariable or variable working condition, and collect online monitoring data, such as speed, load, temperature and vibration signals, which can be used to conduct the bearing fault detection and diagnose in advance, as shown in FIGURE 6. It provides experimental data for the entire bearing life (until complete failure). Three different types of bearings under the same
A total of 21 signal features are extracted from vibration signals acquired in the bearing life test, as shown in Table 1.

### TABLE 1. The features of the bearing.

| Index | Feature | Index | Feature |
|-------|---------|-------|---------|
| 1     | PV=max(abs(x)) | 8 | CF=PV/RMS |
| 2     | PPV=max(x)-min(x) | 9 | CLF=PV/RA |
| 3     | MA=sum(abs(x))/N | 10 | SK=sum((x-mean(x))/variance(x))^3/N |
| 4     | RA=(sum(sqrt(abs(x))))/N | 11 | Kurtosis |
| 5     | RMS=sqrt(sum(x.^2)/N) | 12 | Amplitude spectrum entropy |
| 6     | SF=RMS/MA | 13 | Envelope spectrum entropy |
| 7     | IF=PV/MA | 14-21 | Wavelet packet energy Entropy 1-8 |

(Remarks: (1) x is the signal; (2) N is the length of signal; (3) Features of index 11 to 21 are calculated by certain function; (4) the layer of wavelet packet energy is three.)

All but the first five are dimensionless, and dimensionless features have the advantage of not being affected by bearing size, speed and load. Shannon [19] proposed the concept of information entropy, which can be used to measure the uncertainty of the system, and developed the features of amplitude spectrum entropy and envelope spectrum entropy. On the other hand, Wavelet decomposition is a typical time-frequency analysis method for non-stationary signals. In this paper, wavelet packet energy entropy is a commonly used feature in wavelet analysis [20]. Thus, 11 time domain features, wavelet packet energy entropy, amplitude spectrum entropy and envelope spectrum entropy consist of the feature set.
that before the fitness calculation, each program needs to perform high order polynomial curve fitting, because fitness is concerned with the trend of the program. If not fitting, the program always fluctuates up and down in time series, so the fitness will be very low. In this paper, the order is set to seven for the polynomial curve fitting, which can well show the trend of the program.

C. CASE ONE: THE COMPARISON BETWEEN TRADITIONAL AND IMPROVED GENETIC OPERATION

The feature set from bearing-1 is input. The improved genetic operation is compared with the traditional genetic operation. The numbers of the original feature \( f_{8} \) with the highest fitness and the original feature \( f_{17} \) with the lowest fitness in the 5th and 10th generations are shown in Table 2. It shows that as the iterations progress, for improved GP, the number of \( f_{8} \) with the highest fitness is becoming more in the population, meanwhile, \( f_{17} \) with the lowest fitness is still lower than the average number 95. On the contrary, for traditional GP, the number of features are all close to 95, which indicates that traditional GP can’t assign the suitable weight to each feature. Thus, the improved GP can make the number of key features be more than other original features in the program population. Meanwhile, the rationality of feature weighted matrix is verified, because the matrix gets updated by the number of features through iterations.

| Generation | Genetic operation | \( f_{8} \) | \( f_{17} \) | Average number for each feature |
|------------|-------------------|-------------|-------------|-------------------------------|
| The 5th generation | Improved | 48 | 46 | |
| The 10th generation | Improved | 103 | 109 | 95 |
| Improved | Traditional | 96 | 98 | |

D. CASE TWO: VALIDATION OF THE IMPROVED FITNESS FUNCTION

The feature set from bearing-1 is input. The GP with the improved fitness function is compared to the traditional GP [15] mentioned in section 1, and the iterations is set to 20. The comparison results are shown in FIGURE 9, and in FIGURE 9(c) and (d), the curves are fitted by a seventh order polynomial. Although the fitness of HI output from the traditional GP is 100, which is maximal, the HI is not close to linearity. On the other hand, the fitness of HI output from the GP with the improved fitness function is 98.76, however, the corresponding HI is very close to linearity, so it proved that the improved fitness function is better for discovering the linear HI.

E. CASE THREE: HI AND LIFE PREDICTION

In order to improve the robustness of HI, the number of features in each program is set as four or five in advance. If the number of features is small, the HI are greatly affected by a single feature. Inversely, if the number of features is large, the structure of HI is too complex, which is not conducive to the subsequent evaluation of bearing performance degradation. Eight HIs discovered by GP from bearing-1’s feature set are shown in FIGURE 10, and the expressions are drawn in the Table 3, respectively. Compared with the original features, the fitness of HIs are much higher, and they are 98.55, 97.58, 98.65, 98.04, 97.27, 98.37, 98.14, 98.87, respectively, as shown in FIGURE 11.

In life prediction, it is vital to determine the \( t_{h} \) (threshold) of HI when the bearing approaches the end of life. Assume that every original feature follows Gaussian distribution, such as the program in (6). The probability density function of \( p \) is shown in (14).

\[
\text{pdf}(x) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(x-u)^2}{2\sigma^2}\right) \quad (15)
\]

where

\[
\text{Norm}(\mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(x-u)^2}{2\sigma^2}\right) \quad (15)
\]

is a Gaussian distribution with mean \( u \) and standard deviation \( \sigma \). After that, if HI has an increasing trend [21], the \( t_{h} \) is determined by (16).

\[1 - PFA = \int_{-\infty}^{t_{h}} \text{pdf}(x) \, dx \quad (16)\]

where \( PFA \) is the probability of false alarm. And if HI has a decreasing trend, \( t_{h} \) is shown as (17).

\[1 - PFA = \int_{t_{h}}^{\infty} \text{pdf}(x) \, dx \quad (17)\]
Eight HIs are respectively utilized in life prediction. As HI is very close to the linearity, the least square method is employed to predict the trend of HI. And the focus is on the accuracy of life prediction near bearing failure, thus, the starting time of prediction is set as the 1500th cycle. When HI is reached the \( t_h \), the bearing is broken. Eventually, the prediction result is represented by two curves, one of which is the real life curve, and the value of every point on the curve represents the real RUL at the corresponding cycle. The other one is the prediction life curve, and the value of every point on this curve is the predicted RUL at the corresponding cycle.

MAPE (Mean Absolute Percentage Error) is applied in this paper for judging prediction result, as shown in (18).

\[
MAPE = \frac{1}{m - 1500} \sum_{i=1500}^{m} \frac{|RUL_{real}^i - RUL_{pre}^i|}{RUL_{real}^i}
\]

where \( m \) is the cycle at the bearing failure, and \( RUL_{real}^i \) represents the real RUL at the \( i \)th cycle, and \( RUL_{pre}^i \) represents the predicted RUL at the \( i \)th cycle. To visualize the degradation of bearing more vividly, the cycle of 100 points before failure is set as the demarcation point. Before the demarcation point, the bearing stays at health stage. After the demarcation point, the bearing stays at failure stage, which means the bearing will be broken and should be replaced or repaired immediately. Partial predictions of HIs have accurate results, such as the predictions of the HI-7 and HI-8, as shown in the FIGURE 12. The \( t_h \) of them are \(-3.209\) and \(32.549\), respectively. Nevertheless, the average MAPE of eight HIs is 0.2693. To acquire a HI with more accurate result in the life prediction, further improving the fitness of HI is indispensable.

F. OPTIMIZATION HI IN BEARING-1

This paper puts forward each HI as a single feature, hence, the eight HIs constitute a feature set contained eight features, which is \{HI_1, HI_2, HI_3, HI_4, HI_5, HI_6, HI_7, HI_8\}, then the HI set is input to GP, and the output is a new HI named the optimization HI, as shown in FIGURE 13(a), its expression is shown in (19). Its fitness is up to 98.95, which is higher than the previous HIs, as shown in FIGURE 13(b). Optimization HI is applied to predict the bearing life, and the \( t_h \) is \(-44.17\). And the prediction curve and the real curve are...
FIGURE 12. The prediction of HIs in bearings-1. (a) The prediction of HI-7. (b) The prediction of HI-8.

FIGURE 13. The result of optimization in bearing-1. (a) The curve of optimization HI_1. (b) Comparison of fitness between HIs and optimization HI_1.

FIGURE 14. The life prediction of optimization HI_1 in bearing-1.

shown in FIGURE 14. MAPE is 0.0395. Compared with eight HIs, it is obvious that the optimization HI performs better in the life prediction, as shown in Table 4.

Optimization HI_1

\[
\begin{align*}
\text{Optimization } HI_1 &= \ln (3.72 * HI_2) - 3.77 * HI_3 - \ln(3.62 * HI_1) - 3.62 * HI_8 \\
&= \ln(3.1 * f_9)
\end{align*}
\]

TABLE 4. The prediction MAPE of hi in bearing-1.

| HI     | MAPE  |
|--------|-------|
| HI-1   | 0.1376|
| HI-2   | 0.4106|
| HI-3   | 0.4203|
| HI-4   | 0.2274|
| HI-5   | 0.4609|
| HI-6   | 0.1822|
| HI-7   | 0.1775|
| HI-8   | 0.1379|
| Optimization HI_1 | 0.0395 |

G. OPTIMIZATION HI IN BEARING-2

The same way mentioned in section 3.6 is applied in bearing-2, and the expression of optimization HI is shown in (20). The fitness of optimization HI is 99.02, and the curve of optimization HI is shown in FIGURE 15(a), meanwhile, the comparison of fitness with HIs is shown FIGURE 15(b). The th of optimization HI is −2.61, and the result of life prediction is shown in FIGURE 16. MAPE is 0.0602.

Optimization HI_2

\[
\begin{align*}
\text{Optimization } HI_2 &= (3.89 * HI_6 - \sqrt{4.04 * HI_3 - 3.92 * HI_2}) * 4.02 * HI_5 \\
&= (3.89 * HI_6 - \sqrt{3.92 * HI_2}) * 4.02 * HI_5
\end{align*}
\]

TABLE 5. The expression of HIs from bearing-2.

| HI     | Expression                                      |
|--------|-------------------------------------------------|
| HI_6   | $0.35 * f_14 - 3.12 * f_12 - \ln(3.1 * f_9)$   |
| HI_3   | $(3.12 * f_15 - 3.15 * f_9 - 3.05 * f_20) * \sqrt{3.08 * f_19} (3.15 * f_14)^2/(3.03 * f_11)$ |
| HI_5   | $\sqrt{3.11 * f_5 * 3.03 * f_16}$               |
| HI_6   | $\sqrt{3.05 * f_7} \cdot \sqrt{0.57 * f_6 + 3.14 * f_15}$ |

where these HIs, presented in Table 5, are the HIs discovered by GP from bearing-2’s feature set.
H. Optimization HI in Bearing-3

Repeat the process in the previous section, the expression of optimization HI in bearing-3 is shown in (21). The fitness of optimization HI is 97.82, and the curve of optimization HI is shown in FIGURE 17(a), meanwhile, the comparison of fitness with HIs is shown in FIGURE 17(b). The th of optimization HI is \(-1423.48\), and the result of life prediction is shown in FIGURE 18. MAPE is 0.1002.

\[
\text{Optimization } H_{I_3} = \frac{3.99 * H_{I_2} + (1.31 * H_{I_7})^2}{(3.13 * H_{I_1})^2} - (4.07 * H_{I_8})^2 \tag{21}
\]

where these HIs, presented in Table 6, are the HIs discovered by GP from bearing-3’s feature set.

| HI   | Expression                                                                 |
|------|---------------------------------------------------------------------------|
| H_{I_1} | (ln(3.01 * f21) + ln(3.06 * f17)) * \sqrt{3.02 * f19 - 3.11 * f14} |
| H_{I_2} | ln(3.13 * f6) / (3.15 * f20) - 3.12 * f3                                  |
| H_{I_7} | \sqrt{3.14 * f14} (0.91 * f16 - 1.12 * f9) / 0.81 * f4                  |
| H_{I_8} | \sqrt{3.14 * f12 * ln(3.07 * f16) + 3.2 * f19} / 3.13 * f14            |

\section{H. Validation for Generalization Ability}

For verifying the generalization ability of the novel approach between different types of bearings, the HI set obtained from bearing-1 is employed in bearing-2, subsequently, the HI set of bearing-2 is input into GP and the optimization HI from bearing-2 is output, as shown in FIGURE 19(a), which is applied to predict the bearing life. Its expression is shown in (22), and fitness is 99.05. The th is \(-247.08\), and the prediction curve and the real curve are shown in FIGURE 19(b), MAPE is 0.0568, which mean that the prediction result is also precise. At last, the comparison of optimization HI between different HI sets is shown in Table 7, which indicates that the novel approach has generalization ability.

\[
\text{Optimization } H_{I_4} = (3.89 * H_{I_6} - \sqrt{4.04 * H_{I_3} - 3.92 * H_{I_2}}) * 4.02 * H_{I_5} \tag{22}
\]

\section{IV. Conclusion and Discussion}

In view of the HI construction, which is pivotal in the evaluation of bearing performance degradation, an improved GP is proposed. The efficiency of the algorithm gets promotion by using the weighted feature matrix. Furthermore, it implements the fusion of more features, hence, the robustness of HI is enhanced. Ultimately, the optimization HI is extracted from the HI set. Compared with HIs, the optimization HI is more outstanding in life prediction. To demonstrate the effectiveness of the novel method, the comparisons with other methods including the traditional GP in [15], an important Bayesian machine learning method named Gaussian Process model and support vector regression (SVR) optimized by Particle Swarm Optimization (PSO) are made. The comparison results are shown in Table 8. The results show that our proposed method has a lower error than other methods.

In brief, the approach proposed in this paper provides a new HI construction way, and the HI extracted from original features more directly represents the process of bearing degradation. However, the th setting of HI is a problem, because different types of bearings, errors generated during
bearing installation and working conditions can have significant impacts on the $th$ Thus, how to determine the HI’s $th$ needs further study. On the other hand, the generalization ability of the new approach has limitations. For example, if the HI set obtained from bearing-3 is employed in bearing-2, the optimization HI discovered isn’t close to linearity. The robustness of HI may impair the generalization ability, therefore, it is essential to improve the generalization ability of this method.

### REFERENCES

[1] Y. Lei, N. Li, L. Guo, N. Li, T. Yan, and J. Lin, “Machinery health prognostics: A systematic review from data acquisition to RUL prediction,” Mech. Syst. Signal Process., vol. 104, pp. 799–834, May 2018.

[2] Y. Wang, Y. Peng, Y. Zhi, X. Jin, and K.-L. Tsui, “A two-stage data-driven-based prognostic approach for bearing degradation problem,” IEEE Trans. Ind. Informat., vol. 12, no. 3, pp. 924–932, Jun. 2016.

[3] N. Li, Y. Lei, J. Lin, and S. X. Ding, “An improved exponential model for predicting remaining useful life of rolling element bearings,” IEEE Trans. Ind. Electron., vol. 62, no. 12, pp. 7762–7773, Dec. 2015.

[4] Y. Lei, N. Li, and J. Lin, “A new method based on stochastic process models for machine remaining useful life prediction,” IEEE Trans. Instrum. Meas., vol. 65, no. 12, pp. 2671–2684, Dec. 2016.

[5] Z.-X. Zhang, X.-S. Si, and C.-H. Hu, “An Age- and state-dependent nonlinear prognostic model for degrading systems,” IEEE Trans. Rel., vol. 64, no. 4, pp. 1214–1228, Dec. 2015.

[6] A. H. Elwany and N. Z. Gebraeel, “Sensor-driven prognostic models for equipment replacement and spare parts inventory,” IEEE Trans., vol. 40, no. 7, pp. 629–639, Apr. 2008.

[7] K. Medjaher, N. Zerhouni, and J. Baklouti, “Data-driven prognostics based on health indicator construction: Application to PRONOSTIA’s data,” presented at the Eur. Control Conf. (ECC), Jul. 2013.

[8] P. Boskoski, M. Gasperin, and D. Petelin, “Bearing fault prognostics based on signal complexity and Gaussian process models,” presented at the Prognostics Health Manage. (PHM), IEEE Conf., Jun. 2012.

[9] R. Li, P. Sopon, and D. He, “Fault features extraction for bearing prognostics,” J. Intell. Manuf., vol. 23, no. 2, pp. 313–321, Apr. 2012.

[10] D. Wang and P. W. Tse, “Prognostics of slurry pumps based on a moving-average wear degradation index and a general sequential Monte Carlo method,” Mech. Syst. Signal Process., vol. 65–67, pp. 213–229, May 2015.

[11] A. Widodo and B.-S. Yang, “Application of relevance vector machine and survival probability to machine degradation assessment,” Expert Syst. Appl., vol. 38, no. 3, pp. 2592–2599, Mar. 2011.

[12] H. Ocak, K. A. Loparo, and F. M. Discenzo, “Online tracking of bearing wear using wavelet packet decomposition and probabilistic modeling: A method for bearing prognostics,” J. Sound Vibrat., vol. 302, nos. 4–5, pp. 951–961, May 2007.

[13] H. Guo, L. B. Jack, and A. K. Nandi, “Automated feature extraction using genetic programming for bearing condition monitoring,” in Proc. 14th IEEE Signal Process. Soc. Workshop Mach. Learn. Signal Process., Oct. 2004, pp. 519–528.

[14] H.-C. Tsai, Y.-Y. Tyan, Y.-W. Wu, and Y.-H. Lin, “Determining ultimate bearing capacity of shallow foundations using a genetic programming system,” Neural Comput. Appl., vol. 23, nos. 7–8, pp. 2073–2084, Dec. 2013.

[15] L. Liao, “Discovering prognostic features using genetic programming in remaining useful life prediction,” IEEE Trans. Ind. Electron., vol. 61, no. 5, pp. 2464–2472, May 2014.

[16] W. Banzhaf, F. D. Francone, R. E. Keller, and P. Nordin, Genetic Programming: An Introduction: On the Automatic Evolution of Computer Programs and its Applications. San Mateo, CA, USA: Morgan Kaufmann, 1998.

[17] S. Sette and L. Boullart, “Genetic programming: Principles and applications,” Eng. Appl. Artif. Intell., vol. 14, no. 6, pp. 727–736, Dec. 2001.

[18] R. G. P. Nectoux, K. Medjaher, and E. Ramasso, “PRONOSTIA: A systematic review from data acquisition to RUL prediction,” Mech. Syst. Signal Process., vol. 40, no. 4, pp. 1214–1228, May 2015.

[19] Z. K. Peng and F. L. Chu, “Application of the wavelet transform in machine condition monitoring and fault diagnostics: A review with bibliography,” Mech. Syst. Signal Process., vol. 18, no. 2, pp. 199–221, Mar. 2004.

[20] Z. Li, Z. Wu, Y. He, and C. Fulei, “Hidden Markov model-based fault diagnostics method for bearing prognostics,” J. Sound Vibrat., vol. 302, nos. 4–5,