An Effective Kernel Online Learning Algorithm with Sparse Update for Electronic Devices Condition Prediction

Xing Liu¹,³,*, Wei Zhang²,a, Jianyin Zhao¹,b, Deyong Huang³, c and Peng Guo³, d

¹Naval Aviation University, Yantai, China
²Military Representative Office of Naval Equipment Department in Lanzhou, Lanzhou, China
³91967 troops, Xingtai, China

*Corresponding author e-mail: xinghandeqipan@sina.com, a,hjhy1989@163.com, b13791182798@163.com, chowens@163.com, d278544783@qq.com

Abstract. In order to implement condition prediction for electronic devices, a new online learning algorithm for extreme learning machine with kernel (KELM) is proposed. For key nodes selection, a sparse dictionary with predefined size is constructed adaptively by online minimization of its cumulative coherence. Meanwhile, for model coefficients update, an improved decremental learning algorithm is presented by using elementary transformation of Gram matrix and block matrix inversion formula. The performance of proposed algorithm is compared with several well-known online sequential ELM algorithms. The simulation results show that the proposed algorithm has higher prediction accuracy and better stability. Thus it is suited to online condition monitoring of electronic devices.

1. Introduction

Condition monitoring based on time series prediction is an important manner to improve the early detection ability of electronic system faults. By the early prediction to potential degradation trend, the maintenance costs can be greatly reduced. Meanwhile, some potential accidents can also be effectively prevented [1].

As a novel learning algorithm for single layer feedforward neural networks, extreme learning machine (ELM) has been extensively studied over the years. Recently, Huang et al extended ELM by using kernel functions and presented ELM with kernel (KELM) [2]. The experimental and theoretical analyses show that KELM can produce similar or better performance than ELM.

However, many condition monitoring applications require online, adaptive and real-time operation. Such requirements pose a serious challenge, since aforementioned methods operate in batch model [3]. In order to improve the training efficiency, online learning algorithms have been proposed to meet the actual application demand, such as ReOS-ELM [4], KB-IELM [5].

Unfortunately, a noteworthy drawback in kernel-based online learning methods is that the size of Gram matrix equals the number of training data. So the computational complexity and storage requirement usually grow superlinearly with the number of training samples. In order to overcome this issue, the sparsification methods have been designed to curb the growing kernel function number [6]. FOKELM [7] adopted sliding time window to tackle the matrix expansion of KB-IELM by deleting
the oldest sample when new samples occur continually. It can obtain a compact solution which largely depends on the latest k observed samples [8]. KOS-ELM [9] adopted approximate linear dependency (ALD) criterion used in the kernel adaptive filters field to obtain sparse solution. A major criticism that can be made of ALD criterion is that it leads to costly operations with quadratic complexity in the cardinality m of the dictionary [10]. However, limited studies have been made regarding online KELM learning, which up to now remains an open problem in the ELM theory.

The aim of this paper is to seek a new online learning strategy of KELM for electronic devices condition prediction. In our proposed method, for the sparse dictionary selection, a new criterion based on the cumulative coherence of a dictionary [11] is used. A compact dictionary with predefined size can be selected adaptively by minimizing its cumulative coherence. For model coefficients update, an improved decremental learning algorithm is proposed by using elementary transformation of Gram matrix and block matrix inversion formula to moderate computational complexity at each iteration.

The rest of this paper is organized as follows. In Section 2, we introduce the preliminaries for this paper. The sparsification rule and the kernel weight update scheme are given in Section 3. In Section 4, the proposed algorithms are evaluated by both simulation and real-world data. The conclusion is drawn in Section V.

2. Preliminaries

2.1. Extreme learning machine with kernel

For a given training data set 1{(x_i, y_i)}_{i=1}^N \in \mathbb{R}^d \times \mathbb{R}^N, where x_i is an d-dimensional input vector and y_i is the corresponding scalar observation. The optimization problems for ELM can be mathematically formulated as

Minimize:  
L = \frac{1}{2} |\beta|^2 + c \frac{1}{2} \sum_{i=1}^N \xi_i^2

Subject to:  
h(x_i)\beta = y_i - \xi_i

where \beta = [\beta_1, \ldots, \beta_L]^T is the output weights vector; h(x_i) is a feature mapping from the d-dimensional input space to the L-dimensional hidden-layer feature space; \xi = [\xi_1, \ldots, \xi_N]^T is the training error vector. c is the regularization parameter.

The Karush-Kuhn-Tucker (KKT) is used as optimality conditions. \beta Can be obtained as follows.

\beta = H^T (c^{-1} I + HH^T)^{-1} Y

(1)

Where Y = [y_1, \ldots, y_N]^T indicates the target value vector. H = [h(x_1), \ldots, h(x_N)]^T Is the mapping matrix for all input samples? Mercer’s conditions can be applied on ELM. The kernel matrix is defined as \Omega = HH^T and \Omega(i, j) = h(x_i) \cdot h(x_j) = k(x_i, x_j). Then the output function of KELM can be obtained.

f(\cdot) = [k(\cdot, x_1), \ldots, k(\cdot, x_N)](c^{-1} I + \Omega)^{-1} Y

= \sum_{i=1}^N \theta(i)k(\cdot, x_i)

(2)

where \theta = [\theta(1), \ldots, \theta(N)]^T is kernel weight coefficients.

2.2. Motivation

The main drawback in (2) is that the model order equals the size of the training samples set. This leads us to consider designing a sparse criterion in the learning process to prevent the size of kernel functions being too large.
Supposing there is a sparse dictionary $D_t = \{c_i(j)\}_{j=1}^m$ with $m$ key nodes at the $t$-th training iteration, thus

$$\hat{f}_t(x) = \sum_{j=1}^m \theta(j)k(x, c_j(j))$$

Equation (3) learns a kernel machine in the subspace spanned by the selected key nodes. Obviously, two key issues need to be solved at each iteration, i.e., the selection of key-nodes $\{c_i(j)\}_{j=1}^m$ and the updating of the kernel weight coefficients $\{\theta(j)\}_{j=1}^m$.

3. Online sparse extreme learning machine with kernel

3.1. Incremental learning procedure

Supposing that $D_{t-1} = \{c_i, (j)\}_{j=1}^m$ is the dictionary at time $t-1$, the Gram matrix of $D_{t-1}$ is $\Omega_{t-1}$, and the predefined size of dictionary is $m$. When $m_{t-1}$ less than $m$, the dictionary is is expanded with the new kernel function. Moreover, supposing that $k(.,c(i))$ is unit-norm kernel, i.e., $\forall c(i), k(c(i),c(i))=1$. Let $A_{t-1} = I/c + \Omega_{t-1}$, then we have

$$A_{t-1} = \begin{bmatrix} c^{-1} + k_{i,1} & \cdots & k_{i,m} \\ \vdots & & \vdots \\ k_{m,1} & \cdots & c^{-1} + k_{m,m} \end{bmatrix}$$

(4)

At time $t$, the new sample $(x_t, y_t)$ arrives; thus

$$A_t = \begin{bmatrix} A_{t-1} & V_t \\ V_t^T & v_t \end{bmatrix}$$

(5)

Where $V_t = [k_{i,1}, k_{i,2}, \cdots, k_{i,m}]^T$, $v_t = c^{-1} + 1$.

Using the block matrix inverse lemma, we have

$$A_t^{-1} = \begin{bmatrix} A_{t-1}^{-1} + A_{t-1}^{-1}V_t \rho_t^{-1} V_t^T A_{t-1}^{-1} & -A_{t-1}^{-1} V_t \rho_t^{-1} \\ -\rho_t^{-1} V_t^T A_{t-1}^{-1} & \rho_t^{-1} \end{bmatrix}$$

(6)

Where $\rho_t = v_t - V_t^T A_{t-1}^{-1} V_t$.

So kernel weight coefficients can be updated by (7).

$$\theta_t = A_t^{-1} Y_t$$

(7)

Where $Y_t = [y_1, \cdots, y_{t-1}, y_t]^T$.

3.2. Online key nodes selection

If $D = \{c(j)\}_{j=1}^m$ is a dictionary with $m$ members, the Gram matrix of $D$ is $\Omega$, then the cumulative coherence of $D$ is defined as (8).

$$\mu_\ell(\Omega) = \max_{1 \leq i \leq m} \sum_{1 \leq j \leq m} |k(c(i), c(j))|$$

(8)
where $\mu_i(\mathbf{\Omega})$ denotes the maximum sum of inner products between a kernel function $k(\cdot, c(i))$ and the other $m-1$ kernel functions. $\mu_i(\mathbf{\Omega})$ provides a criterion to determine the overall similarity of the dictionary members. For a dictionary with predefined size, a smaller cumulative coherence means the members are more dissimilar to each other.

If $m_{t-1} = m$ at time $t-1$, then the new training sample $x_t$ is possible to be the dictionary member by means of replacing one of the dictionary members if it can provide smaller cumulative coherence. Otherwise, the dictionary keeps unchanged.

Let $\mathbf{D}_t = \{d_{t,i}, x_t\}$ denote the dictionary with all potential key nodes, $\mathbf{\Omega}_t$ is the Gram matrix of $\mathbf{D}_t$, then we have

$$
\mathbf{\Omega}_t = \begin{bmatrix}
\mathbf{\Omega}_{t-1} & \mathbf{k}_t \\
\mathbf{k}_t^T & 1
\end{bmatrix}_{(m+1) \times (m+1)}
$$

Where $\mathbf{k}_t = [k(c_{t,1}(1), x_t), \cdots, k(c_{t,1}(m), x_t)]^T \in \mathbb{R}^{m \times 1}$. Let

$$
\mathbf{E} = \begin{bmatrix}
0 & 1 & \cdots & 1 \\
\vdots & \vdots & \ddots & \vdots \\
1 & 1 & \cdots & 0
\end{bmatrix}_{(m+1) \times (m+1)}
$$

Which is a matrix with its diagonal elements being 0 and all its off-diagonal elements being 1.

Let $\mathbf{S} = \mathbf{\Omega}_t \times \mathbf{E}$, we have $S_{t,i} = \sum_{1 \leq j \leq m+1} k(c(i), c(j))$. And let $S_u = S - S_d$, where $S_d = \text{diag(diag(S))}$. Supposing that the $i$-th member should be deleted from $\mathbf{D}_t$, and the Gram matrix of the new dictionary is $\mathbf{\Omega}_t^{u}$, then its cumulative coherence can be obtained according to (10).

$$
\mu_i(\mathbf{\Omega}_t^{u}) = \max(S_u(:,i)) - 1 = \|S_u(:,i)\|_\infty - 1
$$

We aim to select the $m$ key nodes among the $m+1$ potential nodes by removing the $i$-th node which can assure a minimum cumulative coherence. The index $i$ can be determined according to (11).

$$
i = \arg\min_j \mu_j(\mathbf{\Omega}_t^{u}) = \arg\min_j \|S_u(:,j)\|_\infty
$$

If $i=m+1$, the dictionary keeps unchanged. If $1 \leq i \leq m$, the $i$-th member of dictionary is replaced by the new one.

3.3. Improved decremental learning procedure

When $m_{t-1} = m$ at time $t-1$, supposing that $i$ is the optimal index searched by (11), and $1 \leq i \leq m$. In order to move the $i$-th row and the $i$-th column of $\mathbf{A}_{t-1}$ to the first row and first column, elementary transformation is performed to $\mathbf{A}_{t-1}$. The transformation process can be formulated as (12).

$$
\mathbf{A}_{t-1}^{'} = \mathbf{P} \mathbf{A}_{t-1} \mathbf{Q}
$$

Where $\mathbf{P}$ and $\mathbf{Q}$ are two $m$-order elementary matrix, i.e,
Because $P$ and $Q$ are orthogonal matrices, $P = Q^{-1}$ and $Q = P^{-1}$. Without loss of generality, we have

$$
(A'_{-i})^{-1} = (PA_{-i}Q)^{-1} = PA_{-i}^{-1}Q
$$

(13)

Moreover, $A'_{-i}$ can be written as the block matrix, i.e.

$$
A'_{-i} = \begin{bmatrix}
\overline{v}_{r-1} & \overline{F}_{r-1} \\
\overline{F}_{r-1}^T & \overline{A}_{r-1}
\end{bmatrix}
$$

(14)

Where $\overline{F}_{r-1} = [k_{1,i}, \ldots, k_{i,r,x-1}, k_{i,r,y}, \ldots, k_{i,m}]^T$ and $\overline{v}_{r-1} = c^{-1} + 1$.

Using the block matrix inverse formula, we have

$$
(A'_{-i})^{-1} = \begin{bmatrix}
\overline{P}_{r-1}^{-1} & -\overline{P}_{r-1}^{-1}\overline{F}_{r-1}^T\overline{A}_{r-1}^{-1} \\
-\overline{A}_{r-1}^{-1}\overline{F}_{r-1}\overline{P}_{r-1}^{-1} & \overline{A}_{r-1}^{-1} - \overline{A}_{r-1}^{-1}\overline{F}_{r-1}\overline{A}_{r-1}^{-1}\overline{F}_{r-1}^T\overline{A}_{r-1}^{-1}
\end{bmatrix}
$$

(15)

Where $\overline{P}_{r-1} = \overline{v}_{r-1} - \overline{F}_{r-1}\overline{A}_{r-1}^{-1}\overline{F}_{r-1}^T$.

According to (13), let $W_{r-1} = PA_{-i}^{-1}Q$. So $(A'_{-i})^{-1}$ can be further written as following form.

$$
(A'_{-i})^{-1} = \begin{bmatrix}
W_{r-1}^{(1,1)} & W_{r-1}^{(1,2:)} \\
W_{r-1}^{(2:1,1)} & W_{r-1}^{(2:1,2:)}
\end{bmatrix}
$$

(16)

Combined (15) with (16), we have

$$
\overline{A}_{r-1}^{-1} = W_{r-1}^{(2:1,2:)} - \frac{W_{r-1}^{(2:1,1)} \times W_{r-1}^{(1,2:)}}{W_{r-1}^{(1,1)}}
$$

(17)

At time $t$, new sample $(x_i, y_i)$ arrives. When compute matrix $A_{r-1}$, we directly replace $A_{r-1}^{-1}$ with $\overline{A}_{r-1}^{-1}$, i.e.,

$$
A_{r-1}^{-1} = \begin{bmatrix}
\overline{A}_{r-1}^{-1}(I + V_i\rho_i^{-1}V_i^T \overline{A}_{r-1}^{-1}) & -\overline{A}_{r-1}^{-1}V_i\rho_i^{-1} \\
-\rho_i^{-1}V_i^T \overline{A}_{r-1}^{-1} & \rho_i^{-1}
\end{bmatrix}
$$

(18)

Where $V_i = [k_{1,i}, \ldots, k_{i,1,x}, k_{i,1,y}, \ldots, k_{i,m}]^T$, $v_i = c^{-1} + 1$ and $\rho_i = v_i - V_i^T \overline{A}_{r-1}^{-1}V_i$.

Obviously, the improved algorithm avoids the computational burden caused by recalculating matrix $A_{r-1}^{-1}$. Compared with the method in [7], the proposed algorithm can delete any sample in the
dictionary, which is much more flexible. Finally, kernel weight coefficients can be obtained by (7), where
\[ Y_{t} \]

4. Experimental analysis
In this section, we will give an examples to demonstrate the effectiveness of the proposed method. In
the experiment, the prediction accuracy is evaluated in terms of the testing root mean square error (RMSE),
testing maximal absolute prediction error (MAPE) and testing mean relative prediction error (MRPE).
\[
\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (\hat{y}(i) - y(i))^2}
\]
\[
\text{MAPE} = \max_{i=1, \ldots, n} \left| \frac{\hat{y}(i) - y(i)}{y(i)} \right|
\]
\[
\text{MRPE} = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{\hat{y}(i) - y(i)}{y(i)} \right|
\]

The proposed algorithm is denoted as SKOELM for simplicity. Its details are summarized as follows.

| Algorithm: SKOELM |
|-------------------|
| **Model parameters:** \( m, \sigma \) and \( c \) |
| **Initialization:** set \( t=1 \), \( D_{0}=(\varepsilon_{(j)})_{m+1}^{m} \); compute \( A_{t}^{1}=(I+cD_{t})^{-1} \) |
| 1 New training sample \( x \) arrives. |
| 2 If \( m_{s}<m \) then update \( m_{s}=m_{s}+1 \) and \( D_{s}=D_{s} \cup \{x\} \); |
| 3 Using (6) to update \( A^{1} \); using (7) to update \( \theta \); |
| 4 Else |
| 5 Using (9) to compute \( \Omega \); using (10) to compute \( \mu_{\Omega}\); |
| 6 Using (11) to determine removable vector index \( i \); |
| 7 If \( i=m+1 \), then \( D_{s}=D_{s} \cup \{\varepsilon_{(i)}\} \); |
| 8 Else |
| 9 Update \( D_{s}=D_{s} \cup \{\varepsilon_{(i)}\} \); |
| 10 Using (13) to compute \( w_{i} \); using (17) to compute \( \lambda_{i} \); |
| 11 Using (18) to update \( A^{+} \); using (7) to update \( \theta_{i} \); |
| 12 End if |
| 13 End if |
| 14 Using (3) to compute \( \hat{y}_{i} \) |
| **Output** \( \hat{y}_{i} \); \( t=t+1 \), go to step 1 |

In order to demonstrate the performance of the proposed method, it is compared with four state-of-
art ELM methods: ReOS-ELM[4], KB-IELM[5], FOKELM[7] and KOS-ELM[9]. In all experiments,
Gaussian kernel is applied as the kernel function. Sigmoid function is applied as additive function in
ReOS-ELM. Moreover, optimal parameters are selected by grid search. All simulation studies are
conducted in MATLAB R2010a environment on a Windows XP PC with core i3 CPU and 2GB RAM.

4.1. Nonlinear dynamical time series prediction
The nonlinear dynamical time series are derived from the following nonlinear difference equation.
\[
y_{s} = \left[0.8 - 0.5 \exp(-y_{s-1}^2)\right]y_{s-1} + \left[0.3 + 0.9 \exp(-y_{s-1}^2)\right]y_{s-2} + 0.1 \sin(y_{s-1} \pi)
\]
The data are generated with an initial condition \((0.1, 0.1)\). The number of sample points is 600, the input is set as \(u = (y_{i-1}, y_{i-2}, y_{i-3})\) to estimate \(y_i = f(u_i)\). In this example, the selected parameters are depicted in Table 1.

| Table 1. SELECTED PARAMETERS FOR NONLINEAR DYNAMICAL TIME SERIES |
|---------------------------------------------------------------|
| Methods          | \(c\) | \(\sigma\) | \(\text{else}\) |
| ReOS-ELM         | 5     | ---        | \(L=50\) |
| KB-IELM          | \(5e^{-2}\) | 1   | ---    |
| FOKELM           | \(5e^{-2}\) | 1   | \(\mu=50\) |
| KOS-ELM          | \(5e^{-2}\) | 1   | \(\delta=5e^{-4}\) |
| SKOELM           | \(5e^{-2}\) | 1   | \(\mu=50\) |

Table 2 shows the prediction results of different algorithms when prediction step size is equal to 200. We can see that, kernel-based methods obviously can improve the prediction accuracy. Compared with KB-IELM, FOKELM and KOS-ELM, the testing RMSEs of our algorithm decrease by 15.8%, 3.0% and 33.3% respectively.

| Table 2. SIMULATION RESULTS OF THE NONLINEAR DYNAMICAL TIME SERIES |
|---------------------------------------------------------------|
| Methods | Training | Testing |
|         | Tr-time | RMSE | Te-time | RMSE | MAPE | MRPE |
| ReOS-ELM | 0.3438 | 0.0231 | 0.0005 | 0.0181 | 0.0309 | 0.0132 |
| KB-IELM | 3.0316 | 0.0116 | 0.0127 | 0.0076 | 0.156 | 0.0028 |
| FOKELM | 0.1164 | 0.0279 | 0.0012 | 0.0066 | 0.0166 | 0.0038 |
| KOS-ELM | 0.1565 | 0.0089 | 0.0008 | 0.0096 | 0.0253 | 0.0034 |
| SKOELM | 0.1437 | 0.0282 | 0.0007 | 0.0064 | 0.0147 | 0.0026 |

4.2. Audio amplifier online condition prediction
Amplifier is an indispensable part of modern electronic systems. In the amplifying circuit, the changes of any parameter, such as mains voltage fluctuation, components aging and semiconductor component temperature drift, will result in output containing some errors. In this section, a fifty-fold high-powered audio amplifier is used as the research object. The experimental principle diagram is shown in Fig.1.

![Figure 1. Experimental principle diagram](image)

Because the temperature of audio amplifier will slowly rise with time changing, the output power will produce some nonlinear fluctuations. In this experiment, the current moment input power, the current moment temperature and the previous moment output power are used as input data; the next moment output power is used as output data. Then FOKELM, KOS-ELM and the proposed method
are online trained by using the training samples one by one. Finally, the system output power is predicted respectively based on three trained models. The selected parameters are depicted in Table 3.

Table 3. SELECTED PARAMETERS FOR AUDIO AMPLIFIER CONDITION PREDICTION

| Methods   | c    | σ    | else |
|-----------|------|------|------|
| FOKELM    | 5e-4 | 1e4  | \(m=5e^1\) |
| KOS-ELM   | 5e-4 | 1e-5 | \(\delta=1e^{-5}\) |
| SKOELM    | 5e-2 | 1e1  | \(m=5e^1\) |

Table 4 shows the prediction results of different algorithms when prediction step size is equal to 200. We can see that proposed method obviously improves the prediction accuracy compared with FOKELM and KOS-ELM. The testing RMSEs decrease by 13.7% and 4.1% respectively. Although the training time of the proposed method is the longest among three sparse methods, the testing time, however, is similar to other methods.

Table 4. THE CONDITION PREDICTION RESULTS OF AUDIO AMPLIFIER

| Methods   | Training | Testing |
|-----------|----------|---------|
|           | Tr-time | RMSE    | Te-time | RMSE | MAPE | MRPE |
| FOKELM    | 0.1391  | 3.9359  | 0.0010  | 0.8384 | 2.4253 | 0.0112 |
| KOS-ELM   | 0.0574  | 0.5869  | 0.0007  | 0.7237 | 2.1547 | 0.0094 |
| SKOELM    | 0.2699  | 0.7819  | 0.0008  | 0.6942 | 1.9311 | 0.0092 |

The learning curves of three algorithms are depicted in Fig.2, where Y-axis denotes testing mean square error (MSE). We can see that, the proposed algorithm has a better performance than other algorithms. The main superiority of our algorithm is that it converge to a more accurate stage when the training step become large.

Figure 2. Learning curves of different algorithms for audio amplifier condition monitoring data

Fig.3 shows that the prediction results and absolute prediction error (APE) curves when the prediction step size is 200. From Fig.3 (a), we can see that FOKELM’s prediction errors will gradually increase. From Fig.3 (b) and Fig.3 (c), KOS-ELM and SKOELM can forecast the output trend well, but the prediction accuracy of the proposed method is higher than KOS-ELM in the local details.
5. Conclusion

In this paper, an online variant of extreme learning machine with kernel is proposed for electronic system online condition prediction. In order to avoid the redundant computation and achieve the model online sparsification, sliding time window is used as basic online modeling strategy, and the cumulative coherence measure is adopted to select key nodes. When the dictionary size exceeds time window length, using the improved decremental learning algorithm to delete old sample in dictionary and update current model with new sample.

Our proposed algorithm is applied to audio amplifier online condition monitoring, the experiment results indicate that the proposed algorithm has higher prediction accuracy and better stability compared with several well-known online sequential ELM algorithms.

Acknowledgments

The research was supported by National Science Foundation of China under Grant Nos. 61571454.

References

[1] Z. Tian, C. Qian, B. Gu, L. Yang, F. Liu, “Electric vehicle air conditioning system performance prediction based on artificial neural network,” Applied Thermal Engineering, vol.89, 2015, pp.101-104, doi:10.1016/j.applthermaleng.2015.06.002.
[2] G.B. Huang, H. Zhou, X. Ding, R. Zhang, “Extreme learning machine for regression and multiclass classification,” IEEE Transactions on Systems, Man and Cybernetics-Part B: Cybernetics, vol.42, APRIL 2012, pp.513-529, doi:10.1109/TSMCB.2011.2168604.
[3] S.Y. Wong, K.S. Yap, H.J. Yap, S.C. Tan, “A truly online learning algorithm using hybrid fuzzy ARTMAP and online extreme learning machine for pattern classification,” Neural Process Lett, vol.42, 2015, pp.585-602, doi:10.1007/s1063-014-9374-5.
[4] H.T. Huynh, Y.G. Won, “Regularized online sequential learning algorithm for single-hidden layer feedforward neural networks,” Pattern Recognition Letters, vol.32, 2011, pp.1930-1935, doi:10.1016 /j.patrec.2011.07.016.
[5] L. Guo, J.H. Hao, M. Liu, “An incremental extreme learning machine for online sequential learning problems,” Neurocomputing, vol.128, 2014, pp.50-58, doi:10.1016/j.neucom.2013.03.055.
[6] H.J Fan, Q. Song, “A sparse kernel algorithm for online time series data prediction,” Expert Systems with Applications, vol.40, 2013, pp.2174-2181, doi:10.1016/j.eswa.2012.10.046.
[7] X.R. Zhou, Z.J Liu, C.X. Zhu, “Online regularized and kernelized extreme learning machines with forgetting mechanism,” Mathematical Problems in Engineering, vol.2014, 2014, pp.1-11, doi:10.1155/2014/938548.
[8] X.R. Zhou, C.S. Wang, “Cholesky factorization based online regularized and kernelized extreme learning machines with forgetting mechanism,” Neurocomputing, vol.174, 2016, pp.1147-1155, doi: 10.1016/j.neucom.2015.10.033.
[9] S. Scardapance, D. Comminiello, M. Scarpiniti, A. Uncini, “Online Sequential extreme learning machine with kernel,” IEEE Transactions on Neural Networks and Learning Systems, vol.26, Sep. 2015, pp.2214-2220, doi:10.1109/TNNLS.2014.2382094.

[10] C. Richard, J. Bermudez, P. Honeine, “Online prediction of time series data with kernels,” IEEE Transactions on Signal Processing, vol.57, MARCH, 2009, pp.1058-1067, doi:10.1109/TSP.2008.2009895.

[11] H.J. Fan, Q. Song, S.B. Shrestha, “Kernel online learning with adaptive kernel width,” Neurocomputing, vol.175, 2016, pp.233-242, doi: 10.1016/j.neucom.2015.10.055.