



University of HUDDERSFIELD

University of Huddersfield Repository

Tran, Van Tung, Yang, Bo-Suk and Tan, Andy Chit Chiow

Multi-step ahead direct prediction for machine condition prognosis using regression trees and neuro-fuzzy systems

Original Citation

Tran, Van Tung, Yang, Bo-Suk and Tan, Andy Chit Chiow (2009) Multi-step ahead direct prediction for machine condition prognosis using regression trees and neuro-fuzzy systems. *Expert Systems With Applications*, 36 (5). pp. 9378-9387. ISSN 0957-4174

This version is available at <http://eprints.hud.ac.uk/id/eprint/16577/>

The University Repository is a digital collection of the research output of the University, available on Open Access. Copyright and Moral Rights for the items on this site are retained by the individual author and/or other copyright owners. Users may access full items free of charge; copies of full text items generally can be reproduced, displayed or performed and given to third parties in any format or medium for personal research or study, educational or not-for-profit purposes without prior permission or charge, provided:

- The authors, title and full bibliographic details is credited in any copy;
- A hyperlink and/or URL is included for the original metadata page; and
- The content is not changed in any way.

For more information, including our policy and submission procedure, please contact the Repository Team at: E.mailbox@hud.ac.uk.

<http://eprints.hud.ac.uk/>

Multi-step ahead direct prediction for machine condition prognosis using regression trees and neuro-fuzzy systems

Van Tung Tran^a, Bo-Suk Yang^{a,*}, Andy Chit Chiow Tan^b

*School of Mechanical Engineering, Pukyong National University,
San 100, Yongdang-dong, Namgu, Busan 608-739, South Korea*

^b*School of Mechanical, Manufacturing and Medical Engineering,
Queensland University of Technology, G.P.O. Box 2343, Brisbane, Qld. 4001, Australia*

Abstract

This paper presents an approach to predict the operating conditions of machine based on classification and regression trees (CART) and adaptive neuro-fuzzy inference system (ANFIS) in association with direct prediction strategy for multi-step ahead prediction of time series techniques. In this study, the number of available observations and the number of predicted steps are initially determined by using false nearest neighbor method and auto mutual information technique, respectively. These values are subsequently utilized as inputs for prediction models to forecast the future values of the machines' operating conditions. The performance of the proposed approach is then evaluated by using real trending data of low methane compressor. A comparative study of the predicted results obtained from CART and ANFIS models is also carried out to appraise the prediction capability of these models. The results show that the ANFIS prediction model can track the change in machine conditions and has the potential for using as a tool to machine fault prognosis.

Keywords: Machine fault prognosis; Long-term time series prediction; ANFIS; CART; Direct prediction methodology.

1. Introduction

The fault progression process of mechanical systems usually consists of a series of degraded states mainly due to component wear and fatigue during the operation process. In order to sustain the operating life of these components, selective maintenance strategies are regularly performed. Maintenance strategy is traditionally carried out by either corrective maintenance or preventive maintenance. The detailed introduction and the ability to apply these strategies could

be found in [1]. Condition based maintenance (CBM) is the second type of preventive maintenance strategy of which structure consists of the modules as follows: sensing and data acquisition, signal processing, condition monitoring, fault diagnosis and health assessment, prognosis, decision support, and presentation [2]. It is based on the actual condition and can assess whether the equipment is in need of maintenance or not; and if it is necessary, determine when the maintenance actions need to be executed. Moreover, with the assistance of prognosis, an alarm level can be set when the predicted values and the actual fault symptom of failure fall within the warning region. This will provide the adequate time for the system operators to take remedial actions, inspect the condition of the equipment, and conduct a repair on the defect before the catastrophic failure occurs. Therefore, an effective and efficient machine condition prognosis is essential for effective maintenance strategy and has become the key component of CBM system [3]. Nevertheless, prognosis has been a difficult task of CBM and can be broadly classified into three categories: experience-based, model-based, and data-driven based.

Experience-based prognostic approaches require the component failure history data or operational usage profile data. They involve in collecting statistical information from a large number of component samples to indicate the survival duration of a component before a failure occurs and use these statistical parameters to predict the remaining useful life (RUL) of individual components. Generally, they are the least complex forms of prognostic techniques and their accuracy is not high because they base solely on the analysis of the past experience.

Model-based prognostic approaches are applicable to where the accurate mathematical models can be constructed based on the physical fundamentals of a system. These approaches use residuals as features, which are the outcomes of consistency checks between the sensed measurements of system and the outputs of a mathematical model [4]. Some of the published researches using these approaches can be found in references [3-8]. However, these techniques are merely applied for some specific components and each requires a different mathematical model. Changes in structural dynamics and operating conditions can affect the mathematical model as it is impossible to model all real-life conditions. Furthermore, a suitable model is difficult to establish to mimic the real life.

The data-driven prognostic approaches are also known as data mining or machine learning techniques. They utilize and require large amount of historical failure data to build a prognostic model that learns the system behavior. Among these techniques, artificial intelligence is regularly used because of its flexibility in generating appropriate model. Schwabacher and Goebel [9] have demonstrated the capability of using artificial intelligent techniques for prognosis. Other outstanding data-driven approaches have been proposed by Vachsevanos and Wang [10]; and Huang et al. [11]. In [10], they used dynamic wavelet neural network to predict

the failure growth based on the vibration signals to estimate the RUL of bearings. Huang et al. [11] applied self-organizing map and back propagation neural networks methods using vibration signals to predict the RUL of ball bearing. Wang et al. [12] utilized and compared the results of two predictors, namely, recurrent neural networks and ANFIS, to forecast the damage propagation trend of rotating machinery. A hybrid approach of fuzzy logic and neural networks was employed in [13] to predict the RUL of bearings of small and medium size induction motors. An approach based on data-driven prognostic technique applied in aircraft systems can be referred in [14].

In our previous work, we have proposed a data-driven prognosis approach that used regression trees as prediction model and one-step ahead (OS) prediction methodology for forecasting the machines' operating conditions [15]. In this paper, a multi-step ahead (MS) prediction methodology is proposed for the same purpose. MS prediction plays a crucial role in industry by providing information on the RUL of machine ranging from system identification to ecology. It is divided into three strategies involved recursive prediction, direct prediction which is dealt with this study, and DirRec prediction [16]. The detail of these strategies will be presented and compared further in the next section.

Other problems to be dealt with MS methodology will also be addressed in this paper as follows: the number of initial observations (embedding dimension) should be used as the inputs for prediction model; the number of steps (time delay) can be predicted by the prediction model to obtain the optimum performance. The former problem can be solved by using the false nearest neighbor method (FNN) [17] or the Cao's method [18] in which FNN is commonly used. The latter can be calculated by using published methods such as auto-correlation [19], average displacement [20], and auto mutual information (AMI) [21]. In this study, AMI is chosen to estimate the time delay. After determining the embedding dimension and time delay, the prediction model is subsequently established. CART [22] and ANFIS [23] are utilized as the prediction model for the purposes of comparing the forecasting ability for MS prediction in the machine condition.

2. Background knowledge

2.1. Multi-step ahead strategies

Multi-step ahead is a difficult task as well as challenging task for time series prediction problem due to the growing uncertainties which arise from various sources, such as the accumulated errors and the lack of information [24]. As mention in previous section, there are three strategies that could be frequently used for creating prediction model. The variance of

these strategies is introduced and compared as follows

2.1.1. Recursive prediction strategy

In order to predict h future values, the prediction model uses the predicted value of the previous step as a known value to forecast iteratively the next value until h future values are obtained. Given the observations $y_t = [x_{t-d+1}, x_{t-d+2}, \dots, x_t]$, the first future value can be predicted by using OS prediction:

$$\hat{y}_{t+1} = f(y_t) = f(x_{t-d+1}, x_{t-d+2}, \dots, x_t) \quad (1)$$

where d denotes the number of inputs or the embedding dimension. For predicting the next value, the same prediction model is used:

$$\hat{y}_{t+2} = f(x_{t-d+2}, x_{t-d+3}, \dots, x_t, \hat{y}_{t+1}) \quad (2)$$

The predicted value \hat{y}_{t+1} is used as a known value in equation (2) for current forecasting time. The similar process is executed iteratively for $h-2$ remaining values. However, the accumulated error in previous predicting process will be added in the following step. Hence, this reduces the performance of prediction accuracy of recursive strategy.

2.1.2. DirRec prediction strategy

The predicting process of this strategy is similar with the above strategy. Nevertheless, the difference is that a new model is generated for each iteration when the predicted value is obtained. For instance, the value \hat{y}_{t+2} is predicted by the new prediction model which is retrained with temporary training set included the initial training set and \hat{y}_{t+1} . This strategy also has the same drawback of recursive strategy even though the new model is created after each step of the predicting process.

2.1.3. Direct prediction strategy

The direct prediction strategy can forecast the sequence of h future values $\hat{y}_{t+h} = [x_{t+1}, x_{t+2}, \dots, x_{t+h}]$ from a given observations $y_t = [x_{t-d+1}, x_{t-d+2}, \dots, x_t]$ by using H different prediction models. In order to generate these models, the training set D is initially created from the time series by using a sliding window of length $d+h$. Let X and Y be the input and output vectors, respectively. The input X corresponds to the first d values of window whilst the output Y is the remaining h values of window. The training set D comprised X and Y vectors is structured in the form shown in [Table 1](#). Therefore, by learning each training set independently, H prediction models are sequentially generated with the same input X but different output Y_i which include all the values in the i th column Y in D . Obviously, the direct

prediction strategy provides a higher accuracy due to the avoidance of the accumulated errors and is therefore used in this paper for MS prognosis. MS prediction with direct prediction strategy has been applied in many fields, such as time series prediction and the state of river flow [24-26].

Table 1 Training data D for direct prediction strategy

2.2. Time delay estimation

There are several methods published in literatures could be used to choose the time delay. However, most of them are based on empirical concepts and is not easy to identify which of the method is the best for a particular task. In this paper, time delay is dealt with AMI method which is mutual information between two measurements taken from a single time series. AMI estimates the degree to which the time series $x(t+\tau)$ on average can be predicted from $x(t)$, i.e. the mean predictability of future values in the time series from the past values [27].

The AMI between $x(t)$ and $x(t+\tau)$ is:

$$I_{XX_\tau} = \sum_{x(t), x(t+\tau)} P_{XX_\tau}(x(t), x(t+\tau)) \ln \left(\frac{P_{XX_\tau}(x(t), x(t+\tau))}{P_X(x(t))P_{X_\tau}(x(t+\tau))} \right) \quad (3)$$

where $P_X(x(t))$ is the normalized histogram of the distribution of values observed for $x(t)$ and $P_{XX_\tau}(x(t), x(t+\tau))$ is the joint probability density for the measurements of $x(t)$ and $x(t+\tau)$.

The rate of decrease of the AMI with increasing time delay is a normalized measure of the complexity of the time series. The first local minimum of the AMI of time series has been used to determine the optimal time delay that makes the coordinates for an embedding procedure less pair-wise dependent in a well controlled manner.

2.3. Determining the embedding dimension

Assuming a time-series of x_1, x_2, \dots, x_N . The time delay vector $y_i(d)$ constructed from this time series with time delay τ and embedding dimension d is defined as follows:

$$y_i(d) = [x_i, x_{i+\tau}, \dots, x_{i+(d-1)\tau}], \quad i = 1, 2, \dots, N - (d-1)\tau \quad (4)$$

The FNN method is based on the concept that in the passage from dimension d to dimension $d+1$, where one can differentiate between points which are ‘true’ or ‘false’ neighbor on the orbit. The criteria for identification of false nearest neighbors can be explained as follows: denote $y_i^r(d)$ as the nearest neighbor of $y_i(d)$ in a d dimensional embedding space. The

nearest neighbor is determined by finding the vector which minimizes the Euclidean distance:

$$R_d = \|y_i(d) - y_i^r(d)\| \quad (5)$$

Considering each of these vectors under a $d+1$ dimensional embedding:

$$y_i(d+1) = [x_i, x_{i+\tau}, x_{i+2\tau}, \dots, x_{i+d\tau}], \quad i = 1, 2, \dots, N - d\tau \quad (6)$$

$$y_i^r(d+1) = [x_i^r, x_{i+\tau}^r, x_{i+2\tau}^r, \dots, x_{i+d\tau}^r], \quad i = 1, 2, \dots, N - d\tau \quad (7)$$

The vectors are separated by the Euclidean distance:

$$R_{d+1} = \|y_i(d+1) - y_i^r(d+1)\| \quad (8)$$

The first criterion of FNN which identifies a false nearest neighbor is:

$$\sqrt{\frac{R_{d+1}^2 - R_d^2}{R_d^2}} = \frac{|x_{i+d\tau} - x_{i+d\tau}^r|}{R_d} > R_{tol} \quad (9)$$

where R_{tol} is a tolerance level.

The second criterion is:

$$\frac{R_{d+1}}{R_A} > A_{tol} \quad (10)$$

where R_A is a measure of the size of the attractor and A_{tol} is a threshold that can be chosen in practice. If both equations (9) and (10) are satisfied, then $y_i^r(d)$ is a false nearest neighbor of $y_i(d)$. Once the total number of FNN is calculated, the percentage of FNN is measured. An appropriate embedding dimension is the value where the percentage of FNN falls to zero.

2.4. Prediction models

2.4.1. Regression trees

CART method has been extensively developed for classification or regression purpose depending on the response variable which is either categorical or numerical. In this study, CART is utilized to build a regression tree model. Beginning with an entire data set, a binary tree is constructed with the repeated splits of the subsets into two descendant subsets according to independent variables. The goal is to produce subsets of the data which are as homogeneous as possible with respect to the response variables. Regression tree in CART is built by using the following two processes: tree growing and tree pruning.

Tree growing: Let L be a learning data which comprises n couples of observations $(y_1, \mathbf{x}_1), \dots, (y_n, \mathbf{x}_n)$, where $\mathbf{x}_i = (x_{i_1}, \dots, x_{i_d})$ is a set of independent variables and

$y_i \in R$ is a response associated with x_i . In order to build the tree, learning data L is recursively partitioned into two subsets by binary split until the terminal nodes are achieved. The result is to move the couples (y, x) to left or right nodes containing more homogeneous responses. The predicted response at each terminal node t is estimated by the mean $\bar{y}(t)$ of the $n(t)$ response variables y contained in that terminal node.

The split selection at any internal node t is chosen according to the node impurity that is measured by within-node sum of squares:

$$R(t) = \frac{1}{n} \sum_{y_i, x_i \in t} (y_i - \bar{y}(t))^2 \quad (11)$$

and,

$$\bar{y}(t) = \frac{1}{n(t)} \sum_{y_i, x_i \in t} y_i \quad (12)$$

When a split is performed, two subsets of observations t_L and t_R are obtained. The optimum split s^* at node t is obtained from the set of all splitting candidates S in order that it verifies:

$$\begin{aligned} \Delta R(s^*, t) &= \max_{s \in S} \Delta R(s, t), \\ \Delta R(s, t) &= R(t) - R(t_L) - R(t_R) \end{aligned} \quad (13)$$

where $R(t_L)$ and $R(t_R)$ are sum of squares of the left and right subsets, respectively.

Tree pruning: The tree gained in tree growing process has many terminal nodes that increase precision of the responses. However, this is frequently too complicated and over-fitting is highly probable. Consequently, it should be pruned back.

Tree pruning process is performed by the following procedure:

Step 1: At every internal node, an error-complexity is found for the number of descendant subtrees. The error-complexity is defined as:

$$R_\alpha(T) = R(T) + \alpha |\tilde{T}| \quad (14)$$

where $R(T) = \frac{1}{n} \sum_{t \in \tilde{T}} \sum_{(y_i, x_i) \in t} (y_i - \bar{y}(t))^2$ is the total within-node sum of squares, \tilde{T} is the set

of current nodes of T and $|\tilde{T}|$ is the number of terminal nodes in T , $\alpha \geq 0$ is the complexity parameter which weights the number of terminal nodes.

Step 2: Using the error-complexity attained in step 1, the internal node with the smallest error is replaced by terminal node.

Step 3: The algorithm terminates if all the internal nodes have converge to a terminal node. Otherwise, it returns to step 1.

Cross-validation for selecting the best tree: There are two possible methods to select the best tree. One is through the use of independent test data and the other is cross-validation that is used in this study.

The learning data L is randomly divided into v approximately equal group, and $(v-1)$ groups are then utilized as the learning data for growing the tree model. The remaining group is employed as testing data for error estimation of tree model. As a result, v errors are obtained by v iterations with variation of the combinations of the learning data and testing data. The mean and standard deviation of the errors are given:

$$R^{CV}(d) = \frac{1}{v} \sum_{i=1}^v R^{ts}(d_i) \quad (15)$$

$$\sigma(R^{CV}(d)) = \sqrt{\frac{1}{v} \sum_{i=1}^v (R^{ts}(d_i) - R^{CV}(d))^2}$$

Here $R^{CV}(\cdot)$ is the average relative error, d is the cross-validation tree, σ is the standard error, and $R^{ts}(\cdot)$ is the testing data error. The best tree T_t selection is adopted:

$$R(T_t) = R^{CV}(T_{\min}) + \sigma(R^{CV}(T_{\min})) \quad (16)$$

where $R(\cdot)$ is the cross-validation error and T_{\min} is the tree with the smallest cross-validation error.

2.4.2. Adaptive Neuro-fuzzy inference system (ANFIS)

Architecture of ANFIS: The ANFIS is a fuzzy Sugeno model put in the framework of adaptive systems to facilitate learning and adaptation [23]. Such framework makes the ANFIS modeling more systematic and less dependent on expert knowledge. In order to present ANFIS architecture, two fuzzy if-then rules based on a first-order Sugeno model are considered:

Rule 1: If (x is A_1) and (y is B_1) then $f_1 = p_1x + q_1y + r_1$.

Rule 2: If (x is A_2) and (y is B_2) then $f_2 = p_2x + q_2y + r_2$.

where x and y are the inputs, f_i are the outputs within the fuzzy region specified by the fuzzy rule, A_i and B_i are the fuzzy sets, $\{p_i, q_i, r_i\}$ is a set of design parameters that are determined during the learning process. The ANFIS architecture to implement these rules consists of five layers as shown in Fig. 1. In this architecture, circles indicate fixed nodes and squares indicate adaptive nodes. Nodes within the same layer perform identical functions as detailed below.

Fig. 1 Schematic of ANFIS architecture

Layer 1: all the nodes are adaptive nodes. The outputs of this layer are the fuzzy membership grade of the inputs, which are given by:

$$\begin{aligned}
O_i^1 &= \mu_{A_i}(x), \quad i=1,2 \\
O_i^1 &= \mu_{B_{i-2}}(y), \quad i=3,4
\end{aligned} \tag{17}$$

Theoretically, $\mu_{A_i}(x)$ and $\mu_{B_i}(y)$ can adopt any fuzzy membership function. For example, if the bell functions are chosen then:

$$\mu_{A_i}(x) = \frac{1}{1 + \left[\left(\frac{x - c_i}{a_i} \right)^2 \right]^{b_i}}, \quad i=1,2 \tag{18}$$

where $\{a_i, b_i, c_i\}$ are the modifiable parameters governing the shape of the membership functions. Parameters in this layer are referred to as premise parameters.

Layer 2: The nodes are fixed nodes denoted as Π , indicating that they perform as a simple multiplier. Each node in this layer calculates the firing strengths of each rule via multiplying the incoming signals and sends the product out. The outputs of this layer can be represented as:

$$O_i^2 = w_i = \mu_{A_i}(x)\mu_{B_i}(y), \quad i=1,2 \tag{19}$$

Layer 3: The nodes are also fixed nodes. They are labeled with N , indicating that they play a normalization role to the firing strengths from the previous layer. The i th node of this layer calculates the ratio of the i th rule's firing strength to the sum of all rules' firing strengths:

$$O_i^3 = \bar{w}_i = \frac{w_i}{\sum w_i} = \frac{w_i}{w_1 + w_2}, \quad i=1,2 \tag{20}$$

Layer 4: The nodes are adaptive nodes. The output of each node in this layer is simply the product of the normalized firing strength and a first order polynomial. Thus, the outputs of this layer are given by:

$$O_i^4 = \bar{w}_i f_i = \bar{w}_i (p_i x + q_i y + r_i) \tag{21}$$

where \bar{w}_i is the output of layer 3, and $\{p_i, q_i, r_i\}$ are consequent parameters.

Layer 5: There is only a single fixed node labeled with Σ . This node performs the summation of all incoming signals. Hence, the overall output of the model is given by:

$$O_i^5 = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i}, \quad i=1,2 \tag{22}$$

Learning algorithm of ANFIS: The task of the learning algorithm for ANFIS architecture is to tune all the modifiable parameters, namely premise parameters $\{a_i, b_i, c_i\}$ and consequent parameters $\{p_i, q_i, r_i\}$, to make the ANFIS output match the training data [28]. From the ANFIS

architecture, it can be observed that when the values of premise parameters are fixed, the output of network can be expressed as a linear combination of the consequent parameters:

$$\begin{aligned} f &= \bar{w}_1 f_1 + \bar{w}_2 f_2 \\ &= (\bar{w}_1 x) p_1 + (\bar{w}_1 y) q_1 + (\bar{w}_1) r_1 + (\bar{w}_2 x) p_2 + (\bar{w}_2 y) q_2 + (\bar{w}_2) r_2 \end{aligned} \quad (23)$$

The least squares method can be easily used to identify the optimal values of these parameters. When the premise parameters are not fixed, the search space becomes larger and convergence of the training becomes slower. A hybrid algorithm combining the least squares method and the gradient descent method is adopted to solve the problem. The hybrid algorithm is composed of a forward pass and a backward pass. In the forward pass, the least squares method is used to optimize the consequent parameters with the fixed premise parameters. Once the optimal consequent parameters are found, the backward pass commences immediately. In the back pass, the gradient descent method is used to adjust the premise parameters corresponding to the fuzzy sets in the input domain, whilst the consequent parameters remain fixed. This procedure is repeated until either the squared error is less than a specified value or the maximum number of training epoch is encountered.

3. Proposed system

The proposed system for prognosis of machine condition comprises four procedures sequentially as shown in Fig. 2: data acquisition, data splitting, training-validating model, and predicting. The role of each procedure is explained as follows:

Step 1 Data acquisition: this procedure is used to obtain the vibration data from machine condition. It covers a range of data from normal operation to obvious faults of the machine.

Step 2 Data splitting: the trending data attained from previous procedure is split into two parts: training set and testing set. Different data is used for different purposes in the prognosis system. Training set is used for creating the prediction models whilst testing set is utilized to test the trained models.

Step 3 Training-validating: this procedure includes the following sub-procedures: estimating the time delay and determining the embedding dimension based on AMI and FNN method, respectively; creating the prediction models and validating those models. Validating the prediction models are used for measuring their performance capability.

Step 4 Predicting: multi-step ahead or long-term direct prediction method is used to forecast the future values of machine condition. The predicted results are measured by the error between predicted values and actual values in the testing set. Models and updated data are also carried out in this procedure for the next prediction time.

Fig. 2 Proposed system for machine fault prognosis.

4. Experiment

The proposed method is applied to a real system to predict the trending data of a low methane compressor which is an important equipment in petrochemical plant. This compressor shown in Fig. 3 is driven by a 440 kW motor, 6600 volt, 2 poles and operating at a speed of 3565 rpm. Other information of the system is summarized in Table 2.

Fig. 3 Low methane compressor: wet screw type.

Table 2 Information of the system

The condition monitoring system of this compressor consists of two types: off-line and on-line. In the off-line system, the vibration sensors are installed along axial, vertical, and horizontal directions at the locations of drive-end motor, non drive-end motor, male rotor compressor and suction part of compressor. In the on-line system, acceleration sensors are located at the same places as in the off-line system but only in the horizontal direction.

The trending data was recorded from August 2005 to November 2005 which included peak acceleration and envelope acceleration data. The average recording duration was 6 hours during the data acquisition process. This data consists of approximately 1200 data points as shown in Figs. 4 and 5, and contains information of machine history with respect to time sequence (vibration amplitude). Consequently, it can be classified as time-series data. The proposed method is employed to predict the future condition of vibration amplitude based on the past and current states.

Fig. 4 The entire peak acceleration data of low methane compressor.

Fig. 5 The entire envelope acceleration data of low methane compressor.

The machine is in normal condition during the first 300 points of the time sequence. After that time, the condition of the machine suddenly changes. This indicates that there are some faults occurring in the machine. These faults were identified as the damages of main bearings of the compressor (notation Thrust: 7321 BDB) due to insufficient lubrication. Consequently, the surfaces of these bearings were overheated and delaminated [15].

5. Results and discussions

With the aim of forecasting the change of machine condition, the first 300 points were used to train the system. Before being used to generate the prediction models, the time delay τ is

initially calculated according to the method mentioned in section 2.2. Theoretically, the optimal time delay is the value at which the AMI obtains the first local minimum. From Fig. 6, the optimal time delay of peak acceleration training data is found as 7. Similarly, 5 is the optimal time delay value of envelope acceleration training data.

Fig. 6 Time delay estimation. (a) Peak acceleration, (b) Envelope acceleration.

Using FNN method described in section 2.3, the optimal time delay τ is subsequently utilized to determine the embedding dimension d . It is noted that the tolerance level R_{tol} and threshold A_{tol} must be initially chosen. In this study, $R_{tol} = 15$ and $A_{tol} = 2$ are used according to [17]. The relationship between the false nearest neighbor percentage and the embedding dimension for both peak acceleration data and envelope data is shown in Fig. 7. From the figure, the embedding dimension d is chosen as 4 for both data sets where the false nearest neighbor percentage reaches to 0.

Fig. 7 The relationship between FNN percentage and embedding dimension. (a) Peak acceleration, (b) Envelope acceleration.

After calculating the time delay and embedding dimension, the process of generating the prediction models is carried out. Based on the time delay and embedding dimension values, the training data is created as mentioned in section 2.1.3. Using this training data, the CART model and the ANFIS model are established. In case of the CART model, the number of response values for each terminal node in tree growing process is 5 and 10 cross-validations are decided for selecting the best tree in tree pruning. Furthermore, in order to evaluate the predicting performance, the root-mean square error (RMSE) is utilized as following

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{N}} \quad (24)$$

where N , y_i , \hat{y}_i represent the total number of data points, the actual value, and predicted value of prediction model in the training data or testing data, respectively.

Figs. 8(a) and 9(a) perform the training and validating results of the CART models for peak acceleration and envelope acceleration data, respectively. The actual values and predicted values are almost identical with very small RMSE values ranging from 0.002217 to 1.3314×10^{-5} . It indicates that the learning capability of CART model is extremely good. Similarly, the ANFIS models are also created for both training set of peak acceleration and envelope acceleration. There are four inputs for each ANFIS model due to the embedding dimension value. For each

input, a bell shape is chosen for each membership function (MF) and the number of MFs is 2. It means that the region value of each input is divided into two, namely, small and large. In order to evaluate the learning process, the convergence of RMSE is utilized. If the decreasing rate of the RMSE as well as the performance is not significant, the learning process can be terminated. In this study, after executing 100 epochs, all RMSEs of the outputs reach the convergent stage for both the peak acceleration data and envelope acceleration data as shown in Fig. 10. Alternatively, the parameters of MFs, which are premise parameters and consequent parameters, are automatically adjusted through the learning in order that the outputs of ANFIS model match the actual values in training data. The changes of MF shapes are depicted in Fig. 11. The training and validating results of ANFIS models for both the peak acceleration data and envelope acceleration data are respectively shown in Figs. 8(b) and 9(b). From these figures, the RMSE values are sequentially 0.00876 and 0.08886. These values are higher than those of the CART models. The reason could be that the number of MFs is improperly chosen. For higher accuracy of RMSEs, the MFs can be increased. Nevertheless, this will also increase the computational complexity and take too much training time.

Fig. 8 Training and validating results of peak acceleration data. (a) CART, (b) ANFIS

Fig. 9 Training and validating results of envelope acceleration data. (a) CART, (b) ANFIS.

Fig. 10 RMSE convergent curve. (a) Peak acceleration, (b) Envelope acceleration.

Fig. 11 The changes of MFs after learning. (a) Peak acceleration, (b) Envelope acceleration.

Figs. 12 and 13 show the predicted results of the CART models and the ANFIS models for peak acceleration and envelope acceleration data. The RMSE values of the CART model and the ANFIS model for those data are summarized in Table 3. Although, the RMSEs of ANFIS models are slightly higher values than those of CART models in both cases of peak acceleration and envelope acceleration data, the predicted results of ANFIS models can keep track with the changes of the operating condition of machine more precisely. This is of crucial importance in industrial application for estimating the time-to-failure of equipments. As mentioned above, the predicted results of ANFIS models can be improved by adjusting the parameters of ANFIS. However, these changes should take into consideration the increase of computational complexity and time-consumption of the training process which may lead to unrealistic application in real life.

Fig. 12 Predicted results of peak acceleration data. (a) CART, (b) ANFIS.

Fig. 13 Predicted results of envelope acceleration data. (a) CART, (b) ANFIS.

Table 3 The RMSEs of CART and ANFIS

5. Conclusion

Machine condition prognosis is extremely essential in foretelling the degradation of operating conditions and trends of fault propagation before they reach the final failure threshold. In this study, multi-step ahead direct prediction for the operating conditions of machine based on data-driven approach has been investigated. The CART models and ANFIS models are validated by its ability to predict future state conditions of a low methane compressor using the peak acceleration and envelope acceleration data. The predicted results of CART models are slightly better than those of ANFIS. Nonetheless, they are incapable of tracking the change of machines' operating conditions with high accuracy as compared to ANFIS models. The tracking-change capability of operating conditions is of crucial importance in estimating the RUL of industrial equipments. This means that ANFIS has the potential for using as a tool to machine condition prognosis.

References

- [1] A.H.C. Tsang, Condition-based maintenance: tools and decision making, *Journal of Quality in Maintenance Engineering*, 1 (1995) 3–17.
- [2] M. Bengtsson, Condition based maintenance system technology – where is development heading, *Proceeding of the 17th European Maintenance Congress*, 11-13 May (2004), Barcelona, Spain, B-19.580-2004.
- [3] M. Abbas, A.A. Ferri, M.E. Orchard, G.J. Vachtsevanos, An intelligent diagnostic/prognostic framework for automotive electrical systems, *Intelligent Vehicles Symposium, 2007 IEEE*, 13-15 June (2007), 352–357.
- [4] F. Tu, S. Ghoshal, J. Luo, G. Biswas, S. Mahadevan, L. Jaw, K. Navarra, PHM integration with maintenance and inventory management systems, *Aerospace Conference, 2007 IEEE*, 3-10 March (2007) 1–12.
- [5] Y. Li, S. Billington, C. Zhang, T. Kurfess, S. Danyluk, S. Liang, Adaptive prognostics for rolling element bearing condition, *Mechanical Systems and Signal Processing* 13 (1999) 103–113.
- [6] Y. Li, T.R. Kurfess, S.Y. Liang, Stochastic prognostics for rolling element bearings, *Mechanical Systems and Signal Processing* 14 (2000) 747–762.
- [7] M. Watson, C. Byington, D. Edwards, S. Amin, Dynamic modeling and wear-based remaining useful life prediction of high power clutch systems, *Tribology Transactions* 48 (2005) 208–217.
- [8] M. Luo, D. Wang, M. Pham, C.B. Low, J.B. Zhang, D.H. Zhang, Y.Z. Zhao, Model-based fault diagnosis/prognosis for wheeled mobile robots: a review, *Industrial Electronics Society, 32nd Annual conference of IEEE*, 6-10 Nov. (2005) 2267–2272.
- [9] M. Schwabacher, K. Goebel, An survey of artificial intelligence for prognostics, *AAAI Fall Symposium on Artificial Intelligence for Prognostics*, 9-11 Nov. (2007).
- [10] G. Vachtsevanos, P. Wang, Fault prognosis using dynamic wavelet neural networks, *AUTOTESTCON Proceedings, IEEE Systems Readiness Technology Conference*, 22-23 Aug. (2001) 857–870.
- [11] R. Huang, L. Xi, X. Li, C. R. Liu, H. Qiu, J. Lee, Residual life prediction for ball bearings

- based on self-organizing map and back propagation neural network methods, *Mechanical Systems and Signal Processing* 21 (2007) 193–207.
- [12] W.Q. Wang, M.F. Golnaraghi, F. Ismail, Prognosis of machine health condition using neuro-fuzzy system, *Mechanical System and Signal Processing* 18 (2004) 813–831.
- [13] B. Satish, N.D.R. Sarma, A fuzzy BP approach for diagnosis and prognosis of bearing faults in induction motors, *IEEE Power Engineering Society General Meeting* 3 (2005) 2291–2294.
- [14] E.R. Brown, N.N. McCollom, E. Moore, A. Hess, Prognostics and health management – a data-driven approach to supporting the F-35 Lightning II, *Aerospace Conference*, 2007 IEEE, 3-10 March (2007) 1–12.
- [15] V.T. Tran, B.S. Yang, M.S. Oh, A.C.C. Tan, Machine condition prognosis based on regression trees and one-step-ahead prediction, *Mechanical Systems and Signal Processing*, in press.
- [16] A. Sorjamaa, A. Lendasse, Time series prediction as a problem of missing values: application to ESTSP and NN3 competition benchmarks, *European Symposium on Time Series Prediction*, 7-9 Feb. (2007) 165–174.
- [17] M.B. Kennel, R. Brown, H.D.I. Abarbanel, Determining embedding dimension for phase-space reconstruction using a geometrical construction, *Physical Review A* 45 (1992) 3403–3411.
- [18] L. Cao, Practical method for determining the minimum embedding dimension of a scalar time series, *Physica D* 110 (1997) 43–50.
- [19] D.S. Broomhead, Extracting qualitative dynamics from experimental data, *Physica D* 20 (1986) 217.
- [20] M.T. Rosenstein, J.J. Collins, C.J.D. Luca, Reconstruction expansion as a geometry-based framework for choosing proper delay time, *Physica D* 73 (1994) 82–89.
- [21] A.M. Fraser, H.L. Swinney, Independent coordinates for strange attractors from mutual information, *Phys. Rev. A* 33 (1986) 1134.
- [22] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, *Classification and regression trees*, Chapman & Hall (1984).
- [23] J.S.R. Jang, 1993, ANFIS: Adaptive-network-based fuzzy inference system, *IEEE Trans. System, Man and Cybernetics* 23 (1993) 665–685.
- [24] A. Sorjamaa, J. Hao, N. Reyhani, Y. Ji, A. Lendasse, Methodology for long-term prediction of time series, *Neurocomputing* 70 (2007) 2861–2869.
- [25] S.E. Shoura, M.E. Sherif, A. Atiya, S. Shaheen, Neural networks in forecasting model: Nile river application, *Proceeding of Circuits and Systems*, 1998 Midwest Symposium, 9-12 Aug. (1998) 600–603.
- [26] Y. Ji, J. Hao, N. Reyhani, A. Lendasse, Direct and recursive prediction of time series using mutual information selection, *Lecture Notes in Computer Science* 3512 (2005) 1010–1017.
- [27] J. Jeong, J. C. Gore, B. S. Peterson, Mutual information analysis of the EEG in patients with Alzheimer’s disease, *Clinical Neurophysiology* 112 (2001) 827–835.
- [28] E. Ikonen, K. Najim, Fuzzy neural networks and application to the FBC process, *IEE Proceeding, Control Theory Application* 143 (1996) 259–269.

List of figures

Fig. 1 Schematic of ANFIS architecture.

Fig. 2 Proposed system for machine fault prognosis.

Fig. 3 Low methane compressor: wet screw type.

Fig. 4 The entire of peak acceleration data of low methane compressor.

Fig. 5 The entire of envelope acceleration data of low methane compressor.

Fig. 6 Time delay estimation. (a) Peak acceleration, (b) Envelop acceleration.

Fig. 7 The relationship between FNN percentage and embedding dimension. (a) Peak acceleration, (b) Envelop acceleration.

Fig. 8 Training and validating results of peak acceleration data. (a) CART, (b) ANFIS.

Fig. 9 Training and validating results of envelop acceleration data. (a) CART, (b) ANFIS.

Fig. 10 RMSE convergent curve. (a) Peak acceleration, (b) Envelop acceleration.

Fig. 11 The changes of MFs after learning. (a) Peak acceleration, (b) Envelop acceleration.

Fig. 12 Predicted results of peak acceleration data. (a) CART, (b) ANFIS.

Fig. 13 Predicted results of envelop acceleration data. (a) CART, (b) ANFIS.

List of tables

Table 1 Training set D for direct prediction strategy.

Table 2 Information of the system.

Table 3 The RMSEs of CART and ANFIS.

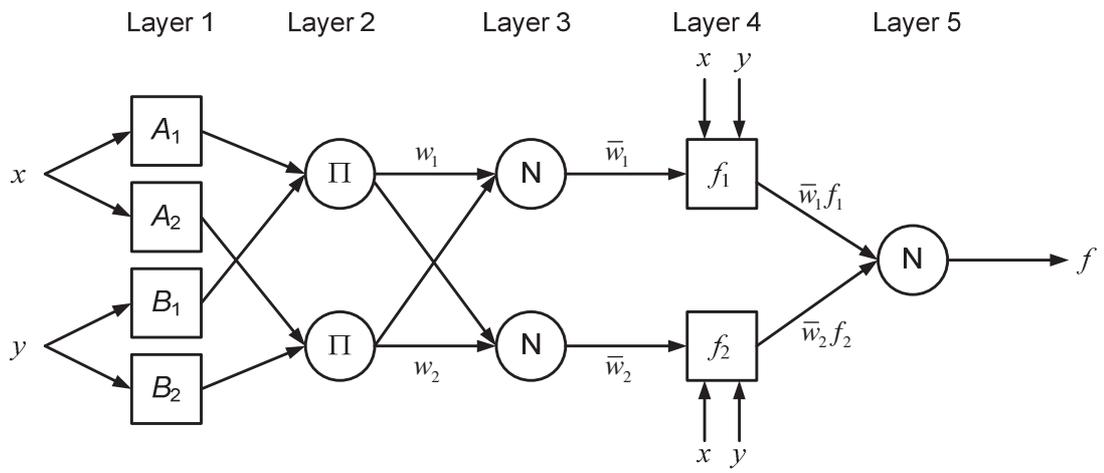


Fig. 1 Schematic of ANFIS architecture.

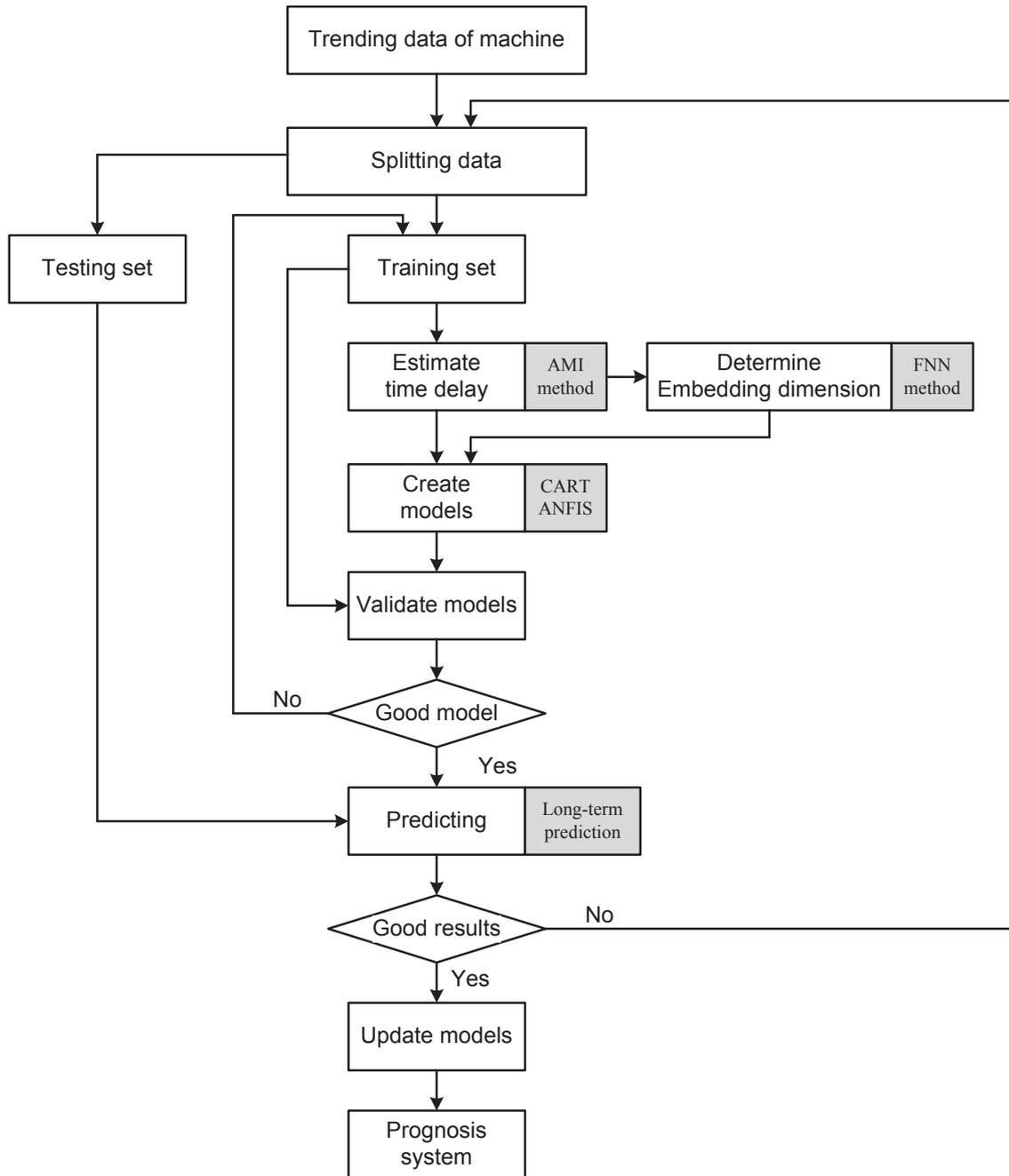


Fig. 2 Proposed system for machine fault prognosis.

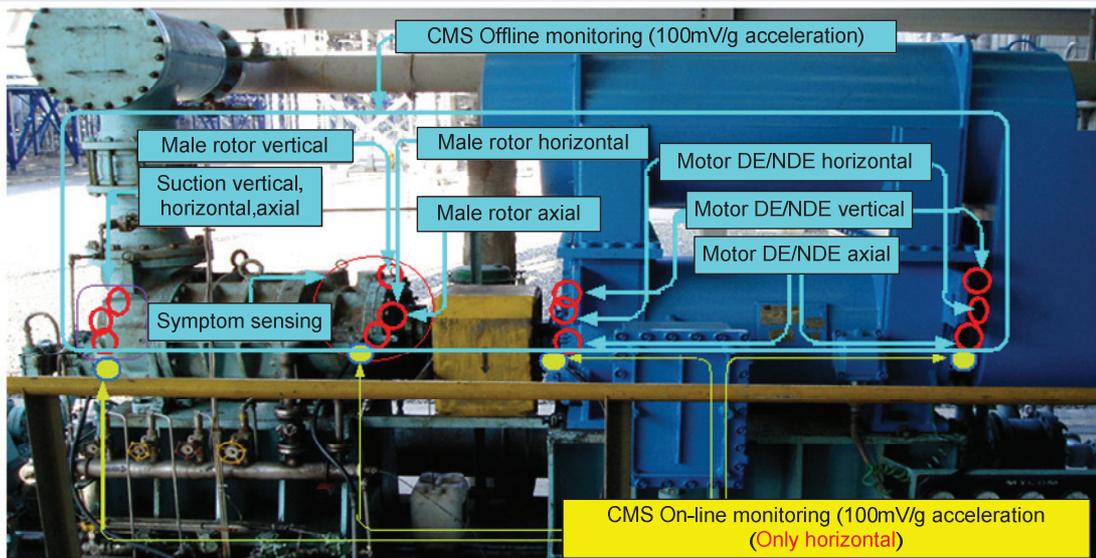


Fig. 3 Low methane compressor: wet screw type.

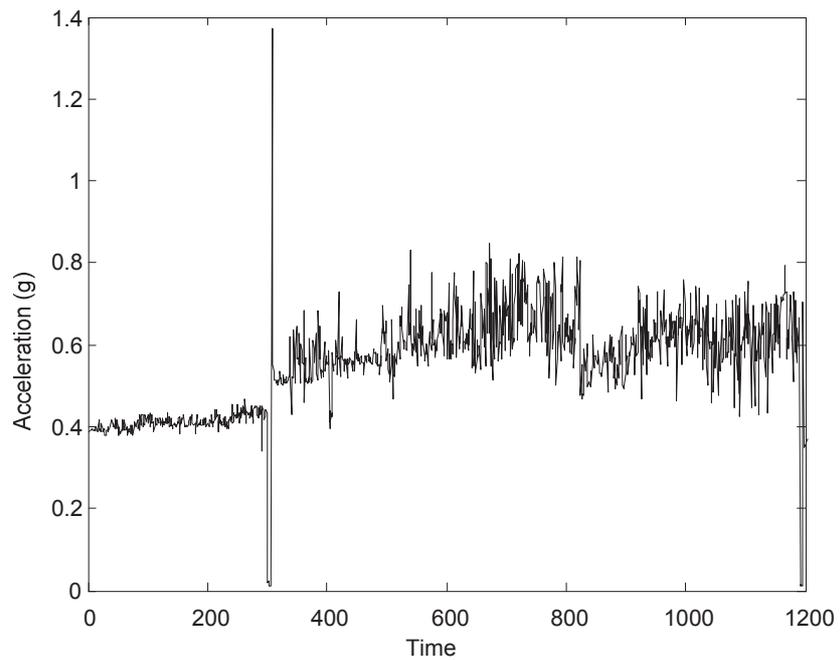


Fig. 4 The entire of peak acceleration data of low methane compressor.

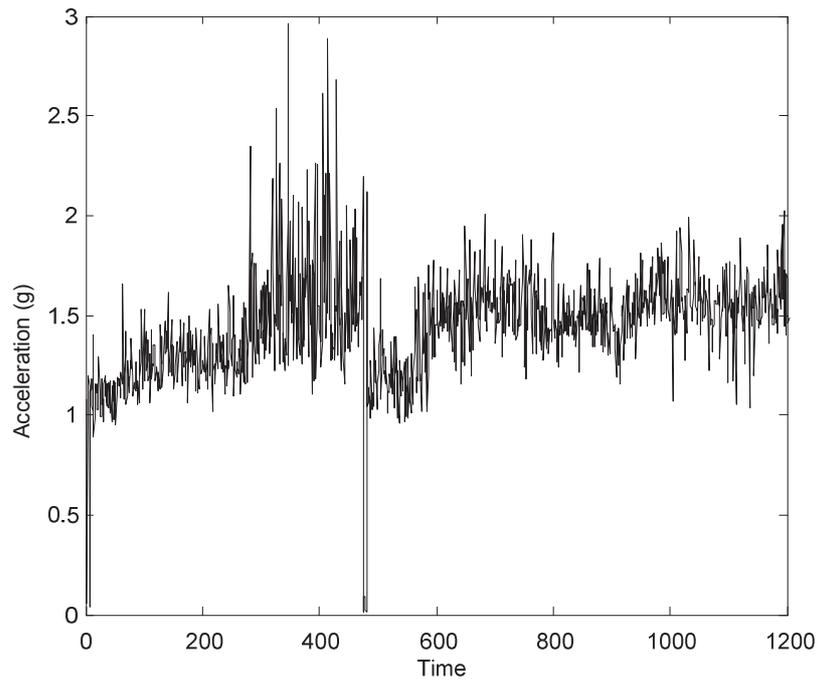
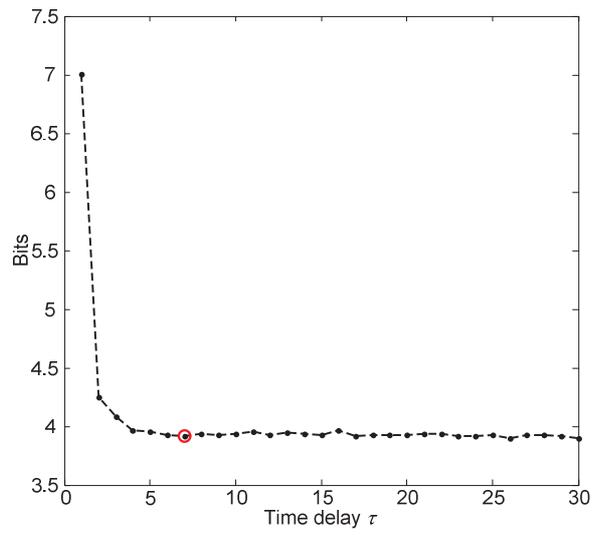
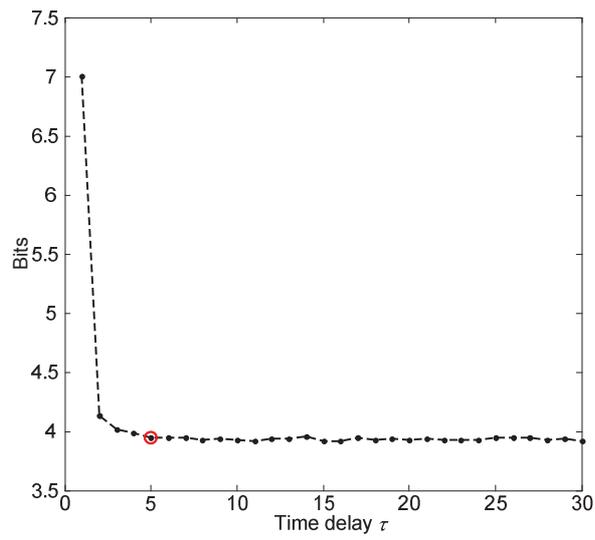


Fig. 5 The entire of envelope acceleration data of low methane compressor.

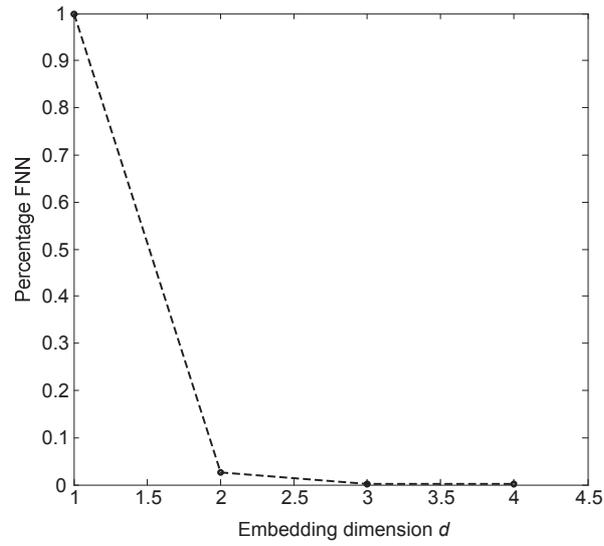


(a)

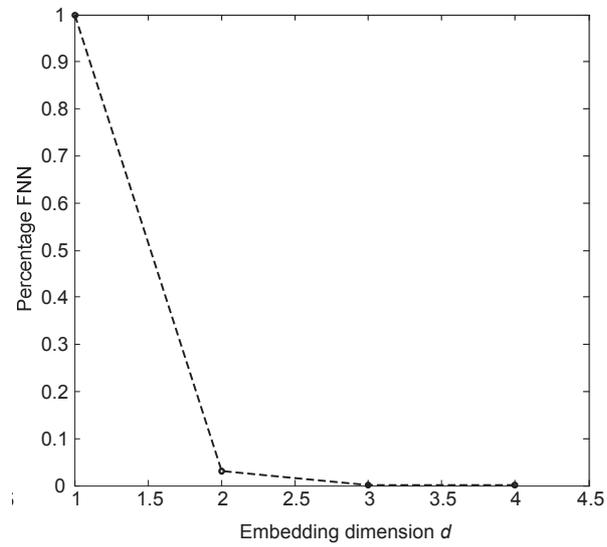


(b)

Fig. 6 Time delay estimation. (a) Peak acceleration, (b) Envelop acceleration.



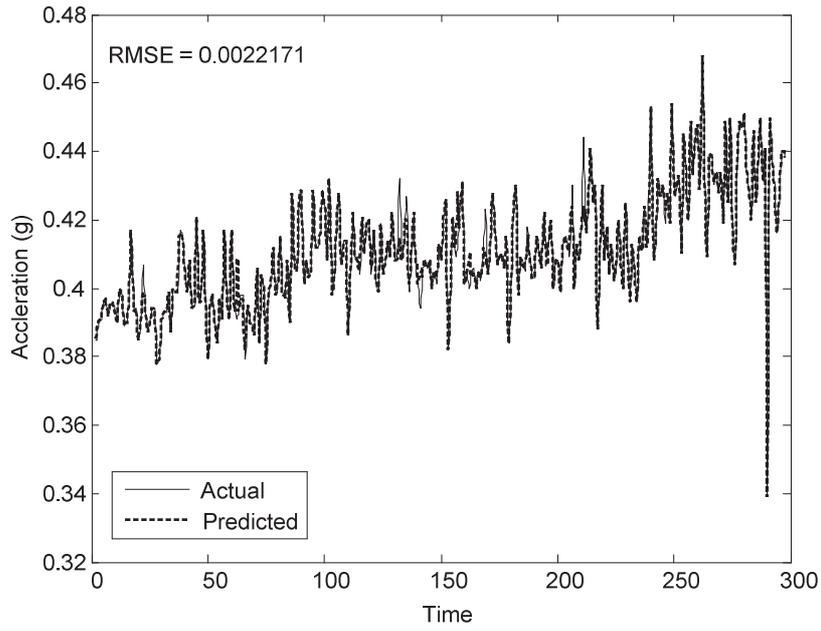
(a)



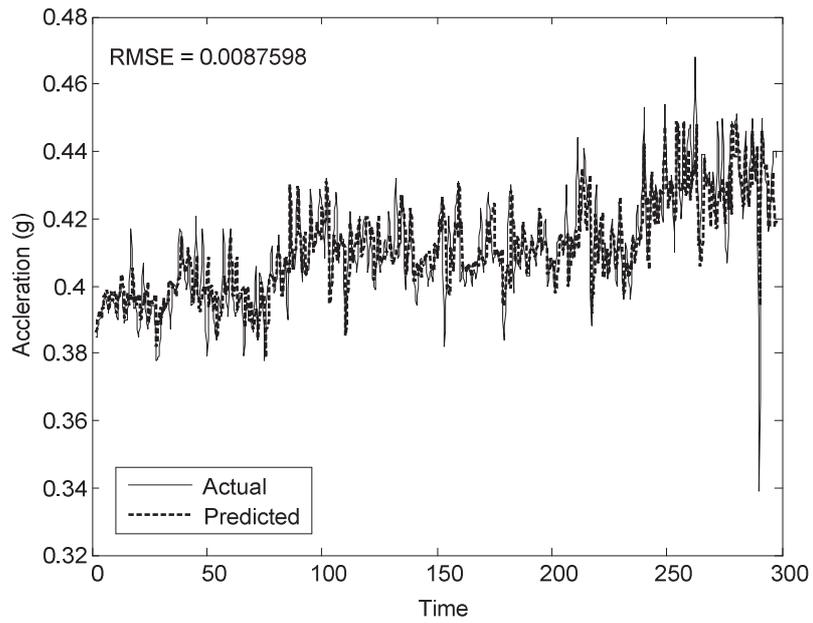
(b)

Fig. 7 The relationship between FNN percentage and embedding dimension.

(a) Peak acceleration, (b) Envelop acceleration.

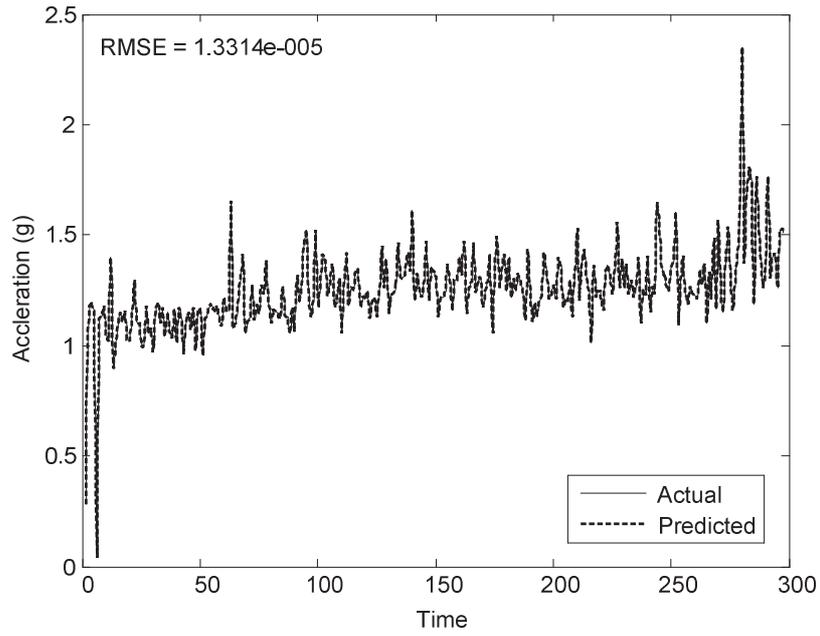


(a)

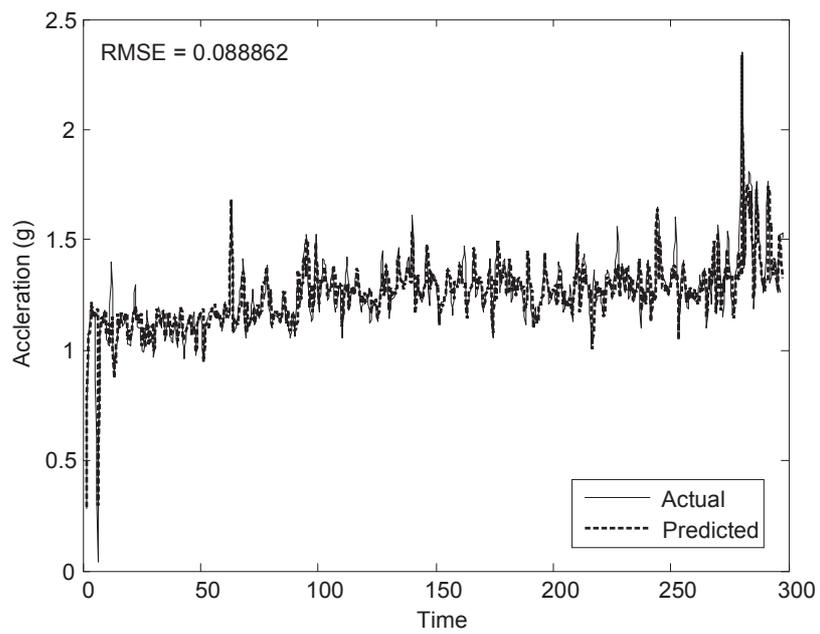


(b)

Fig. 8 Training and validating results of peak acceleration data. (a) CART, (b) ANFIS.

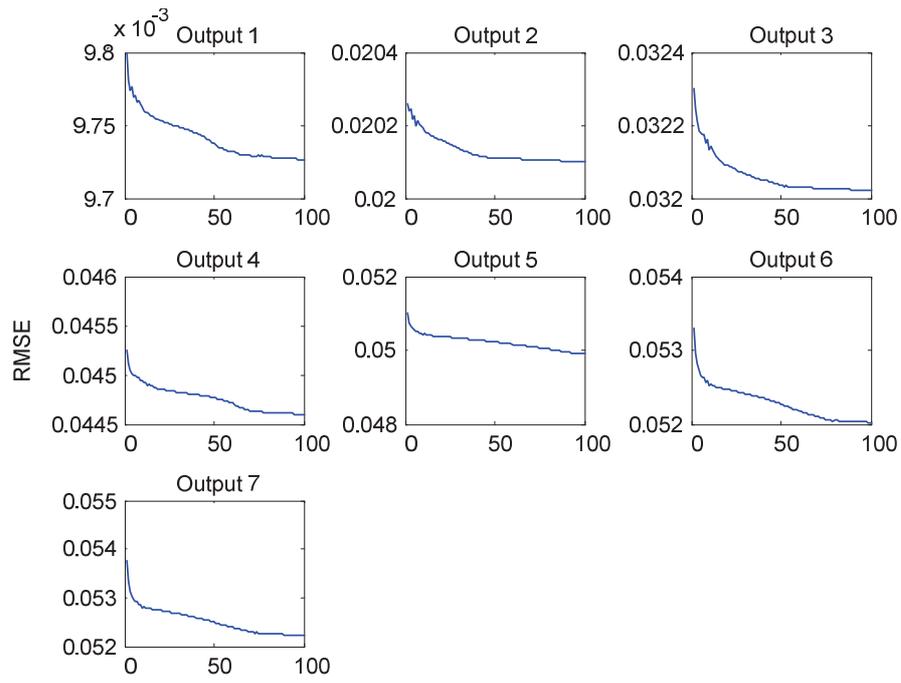


(a)

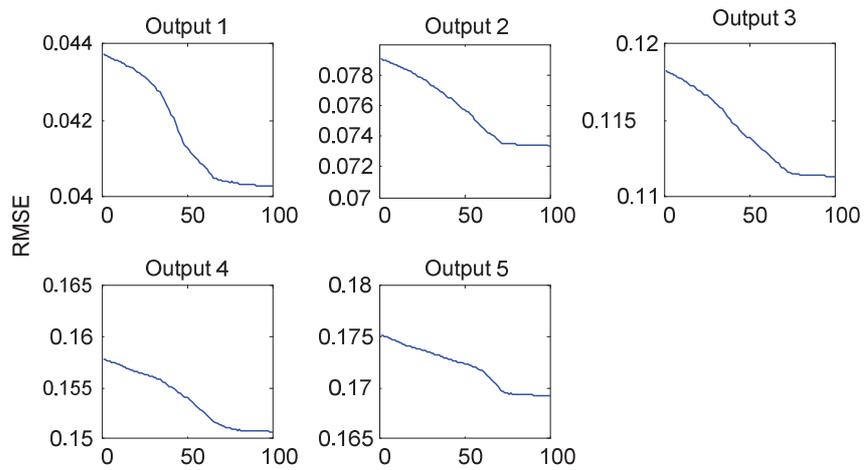


(b)

Fig. 9 Training and validating results of envelop acceleration data. (a) CART, (b) ANFIS



(a)



(b)

Fig. 10 RMSE convergent curve. (a) Peak acceleration, (b) Envelop acceleration.

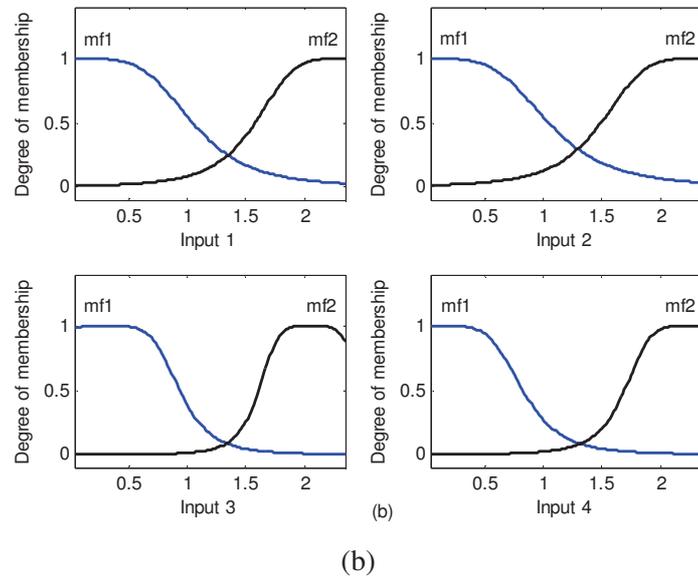
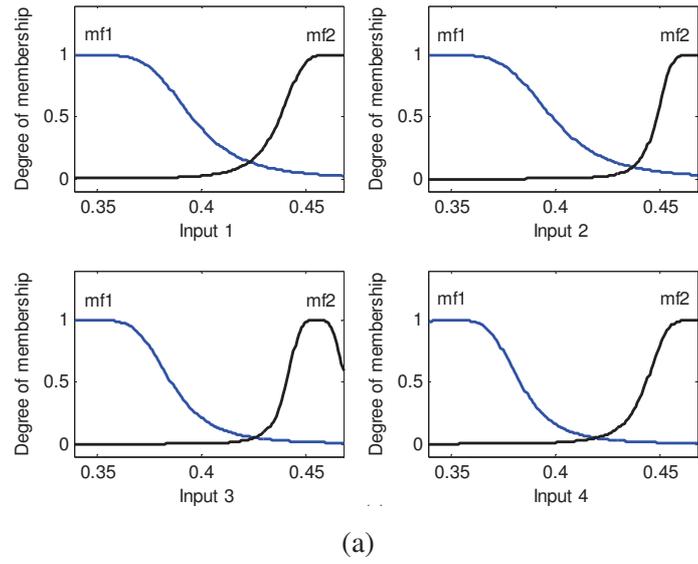
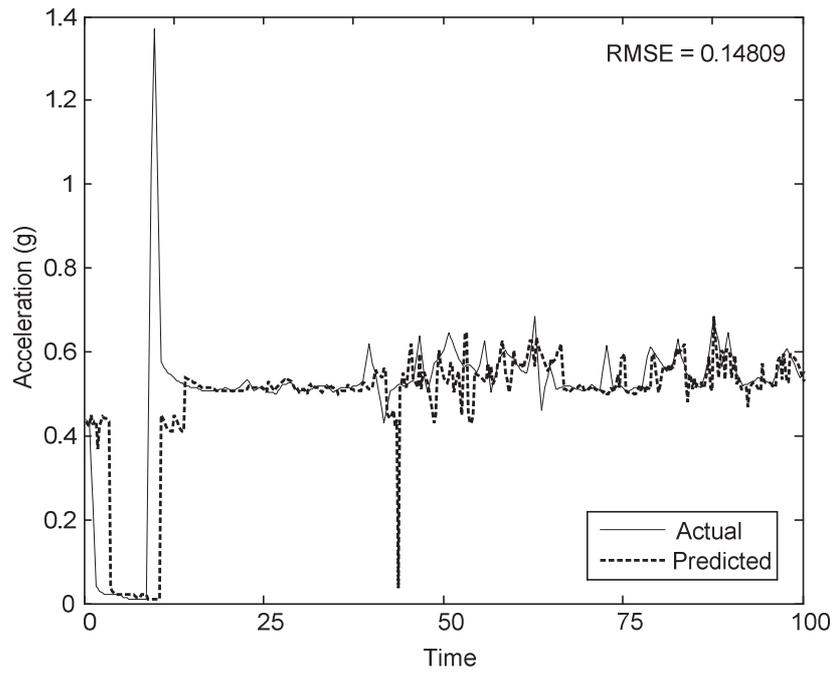
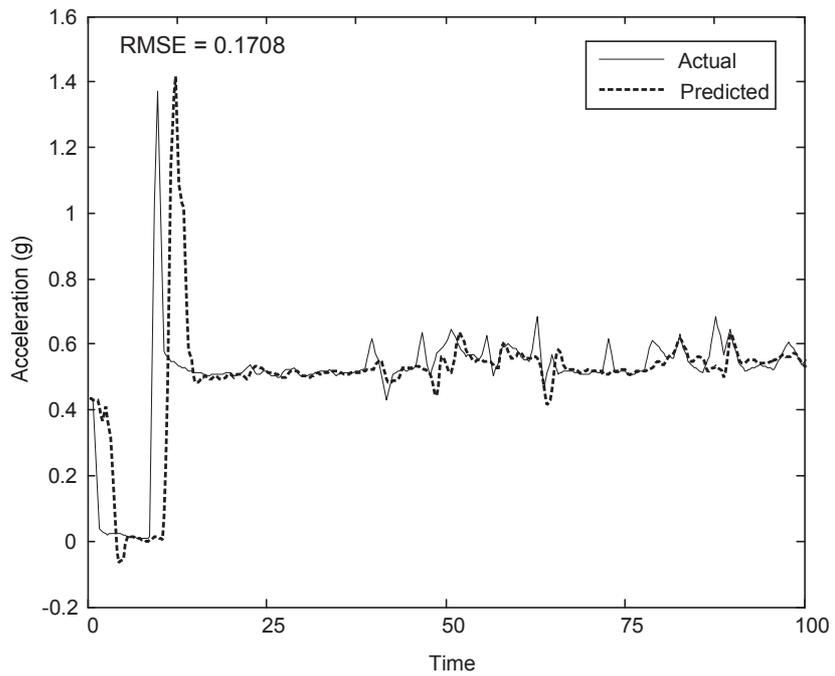


Fig. 11 The changes of MFs after learning. (a) Peak acceleration, (b) Envelop acceleration.

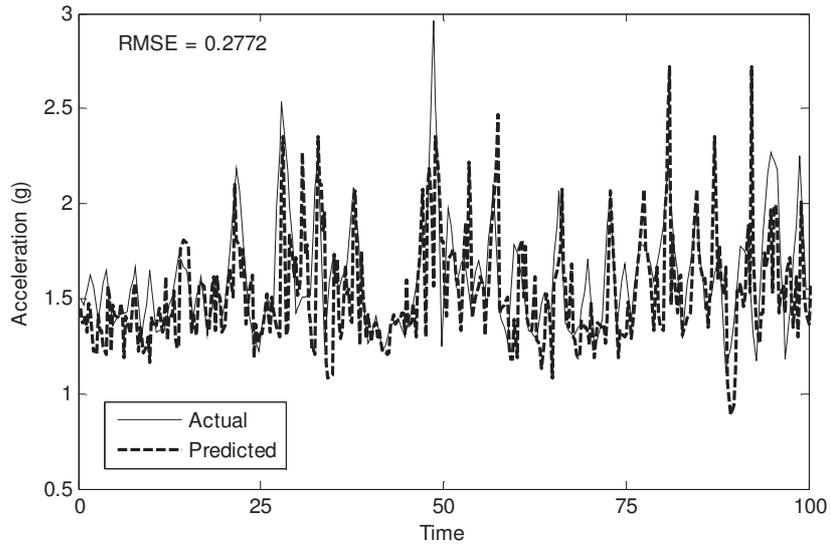


(a)

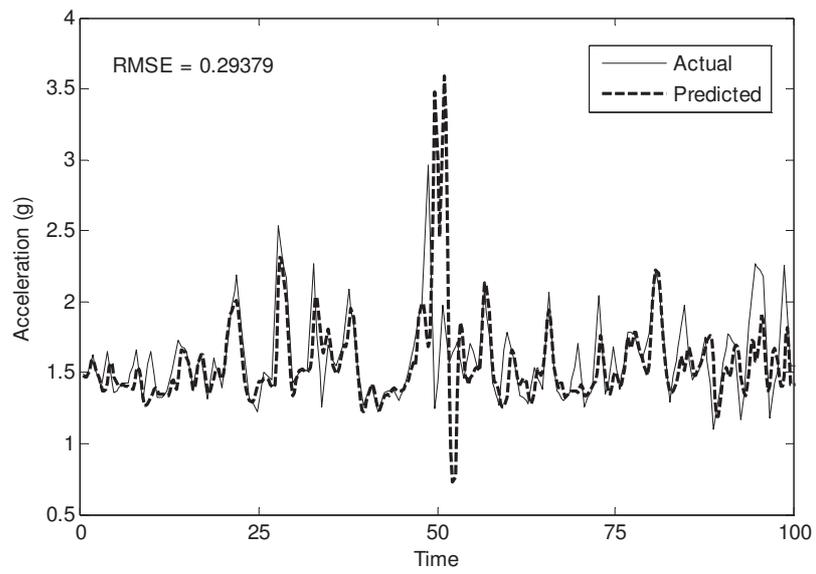


(b)

Fig. 12 Predicted results of peak acceleration data. (a) CART, (b) ANFIS.



(a)



(b)

Fig. 13 Predicted results of envelop acceleration data. (a) CART, (b) ANFIS.

Table 1 Training set D for direct prediction strategy

Input $X = [X_1, X_2, \dots, X_d]$	Output $Y = [Y_1, Y_2, \dots, Y_h]$
$[x_1, x_2, \dots, x_d]$	$[x_{d+1}, x_{d+2}, \dots, x_{d+h}]$
$[x_2, x_3, \dots, x_{d+1}]$	$[x_{d+2}, x_{d+3}, \dots, x_{d+h+1}]$
...	...
$[x_{t-h-d+1}, x_{t-h-d+2}, \dots, x_{t-h}]$	$[x_{t-h+1}, x_{t-h+2}, \dots, x_t]$

Table 2 Information of the system

Electric motor		Compressor	
Voltage	6600 V	Type	Wet screw
Power	440 kW	Lobe	Male rotor (4 lobes)
Pole	2 Pole		Female rotor (6 lobes)
Bearing	NDE:#6216, DE:#6216	Bearing	Thrust: 7321 BDB
RPM	3565 rpm		Radial: Sleeve type

Table 3 The RMSEs of CART and ANFIS

Data type	Training		Testing	
	CART	ANFIS	CART	ANFIS
Peak acceleration	0.002217	0.00876	0.14809	0.1708
Envelop acceleration	1.3314×10^{-5}	0.08886	0.2772	0.2938