Online prediction of Characteristic Parameters based on Kernel Extreme Learning Machine with Incremental Sparse Structure

Min Liu, Hongbo Fan*, Yingtang Zhang, Zhining Li
Army Engineering University of PLA
Vehicle and Electrical Engineering Department
Shijiazhuang China
hunter1848@163.com
*Corresponding author

***Abstract*—Online prediction model should be able to update its own prediction ability and knowledge automatically in real time. But due to the continuously accumulation of samples in online modeling process, many traditional online prediction methods will meet the problems of model inflation, excessive updating time and poor generalization performance. To address the problems, a novel online prediction method based on Kernel Extreme Learning Machine with incremental sparse structure (ISKELM) is proposed. Large-scale samples become sparse through the proposed method of instantaneous information measurement and the model of ISKELM is updated incrementally and recursively using the presented method consisting of incremental and decrement learning algorithms. Prediction experiments on chaotic time series and engine condition parameters are performed and results indicate that compared with several existing online prediction models, ISKELM distinctly improves the online modeling speed and model generalization performance while ensuring high prediction accuracy.**

Keywords-sample sparseness; instantaneous information measurement; incremental learning; decremental learning

I. INTRODUCTION

During the operation of equipments, the condition change rule is not immutable, especially for strongly time-varying system. So it is necessary to update the online prediction model in real time with newly generated condition samples which can reflect the current changing trend of the equipment condition. However, due to the new samples are obtained in online sequential way and become more and more as the online prediction process goes on, the traditional batch learning method based on all samples leads to model inflation, poor generalization performance and excessive updating time of the model [1,2].

According to the literature survey, it can be summarized that there are two key points to solve the above problems: sample sparseness and incremental modeling method [3,4,5]. Sample sparseness can select effective new samples and delete redundant samples, which can effectively reduce sample size and model updating time. The main sample sparseness methods include consistency criterion [4], accumulation consistency criterion [6], Approximate Linear Independence criterion (ALD) [7], Surprise measurements [8] and so on. Incremental modeling method can incrementally update the current model based on the priori knowledge, which can effectively reduce the repeated training time of batch learning mode and distinctly improve the online modeling speed. Main incremental modeling methods include inversion matrix [5], Cholesky factorization [9] and so on.

A large number of online prediction models have been established based on the traditional machine learning methods, such as decision tree [10], Markov Chain[11], Artificial Neural Network(ANN) [12], Support Vector Regression (SVR) [13]. Huang proposed Kernel Extreme Learning Machine (KELM) [14] which have higher learning speed and accuracy than traditional methods, and it is widely used in classification and prediction [14,15]. For online modeling, Richard [1] presented online sequential learning algorithm to establish the model of Online Sequential Kernel Extreme Learning Machine (OS-KELM). Zhou [9] proposed Online Kernel Extreme Learning Machine with Forgotten Mechanism (FM-OKELM) that updates the current prediction model using Cholesky factorization method based on all the new samples within a sliding time window. Simone [15] designed Online Sequential Kernel Extreme Learning Machine based on ALD (ALD-OSKELM) that selects effective new samples according to the predefined prediction error threshold to update the model incrementally. Zhang [16] presented Online Kernel Extreme Learning Machine based on Fast Leave One Out-Cross Validation (FL-OKELM) that selects effective new samples adaptively by calculating the cross-validation error of samples within the time window as the error threshold to incrementally update the model in real time.

All the above methods have improved the online prediction performance to some extent. But because they don't pay full attention to deleting redundant old samples and limiting model

978-1-7281-0860-5/19/$31.00 ©2019 IEEE

inflation, the efficiency of sample sparseness and online modeling still need to be further improved. Therefore, Kernel Extreme Learning Machine with Incremental Sparse Structure (ISKELM) is proposed in this paper. The remaining paper is organized as follows. Section 2 presents the basic principle of ISKELM. Method of constructing the sparse kernel function dictionary based on instantaneous information measurement is proposed in Section 3. Section 4 puts forward the improved incremental recursive updating method for the kernel function weight matrix of ISKELM. Online prediction Experiments on chaotic time series and engine condition parameters are shown in Section 5. Section 6 draws conclusions and proposes future work.

II. Kernel Extreme Learning Machine with Incremental Sparse Structu

Given a time series $\boldsymbol{S}=\{(\boldsymbol{u}_i,d_i)\}_{i=1}^{t}$, where $\boldsymbol{u}_i \in \text{¡}^{n}$ is the input data, and $d_i \in \text{¡}$ is the desired output data, the regression model of KELM with multi-input and single-output structure can be expressed as

$$\begin{aligned} f(\cdot) &= \boldsymbol{h}(\cdot)\boldsymbol{H}^T\left(\gamma^{-1}\boldsymbol{I}+\boldsymbol{H}\boldsymbol{H}^T\right)^{-1}\boldsymbol{d} \\ &= \boldsymbol{k}(\cdot)\boldsymbol{\alpha} \\ &= \sum\nolimits_{i=1}^{t}\alpha_i k(\boldsymbol{u}_i,\cdot) \end{aligned} \tag{1}$$

where $\boldsymbol{H}=[\boldsymbol{h}(\boldsymbol{u}_1)^T,\cdots,\boldsymbol{h}(\boldsymbol{u}_t)^T]^T$ is the mapping matrix of all the input samples, $\boldsymbol{d}=[d_1,d_2,\cdots,d_t]^T$ is the desired output vector, γ is the penalty factor, $\boldsymbol{k}(\cdot)=[k(\boldsymbol{u}_1,\bullet),\cdots,k(\boldsymbol{u}_t,\bullet)]$ is the kernel function matrix consisting of kernel operation results between every two samples, and $\boldsymbol{\alpha}=[\alpha_1,\alpha_2,\cdots,\alpha_t]=(\gamma^{-1}\boldsymbol{I}+\boldsymbol{G})^{-1}\boldsymbol{d}$ is the output weight matrix of KELM.

According to Eq. (1), the computational complexity of the model will become larger with the continuous increase of samples. It makes the online modeling process more and more difficult. Therefore, sample sparseness is necessary for reducing the sample size and improve the online modeling efficiency. According to the theory of dictionary learning based on sparse representation [17], a sparse kernel function dictionary based on $\boldsymbol{k}(\cdot)$ can be defined as

$$D_t=\{k(\boldsymbol{c}_1,\cdot),k(\boldsymbol{c}_2,\cdot),\mathrm{L}\ ,k(\boldsymbol{c}_{m_t},\cdot)\} \tag{2}$$

The output of ISKELM at t moment can be described as

$$\hat{f}_t(\cdot)=\sum\nolimits_{i=1}^{m_t}\alpha_{i,t}k(\boldsymbol{c}_i,\cdot) \tag{3}$$

where $\{\boldsymbol{c}_1,\mathrm{L}\ ,\boldsymbol{c}_{m_t}\}\subset\{\boldsymbol{u}_1,\mathrm{L}\ ,\boldsymbol{u}_t\}$ are the effective samples, $\boldsymbol{c}_i$ is the center of the *i-th* kernel function, $m_t=t$ is the order of the current model, and $\alpha_{i,t}$ is the weight of the *i-th* kernel function at t moment. The output corresponding to the input at $t+1$ moment can be expressed as

$$\hat{f}_t(\boldsymbol{u}_{t+1})=\sum\nolimits_{i=1}^{m_t}\alpha_{i,t}k(\boldsymbol{c}_i,\boldsymbol{u}_{t+1}) \tag{4}$$

To implement online modeling of ISKELM, we propose a method of instantaneous information measurement to construct the sparse kernel function dictionary of ISKELM and an incrementally updating method consisting of incremental and decrement learning algorithms to update the kernel function weight matrix of ISKELM.

III. Sparse kernel function dictionary construction method

The learning model at t moment is recorded as $T(f_t,\alpha_t,D_t)$, which is simply marked as T_t. At $t+1$ moment, a new kernel function is obtained when a new observed sample $\boldsymbol{u}_{t+1}$ is generated. The potential dictionary is recorded as $\bar{D}_t=\{D_t,k(\boldsymbol{u}_{t+1},\cdot)\}$ at this moment. To determine whether to accept $k(\boldsymbol{u}_{t+1},\cdot)$, firstly give two definitions as follows.

Definition 1. Assuming that the instantaneous posterior probability of $\boldsymbol{u}_{t+1}$ under T_t is $p_t(\boldsymbol{u}_{t+1}\,|\,T_t)$, the amount of information in $\boldsymbol{u}_{t+1}$ that can be transferred to the current dictionary is defined as the instantaneous conditional self-information of $\boldsymbol{u}_{t+1}$, which can be described as

$$I(\boldsymbol{u}_{t+1}\,|\,T_t)=-\log p_t(\boldsymbol{u}_{t+1}\,|\,T_t) \tag{5}$$

Definition 2. Assuming that the number of elements in the dictionary is m_t, and the instantaneous posteriori probability of the kernel center $\boldsymbol{c}_i(1\le \boldsymbol{i}\le m_t)$ under T_t is $p_t(\boldsymbol{c}_i\,|\,T_t)$, the average self-information of the dictionary D_t at the t moment is defined as its instantaneous conditional entropy, which can be described as

$$H(D_t\,|\,T_t)=-\sum\nolimits_{i=1}^{m_t}p_t(\boldsymbol{c}_i\,|\,T_t)\log p_t(\boldsymbol{c}_i\,|\,T_t) \tag{6}$$

Kernel Density Estimator (KDE) is used to calculate the Probability Distribution Function (PDF) of $\boldsymbol{u}_{t+1}$ and $\boldsymbol{c}_i$. It is known that for a data sequence $\boldsymbol{U}=\{\boldsymbol{u}_1,\boldsymbol{u}_2,\mathrm{L}\ ,\boldsymbol{u}_N\}\in ?^{n}$, its KDE-based PDF can be expressed as $p(\boldsymbol{u}\,|\,\theta,\boldsymbol{w})=\sum\nolimits_{i=1}^{N}w_i k_\theta(\boldsymbol{u},\boldsymbol{u}_i)$. where $\sum\nolimits_{i=1}^{N}w_i=1$, θ is kernel width and w_i is weight coefficient. The kernel coefficient is calculated using the maximum neighborhood estimation criterion [18]. Hence the PDF can be expressed as

$$p(\boldsymbol{u}\,|\,\theta,\boldsymbol{w})=\frac{1}{N}\sum_{i=1}^{N}k_\theta(\boldsymbol{u},\boldsymbol{u}_i) \tag{7}$$

For the dictionary $D_t=\{k(\boldsymbol{c}_1,\cdot),\mathrm{L}\ ,k(\boldsymbol{c}_{m_t},\cdot)\}$ under T_t, the instantaneous conditional KDE-based PDF of the kernel center $\boldsymbol{c}$ can be expressed as

$$p_t(\boldsymbol{c}\,|\,\theta,T_t)=\frac{1}{m_t}\sum_{i=1}^{m_t}k_\theta(\boldsymbol{c},\boldsymbol{c}_i) \tag{8}$$

Thus, Eq. (5) and (6) can be redefined as

$$I(\boldsymbol{u}_{t+1}\,|\,\theta,T_t)=-\log\frac{1}{m_t}\sum_{i=1}^{m_t}k_\theta(\boldsymbol{u}_{t+1},\boldsymbol{c}_i) \tag{9}$$

$$H(D_t \mid \theta, T_t) = -\sum_{i=1}^{m_t} \left\{ \left[\frac{1}{m_t} \sum_{j=1}^{m_t} k_\theta(\boldsymbol{c}_i, \boldsymbol{c}_j) \right] \log \left[\frac{1}{m_t} \sum_{j=1}^{m_t} k_\theta(\boldsymbol{c}_i, \boldsymbol{c}_j) \right] \right\} \tag{10}$$

In this paper, we select the effective samples and eliminate the redundant samples according to Eq. (9) and (10) in order to construct a sparse kernel function dictionary online. The kernel function used in this paper is the unit norm kernel function which is expressed as $\forall \boldsymbol{u} \in \boldsymbol{U}, k(\boldsymbol{u}, \boldsymbol{u}) = 1$.

A. Online expansion strategy for sparse kernel function dictionary

Given $\boldsymbol{e}_t = [1, \cdots, 1]^T \in \mathbb{R}^{m_t \times 1}$, the Gram matrix of dictionary D_t is denoted as $\boldsymbol{G}_t$, so the matrix $\boldsymbol{S}_t$ can be calculated according to $\boldsymbol{S}_t = \boldsymbol{G}_t \times \boldsymbol{e}_t$. The calculation result can be written as

$$\boldsymbol{S}_t = \left[\sum_{i=1}^{m_t} k_\theta(\boldsymbol{c}_1, \boldsymbol{c}_j), \sum_{i=1}^{m_t} k_\theta(\boldsymbol{c}_2, \boldsymbol{c}_j), \mathrm{L} \ \sum_{i=1}^{m_t} k_\theta(\boldsymbol{c}_{m_t}, \boldsymbol{c}_j) \right]^T \tag{11}$$

According to KDE, the instantaneous conditional probability of the $\boldsymbol{c}_i$ in D_t under T_t is $p_t(\boldsymbol{c}_i \mid \theta, T_t) = \boldsymbol{S}_t(i) / m_t$. So the instantaneous conditional entropy of D_t is

$$H(D_t \mid \theta, T_t) = -(\frac{\boldsymbol{S}_t^T}{m_t}) \log(\frac{\boldsymbol{S}_t}{m_t}) \tag{12}$$

At time $t+1$, the Gram matrix of the potential dictionary $\bar{D}_t = \{D_t, k(\boldsymbol{u}_{t+1}, \cdot)\}$ made up of all kernel functions is denoted as

$$\bar{\boldsymbol{G}}_t = \begin{bmatrix} \boldsymbol{G}_t & \boldsymbol{k}_t \\ \boldsymbol{k}_t^T & 1 \end{bmatrix} \tag{13}$$

Given $\bar{\boldsymbol{e}}_t = [1, \mathrm{L} \ , 1]^T \in ?^{(m_t+1) \times 1}$, according to $\bar{\boldsymbol{S}}_t = \bar{\boldsymbol{G}}_t \times \bar{\boldsymbol{e}}_t$, $\bar{\boldsymbol{S}}_t$ can be described as

$$\bar{\boldsymbol{S}}_t = \begin{bmatrix} \boldsymbol{G}_t & \boldsymbol{k}_t \\ \boldsymbol{k}_t^T & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{e}_t \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{S}_t + \boldsymbol{k}_t \\ 1 + \sum \boldsymbol{k}_t \end{bmatrix} \tag{14}$$

where $\boldsymbol{k}_t = [k_\theta(\boldsymbol{c}_1, \boldsymbol{u}_{t+1}), \mathrm{L} \ , k_\theta(\boldsymbol{c}_{m_t}, \boldsymbol{u}_{t+1})]^T \in ?^{m_t \times 1}$, the result of $\bar{\boldsymbol{S}}_t$ can be obtained by using Eq. (11) and Eq. (14).

The instantaneous conditional probability of the *i-th* kernel core of $\bar{D}_t$ is $p_t(\boldsymbol{c}_i \mid \theta, \bar{T}_t) = \bar{\boldsymbol{S}}_t(i) / (m_t + 1)$. So the instantaneous conditional entropy of $\bar{D}_t$ is

$$H(\bar{D}_t \mid \theta, \bar{T}_t) = -(\frac{\bar{\boldsymbol{S}}_t^T}{m_t + 1}) \log(\frac{\bar{\boldsymbol{S}}_t}{m_t + 1}) \tag{15}$$

According to the relevant definition, the redundancy values of the D_t and $\bar{D}_t$ can be respectively defined as

$$R_t = 1 - \frac{H(D_t \mid \theta, T_t)}{\log |D_t|} = 1 - \frac{H(D_t \mid \theta, T_t)}{\log |m_t|} \tag{16}$$

$$\bar{R}_t = 1 - \frac{H(\bar{D}_t \mid \theta, \bar{T}_t)}{\log |\bar{D}_t|} = 1 - \frac{H(\bar{D}_t \mid \theta, \bar{T}_t)}{\log |m_t + 1|} \tag{17}$$

If $\bar{R}_t < R_t$, it indicates that the new added kernel function reduces the redundancy of the dictionary, and increases the average amount of self-information of the dictionary. Hence the new training sample is added into the learning model, and there are $D_{t+1} = \{D_t, k(\boldsymbol{u}_{t+1}, \cdot)\}$, $m_{t+1} = m_t + 1$, $R_{t+1} = \bar{R}_t$, $\boldsymbol{S}_{t+1} = \bar{\boldsymbol{S}}_t$, $G_{t+1} = \bar{\boldsymbol{G}}_t$ and $H(D_{t+1} \mid \theta, T_t) = H(\bar{D}_t \mid \theta, \bar{T}_t)$. Otherwise, the new training sample is deleted directly as a redundant sample, and the above parameters remain unchanged.

B. Online pruning strategy for sparse kernel function dictionary

When the size of the dictionary reaches a predetermined size, the pruning strategy is executed at the $t+1$ moment when $m_t = m$. By selecting m elements from $m+1$ potential elements, the selected element can obtain the largest instantaneous conditional self-information.

Defining the matrix $\bar{\boldsymbol{E}}_t = (\bar{\boldsymbol{e}}_t \times \bar{\boldsymbol{e}}_t^{\,T} - \boldsymbol{I}_t) \in ¡^{(m+1) \times (m+1)}$, where $\bar{\boldsymbol{e}}_t = [1, \mathrm{L} \ , 1]^T \in ?^{(m+1) \times 1}$, and $\boldsymbol{I}_t$ is the unit matrix with $(m+1)$ order. $\bar{\boldsymbol{F}}_t$ can be defined as

$$\begin{aligned} \bar{\boldsymbol{F}}_t &= \bar{\boldsymbol{G}}_t \times \bar{\boldsymbol{E}}_t \\ &= \bar{\boldsymbol{G}}_t \times (\bar{\boldsymbol{e}}_t \times \bar{\boldsymbol{e}}_t^{\,T}) - \bar{\boldsymbol{G}}_t \\ &= \begin{bmatrix} \boldsymbol{G}_t & \boldsymbol{k}_t \\ \boldsymbol{k}_t^T & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{e}_t & \cdots & \boldsymbol{e}_t \\ 1 & \cdots & 1 \end{bmatrix} - \bar{\boldsymbol{G}}_t \\ &= \underbrace{\begin{bmatrix} \bar{\boldsymbol{S}}_t & \cdots & \bar{\boldsymbol{S}}_t \end{bmatrix}}_{m+1} - \bar{\boldsymbol{G}}_t \end{aligned} \tag{18}$$

$\bar{\boldsymbol{G}}_t$ and $\bar{\boldsymbol{S}}_t$ are respectively obtained by solving Eq. (13) and Eq. (19). The result of $\bar{\boldsymbol{F}}_t$ can be redefined as Eq. (19) by substituting $\bar{\boldsymbol{G}}_t$ and $\bar{\boldsymbol{S}}_t$ into Eq. (18).

$$\bar{\boldsymbol{F}}_t = \begin{bmatrix} \sum\limits_{1 \le j \le m+1}^{j \ne 1} k_\theta(\boldsymbol{c}_1, \boldsymbol{c}_j) & \cdots & \sum\limits_{1 \le j \le m+1}^{j \ne m} k_\theta(\boldsymbol{c}_1, \boldsymbol{c}_j) & \sum\limits_{1 \le j \le m+1}^{j \ne m+1} k_\theta(\boldsymbol{c}_1, \boldsymbol{c}_j) \\ \vdots & \ddots & \vdots & \vdots \\ \sum\limits_{1 \le j \le m+1}^{j \ne 1} k_\theta(\boldsymbol{c}_m, \boldsymbol{c}_j) & \cdots & \sum\limits_{1 \le j \le m+1}^{j \ne m} k_\theta(\boldsymbol{c}_m, \boldsymbol{c}_j) & \sum\limits_{1 \le j \le m+1}^{j \ne m+1} k_\theta(\boldsymbol{c}_m, \boldsymbol{c}_j) \\ \sum\limits_{1 \le j \le m+1}^{j \ne 1} k_\theta(\boldsymbol{u}_{t+1}, \boldsymbol{c}_j) & \cdots & \sum\limits_{1 \le j \le m+1}^{j \ne m} k_\theta(\boldsymbol{u}_{t+1}, \boldsymbol{c}_j) & \sum\limits_{1 \le j \le m+1}^{j \ne m+1} k_\theta(\boldsymbol{u}_{t+1}, \boldsymbol{c}_j) \end{bmatrix} \tag{19}$$

When the *i-th* $(1 \le i \le m+1)$ element of $\bar{D}_t$ is deleted, the instantaneous conditional KDE-based PDF of the *l-th* $(l \ne i)$ element can be expressed as

$$p_t(\boldsymbol{c}_l \mid \theta, \bar{T}_t^{-i}) = \frac{1}{m} \sum_{1 \le j \le m+1}^{j \ne i} k_\theta(\boldsymbol{c}_l, \boldsymbol{c}_j) \tag{20}$$

According to Eq. (18), Eq. (8) can be redefined as

$$p_t(\boldsymbol{c}_l \mid \theta, \bar{T}_t^{-i}) = \frac{1}{m} \bar{\boldsymbol{F}}_t(l, i) = \frac{1}{m} (\bar{\boldsymbol{S}}_t - \bar{\boldsymbol{G}}_t(:, i))(l) \tag{21}$$

Thus, the instantaneous conditional self-information matrix of all the elements in the dictionary $\bar{D}_t^{-i}$, which consists of the remaining elements after removing the *i-th* element can be defined as

$$\boldsymbol{I}(\boldsymbol{c}\mid\theta,\bar{T}_t^{-i}) = -\log\frac{1}{m}\begin{bmatrix} (\bar{\boldsymbol{S}}_t-\bar{\boldsymbol{G}}_t(:,i))(1) \\ \mathrm{M} \\ (\bar{\boldsymbol{S}}_t-\bar{\boldsymbol{G}}_t(:,i))(i-1) \\ (\bar{\boldsymbol{S}}_t-\bar{\boldsymbol{G}}_t(:,i))(i+1) \\ \mathrm{M} \\ (\bar{\boldsymbol{S}}_t-\bar{\boldsymbol{G}}_t(:,i))(m+1) \end{bmatrix} \tag{22}$$

The minimum value of Eq. (22) is written as

$$\mu_t^{-i} = \min \boldsymbol{I}(\boldsymbol{c}\mid\theta,\bar{T}_t^{-i}) \tag{23}$$

The more instantaneous self-information each vector possesses in the dictionary, the less similar it is to one another, and the more information the dictionary has [14]. To maximize the amount of instantaneous condition self-information for each element in the dictionary, the subscript of the element to be deleted can be determined by Eq. (24).

$$i = \underset{1\le i\le m+1}{\arg\ \max}(\mu_t^{-1},\mathrm{L}\ ,\mu_t^{-(m+1)}) \tag{24}$$

If $i = m+1$, the new kernel function is removed from the potential dictionary. The original dictionary and parameters remain unchanged. Instead, replace the *i-th* kernel function $k(\boldsymbol{c}_i,\cdot)$ with $k(\boldsymbol{u}_{t+1},\cdot)$, and there are $D_{t+1} = \bar{D}_t^{-i}$, $\boldsymbol{S}_{t+1} = \bar{\boldsymbol{S}}_t^{-i}$, $\boldsymbol{G}_{t+1} = \boldsymbol{A}_{t+1} - \gamma^{-1}\boldsymbol{I}$. $\bar{\boldsymbol{S}}_t^{-i}$ can be obtained by solving the following equation group.

$$\begin{cases} \bar{\boldsymbol{S}}_t^{-i}(1:i-1) = (\bar{\boldsymbol{S}}_t-\bar{\boldsymbol{G}}_t(:,i))(1:i-1) \\ \bar{\boldsymbol{S}}_t^{-i}(i:m) = (\bar{\boldsymbol{S}}_t-\bar{\boldsymbol{G}}_t(:,i))(i+1:m+1) \end{cases} \tag{25}$$

IV. INCREMENTAL UPDATING METHOD FOR THE KERNEL FUNCTION WEIGHT MATRIX OF ISKELM

In order to match the dictionary expansion and pruning process， limit model inflation and update kernel function weight matrix fast, an incremental recursive updating method based on the traditional incremental modeling method is proposed in this paper, which consists of the incremental learning algorithm and improved decrement learning algorithm.

A. Incremental learning algorithm

The incremental learning process of the kernel weight matrix is corresponding to the online expansion process of the sparse kernel function dictionary. When the dictionary size is less than m, new training samples are used to extend the dictionary. At the same time, the kernel function dictionary corresponding to the dictionary is updated. In ISKELM , the kernel weight matrix is written as $\boldsymbol{\alpha} = (\gamma^{-1}\boldsymbol{I}+\boldsymbol{G})^{-1}\boldsymbol{d}$. At t moment, we define $\boldsymbol{A}_t$ as $\boldsymbol{A}_t = \gamma^{-1}\boldsymbol{I}+\boldsymbol{G}_t$.

At $t+1$ moment, $\boldsymbol{A}_{t+1}$ for training sample $(\boldsymbol{u}_{t+1},d_{t+1})$ can be defined as

$$\boldsymbol{A}_{t+1} = \begin{bmatrix} \boldsymbol{A}_t & \boldsymbol{k}_t \\ \boldsymbol{k}_t^T & v_t \end{bmatrix} \tag{26}$$

where $v_t = \gamma^{-1} + k_{t+1,t+1}$, $v_t = \gamma^{-1} + k_{t+1,t+1}$.

The inverse matrix of $\boldsymbol{A}_{t+1}$ can be obtained as Eq. (27) using the formula of calculating the inverse matrix of a block matrix.

$$\boldsymbol{A}_{t+1}^{-1} = \begin{bmatrix} \boldsymbol{A}_t^{-1} + \boldsymbol{A}_t^{-1}\boldsymbol{k}_t\rho_t^{-1}\boldsymbol{k}_t^T\boldsymbol{A}_t^{-1} & -\boldsymbol{A}_t^{-1}\boldsymbol{k}_t\rho_t^{-1} \\ -\rho_t^{-1}\boldsymbol{k}_t^T\boldsymbol{A}_t^{-1} & \rho_t^{-1} \end{bmatrix} \tag{27}$$

where $\rho_t = v_t - \boldsymbol{k}_t^T\boldsymbol{A}_t^{-1}\boldsymbol{k}_t$.

B. Improved decrement learning algorithm

When the dictionary size is greater than m, this paper presents an improved decrement learning algorithm in order to recursively update model parameters online, while maintaining the dictionary size is equal to m. Move the ***i-th*** row to the first row and move the ***i-th*** column to the first column in $\boldsymbol{A}_t$ to obtain $\tilde{\boldsymbol{A}}_t$. This conversion process can be expressed as $\tilde{\boldsymbol{A}}_t = \boldsymbol{P}\boldsymbol{A}_t\boldsymbol{Q}$, where $\boldsymbol{P}$ and $\boldsymbol{Q}$ are two orthogonal matrices with m order, $\boldsymbol{P}\boldsymbol{P}^T = \boldsymbol{E}$ and $\boldsymbol{Q}\boldsymbol{Q}^T = \boldsymbol{E}$. According to the nature of orthogonal matrix, there are $\boldsymbol{P}^{-1} = \boldsymbol{P}^T$, $\boldsymbol{Q}^{-1} = \boldsymbol{Q}^T$. And then, $\boldsymbol{Q}^{-1} = \boldsymbol{P}$, $\boldsymbol{P}^{-1} = \boldsymbol{Q}$, because $\boldsymbol{P} = \boldsymbol{Q}^T$. So the inverse matrix of $\tilde{\boldsymbol{A}}_t$ can be calculated by

$$\tilde{\boldsymbol{A}}_t^{-1} = (\boldsymbol{P}\boldsymbol{A}_t\boldsymbol{Q})^{-1} = \boldsymbol{P}\boldsymbol{A}_t^{-1}\boldsymbol{Q} \tag{28}$$

$\tilde{\boldsymbol{A}}_t$ can be expressed as a block matrix.

$$\tilde{\boldsymbol{A}}_t = \begin{bmatrix} s_t & \boldsymbol{v}_t \\ \boldsymbol{v}_t^T & \boldsymbol{A}_t^{(-i)} \end{bmatrix} \tag{29}$$

where $\boldsymbol{v}_t = [k_{i,1},\mathrm{L}\ ,k_{i,i-1},k_{i,i+1},\mathrm{L}\ ,k_{i,m}]$, $s_t = \gamma^{-1} + k_{i,i}$, and $\boldsymbol{A}_t^{(-i)}$ is the matrix after deleting the ***i-th*** row and the ***i-th*** column. Further, Eq. (30) can be obtained.

$$\tilde{\boldsymbol{A}}_t^{-1} = \beta_t^{-1}\begin{bmatrix} 1 & -\boldsymbol{v}_t(\boldsymbol{A}_t^{(-i)})^{-1} \\ -(\boldsymbol{A}_t^{(-i)})^{-1}\boldsymbol{v}_t^T & (\boldsymbol{A}_t^{(-i)})^{-1}\boldsymbol{v}_t^T\boldsymbol{v}_t(\boldsymbol{A}_t^{(-i)})^{-1} \end{bmatrix} + \begin{bmatrix} 0 & \boldsymbol{O} \\ \boldsymbol{O} & (\boldsymbol{A}_t^{(-i)})^{-1} \end{bmatrix} \tag{30}$$

where $\beta_t = s_t - \boldsymbol{v}_t(\boldsymbol{A}_t^{(-i)})^{-1}\boldsymbol{v}_t^T$.

At time $t+1$, for the training sample $(\boldsymbol{u}_{t+1},d_{t+1})$, $\boldsymbol{A}_{t+1}$ can be written as

$$\boldsymbol{A}_{t+1} = \begin{bmatrix} \boldsymbol{A}_t^{(-i)} & \bar{\boldsymbol{k}}_t \\ \bar{\boldsymbol{k}}_t^T & v_t \end{bmatrix} \tag{31}$$

$\boldsymbol{A}_{t+1}^{-1}$ can be obtained through the block matrix inversion formula. It can be represented as

$$\boldsymbol{A}_{t+1}^{-1} = \bar{\rho}_t^{-1}\begin{bmatrix} (\boldsymbol{A}_t^{(-i)})^{-1}\bar{\boldsymbol{k}}_t\bar{\boldsymbol{k}}_t^T(\boldsymbol{A}_t^{(-i)})^{-1} & -(\boldsymbol{A}_t^{(-i)})^{-1}\bar{\boldsymbol{k}}_t \\ -\bar{\boldsymbol{k}}_t^T(\boldsymbol{A}_t^{(-i)})^{-1} & 1 \end{bmatrix} + \begin{bmatrix} (\boldsymbol{A}_t^{(-i)})^{-1} & \boldsymbol{O} \\ \boldsymbol{O} & 0 \end{bmatrix} \tag{32}$$

where $\bar{\boldsymbol{k}}_t = [k_{1,t+1}, L\ , k_{i-1,t+1}, k_{i+1,t+1}, L\ , k_{m,t+1}]^T$. At this moment, the kernel function weight vector $\boldsymbol{\alpha}_t$ is updated to $\boldsymbol{\alpha}_{t+1} = \boldsymbol{A}_{t+1}^{-1}\boldsymbol{d}_{t+1}$, where $\boldsymbol{d}_{t+1} = [\hat{d}_1, L\ , \hat{d}_{i-1}, \hat{d}_{i+1}, L\ , \hat{d}_m, d_{t+1}]^T$.

According to the above analysis, the online modeling process of ISKELM proposed in this paper can be described as Fig. 1.

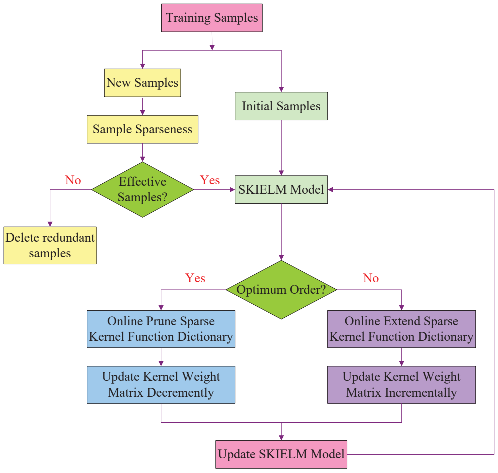
Training Samples
New Samples
Initial Samples
Sample Sparseness
No
Effective Samples?
Yes
SKIELM Model
Delete redundant samples
Yes
Optimum Order?
No
Online Prune Sparse Kernel Function Dictionary
Online Extend Sparse Kernel Function Dictionary
Update Kernel Weight Matrix Decremently
Update Kernel Weight Matrix Incrementally
Update SKIELM Model

Fig. 1 Online modeling process of ISKELM

V. Experimental Analysis

A. Online prediction of non-stationary chaotic time series

Online prediction experiments on chaotic time series is done to validate performance of the proposed method in this paper. Chaotic time series of Mackey Glass can be described as:

$$\frac{dx_{MG}(t)}{dt} = -0.1x_{MG}(t) + \frac{0.2x_{MG}(t-17)}{(1+x_{MG}^{10}(t-17))} \tag{33}$$

The sampling interval is set to $T_s = 1s$, and 1010 data points are acquired totally. Then set the embedded dimension to 10 and get 1000 experiment samples. The first 200 samples are selected as training samples to initialize the prediction model, and the remaining 800 samples are used as testing samples to simulate online data to update the initial model online. Prediction models of FM-OKELM, FL-OKELM and ISKELM are respectively established as contrast experiments to predict the time series.

In the training stage, some structure parameters of the model are initialized. Firstly set the kernel function of KELM to $k(\boldsymbol{u}_i, \boldsymbol{u}_j) = \exp(-\|\boldsymbol{u}_i - \boldsymbol{u}_j\|^2 / \theta)$, where θ is the kernel parameter. Set (γ, θ) of KELM to $(2,1)$ randomly, set the time window length L of FL-OKELM to $L \in (1, 40)$ and set the dictionary size m of ISKELM to $m \in (1, 40)$. Then, establish the prediction model of FL-OKELM and ISKELM using different values of L and m with 200 training samples, and investigate the correspondence between the training RMSE and the values of L and m in order to determine their optimal values. The relationship curves are shown in Fig. 2.

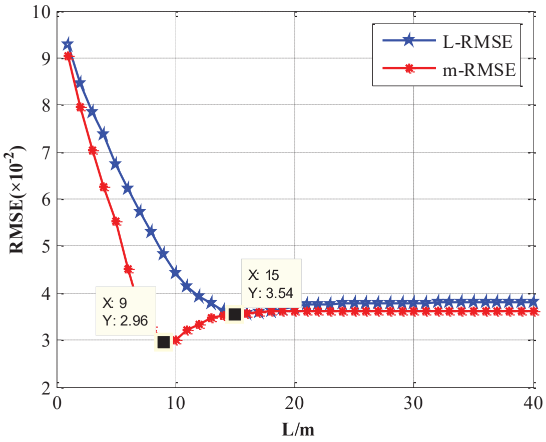

Fig. 2 Curves of prediction RMSE varying with the change of L and m for Mackey Glass time series

According to Fig. 2, the optimal values of L and m are obtained as $L = 15$ and $m = 9$, which can get the highest prediction accuracy, make the model size minimum, and avoid the model inflation. Then, search the optimal values of (γ, θ) for FL-OKELM and ISKELM through grid searching method, and the searching results are respectively $(987, 3.33)$ and $(865, 2.24)$. Finally, the optimal prediction models of FL-OKELM and ISKELM are established based on the above parameters. Similarly, the optimal values of (γ, θ) for FM-OKELM are obtained as $(513, 1.76)$ and the prediction model is established. In the testing stage, the above three models are used to predict the chaotic time series by 800 steps based on the testing samples.

Fig. 3 shows the curves of calculation time varying with prediction step within 800-step prediction for the three methods, where "+" represents effective samples for model updating in 800 samples. It can be seen that ISKELM only selects a part of samples as effective ones to update the model, and the longer the prediction step size, the higher the efficiency of sample sparseness. Therefore the online prediction time of ISKEL is distinctly reduced.

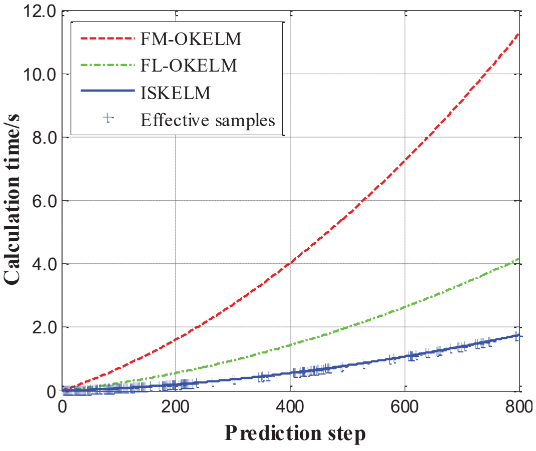

Fig. 3 Online prediction time of three methods

Root Mean Square Error (RMSE) is defined as the indicator of the prediction accuracy.

$$\mathrm{RMSE}=\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left|\hat{y}(i)-y(i)\right|^{2}} \tag{39}$$

where n is the number of samples, $y(i)$ is the actual value, and $\hat{y}(i)$ is the predicted value.

The total calculation time and prediction RMSE of various methods mentioned above are shown in Table I.

TABLE I. PREDICTION RESULTS OF MACKEY GLASS TIME SERIES

Prediction methods	Training results		Testing results	
	Training time/s	*RMSE*	*Testing time/s*	*RMSE*
FM-OKELM	1.6102	0.0501	11.2203	0.0497
FL-OKELM	0.5327	0.0354	4.1526	0.0356
ISKELM	0.2061	0.0296	1.7626	0.0297

As is shown in Table I, in the testing stage, compared with FM-OKELM and FL-OKELM, the prediction speed of ISKELM is respectively increased by 84.29% and 57.55%, and the prediction accuracy is respectively improved by 40.24% and 16.38%. This is due to the fact that FM-OKELM doesn't select effective new samples and FL-OKELM doesn't have effective strategies for deleting redundant samples and limiting the model size. In ISKELM, effective samples are selected, redundant samples are deleted and its structure matrix is incrementally updated, which distinctly improves the sample sparseness efficiency and online modeling speed .

In order to further illustrate the prediction effect of each method, the prediction curve of the first 200 test samples is shown in Fig. 4. It can be seen that the three methods can match the target sequence as a whole, but the matching effect of ISKELM is better, and its prediction RMSE can be controlled within 0.03. The prediction results of the chaotic time series show that ISKELM is more effective in prediction application.

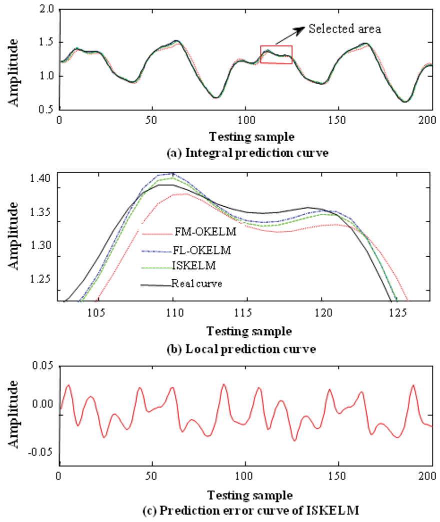

Fig. 4 Prediction curves of Mackey Glass time series

B. *Online prediction of the engine condition parameters*

Prediction experiment is conducted on 8V150ZAL-type diesel engine with the rotate speed changing from 1000 to 1900. 201 sets of engine condition parameters including speed, gear-lever shift, water temperature and exhaust gas temperature are collected under the variable speed condition, and part of them are shown in Table II.

TABLE II. PARTIAL ENGINE CONDITION PARAMETERS

Index	Exhaust gas temperature /℃	Rotation speed/(r/min)	Gear-lever shift /mm	Water temperature/℃
1	373	1258	8.3	86
2	372	1510	14.1	86
3	381	1693	11.0	86
4	390	1668	6.5	86

Set the embedded dimension to 1 to get 200 sets of samples. The first 70 samples are selected as training samples and the remaining 130 samples are used as testing samples. Taking the online prediction of exhaust gas temperature as an example, three prediction models of FM-OKELM, FL-OKELM and ISKELM are established respectively to predict the changing trend of the temperature. In the training stage, firstly set (γ,θ) of KELM randomly to $(2,1)$, set L of FL-OKELM to $L\in(1,20)$, and set m of KELM to $m\in(1,20)$. The curves of training RMSE varies with the change of L and m are shown in Fig. 5.

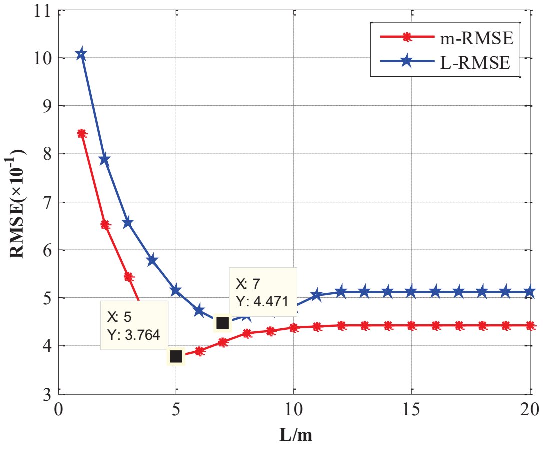

Fig. 5 Curves of prediction RMSE varying with the change of L and m for exhaust gas temperature

As can be seen from Fig. 5, the optimal values of L and m are respectively obtained as $L=7$ and $m=5$. Searching results of (γ,θ) for FL-OKELM, ISKELM and FM-OKELM are respectively $(987,3.33)$, $(865,2.24)$ and $(513,1.76)$. Finally, three optimal prediction models are established based on the above parameters. In the testing process, the above three models are used to predict the exhaust gas temperature by 130 steps online. The total calculation time and prediction RMSE of various methods mentioned above are shown in Table III.

TABLE III. THE ONLINE PREDICTION RESULTS OF ENGINE EXHAUST GAS TEMPERATURE

Prediction methods	Training results		Testing results	
	Training time/s	*RMSE*	*Testing time/s*	*RMSE*
FM-OKELM	0.7198	0.8421	1.5170	0.8420
FL-OKELM	0.2376	0.4471	0.4326	0.4473
ISKELM	0.1523	0.3764	0.2954	0.3764

As is shown in Table III, compared with the other two methods, ISKELM has the highest prediction speed and accuracy. In the testing stage, compared with FM-OKELM and FL-OKELM, the prediction speed of ISKELM is respectively increased by 80.50% and 31.72%, and the prediction accuracy is respectively improved by 48.56% and 15.81%.

The exhaust gas temperature prediction curves of three methods are shown in Fig. 6. It shows that the prediction results of ISKELM best match the real sequence of exhaust gas temperature, and the more the prediction steps, the more obvious the advantages. The results further illustrate the advantages of the proposed method in online prediction.

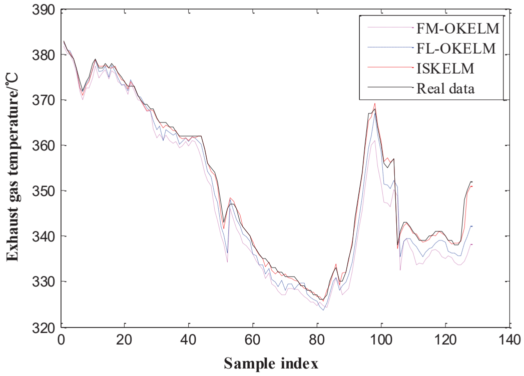

Fig. 6 Prediction curves of exhaust gas temperature for different methods

VI. Conclusions

In this paper, an online modeling method of ISKELM is proposed to solve the problems of low sample sparseness efficiency, model inflation and slow model updating speed in online prediction. ISKELM has the characteristics of sparse structure and finite size. In ISKELM, sample sparseness is adaptively implemented based on the method of instantaneous information measurement, the sparse kernel function dictionary is expanded and pruned online within the optimal order and the structure matrix is updated incrementally through the incremental and decrement learning algorithms. Therefore, the sample sparseness efficiency, online modeling speed and model generalization performance of ISKELM is improved distinctly. Experiment results show that ISKELM has better online prediction speed and accuracy compared with some existing online prediction models such as FM-OKELM and FL-OKELM.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (No. 51305454).

References

[1] C. Richard, M. Bermudez, P. Honeine, Online prediction of time series data with kernels, IEEE Transactions on Signal Processing. 57 (3) (2009) 1058-1067.

[2] T. Diethe, M. Girolami, Online learning with (multiple) kernels: a review, Neural Computation. 25 (2013) 567-625.

[3] M.X. Liu, J. Zhang, X.C. Guo, et al., Hypergraph regularized sparse feature learning, Neurocomputing. 237 (2017) 185-192.

[4] K.H. Hui, C.L. Li, L. Zhang, Sparse neighbor representation for classification, Pattern Recognition Letters. 33 (2012) 661-669.

[5] G. Yin, Y.T. Zhang, Z.N Li, et al., Online fault diagnosis method based on Incremental Support Vector Data Description and Extreme Learning Machine with incremental output structure, Neurocomputing. 128 (2014) 224-231.

[6] W.F. Liu, I. Park, J.C. Principe, An information theoretic approach of designing sparse kernel adaptive filters, IEEE Transactions on Neural Networks. 20 (12) (2009) 1950-1961.

[7] P. Honeine, Analyzing sparse dictionaries for online learning with kernels, IEEE Transactions on Signal Processing. 63 (23) (2015) 6343-6353.

[8] Y.F. Chen, J.B. Su, Sparse embedded dictionary learning on face recognition, Pattern Recognition. 64 (2017) 51-59.

[9] X.R. Zhou, C.H. Wang, Cholesky factorization based online regularized and kernelized extreme learning machines with forgetting mechanism, Neurocomputing. 174 (2016) 1147-1155.

[10] J. Liu, R.J. Stones, G. Wang, et al., Hard drive failure prediction using Decision Trees, Reliability Engineering & System Safety. 164 (1) (2017) 55-65.

[11] Q. Liu, M. Dong, W. Lv, et al., A novel method using adaptive hidden semi-Markov model for multi-sensor monitoring equipment health prognosis, Mech. Syst. Signal Process. 64–65 (2015) 217–232.

[12] Y. Chen, R.L. Sun, Y. Gao, et al., A nested-ANN prediction model for surface roughness considering the effects of c utting forces and tool vibrations, Measurement. 98 (2017) 25-34.

[13] J. Liu, E. Zio, An adaptive online learning approach for Support Vector Regression: Online-SVR-FID, Mechanical Systems and Signal Processing. 76-77 (2016) 796–809.

[14] G.B. Huang, H.M. Zhou, X.J. Ding, et al., Extreme Learning Machine for Regression and Multiclass Classification. IEEE Transactions on Systems Man and Cybernetics-Part B: Cybernetics. 42 (2) (2012) 513-529.

[15] S. Simone, C. Danilo, S. Michele, et al., Online sequential extreme learning machine with kernel, IEEE Transactions on Neural Networks and Learning Systems. 26 (9) (2015) 2214-2220.

[16] Y.T. Zhang, C. Ma, Z.N. Li, et al., Online modeling of kernel extreme learning machine based on fast leave-one-out cross-validation, Journal of Shanghai Jiaotong University. 48 (5) (2014) 641-646 (in Chinese).

[17] W. Gao, J. Chen, C. Richard, et al., Online dictionary learning for kernel LMS, IEEE Transactions on Signal Processing. 62 (11) (2014) 2765-2777.

[18] M.C. Jones, D.A. Henderson, Maximum likelihood kernel density estimation: on the potential of convolution sieves, Computational Statistics and Data Analysis. 53 (2009) 3726-3733.