Hindawi
Shock and Vibration
Volume 2022, Article ID 9010419, 16 pages
https://doi.org/10.1155/2022/9010419

Research Article

An Improved Recursive ARIMA Method with Recurrent Process for Remaining Useful Life Estimation of Bearings

Zeyu Luo,[1] **Xian-Bo Wang**,[1,2] **and Zhi-Xin Yang**[1]

[1]*State Key Laboratory of Internet of Things for Smart City, University of Macau, Macau SAR 999078, China*
[2]*College of Electrical Engineering, Henan University of Technology, Zhengzhou 450001, China*

Correspondence should be addressed to Zhi-Xin Yang; zxyang@um.edu.mo

Received 13 July 2021; Revised 18 January 2022; Accepted 26 January 2022; Published 21 February 2022

Academic Editor: Alba Sofi

A typical way to predict the remaining useful life (RUL) of bearings is to predict certain health indicators (HIs) according to the historical HI series and forecast the end of life (EOL). The autoregressive neural network (ARNN) is an early idea to combine the artificial neural network (ANN) and the autoregressive (AR) model for forecasting, but the model is limited to linear terms. To overcome the limitation, this paper proposes an improved autoregressive integrated moving average with the recurrent process (ARIMA-R) method. The proposed method adds moving average (MA) components to the framework of ARNN, adding the long-range dependence and nonlinear factors. To deal with the recursive characteristics of the MA term, a process of MA component estimating is constructed based on the expectation-maximum method. In the concrete realization of the method, the rotation tree (RTF) is introduced in place of ANN to improve the prediction performance. The experiment on FEMTO datasets reveals that the proposed ARIMA-R method outperforms the ARNN method in terms of predictive performance evaluation indicators.

1. Introduction

Bearings are critical parts of most rotating machines. They carry the resolves of shafts, but also bear the largest pressure. 44% of failure of some rotating machines are due to the malfunction of bearings, which cause force focusing on other fragile parts and then lead to systematic failure [1–4]. A difficulty in the maintenance is that the replacement of bearings needs disassembly of all related parts and requires planned maintenance based on their health condition. The necessity of prediction of the degradation process, therefore, arises that if the degradation can be predicted somehow, maintenance can be arranged to examine the least parts and repair most potential faults. Therefore, the estimation of the time before failure, that is, the remaining useful life (RUL) of bearings, is an important part of rotating machines' prognostics and health management (PHM) [5–7].

In the prediction of RUL, because of the nonlinearity and condense noise mixed within, certain forms of health indicators (HIs) are proposed to simplify the prediction. HIs are generally quantities that can be calculated according to life data, and they are designed to be more predictable than RULs and keep consistency with RULs. That is, the time HIs reach a given threshold corresponds to the time RULs are depleted, and the end of life (EoL) of the measuring equipment is met [8].

Although the end of life (EoL) of the equipment can be judged by the failure criterion, the service life itself is not a physical quantity that can be directly observed. It needs to be achieved through indirect methods (i.e., the state observation signal is processed to obtain indicators that reflect the degradation state and then the health indicators (HIs) are used to determine the EoL location with HI degradation trend prediction).

Generally, degradation prediction methods of bearings can be classified as physical model-based and data-driven-based approaches. Physical model-based approaches describe the whole system with a comprehensive mathematical model, indicating the characteristics and failure patterns of the system, especially the occurrence and growth of cracks, and the modes and energy of vibration [9–11]. Alternatively, data-driven methods dynamically build a model for the

observed machine based on the signals acquired, with a certain prior knowledge that usually comes from previous data collected. Compared with the model-driven methods, the digital model based on the data-driven monitoring method is not accurate, but from the perspective of feasibility, the observation data required by these methods are easy to be obtained, are more versatile, and can provide satisfactory prediction requirements [12–17].

It is a very popular path to build a hybrid model combining the ML-based models with time series analysis (TSA) [18–20]. Methods based on TSA deem the degradation of the equipment like a process continuous in time. The autoregressive (AR) model and its successive models are widely used time series models in prognostics. In this family of models, the past signal is linearly mapped to the current signal, and the current signal is the linear polynomial summing up the historical signals and other random factors. The moving average (MA) models are combined with the AR model to introduce long-range dependence missed from the AR model, which becomes the ARMA model [21]. Derived from this combination, the differential process is further added to it, giving rise to the autoregressive integrated moving average (ARIMA). Researchers solve forecasting tasks with all types of ARIMA models since the time they are invented [22–24]. The long-range dependence expressed in the MA terms of these models is critical in the prognostics of bearings [25]. In addition, Qiu et al. [26] compared the methods of predicting HIs and suggested that an ARMA model achieves well balances in computing complexity and performance.

Nevertheless, the traditional ARMA model solving procedure needs to repeatedly fit a variety of AR and ARMA models of different orders, which are lengthy and difficult for large data. So, it is appealing to combine ARIMA with ANN to simplify the solving. Autoregressive neural network (ARNN) is an early combination of ML and TSA [27]. The ARNN model directly uses historical observation data as the input of the neural network to solve the next observation, but different from the original model, the outputs are not necessarily obtained by a linear combination of inputs. It mends AR with nonlinear trait and black box solving of parameters and confines the input features of ANN to reduce overfit and difficulties on training. Because of the nonlinear trait of ARNN, it is also called nonlinear autoregressive neural network (NARNN) [28–31].

ARNN is based on the AR model, so the long-term coefficients described by the MA part of the ARMA model will be neglected. As [25] suggests, such leakages will lead to inaccurate prediction of HIs. Because ARMA describes more complicated models and the ARIMA is more universal than ARMA, it is natural to expect that a machine learning method combines with the ARMA or ARIMA model.

The main obstacle of such a combination of machine learning algorithms and the ARMA or ARIMA model is that the MA parts are basically the error terms between real and predicted observation, so it refers to the model result, which is still unknown when solving the model. Thus, the family of ARMA models is hard to be modeled by the ANN and other similar ML structures [32–34]. In the application of time series models to HI or RUL prediction, researchers mainly combine the ARMA family and ML methods as two independent parts, such as combining ARMA and SVM [35] and particle filter (PF) [23]. Even the ARNN has been used for such simply combination [36], or adding complexity from exogenous input to it [37]. In this type of combination, parameters and outputs of ARMA models are calculated by the traditional method and the outputs are added to predictions of other models to obtain the final result.

The contribution of this work lies in the following two aspects: first, this paper proposes an ARIMA-R method as the implementation framework of approximating the ARIMA model with ML, which is realized with the moving average approximation by recurrent means. Second, a test in series generated from an ARIMA model is carried out to examine the proposed ARIMA-R method, and the result shows that the method can estimate the next observation in such series with high accuracy.

The rest of this paper is organized as follows: Section 3 describes the ARIMA model and proposes the ARIMA-R method on its basis; Section 4 expresses the verification on the ability of the ARIMA-R model in describing generated ARIMA series and then uses the method to estimate bearing HIs through the public FEMTO dataset, with analysis of the parameter settings of the method; Section 5 concludes the whole study.

2. Methodology

2.1. ARIMA and ARNN. The ARIMA model originates from the AR model and MA model [21]. These models belong to the same type, depicting the autocorrelation of series data, which is the correlation between the data and its "lagged" copy. If autocorrelation exists, those data of the past will influence the future in some ways, usually described by functional relationships. Autocorrelation is common in times series analysis since most time series data reflects the inherited causality of the observation subject.

Assuming the autocorrelation in series is linear, the AR model and MA model can be obtained. In the AR model, the result is related to its previous observation and expressed by the linear combination of historical observations. Different from the AR model, the output in the MA model is explained by the errors of historical predictions. The prediction errors, implying the relationship between output and its trend or expectation, include accumulative historical effects and long-range dependency, as well as the influences of random history.

The combination of AR model and MA model is the ARMA model, while its improvement by applying differencing to both inputs and outputs results in the ARIMA model. The ARMA model includes short-term autoregressive and long-term moving average, which can better describe time series. However, ARMA inherits the strict requirements of the AR and MA models for the sequence to ensure the existence of autocorrelation. The sequence subject to the ARMA model is required to have weak stationarity, that is, zero mean and constant variance. The ARIMA model weakens this requirement by introducing the difference of

the sequence. It only needs the sequence to have differencing stationarity; in other words, it can become a stationary sequence after a finite number of differences.

For convenience, we denote the lag operator L with $L^d x_t = x_{t-d}$. Then, given a time series observation $X = x_1, x_2, \ldots, x_t$, the ARIMA model describes the series as

$$\begin{aligned} y_t &= (1-L)^d x_t \left(1 - \phi_1 L - \cdots - \phi_p L^p\right), \\ y_t &= c + \left(1 + \theta_1 L + \cdots + \theta_q L^q\right)\varepsilon_t, \end{aligned} \tag{1}$$

where c is the base value of X; $\varepsilon_s = \hat{y}_s - y_s$ is the historical prediction error (assumed to be white noise when $s = t$); p is the order of AR terms $(1 - \phi_1 L - \cdots - \phi_p L^p) y_t$, q of MA terms $(1 + \theta_1 L + \cdots + \theta_q L^q)\varepsilon_t$, and d of differencing orders; ϕ and θ are the parameters of the AR and MA (q) model, respectively. To emphasize the order parameters, the model may be written as follows: ARIMA (p, d, q).

This description takes the series as the d order sum of linear combination by multiple predictors in terms of AR, MA, and random noise. The orders of AR, MA, and the sum determine the specific form of ARIMA model, while the parameters of each term are to be solved to provide the prediction. With the variety of orders, the ARIMA model can generate time series ranging from white noise and random walk to complicated drift with the quadratic trend.

The simplicity of ARIMA model requires certain features of the subject data. The critical requirement is the autocorrelation of the series, so the autocorrelation function (ACF) and partial autocorrelation function (PACF) need to be calculated from the data to determine which order of AR and MA terms are available in the prediction.

A reliable procedure has been described by Box and Jenkins [38] to build the ARIMA model, including the following three steps: (1) model identification and determination of the orders p, d, and q, (2) parameter estimation of AR and MA terms, and (3) prediction checking and performance evaluation.

Proposed by [27], the ARNN attempts to use ML methods to solve AR models. The output variable (i.e., target) (p) can be expressed as follows:

$$\hat{y}_t = h\left(c, y_{t-1}, \ldots, y_{t-p}\right), \tag{2}$$

where $h(\cdot)$ is the nonlinear transformation obtained by the neural network.

Because the neural network can produce nonlinear features by introducing activation functions with strong nonlinear features, the application of activation functions with weak nonlinear characteristic in ARNN can achieve an effect that is almost equivalent to the AR model. However, this method is not expanded to the ARMA model because simply using the input data cannot directly obtain the prediction error features.

2.2. Rotation Forest. Rotation forest is an ensemble ML method, which integrates weak learners to gain accurate results. It is originally proposed only for classification purposes in [39], but the regression version was implemented later in [40].

In the field of ML, ensemble methods mainly include bagging (which is to train the classifiers with partial sample) and boosting (which is used to chain the weak classifiers for reinforcement). On the basis of bagging, the rotation forest constructs different features by rotating the features in feature space, thereby generating diversity on the classifiers. Different from bagging, although bootstrapping is used in the rotation forest method, instances are not discarded during classification, and all information about features is retained. The procedure of rotation forest can be depicted as follows.

Let $X = [x_1, x_2, \ldots, x_N]^T$ be a dataset with N instances, where each x_i instance includes m features in m-dimensional feature space F^m, $x_i = [Fx_{ii,1}, Fx_{i,2}, \ldots, Fx_{i,m}]^T \in F^m$. Therefore, X is an $N \times m$ matrix. Let $Y = [y_1, y_2, \ldots, y_N]^T$ be the regression target with respect to X.

Rotation forest generates l decision trees, D_1, D_2, D_l, to form a rotation tree.

The training set for an individual DT is processed with the following steps:

Step 1: randomly divide the feature set F into k subset $F_1, F_2, \ldots, F_k$. Assume each subset contains $q = m/k$ features.

Step 2: for each subset F_j, let X_j be the training set with only features in F_j and a bootstrap subset of X_j is drawn and used to form a new training set denoted by X'_j.

Step 3: principal component analysis (PCA) is applied to each X'_j and gives weight matrix C_j.

Step 4: all k matrices, C_1 to C_k, is used to construct a rotation matrix R in the form of diagonal block matrix that C_1 to C_k is put to the main diagonal blocks, and set all other blocks to zero as shown in equation (3). The columns of R, which are divided from feature F, are then rearranged according to F, denoted as R'.

$$R = \begin{vmatrix} C_1 & \{0\} & \ldots & \{0\} \\ \{0\} & C_2 & \ldots & \{0\} \\ \ldots & \ldots & \ldots & \ldots \\ \{0\} & \{0\} & \ldots & C_k \end{vmatrix}. \tag{3}$$

Step 5: the training set for a DT is given by $T = XR\prime$. DT is then trained by T and Y.

All l trees estimate the regression target with y^i. The average of all y^i is taken as the output of the rotation tree as in

$$\hat{y} = \sum_i \widehat{y^i}. \tag{4}$$

For the entire training set X_t, randomly generate a subset F_j of all features F_t for K times. For each subset F_j, the features in the intersection of F_j and X are selected to create the corresponding subset X_j. A random replacement

sampling is performed in X_j to produce the bootstrap sample X'_j. With K bootstrap sample $X'j$, PCA is used on $X'j$ to obtain the component matrix C_j, and all C_j rearrange internally to form a block diagonal matrix R as the rotation matrix. A rotation regression tree D is trained based on the sample X, Y and matrix R:

$$D = f\colon T \longrightarrow Y = f\colon XR' \longrightarrow Y. \tag{5}$$

To further increase the number of ensemble samples, repeat the above process L times to obtain L rotating trees and then use all rotating trees to form a rotation forest. The average prediction of all trees is the output of the rotation forest.

3. ARIMA-R

3.1. Flowchart and Basics of Proposed ARIMA-R. Compared with the AR model, the solution of the ARIMA model is more difficult and complicated. To determine the three order parameters of p, d, and q, it is necessary to enumerate different differential order d and search for the proper combination of p and q. Given a tentative d, the inputs and outputs are differenced to d order for successive calculation. The autocorrelation function (ACF) is calculated with respect to all possible lags and then another tentative middleware; the AR model is fit by the differential data for the partial autocorrelation function (PACF) or directly maximum likelihood estimation (MLE). The tentative p and q are determined by the result of ACF and PACF; then, the still tentative ARIMA model is finally solved only for evaluating the criterion such as Akaike information criterion (AIC). After testing multiple combinations of parameters, the order parameters are finally determined.

As shown in Figure 1, to simplify the solving of ARIMA model, a framework called ARIMA-R is constructed based on the ARNN method. The framework takes series data as input and provides its prediction as output.

Given a time series S and its observation X, where the time series is discrete and X contains observations until time $t-1$, $X = \{x_{t-1}, x_{t-2}, \ldots, x_1\}$, the prediction target is the observation x_t at time t. According to the ARIMA model, given orders p, d, and q, the target x_t can be expressed as follows:

$$\begin{aligned}\left(1 - \phi_1 L - \cdots - \phi_p L^p\right)(1-L)^d x_t &= c \\ &+ \left(1 + \theta_1 L + \cdots + \theta_q L^q\right)\varepsilon_t, \\ \varepsilon_\tau &= (1-L)^d \widehat{x}_\tau - (1-L)^d x_\tau,\end{aligned} \tag{6}$$

where $\phi_1, \phi_2, \ldots, \phi_p$ are the coefficients of AR terms, $\theta_1, \theta_2, \ldots, \theta_q$ are the coefficients of MA terms, ε_τ is the prediction error at time point τ, when $\tau = t$, ε_τ meets the Gaussian distribution, L is the lag operator, and c is the constant. For convenience, assume $y_t = (1-L)^d x_t$ as the observation, and the prediction error can be written as $\varepsilon_\tau = \widehat{y}_\tau - y_\tau$.

For calculation of ε_τ, we introduce the recurrent moving average (RMA) terms $(1 + \theta_1 L + \cdots + \theta_q L^q)\varepsilon_t^{(n)}$, where $\varepsilon^{(n)} = \widehat{y}^{(n-1)} - y$ is the nth generation error of prediction and real history.

For recurrent prediction, the model is depicted as follows:

$$\begin{aligned}&\left(1 - \phi_1 L - \cdots - \phi_p L^p\right)\widehat{y}_t^{(n)} \\ &\quad = c + \left(1 + \theta_1 L + \cdots + \theta_q L^q\right)\varepsilon_t^{(n)} \\ &\quad = c + \left(1 + \theta_1 L + \cdots + \theta_q L^q\right)\widehat{y}^{(n-1)} - \left(1 + \theta_1 L + \cdots + \theta_q L^q\right)y.\end{aligned} \tag{7}$$

The input term $F_t^{(n)}$ of ARIMA-R can be described as follows:

$$F_t^{(n)} = \left\{c, Ly_t, L^2 y_t, \ldots, L^p y_t, \varepsilon_t^{(n)}, L\varepsilon_t^{(n)}, \ldots, L^q \varepsilon_t^{(n)}\right\}. \tag{8}$$

Using the rotation forest based on regression tree, the prediction output of this generation can be given as follows:

$$\widehat{Y}^{(n)} = \widehat{y}_1^{(n)}, \widehat{y}_2^{(n)}, \ldots, \widehat{y}_t^{(n)} \text{where } \widehat{y}_t^{(n)} = \frac{1}{L}\sum_{i=1}^{L} D_i\left(F_t^{(n)}\right), \tag{9}$$

where L is the number of rotation trees (e.g., the magnitude of ensemble) and D_i is the single regression tree found by rotation forest algorithm.

After each iteration, if the parameters of MA terms converge or the maximum iteration generation R is reached, the recursion ends. In the end, calculate x_t from $\widehat{Y}^{(n)}$ referring to the reverse of $y_t = (1-L)^d x_t$. The first prediction $\widehat{y}^{(0)}$ needs to be estimated by other means. Considering that the evolution of regression tree may fall into local maxima, a more accurate estimate of y should be selected for the preliminary prediction value. The most universal way is to set it as equal to the last historical data x_{t-1}, which can be regarded as a coarse solution of AR(1) model. Given the condition of stationary including zero-mean sequence, an estimation of always 0 is also acceptable.

3.2. Order Determination in Proposed ARIMA-R. The main purpose of order determination is to estimate the polynomial form when building the model. Too few model variables cannot give a model that can describe the data well; too many variables are avoided in traditional modeling based on the principle of reducing assumptions, but in ML because the solution is approximate, too many feature variables are likely to cause overfitting. In the method, we choose to use an integrated algorithm based on regression decision tree-rotating tree algorithm to alleviate this problem when solving the model.

The regression tree algorithm uses features to divide the output space, and it applies the information entropy to divide the input feature space. If some features do not produce meaningful division, the pruning algorithm will be used to remove the judging branch of the feature. This study applies the bagging strategy to generate samples with diverse characteristics through random sampling. A shallow weak decision tree is generated for each sample, and the final regression result is generated. Since the weak decision tree naturally tends to be underfitting, plus that there are pruning

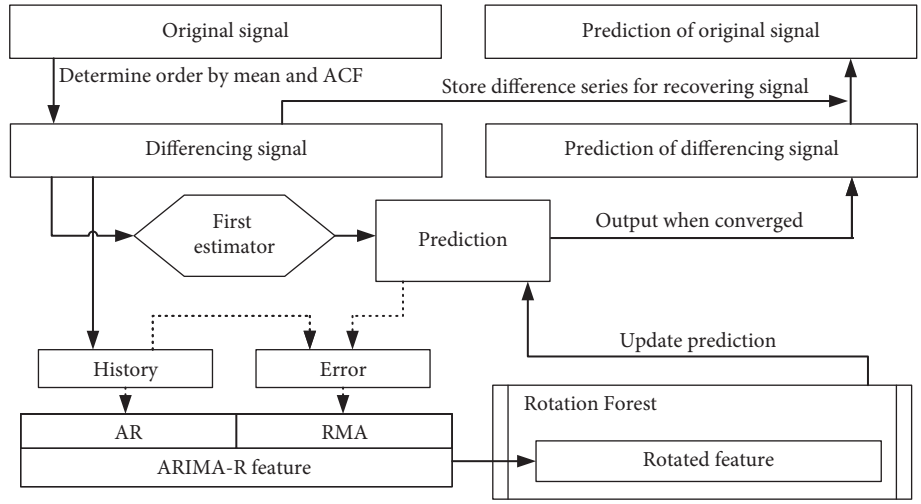

Figure 1: Flowchart of ARIMA-R.

algorithms to further reduce the decision branches, the integrated decision tree algorithm can effectively alleviate the overfitting problem caused by too many features.

3.3. Discussion about MA Terms. As shown in (1), it is easy to see that the different process in the model only uses the different observation data and the prediction error of the different data as input and the difference of the prediction result as the output. In other words, this process equally affects historical data as input and prediction results as output. Therefore, the actual problem to be predicted after d-order differential process is the same in form as the ARMA model. In this way, it is also possible to study the ARIMA model in the same way as the ARMA model.

When trying to use ML methods such as neural networks to model the ARMA model, the MA term in the ARMA model, that is, the random error term, is difficult to deal with. This term represents the error between the past predicted results and the actual results. Before modeling this term, one must first obtain all the predicted outputs before this input when training each input. This also means that a model for each data point needs to be trained step by step, instead of training a model for all data at once. In addition, in this case, there will be no consistency between models for each sequential data point and it is possible that a model only for this moment may be trained, and it has no predictive ability for data in other time periods [18, 41].

According to the definition of the MA term, the MA term is the error between the predicted result and the actual result, so the ARMA model can be rewritten as follows:

$$\begin{aligned}&\left(\left(1-\phi_1 L-\cdots-\phi_p L^p\right)+\left(1+\theta_1 L+\cdots+\theta_q L^q\right)\right)y_t\\&\quad=c+\left(1+\theta_1 L+\cdots+\theta_q L^q\right)\widehat{y}_t.\end{aligned}\tag{10}$$

As shown in (10), there are p groups of parameters for AR terms and q parameters for MA terms to be solved in the building of the model. It can be seen that the ARMA model is actually a linear model about q items of historical prediction results and $h = \max\{p, q\}$ items of historical observation data. The long-range dependence of the MA item is also because if the historical observation term is expanded each time the MA term is expanded, the previous h items of historical data will be added to the model. This also explains why the MA term should not be directly assumed as a series of uniformly distributed random variables. Although the historical prediction $\widehat{y}$ is an estimate of historical data y, it is generally assumed that the estimated error $y - \widehat{y}$ is a normal distribution with a fixed variance of zero means, but in fact, if the error is directly assumed to come from a zero-mean normal distribution, or if it is assumed to be zero, it will lose earlier observations and cause greater losses.

According to the above conclusions, the ARMA model can be established and solved by ML methods. However, the problem of how to calculate the predicted results before giving the model parameters and solving the model is still to be solved. Traditionally, the parameters are solved by calculating their maximum likelihood estimation (MLE) or least-squares estimation (LSE). However, considering that what we really need is the single-step estimation of time series prediction, it is also reasonable to calculate the result by training an artificial neuron network (ANN) or other regressors. However, in the ARNN model, MA term is excluded, so there is no solution in ARNN implementation.

This work introduces the idea of expectation maximization (EM) algorithm to carry out the iterative process [42]. The EM algorithm can calculate the MLE when the parameters are incomplete. For the statistical model of the observation data X and the unknown data Z, the parameter Θ needs to be estimated. Each iteration of the EM algorithm consists of the following two steps:

(1) Step E is to calculate the expected value of Z based on the existing model data X and the parameter th and use the expected value to replace the true value of the missing data and calculate the MLE of the parameter th

Input: $X = x_1, x_2, \ldots, x_{t-1}$
(1) $d \longleftarrow 0$
(2) **repeat**
(3) Y: $(1 - L)^d X$
(4) $D_d = Y$
(5) Calculate ACF of y
(6) $d \longleftarrow d + 1$
(7) **until** $E(y) - 0 <$ eps and ACF converge \(\triangleright\) the limit eps depends on practical situation.
(8) First guess:
(9) **for** training and testing set do
(10) Calculate Y_0 by $Y_0 = L^1 Y$ \(\triangleright\) other prediction methods also work.
(11) : **end for**
(12) : $k = 1$
(13) **repeat**
(14) **for** training and testing set do
(15) Calculate **RMA**
(16) $e\langle k\rangle = y - y\langle k - 1\rangle$
(17) Construct feature
(18) $F^{(k)}$: $F_t^{(k)} = \left\{y_{t-1}, \ldots, y_{t-p}, e_{t-1} \ldots e_{t-q}\right\}$
(19) **end for**
(20) Train the Rotation Forest in training set
(21) **for** training set do
(22) $y = D^{(k)}(F^{(k)})$
(23) **end for**
(24) Calculate $y\langle k\rangle$
(25) **for** training and testing set do
(26) $y = D^{(k)}(F^{(k)})$
(27) **end for**
(28) Calculate changes
(29) $s = \mathrm{MSE}(y^{(k)}, y^{(k)})$
(30) **until** $k > k_max$ or $s < eps$
(31) Construct $F<k>$
(32) **for** j from 1 to k do
(33) Calculate $e^{(k)}, F^{(k)}, y^{(k)}$
(34) **end for**
(35) $x_t = D(F^{(k)}) + D_1 + \cdots + D_{d-1}$
Output: estimated output x_t

Algorithm 1: ARIMA-R.

(2) Step M is to use the MLE of the parameter Θ given in Step E to find the parameter Θ when the MLE is minimized

After each iteration, update the parameter Θ and check whether it has converged. If it converges, the estimation is ended. In the EM method, the missing data Z is the expected value calculated based on other data, and the initial value of the parameter Θ is a random value.

On the one hand, the data to be estimated in step E are historical forecast data $\hat{y}$, and the expected value of $\hat{y}$ is consistent with the historical data y, which will cause the MA item to always be zero. On the other hand, if the estimated value of $\hat{y}$ is related to earlier historical data, it will speed up the convergence of the prediction model. In addition, if the parameter values of the ARMA model are initialized randomly, it is likely that the calculation result completely deviates from the true value, leading to overfitting or falling into local optimal problems. Therefore, the choice here is to use an AR(1) model to predict y and use this prediction result as the historical prediction data $\hat{y}$ and then directly solve the model parameters. Since the ARMA model describes a zero-mean sequence with stationarity, it is feasible to use zero as the prediction result.

Step M is obtained by solving and optimizing the MLE in the EM algorithm. In the method proposed here, the rotating forest can be used directly to obtain the model parameters, so only the model parameters need to be saved for the next iteration.

Only for single-step prediction, the proposed framework is useful, while on the multiple-step prediction, there is a limit to the ARIMA-R method. Since the ARIMA model is modeling the next value of observation, such a model is not fit to direct regression of future value. The only method is to predict the future step by step. In the calculation of prediction, the future errors are assumed to be zero and the future observations are replaced with the prediction before. Such treatment means that, on prediction, an ARIMA (p, d, q) model is degraded to an ARIMA $(p, d, 0)$ model. This means that the proposed ARIMA-R model must

similarly reduce to the ARIMA $(p, d, 0)$ model. Therefore, it turns out that, on multiple-step prediction of enough length, the recursive order r should be 1 or 0. In such a situation, this model is indiscriminate from the normal ARNN model.

3.4. Analysis of Computational Complexity. To analyse the computational complexity of ARIMA-R, we adopt the big O notation to describe the relation between operation quantities and input sizes. A complexity $O(p(N))$ means that when the input size N approaches infinite, the time of operation is infinite with the same order as $p(N)$.

For a signal series with length N, given orders p and q, our method needs r times of iteration in which a rotation forest is trained, where r is a small constant. Each time a rotation forest is trained, L trees are calculated. Every rotation tree needs K times of PCA and to train a decision tree based on the rearranged PCA results. In this process, L and K are predetermined parameters. Therefore, the algorithm needs to calculate $K \times L$ times of PCA and train L decision trees, in addition to L times of extra matrix multiplication. Reference [43] shows that the complexity of PCA with d-dimension vectors is $O(d^2N + d^3)$; reference [44] indicates that training a balanced regression tree on d-dimension data requires $O(dN\ logN)$ times of multiplication. We also know that a matrix multiplication between an m-by-n matrix and an n-by-p matrix is $O(\text{mnp})$. Summing up the results, the complexity is

$$O\Big((p+q)^2N + (p+q)^3 + (p+q)N\ \log\ N + (p+q)^2N\Big), \tag{11}$$

which is rewritten as

$$O\Big(s^2N + s^3 + sN\ \log\ N\Big) = O(N\ \log\ N), \tag{12}$$

where $s = p + q$ is relatively small.

Then, we come to ARIMA. As indicated by [45], the computational complexity of ARIMA itself is $O((N-p)p^2 + (N-p)q^2)$, which is better than the proposed ARIMA-R with fixed p and q. Nonetheless, the computing burden lies on the order determination part. Following the Box–Jenkins method (if assume p_max and q_max), given fixed differential parameter i, $p_max q_max$ times of searches are necessary to determine the proper order parameters p and q for ARIMA. For each parameter combination, ACF and PACF to certain orders, such as p_max order, should be calculated to obtain the AIC and determine parameters. Reference [46] suggested that if we use the common Levinson–Durbin method to solve ACF and PACF, the complexity of ACF is $O(N^2)$ and that of PACF is $O(q^2N^2)$. Because the total times of ACF and PACF solving in the Box–Jenkins method are $p_{\max}q_{\max}$, the complexity in this part is $O(p_{\max}q_{\max}q^2N^2) = O(N^2)$. In total, the complexity of ARIMA is

$$O\Big((N-p)p^2 + (N-p)q^2 + p_{\max}q_{\max}q^2N^2\Big) = O\Big(N^2\Big). \tag{13}$$

It is worth noting that the ACF and PACF solving is also needed in ARIMA-R, while the total times are reduced to $p_{\max}$. With the common operations eliminated, traditional ARIMA with the Levinson–Durbin method still needs $p_{\max}(q_{\max} - 1)$ times of searches, so the result keeps unchanging.

The above analysis suggests that the proposed ARIMA-R algorithm reduced the computing burden in order determination, in the cost of complexity increasing in parameter approximation. Overall, the computational complexity of ARIMA-R is reduced.

4. Experimental Verification in Prognostics of Bearings

Three experiments are conducted to verify the ability of the proposed ARIMA-R method: an experiment with simulated data, a single-step forecast experiment based on bearing remaining useful life datasets, and a long-term prediction experiment.

To compare the prediction errors of ARNN and that of ARIMA-R for each sample, four indicators, mean square error (MSE), mean absolute error (MAE), coefficient of determination (R^2), and mean absolute percentage error (MAPE), are introduced. Among the indicators, MSE, MAE, and MAPE are better if they are closer to 0, which means the prediction result is closer to the predicting target. R^2 indicates the proportion of the target that can be explained by the prediction, so the value of R^2 should close to 1. Mathematical expressions of these indicators are shown as

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \widehat{y}_i\right)^2, \tag{14}$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \widehat{y}_i\right|, \tag{15}$$

$$\begin{aligned} R^2 &= 1 - \frac{\sum_{i=1}^{n}\left(y_i - \widehat{y}_i\right)^2}{\sum_{i=1}^{n}\left(y_i - \overline{y}\right)^2} \\ &= 1 - \frac{\text{MSE}}{\text{Var}}, \end{aligned} \tag{16}$$

$$\begin{aligned} \text{MAPE} &= \frac{|y - \widehat{y}|}{|y|} \\ &= \frac{1}{n}\sum_{i=1}^{n}\frac{\left|y_i - \widehat{y}_i\right|}{|y|}. \end{aligned} \tag{17}$$

4.1. Verification with Simulated Data. To verify the prognostic capacity of the proposed ARIMA-R method, this paper selects the following parameters to establish a curve that meets the ARIMA model, as shown in Table 1.

Since the MA term actually uses random noise to replace the unknown past data when generating the ARIMA curve in simulation, other forms of the curve may also be generated with the same parameters and starting point. ARNN and ARIMA-R were used to build a model to solve the first 100 data points of the curve and then used to make a single-step prediction for the last 100 data points. The results are

Table 1: Parameters to simulate ARIMA data.

Term	Order	Order	AR param.	MA param.
AR	2	1	0.5	0.65
MA	4	2	−0.75	−0.5
Diff.	2	3		−0.2
		4		−0.1

shown in Figure 2. This experiment is a single-step prediction, that is, for each time point t, the data up to time point $t-1$ is used to predict the observation result. The x-axis in the figure is the order of observation samples, and the y-axis is the actual HI value and the difference value of HI, respectively. Figure 2(a) shows that the 1–100 observations are used as training data and the 101–200 data are used as test data. Figure 2(b) shows the different prediction results of the test data part. It can be clearly seen that the prediction of the ARIMA-R method is more accurate than that of the ARNN method.

However, it should be emphasized that such high performance comes from that the data are purely generated by the ARIMA model and in single-step prediction. For long-term prediction, random factors will dominate the result. The long-term result in Figure 3 indicates that the prediction error blooms gradually after about 15 iterations of long-term prediction.

4.2. Experiment Setup and Procedure for Single-Step Forecast. In the prognostic experiment, vibration signals of bearings at the run-to-fail situation are used. The bearing dataset is provided by FEMTO Institute, published on the NASA Prognostics Center of Excellence (PCoE) [47]. Two orthogonally installed accelerometers in horizontal and vertical directions on the housing of bearing collect vibration signals in the form of acceleration, as shown in Figure 4. For both signal channels, the sampling frequencies of signals are 25.6 kHz, while the data are collected every 10 seconds for 1 second. For all conditions and bearings, the data acquisition is begun at the normal state of bearings and ended when the bearings fail. The detail of the platform and the experiments can be obtained in [47]. The dataset includes 17 subsets under three different conditions, as shown in Table 2.

The root mean square (RMS) of the vibration signals is calculated and provided as the HI prediction target. Each set of data is a series of observation samples arranged in time order, so the time series analysis is used for processing the signals. The bearing 1_4 data are taken as an example in Figure 5. The RMS data are a flat small value at the beginning, namely, platform period, and in the middle of the wearing period, abnormal values that far exceed the average value of the platform period frequently appear; finally, in the final stage, the RMS value increases abnormally, until the bearing fails completely.

The division of stages is carried out according to the position of the node where the trend of the data changes suddenly. Since the ARIMA model is a model based on sequence prediction, it is not suitable for predicting the data when sudden changes occur. So, it is better not to make cross-stage predictions. Here, the data of the failure stage are mainly selected as the prediction object for method verification. In the process of model training using ML algorithms, the data need to be divided into a training set and a validation set. Under the assumption that the samples of the two datasets follow the same model, the training set is used to determine the model parameters and then the validation set is used to verify the performance of the model. Generally speaking, the training set and the validation set are randomly selected regardless of the order of the data, but the ARIMA model is a time series model; thus, training the model with future data will result in inaccurate performance in the past data in the validation set. Therefore, a certain length of data is selected as the training set from the beginning of the overall sequence, and the subsequent sequence is used as the verification set.

In this experiment, after selecting the training data, the key parameters can be determined through data research, mainly to determine the order of the ARIMA model, including AR order p, MA order q, and differencing order d. As mentioned in the problem of order determination in Section 2.2, the ARIMA-R method does not require precise order determination, but it is still necessary to carry out the difference and calculate the ACF to determine the differencing order d and determine the reference values of p and q.

The data are generally divided into a training set and a validation set for training the model and a test set for testing the performance of the model. However, a special division is required here. The data are first divided into normal phase data and abnormal phase data according to the trend, and then the abnormal data are used to train the model [48]. The reason is that the data of different phases cannot be regarded as in the same distribution, so it cannot be learnt by the same model. In this study, bearing datasets 1_4, 1_6, and 3_3 are used as the training target for prediction, where first 1/3 of data are selected as the training samples, and the rest are for testing. The data of these bearings in the abnormal state are shown in Figure 6.

The ACF and PACF of RMS itself and its difference are represented in Figure 7, from dataset 1–7. This dataset is not used in the prediction, and the ACF/PACF results are taken as reference in order determination. In order to be processed using the ARIMA-R model, the sequence used for model prediction needs to be a sequence of approximately zero mean after differencing, so the ACF function of the sequence is calculated first, and the PACF function is calculated based on the AR (15) model, whose orders are determined by the result of ACF function. According to the lags that the ACF/PACF value is above the significance line, the orders of model are determined as shown in the ARIMA-R part of Table 3.

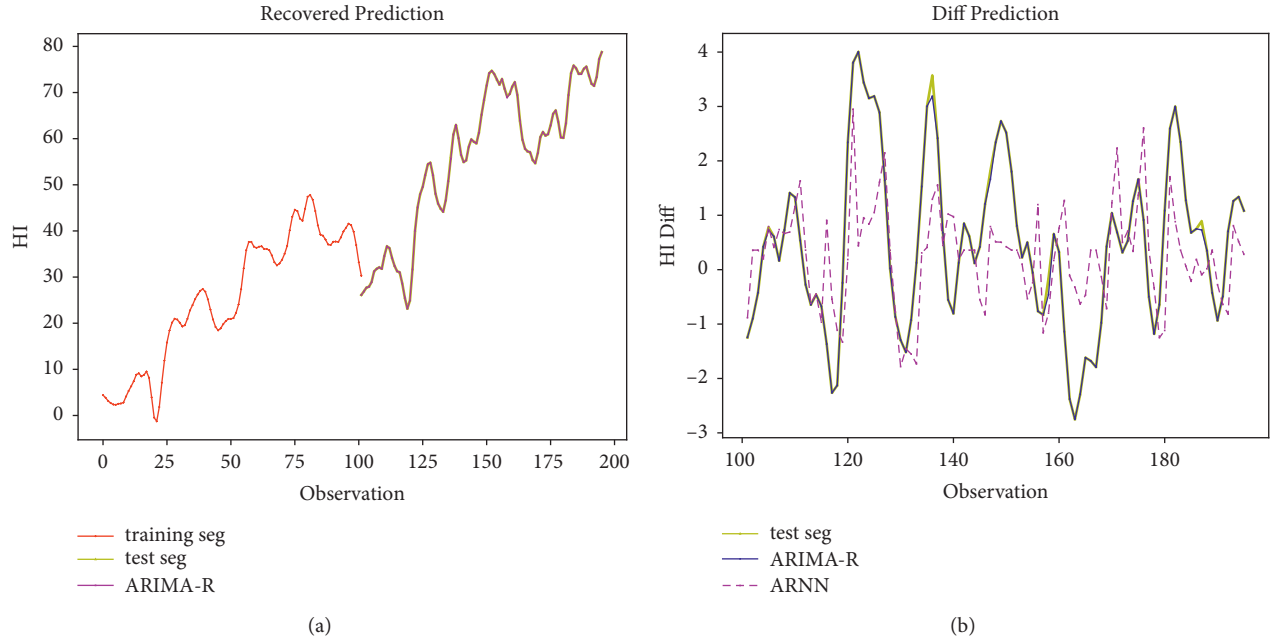

Figure 2: The single-step prediction result of the generated data. (a) The prediction result after the difference is restored. (b) The prediction result of the differenced data.

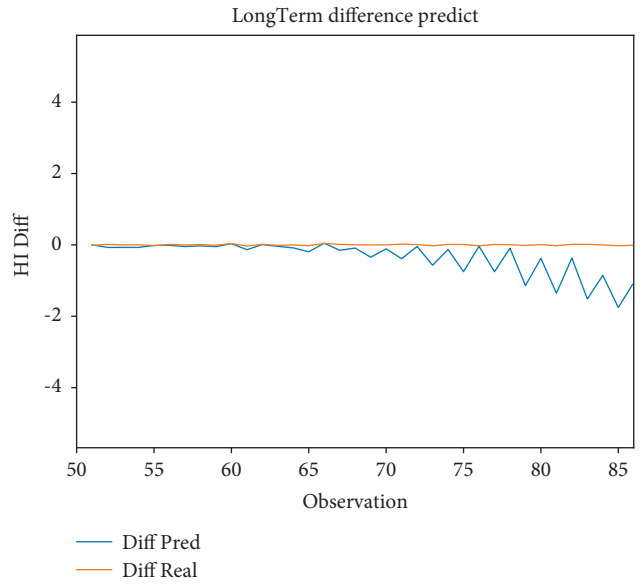

Figure 3: A part of long-term prediction by ARIMA-R.

After determining the order of the ARIMA-R model, the ML algorithm be used to train the model, so the data must be divided first. The ML algorithm inside the model is selected to be Rotation Forest Regressor (RTF), and the parameter settings of the two models are shown in Table 3. In addition, as the reference for RMA, the Lag 1 observation, that is, the last one, is used in calculating RMA for the first estimate of this observation. Since the maximum value of the three parameters d, p, and q is 20, the model can be regarded as a single-step forecast using 20 historical data.

The prediction directly given by RTF or other algorithms is the d-order difference of the observed data. The actual predicted value can be obtained by adding the predicted difference value to the sum of the observed value and difference of this observation.

Taking the data of prediction in bearing 1–4 as an example, the prediction results are shown in Figure 8(b), and the prediction result obtained after accumulating the observation value and the difference value is shown in Figure 8(a). In Figure 8(b), the comparison is the prediction result of ARNN method. Except for the first 50 data where due to sudden

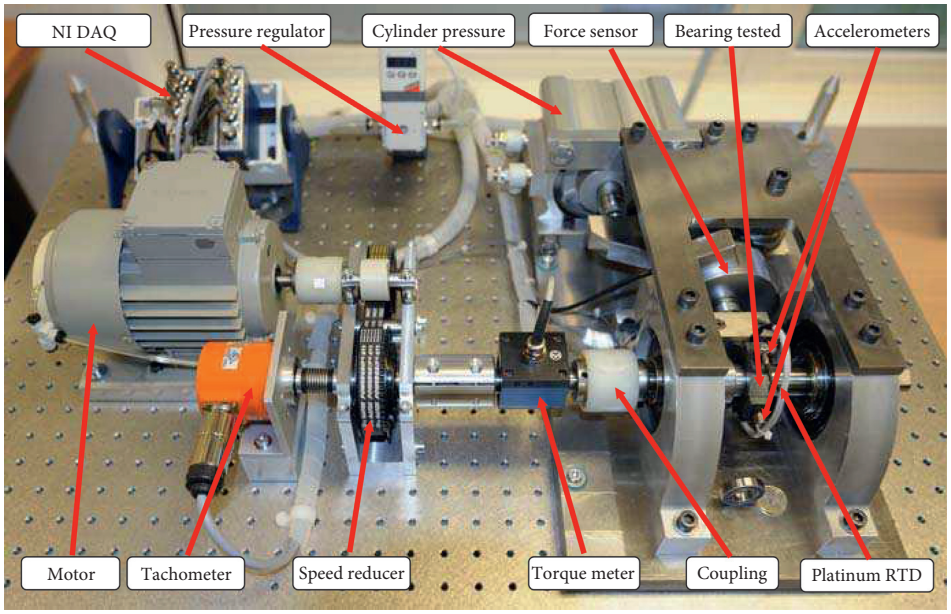

Figure 4: The experimental platform PRONOSTIA. In the accelerated degradation experiment of bearings carried out on the PRONOSTIA platform, the data collected from two mutually orthogonal acceleration sensors are transmitted to the computer through DAQ equipment. In order to accelerate the aging of the bearing, radial force is exerted on the bearing [47].

Table 2: FEMTO datasets.

Operation cond.	Condition1	Condition2	Condition3
Datasets	Bearing1_1	Bearing2_1	
	Bearing1_2	Bearing2_2	
	Bearing1_3	Bearing2_3	Bearing3_1
	Bearing1_4	Bearing2_4	Bearing3_2
	Bearing1_5	Bearing2_5	Bearing3_3
	Bearing1_6	Bearing2_6	
	Bearing1_7	Bearing2_7	

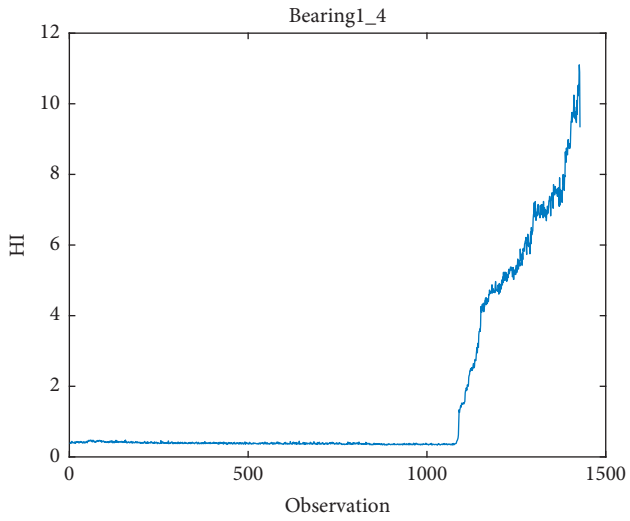

Figure 5: RMS of bearing 1_4.

changes in data and lack of historical data, the predictions of both methods are identically zero, the ARIMA-R method has certain advantages over the ARNN method. Because it is a single-step forecast, it can be seen that the forecast results are in good agreement with the actual data, but if the model is extended to multistep, serious errors will surely occur.

4.3. Experiment Performance and Analysis. The single-step prediction results of the ARIMA-R model for several test samples are shown in Table 4. Comparing the performance of the two methods in the table, it can be seen that the overall performance of the ARIMA-R method is better than that of ARNN, but there is no significant advantage. This is because the data used for testing have obvious noise, which affects the learning of the model. The performance of the ARNN algorithm in different data is unstable. This is because the number of samples is insufficient, resulting in overfitting.

The combination of the ARIMA-R model and various ensemble tree algorithms can be used to train the prediction model, and the comparison of the results is shown in Table 5. The table demonstrates the results replacing RTF in ARIMA-R with different ensemble ML algorithms. Like previous experiment, the result comes from single-step prediction on the bearing dataset, and the result from the ARNN algorithm is provided as a reference.

The algorithms chosen to combine with include Gradient Boosting Tree (GBT), AdaBoost Tree (ABT), and Random Forest (RF), in addition to ANN. The result suggests that the RTF algorithm has some advantages over the others in the framework of ARIMA-R, but is not always better. This is as expected because these ensemble trees have similar learning capabilities, and that is similar to the ability of ANN.

4.4. Long-Term Prediction. A long-term prediction by ARIMA-R is conducted on the FEMTO data, aiming to further examine the performance of the proposed method.

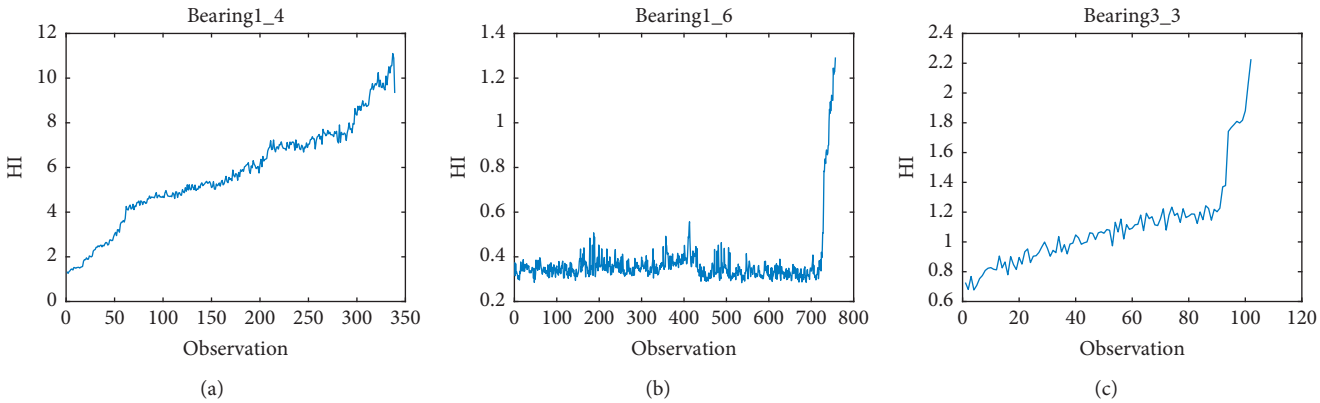

Figure 6: RMS in abnormal state. (a) Bearing 1_4. (b) Bearing 1_6. (c) Bearing 3_3.

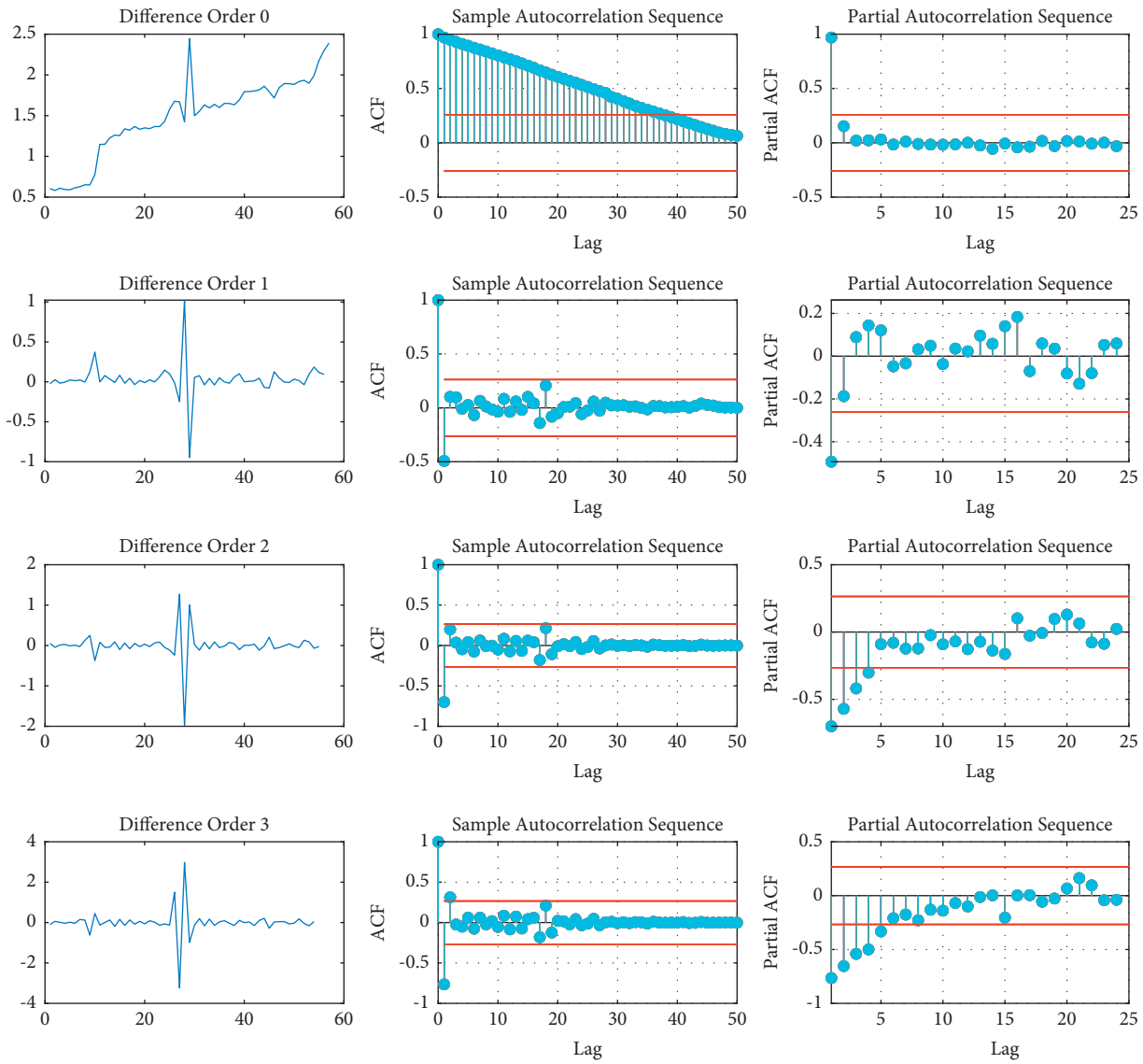

Figure 7: ACF analysis of bearing 1_7. From up to down: the 0,1,2,3 order difference of RMS.

Table 3: Parameters of ARIMA-R and algorithm associated.

ARIMA-R		Random forest and rotation forest		AdaBoost and Gradient Boost		ARNN	
p	20	Estimators	800	Estimators	200	Hidden layer	(10, 30)
q	20			Max depth	10	Activation	ReLU
d	2			Learning rate	0.2	Solver	Adam
r	2					Learning rate	Adaptive

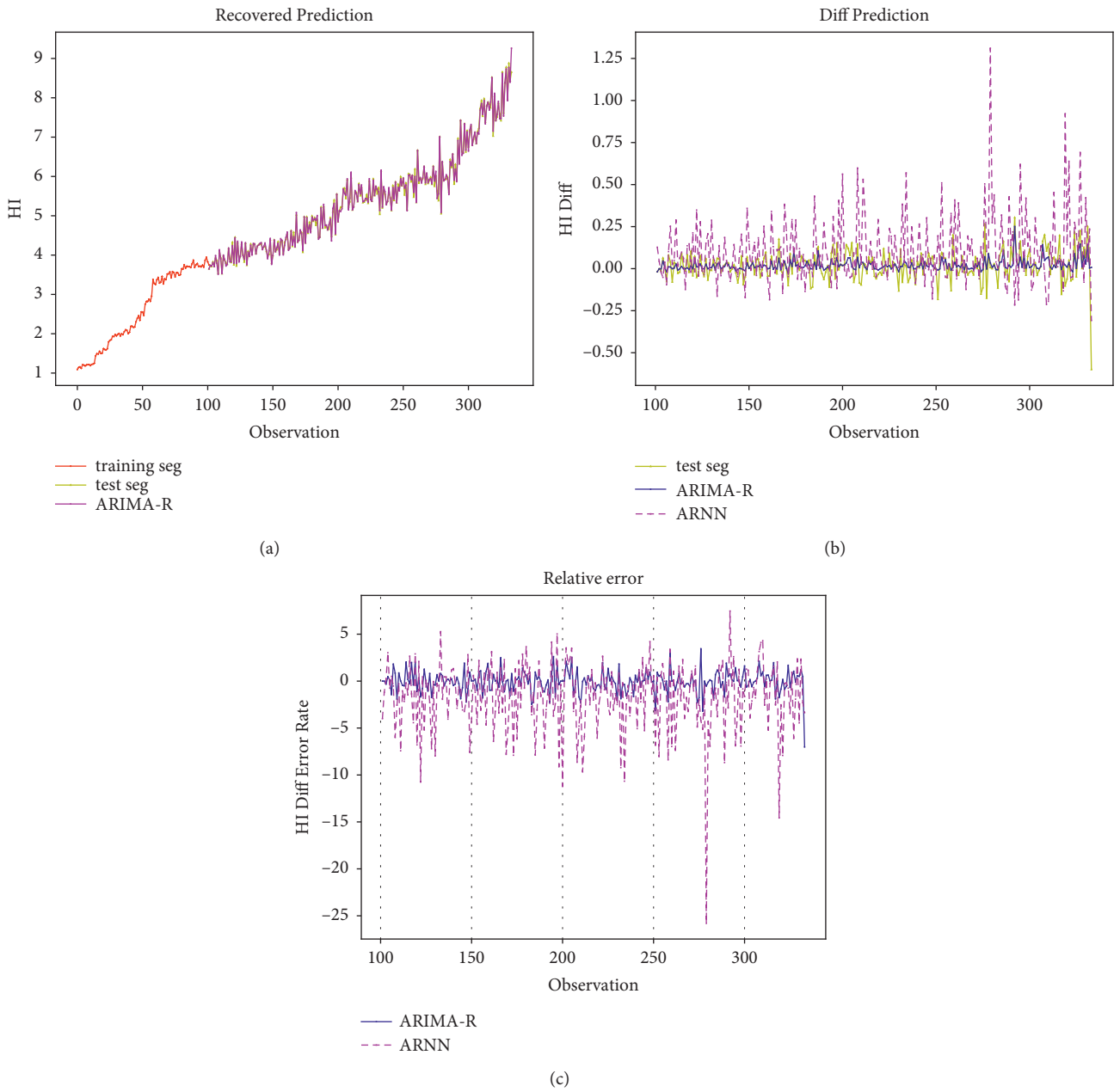

Figure 8: Single-step prediction of ARIMA-R and ARNN. (a) RMS. (b) Difference. (c) Error rate (%) of difference.

A rule similar to IEEE PHM 2012 Data Challenge is adopted. The data from bearing 1_3 to 1_7, 2_3 to 2_7, and 3_3 are taken as test data, in which only the given length of data is available, and the RUL at the end of test data is to be predicted.

As implied before, the proposed method is not fit to predict a long sequence directly. To compensate for this, a 2-order linear regression model based on ordinary least-squares (OLS), provided by scikit-learn [49], is used for initial prediction. The rotation forest regressor then modifies

Table 4: Single-step prediction of ARIMA-R and ARNN.

Algorithm	R	1_6		1_4		3_3	
		MSE	MAE	MSE	MAE	MSE	MAE
ARNN		3.58	53.30	**3.58**	53.30	2.57	45.08
ARIMA	1	1.60	30.15	3.91	55.23	**1.32**	**23.49**
ARIMA	2	**1.54**	**26.04**	5.21	**46.48**	2.57	30.91

Table 5: RMA method comparison.

Data	Algorithm		MSE	MAE	R^2	MAPE
1_4	ARNN	GBT	5.39	55.16	0.968	6.13
		ABT	8.95	72.25	0.990	1.70
		RF	3.91	55.23	0.991	1.57
	ARIMA-R	NN	4.84	64.57	0.993	1.37
		RTF	**3.58**	53.30	0.977	2.45
			5.21	**46.48**	**0.997**	**0.84**
1_6	ARNN	GBT	5.34	55.76	0.960	6.26
		ABT	2.01	30.68	0.977	4.79
		RF	1.60	30.15	0.985	3.63
	ARIMA-R	NN	1.60	31.69	0.986	3.64
		RTF	4.01	47.67	0.976	5.35
			1.54	**26.04**	**0.991**	**2.76**
3_3	ARNN	GBT	15.55	96.15	0.914	5.60
		ABT	4.55	48.25	0.975	2.58
		RF	4.15	43.91	0.977	2.29
	ARIMA-R	NN	4.00	41.64	0.978	2.17
		RTF	10.74	74.34	0.941	3.98
			2.57	**30.91**	**0.986**	**1.63**

Table 6: Prediction performance score comparison on the FEMTO dataset.

Dataset	Predicted RUL(s)	Authentic RUL(s)	Relative error (%)					
			A [51]	B [52]	C [53]	D [54]	E [50]	ARIMA-R
Bearing1_3	3973	5730	43.28	37	−1.04	−0.35	1.05	30.66
Bearing1_4	247	339	67.55	80	−20.94	5.6	20.35	27.14
Bearing1_5	1403	1610	−22.98	9	−278.26	100	11.18	12.85
Bearing1_6	1459	1460	21.23	−5	19.18	28.08	34.93	**0.05**
Bearing1_7	7338	7570	17.83	−2	−7.13	−19.55	29.19	**3.07**
Bearing2_3	4361	7530	37.84	64	10.49	−20.19	57.24	42.08
Bearing2_4	1302	1390	−19.42	10	51.8	8.63	−1.44	6.37
Bearing2_5	333	3090	54.37	−440	28.8	23.3	−0.65	89.23
Bearing2_6	822	1290	−13.95	49	−20.93	58.91	−42.64	36.30
Bearing2_7	349	580	−55.17	−317	44.83	5.17	8.62	39.86
Bearing3_3	589	820	3.66	90	−3.66	40.24	−1.22	28.17
Score	-	-	0.263	0.307	0.355	0.429	**0.569**	0.479

the prediction thereafter. The modification part is similar to the recurrent process in the proposed method. The prediction result is evaluated by the criterion introduced by [47], as depicted in the following equation:

$$\text{score} = \frac{1}{n}\sum_{i=1}^{n} e^{\ln(0.5)C_i\varepsilon_i}, \tag{18}$$

where ϵ_i is the ith relative error of predicted RUL in percentage and C_i is the coefficient that depends on ϵ_i:

$$C_i = \begin{cases} \dfrac{-1}{5}, & \epsilon_i \le 0, \\ \dfrac{1}{20}, & \epsilon_i > 0. \end{cases} \tag{19}$$

The result and criterion score are compared with results from previous research studies [50–54], as shown in Table 6. In the table, predicted RULs, authentic RULs of all test datasets, and relative errors in percentage are presented. The

score (18)of each group of predictions is shown at the bottom of the table.

We can see from the score that the proposed method has an advantage on method A-D, especially for data from bearing1_6 and bearing1_7. That is because, at the end of these datasets, the trend of sequence conforms with the weak stationarity requirement of ARIMA, e.g., the order stationarity. With such stationarity, ARIMA constructed from history observation can reflect the future closer. On the other side, bearing datasets 2_4 and 3_3 illustrate a clear situation transition during the final stage of degradation, and the predictions are far from the true RULs.

In fact, the basic assumption of ARIMA model includes certain types of stationarity on the series. Moreover, sufficient information on the history series is required to obtain a more accurate forecast. In the situation that the monitoring bearings are seriously damaged, neither stationarity nor information is sufficient. Thus, the prediction result is not only because of the predicting ability of the model itself but also due to the nonlinear component introduced by the machine learning method when the rotation forest is used to solve the model parameters.

According to the results, further analysis of the combination between the time series model represented by ARIMA and the machine learning algorithm will be important in the research hereafter.

5. Conclusion

An implementation method of approximating the ARIMA model with ML estimation is proposed in this paper, named ARIMA-R. The advantage of this method is that it does not need to calculate and predict the entire history of the training samples step by step to obtain the MA term of ARIMA model, so it is suitable for general nonrecursive ML methods. This method also adds feature terms to the ARNN model corresponding to moving average terms, improving the ability to describe RUL indicators. At the same time, the idea of using an ensemble regression tree algorithm, rotation forest, instead of ANN to find parameters is proposed to improve the predictive ability when the quantity of data is small. This modification is fit to the data accessibility in bearing HI prediction. The proposed ARIMA-R method combines ARIMA time series prediction model and ensemble regression trees and is used to predict bearing health indicators. The experimental results show that its prediction outperforms the ARNN algorithm. This method constructs the RMA quantity in an iterative way to replace the MA term in the ARIMA algorithm, which is the prediction error term. At the same time, it is proposed to use an ensemble tree instead of ANN in ARNN to improve the fitting ability of small sample training. Through this method, the major benefit is that the ARIMA model can be implemented more completely in the nonrecursive ML algorithm. Therefore, the ARIMA model and similar models such as ARMA, SARIMA, and FARIMA can be used to improve accuracy and efficiency of large-scale data fitting using ML. The further work can be included at the following three aspects. First, more samples can be used to verify the performance of the proposed method. Second, how to combine this method with other ML algorithms and life prediction frameworks is worth exploring. Third, it is meaningful that in-depth research of the relationship between preliminary estimation and the prediction results of the iterative model needs to be studied.

Data Availability

The datasets generated during and/or analysed during the current study are available in the NASA Prognostics Data Repository. URL: https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/#femto.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was funded in part by the Science and Technology Development Fund, Macau SAR (Nos. 0018/2 019/AKP, 0008/2 019/AGJ, FDCT/194/2 017/A3, and SKL-IOTSC-2018-2020), the University of Macau (Grant nos. MYRG2018-00 248-FST and MYRG2019-0137-FST), the Science Foundation of Henan University of Technology (Grant no. 2019BS004), the Cultivation Programme for Young Backbone Teachers in Henan University of Technology (Grant no. 21 420 171), the Natural Science Project of Henan Province (Grant no. 202 102 210 136), and the Key Laboratory of Grain Information Processing and Control, Ministry of Education, Henan University of Technology (Grant no. KFJJ-2016-110).

References

[1] M. Cerrada, R.-V. Sánchez, C. Li et al., "A review on data-driven fault severity assessment in rolling bearings," *Mechanical Systems and Signal Processing*, vol. 99, pp. 169–196, Jan. 2018.

[2] X.-B. Wang, Z.-X. Yang, and X.-A. Yan, "Novel particle swarm optimization-based variational mode decomposition method for the fault diagnosis of complex rotating machinery," *IEEE*, vol. 23, no. 1, pp. 68–79, 2017.

[3] Z.-X. Yang, X. Wang, and P. K. Wong, "Single and simultaneous fault diagnosis with application to a multistage gearbox: a versatile dual-elm network approach," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 12, pp. 5245–5255, 2018.

[4] X.-B. Wang, L. Luo, L. Tang, and Z.-X. Yang, "Automatic representation and detection of fault bearings in in-wheel motors under variable load conditions," *Advanced Engineering Informatics*, vol. 49, Article ID 101321, 2021.

[5] Z.-X. Yang and P.-B. Zhang, ""Elm meets rae-elm: a hybrid intelligent model for multiple fault diagnosis and remaining useful life predication of rotating machinery," in *Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN)*, pp. 2321–2328, IEEE, Vancover, Canada, July 2016.

[6] J. Zhong, Z. Yang, and S. Wong, "Machine condition monitoring and fault diagnosis based on support vector machine," in *Proceedings of the 2010 IEEE International Conference on*

Industrial Engineering and Engineering Management, pp. 2228–2233, IEEE, Xiamen, China, October 2010.

[7] Y. Jiang, C. Li, Z. Yang, Y. Zhao, and X. Wang, "Remaining useful life estimation combining two-step maximal information coefficient and temporal convolutional network with attention mechanism," *IEEE Access*, vol. 9, pp. 16323–16336, 2021.

[8] G. Prakash, X.-X. Yuan, B. Hazra, and D. Mizutani, "Toward a big data-based approach: a review on degradation models for prognosis of critical infrastructure," *Journal of Nondestructive Evaluation, Diagnostics and Prognostics of Engineering Systems*, vol. 4, 2021.

[9] M. R. M. Akramin, M. S. Marizi, M. N. M. Husnain, and M. Shamil Shaari, "Analysis of surface crack using various crack growth models," *Journal of Physics: Conference Series*, vol. 1529, no. 4, Article ID 042074, 2020.

[10] T. E. Tallian and J. I. McCool, "An engineering model of spalling fatigue failure in rolling contact," *Wear*, vol. 17, no. 5-6, pp. 447–461, 1971.

[11] J. Liu, R. Pang, S. Ding, and X. Li, "Vibration analysis of a planetary gear with the flexible ring and planet bearing fault," *Measurement*, vol. 165, Article ID 108100, 2020.

[12] 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," *Mechanical Systems and Signal Processing*, vol. 104, pp. 799–834, 2018.

[13] P. Liang, C. Deng, J. Wu, Z. Yang, J. Zhu, and Z. Zhang, "Single and simultaneous fault diagnosis of gearbox via a semi-supervised and high-accuracy adversarial learning framework," *Knowledge-Based Systems*, vol. 198, Article ID 105895, 2020.

[14] Z.-X. Yang, X.-B. Wang, and J.-H. Zhong, "Representational learning for fault diagnosis of wind turbine equipment: a multi-layered extreme learning machines approach," *Energies*, vol. 9, no. 6, p. 379, 2016.

[15] J.-H. Zhong, P. Wong, and Z.-X. Yang, "Simultaneous-fault diagnosis of gearboxes using probabilistic committee machine," *Sensors*, vol. 16, no. 2, p. 185, 2016.

[16] Z. Yang, P. K. Wong, C. M. Vong, J. Zhong, and J. Liang, "Simultaneous-fault diagnosis of gas turbine generator systems using a pairwise-coupled probabilistic classifier," *Mathematical Problems in Engineering*, vol. 2013, 2013.

[17] Z. Yang, W. I. Hoi, and J. Zhong, "Gearbox fault diagnosis based on artificial neural network and genetic algorithms," in *Proceedings of the 2011 International Conference on System Science and Engineering*, pp. 37–42, IEEE, Noida, India, December 2011.

[18] A. Tealab, "Time series forecasting using artificial neural networks methodologies: a systematic review," *Future Computing and Informatics Journal*, vol. 3, no. 2, pp. 334–340, 2018.

[19] P. Liang, C. Deng, J. Wu, G. Li, Z. Yang, and Y. Wang, "Intelligent fault diagnosis via semisupervised generative adversarial nets and wavelet transform," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 7, pp. 4659–4671, 2019.

[20] P. Liang, C. Deng, J. Wu, Z. Yang, J. Zhu, and Z. Zhang, "Compound fault diagnosis of gearboxes via multi-label convolutional neural network and wavelet transform," *Computers in Industry*, vol. 113, Article ID 103132, 2019.

[21] R. Hyndman and G. Athanasopoulos, *Arima Models," in Forecasting: Principles and Practice*, OTexts, Melbourne, Australia, 2nd edition, 2018.

[22] Y.-S. Lee and L.-I. Tong, "Forecasting time series using a methodology based on autoregressive integrated moving average and genetic programming," *Knowledge-Based Systems*, vol. 24, no. 1, pp. 66–72, Feb. 2011.

[23] Q. Li and S. Liang, "Degradation trend prediction for rotating machinery using long-range dependence and particle filter approach," *Algorithms*, vol. 11, no. 7, p. 89, 2018.

[24] A. Jimenez-Cortadi, F. Boto, I. Irigoien, B. Sierra, and G. Rodriguez, "Time series forecasting in turning processes using ARIMA model," in *Intelligent Distributed Computing XII*, J. Del Ser, E. Osaba, M. N. Bilbao, J. J. Sanchez-Medina, M. Vecchio, and X.-S. Yang, Eds., Springer International Publishing, New York, NY, USA, 2018.

[25] Q. Li, S. Liang, J. Yang, and B. Li, "Long range dependence prognostics for bearing vibration intensity chaotic time series," *Entropy*, vol. 18, no. 1, p. 23, 2016.

[26] G. Qiu, Y. Gu, and J. Chen, "Selective health indicator for bearings ensemble remaining useful life prediction with genetic algorithm and Weibull proportional hazards model," *Measurement*, vol. 150, Article ID 107097, 2020.

[27] J. T. Connor, R. D. Martin, and L. E. Atlas, "Recurrent neural networks and robust time series prediction," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 240–254, 1994.

[28] R. Taherdangkoo, A. Tatomir, M. Taherdangkoo, P. Qiu, and M. Sauter, "Nonlinear autoregressive neural networks to predict hydraulic fracturing fluid leakage into shallow groundwater," *Water*, vol. 12, no. 3, p. 841, 2020.

[29] F. Pereira, F. Bezerra, S. Junior et al., "Nonlinear autoregressive neural network models for prediction of transformer oil-dissolved gas concentrations," *Energies*, vol. 11, no. 7, p. 1691, 2018.

[30] A. Fentis, L. Bahatti, M. Tabaa, and M. Mestari, "Short-term nonlinear autoregressive photovoltaic power forecasting using statistical learning approaches and in-situ observations," *International Journal of Energy and Environmental Engineering*, vol. 10, no. 2, pp. 189–206, 2019.

[31] L. Ruiz, M. Cuéllar, M. Calvo-Flores, and M. Jiménez, "An application of non-linear autoregressive neural networks to predict energy consumption in public buildings," *Energies*, vol. 9, no. 9, p. 684, 2016.

[32] M. R. Cogollo and J. D. Velasquez, "Are neural networks able to forecast nonlinear time series with moving average components?" *IEEE Latin America Transactions*, vol. 13, no. 7, pp. 2292–2300, 2015.

[33] A. N. Burgess and A.-P. N. Refenes, "Modelling non-linear moving average processes using neural networks with error feedback: an application to implied volatility forecasting," *Signal Processing*, vol. 74, no. 1, pp. 89–99, 1999.

[34] W. Waheeb, R. Ghazali, and H. Shah, ""Nonlinear autoregressive moving-average (NARMA) time series forecasting using neural networks," in *Proceedings of the 2019 International Conference on Computer and Information Sciences (ICCIS)*, pp. 1–5, Sakaka, Saudi Arabia, April 2019.

[35] C. Ordóñez, F. Sánchez Lasheras, J. Roca-Pardiñas, and F. J. d. C. Juez, "A hybrid ARIMA-SVM model for the study of the remaining useful life of aircraft engines," *Journal of Computational and Applied Mathematics*, vol. 346, pp. 184–191, 2019.

[36] I. Rojas, O. Valenzuela, F. Rojas et al., "Soft-computing techniques and ARMA model for time series prediction," *Neurocomputing*, vol. 71, no. 4-6, pp. 519–537, 2008.

[37] H. T. Pham, V. T. Tran, and B.-S. Yang, "A hybrid of nonlinear autoregressive model with exogenous input and autoregressive moving average model for long-term machine

state forecasting," *Expert Systems with Applications*, vol. 37, no. 4, pp. 3310–3317, 2010.

[38] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*, Wiley, Hoboken, NJ, USA, 5th edition, 2015.

[39] J. J. Rodriguez, L. I. Kuncheva, and C. J. Alonso, "Rotation forest: a new classifier ensemble method," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 10, pp. 1619–1630, 2006.

[40] C. Pardo, J. F. Diez-Pastor, C. García-Osorio, and J. J. Rodríguez, "Rotation forests for regression," *Applied Mathematics and Computation*, vol. 219, no. 19, pp. 9914–9924, 2013.

[41] A. Tealab, H. Hefny, and A. Badr, "Forecasting of nonlinear time series using ANN," *Future Computing and Informatics Journal*, vol. 2, no. 1, pp. 39–47, 2017.

[42] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via theEMAlgorithm," *Journal of the Royal Statistical Society: Series B*, vol. 39, no. 1, pp. 1–22, 1977.

[43] I. T. Jolliffe and J. Cadima, "Principal component analysis: a review and recent developments," *Philosophical Transactions of the Royal Society A: Mathematical, Physical & Engineering Sciences*, vol. 374, no. 2065, Article ID 20150202, 2016.

[44] J. Franklin, "The elements of statistical learning: data mining, inference and prediction," *The Mathematical Intelligencer*, vol. 27, no. 2, pp. 83–85, 2005.

[45] V. B. Gavirangaswamy, G. Gupta, A. Gupta, and R. Agrawal, "Assessment of arima-based prediction techniques for road-traffic volume," in *Proceedings of the fifth international conference on management of emergent digital EcoSystems*, pp. 246–251, Neumünster Abbey, Luxembourg, October 2013.

[46] A. W. Bojanczyk, R. P. Brent, F. R. De Hoog, and D. R. Sweet, "On the stability of the bareiss and related toeplitz factorization algorithms," *SIAM Journal on Matrix Analysis and Applications*, vol. 16, no. 1, pp. 40–57, 1995.

[47] P. Nectoux, R. Gouriveau, K. Medjaher et al., ""PRONOSTIA: an experimental platform for bearings accelerated degradation tests," in *Proceedings of the IEEE International Conference on Prognostics and Health Management, PHM'12.*, pp. 1–8, IEEE Catalog Number: CPF12PHM-CDR, Denver, CO, USA, 2012.

[48] G. Aydemir and B. Acar, "Anomaly monitoring improves remaining useful life estimation of industrial machinery," *Journal of Manufacturing Systems*, vol. 56, pp. 463–469, 2020.

[49] F. Pedregosa, G. Varoquaux, A. Gramfort et al., "Scikit-learn: machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[50] Y. Yoo and J.-G. Baek, "A novel image feature for the remaining useful lifetime prediction of bearings based on continuous wavelet transform and convolutional neural network," *Applied Sciences*, vol. 8, no. 7, p. 1102, 2018.

[51] L. Guo, N. Li, F. Jia, Y. Lei, and J. Lin, "A recurrent neural network based health indicator for remaining useful life prediction of bearings," *Neurocomputing*, vol. 240, pp. 98–109, 2017.

[52] E. Sutrisno, H. Oh, A. S. S. Vasan, and M. Pecht, "Estimation of remaining useful life of ball bearings using data driven methodologies," in *Proceedings of the 2012 ieee conference on prognostics and health management*, pp. 1–7, IEEE, Hyatt Regency Minneapoli, September 2012.

[53] S. Hong, Z. Zhou, E. Zio, and K. Hong, "Condition assessment for the performance degradation of bearing based on a combinatorial feature extraction method," *Digital Signal Processing*, vol. 27, pp. 159–166, 2014.

[54] Y. Lei, N. Li, S. Gontarz, J. Lin, S. Radkowski, and J. Dybala, "A model-based method for remaining useful life prediction of machinery," *IEEE Transactions on Reliability*, vol. 65, no. 3, pp. 1314–1326, 2016.