Online Sequential Extreme Learning Machine with Generalized Regularization and Adaptive Forgetting Factor for Time-Varying System Prediction

Wei Guo,1,2 Tao Xu,2,3 Keming Tang,1 Jianjiang Yu,1 and Shuangshuang Chen1

1 School of Information Engineering, Yancheng Teachers University, Yancheng 224002, China
2 School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China
3 School of Computer Science and Technology, Civil Aviation University of China, Tianjin 300300, China

Correspondence should be addressed to Wei Guo; weigu031@163.com

Received 10 December 2017; Revised 27 March 2018; Accepted 26 April 2018; Published 31 May 2018

1. Introduction

In many real world applications such as financial data analysis, industrial process monitoring, weather forecast, and customer behavior prediction, the samples arrive successively in the form of a data stream [1]. Moreover, these systems are usually nonstationary and exhibit time-varying behaviors, that is, the underlying distribution or the trend of the data changes over time [2]. In such situations, an online learning algorithm would be a more appropriate selection over batch learning algorithm since the online learner can be incrementally updated to acquire new knowledge or to suit changing patterns in evolving data stream and it does not require retraining whenever a new sample is received [3–5].

Among the existing online learning algorithms, the online sequential extreme learning machine (OSELM) [6] is a very famous one for its efficient and powerful learning abilities. OSELM originates from the batch extreme learning machine (ELM) [7] algorithm which has been demonstrated to be extremely fast with good generalization performance. As a sequential implementation of ELM, OSELM is capable of learning the training data one-by-one or chunk-by-chunk with fixed or varying length, the output weights of which...
are constantly updated by adopting a recursive least squares (RLS) algorithm. Compared with other popular online learning algorithms, OSELM can provide better generalization performance at a much faster learning speed. Depending upon these advantages, OSELM has been successfully applied in the field of system modeling and prediction, such as online nonlinear system identification [8, 9], ship roll motion prediction [10], consumer sentiments prediction [11], and time series prediction [12–15]. Despite an excellent online learning algorithm, the OSELM may still suffer from a drawback of instability due to the potential ill-conditioned matrix inversion, and its stability and generalization performance could be greatly influenced once the autocorrelation matrix of the hidden layer output matrix is singular or ill-conditioning. Regularization technique is an effective way to cope with the ill-posed problem, and a regularized OSELM (R-OSELM) based on the biojective optimization with Tikhonov regularization was proposed in [16]. The R-OSELM successfully overcomes the potential ill-posed problem and tends to provide good generalization performance and stability, and it has become a practical online modeling method in real applications.

In time-varying environment, data evolve over time; accordingly the recent samples are more relevant to the actual target concept and the out-of-date samples may be useless in addition to introducing noise [17]. Therefore, we should consider the increase in the importance of recent samples in contrast with the distant ones and reduce the adverse effects of the outdated data. In order to deal with the time-varying scenario, some improved OSELM algorithms incorporating sliding window strategy have been proposed, such as FOS-ELM [18], OS-ELMK [19], FORELM, and FOKELM [20]. By using the sliding window strategy, these algorithms can discard the outdated data in the learning process so as to reduce their adverse effects to the subsequent online learning and prediction. Nevertheless, since an extra incremental learning procedure is required to remove the oldest sample from the sliding window whenever a new sample is added, the sliding window-based OSELM algorithms are more complex and more computation times are required compared with the original OSELM algorithm. Moreover, in these algorithms all the data samples within the sliding window are still equally treated.

Apart from the sliding window strategy, forgetting factor (FF) method is another more efficient and flexible means to reflect the variations of the observable data in a time-varying environment by assigning a different weight to each data sample according to their time order, and it has been widely applied in the adaptive filtering theory [21]. Analogously, the concept of FF has also been introduced into the OSELM for dynamic system estimation and prediction [22–24]. In [22], a low complexity adaptive forgetting factor OSELM (LAFF-OSELM) algorithm was proposed for time-varying nonlinear system identification, and it was found that the proposed algorithm can deal with the variation effectively by adaptive forgetting mechanism. In [23], the authors presented a new variable forgetting factor OSELM using the directional FF method (DFF-OSELM), which achieved superior prediction performance in industrial applications when compared to the OSELM algorithm. However, in both LAFF-OSELM and DFF-OSELM, the intrinsic ill-posed problem is not considered, so they may encounter a potential instability. To provide a more stable and reliable modeling approach for time-varying system, the FF method and the regularization technique were simultaneously incorporated into the OSELM, and then the FR-OSELM algorithms were developed [24]. With the help of the FF, the FR-OSELM can maintain a good tracking ability for the time-varying system; meanwhile the stability of the algorithm is also enhanced by using the regularization approach. Nevertheless, since the FR-OSELM adopts a special exponential forgetting regularization, its regularization effect will fade gradually until up to nullification with time passing by, and ultimately the FR-OSELM may still encounter the similar or more serious problem of instability as the OSELM due to the probable operation of ill-conditioned matrix inversion. That is to say, the FR-OSELM is stable and workable only within limited period.

In this paper, a novel OSELM with generalized regularization and adaptive forgetting factor (AFGR-OSELM) is proposed for the modeling and prediction of the time-varying system. Our contributions are as follows. (i) Different from the previous FR-OSELM algorithms which use a special exponential forgetting regularization, a more generalized $l_2$ regularization is employed instead in our AFGR-OSELM. The built-in generalized regularization approach makes the AFGR-OSELM have a constant regularization effect without fading in the whole learning process; as a result the potential ill-posed problem of the algorithm can be completely overcome and a persistent stability can be maintained in all online learning stages. (ii) Moreover, to better track the dynamic behaviors of the complex time-varying system, a new adaptive FF method under the condition of the generalized regularization is derived and incorporated in the AFGR-OSELM algorithm. In the sequential learning process of the AFGR-OSELM, the FF can be adaptively tuned in a recursive way to timely suit the dynamic changes of the time-varying system; therefore a desirable prediction performance of the algorithm can be anticipated. (iii) The effectiveness and practicability of the AFGR-OSELM algorithm are evaluated with both artificial and real world data sets, and the results show that the proposed algorithm can obtain more superior prediction accuracy with better stability compared with other representative models.

The rest of the paper is organized as follows. The related works including OSELM, R-OSELM, and FR-OSELM are revisited in Section 2. Section 3 presents the details of the proposed AFGR-OSELM. In Section 4, the performances of the proposed AFGR-OSELM are demonstrated by six illustrative examples. Finally, the conclusions and future work are given in Section 5.

2. OSELM and Related Algorithms

In this section, we provide a brief review of the related works including OSELM, R-OSELM, and FR-OSELM. For the sake of simplicity, all these OSELM-related algorithms are considered for regression with single output.
2.1. OSELM. OSELM [6] is a sequential implementation of the batch ELM algorithm [7] which is originally developed from the study of single hidden layer feedforward networks (SLFNs). For \( N \) arbitrary distinct samples \((x_i, t_i) \in \mathbb{R}^d \times \mathbb{R}^1\), SLFNs with \( n \) hidden nodes are mathematically modeled as

\[
\sum_{i=1}^{n} \beta_i g_i(x_i) = \sum_{i=1}^{n} \beta_i G(a_i, b_i, x_i), \quad j = 1, 2, \ldots, N, \tag{1}
\]

where \( a_i \) is the weight vector connecting the \( i \)th hidden node and the input nodes, \( b_i \) is the threshold of the \( i \)th hidden node, and \( \beta_i \) is the weight vector connecting the \( i \)th hidden node and the output nodes; \( g_i(x_i) = G(a_i, b_i, x_i) \) denotes the output function of the \( i \)th hidden node with respect to input \( x_i \).

That SLFNs can approximate these \( N \) samples with zero error which means that there exist \((a_i, b_i)\) and \( \beta_i \) such that

\[
\sum_{i=1}^{n} \beta_i G(a_i, b_i, x_i) = t_j, \quad j = 1, 2, \ldots, N. \tag{2}
\]

The above \( N \) equations can be written compactly as

\[
H\beta = T, \tag{3}
\]

where

\[
H = \begin{bmatrix}
\vdots \\
h_1 \\
\vdots \\
h_N
\end{bmatrix}
= \begin{bmatrix}
G(a_1, b_1, x_1) & \cdots & G(a_n, b_n, x_1) \\
\vdots & \ddots & \vdots \\
G(a_1, b_1, x_N) & \cdots & G(a_n, b_n, x_N)
\end{bmatrix}_{N \times n},
\]

\[
\beta = \begin{bmatrix}
\beta_1^T \\
\vdots \\
\beta_n^T
\end{bmatrix}_{1 \times N},
\]

\[
T = \begin{bmatrix}
t_1^T \\
\vdots \\
t_N^T
\end{bmatrix}_{1 \times N}
\]

\( H \) is called the hidden layer output matrix of the network; the \( j \)th row of \( H \) is the output vector of the hidden layer with respect to input \( x_j \), and the \( i \)th column of \( H \) is the \( i \)th hidden node’s output vector with respect to inputs \( x_1, x_2, \ldots, x_N \).

It has been mathematically proved in [7] that SLFNs with random hidden nodes have the universal approximation capability, the hidden nodes can be randomly generated independent of the training data and remain fixed; then the hidden layer output matrix \( H \) is a constant matrix. Thus, training an SLFN is simply equivalent to finding a least squares solution \( \hat{\beta} \) of the linear system \( H\beta = T \):

\[
\|H\hat{\beta} - T\| = \min_{\beta} \|H\beta - T\|, \tag{5}
\]

where \( \| \cdot \| \) is a norm in Euclidean space. The ELM adopts the smallest norm least squares solution of the above linear system as the output weights; that is,

\[
\hat{\beta} = H^T T, \tag{6}
\]

where \( H^T \) is the Moore-Penrose generalized inverse of matrix \( H \). If \( H^T H \) is nonsingular, then (6) can be further written as

\[
\hat{\beta} = (H^T H)^{-1} H^T T. \tag{7}
\]

Furthermore, to suit the online learning scenario, an online version of ELM named OSELM is implemented to learn the training samples successively and incrementally. The learning procedure of OSELM consists of an initialization phase and a following sequential learning phase, and the one-by-one OSELM is summarized as follows.

In initialization phase, given an initial training set \( \Omega_{k-1} = \{(x_j, t_j) \mid j = 1, \ldots, k-1\} \), according to (7), the initial output weights are given by

\[
\hat{\beta}_{k-1} = \mathbf{P}_{k-1} H_{k-1}^T T_{k-1}, \tag{8}
\]

where \( \mathbf{P}_{k-1} = (H_{k-1}^T H_{k-1})^{-1}, \quad H_{k-1} = [h_{k-1}^T \ h_{k-2}^T \ \cdots \ h_{k-1}^T]^T \), and \( T_{k-1} = [t_1 \ t_2 \ \cdots \ t_{k-1}]^T \).

In the sequential learning phase, the RLS algorithm is used to update the output weights in a recursive way. Suppose now that we receive another sample \((x_k, t_k)\); the corresponding partial hidden layer output matrix is calculated as \( h_k = [G(a_1, b_1, x_k) \ \cdots \ G(a_n, b_n, x_k)] \), and then the output weights update equations are determined by

\[
\begin{align*}
\mathbf{P}_k &= \mathbf{P}_{k-1} - \frac{\mathbf{P}_{k-1} h_k^T h_k \mathbf{P}_{k-1}}{1 + h_k^T \mathbf{P}_{k-1} h_k}, \\
\hat{\beta}_k &= \hat{\beta}_{k-1} + \frac{1}{\mathbf{P}_k} h_k (t_k - h_k \hat{\beta}_{k-1}).
\end{align*} \tag{9}
\]

As seen from (9), the output weights of OSELM are recursively updated based on the intermediate results in the last iteration and the newly arrived data, which can be discarded immediately as soon as they have been learnt, so the computation overhead and the memory requirement of the algorithm are greatly reduced. The above one-by-one OSELM algorithm can be easily extended to chunk-by-chunk type.

2.2. R-OSELM. Though the OSELM algorithm seems perfect in theory, it still has deficiency when directly applied to real world applications. As seen from (7) and (8), the derivation of OSELM is based on the assumption that the autocorrelation matrix \( H^T H \) is nonsingular; however, this assumption often does not hold in many practical applications. Once the singular or ill-posed problem occurs, the generalization performance of OSELM will deteriorate significantly. In order to overcome this problem, Huyhn and Won [16] proposed a regularized OSELM (R-OSELM) by using the Tikhonov regularization. The learning procedure of the R-OSELM is almost the same as the OSELM, just adding a regularization item to the autocorrelation matrix \( H^T H \) to avoid the singular
or ill-posed problem so as to improve the stability of the algorithm. The R-OSELM can be briefly retold below.

For the initial training set \( \Omega_{k-1} = \{(x_j, t_j) \mid j = 1, \ldots, k - 1\} \) at time \( k - 1 \), let \( H_{k-1} = \begin{bmatrix} h_1^T \ h_2^T \cdots h_{k-1}^T \end{bmatrix}^T \) and \( T_{k-1} = [t_1 \ t_2 \cdots t_{k-1}]^T \); then the initial output weights can be written as

\[
P_{k-1} = (H_{k-1}^T H_{k-1} + \delta I)^{-1},
\]

\[
\beta_{k-1} = P_{k-1} H_{k-1}^T T_{k-1},
\]

where \( \delta \) is a positive real value called the regularization parameter and \( I \) is an identity matrix with the same dimensions as \( H^T H \).

Next in the sequential learning process, whenever a new sample \( (x_k, t_k) \) arrives, then \( H_k = [H_{k-1}^T \ h_k^T]^T \) and \( T_k = [T_{k-1}^T \ t_k]^T \), and the current output weights are recursively computed as

\[
P_k = (H_k^T H_k + \delta I)^{-1} = (H_{k-1}^T H_{k-1} + h_k^T h_k + \delta I)^{-1} = (P_{k-1} + h_k^T h_k P_{k-1})^{-1} P_{k-1} - \frac{h_k^T h_k P_{k-1} P_{k-1}^{-1} h_k^T h_k}{1 + h_k^T P_{k-1} P_{k-1}^{-1} h_k^T},
\]

\[
\beta_k = P_k H_k^T T_k = P_k (H_{k-1}^T T_{k-1} + h_k^T t_k)
\]

\[
\beta_k = P_k (P_{k-1} - h_k^T h_k P_{k-1}^{-1} h_k^T h_k) \beta_{k-1} + h_k^T t_k
\]

\[
\beta_k = P_k \left( \beta_{k-1} + h_k^T (t_k - h_k \beta_{k-1}) \right).
\]

Compare (11) and (12) with (9); though the finally obtained output weights update equations of R-OSELM seem the same as those in OSELM, we should keep in mind that in the sequential learning procedure of R-OSELM the regularization item \( \delta I \) is embedded in the recursive formulas through iterative computations (see (12)). Besides, if the regularization parameter \( \delta \) equals zero, then the R-OSELM degenerates to the original OSELM.

2.3. FR-OSELM. For time-varying system, the system behaviors often evolve over time; correspondingly the new samples are more effective in reflecting the changing trends of the system compared to the old ones. To better express the timeliness of the samples and further improve the prediction performance for the time-varying system, the FF method as well as the regularization technique is introduced into the OSELM, and then the FR-OSELM algorithm is proposed [24]. Different from the OSELM and the R-OSELM which treat all the training samples equally, the FR-OSELM assigns a different weight to each sample, respectively, according to their time order. That is, the recent samples are assigned higher weights while the old samples are assigned lower weights so as to represent their different contributions to the learning model.

The implementation of FR-OSELM can be briefly stated as follows. Suppose we have got the initial output weights \( \beta_{k-1} \) using (10); now a new training sample \( (x_k, t_k) \) arrives; then the corresponding \( \beta_k \) is originally expressed as

\[
\beta_k = (H_{k-1}^T H_{k-1} + \delta I + h_k^T h_k)^{-1} \left( H_{k-1}^T T_{k-1} + h_k^T t_k \right).
\]

Considering the different effectiveness of the newly arrived sample and the old samples to the model, (13) can be rewritten as

\[
\beta_k = \left( \lambda H_{k-1}^T H_{k-1} + \delta I + h_k^T h_k \right)^{-1} \frac{\lambda \delta I + h_k^T h_k}{\lambda (\lambda + h_k^T h_k)^{-1}}.
\]

where \( \lambda \in (0, 1] \) is called the forgetting factor (FF), which is used to weaken the influences of the old samples and indirectly enhance the effects of the latest one to the model so as to better depict the current state of the time-varying system. Let

\[
P_k = \left( \lambda H_{k-1}^T H_{k-1} + \lambda \delta I + h_k^T h_k \right)^{-1},
\]

and invert (15) on both sides; we have

\[
P_k = \lambda P_{k-1}^{-1} + h_k^T h_k.
\]

Applying Sherman-Morrison formula [25] to (16) yields

\[
P_k = \frac{P_{k-1} - h_k^T h_k P_{k-1}^{-1}}{\lambda (\lambda + h_k^T h_k)^{-1}}.
\]

Substituting (15) into (14), we obtain

\[
\beta_k = P_k \left( \frac{h_k^T (t_k - h_k \beta_{k-1})}{\lambda (\lambda + h_k^T h_k)^{-1}} \right)
\]

\[
\beta_k = P_k \left( \frac{h_k^T (t_k - h_k \beta_{k-1})}{\lambda (\lambda + h_k^T h_k)^{-1}} \right)
\]

Equations (17) and (18) give the recursive formulas for updating the output weights of FR-OSELM. When \( \lambda = 1 \), the FR-OSELM degenerates to the general R-OSELM. When \( \lambda = 1 \) and \( \delta = 0 \), the FR-OSELM degenerates to the original OSELM.

3. Proposed AFGR-OSELM

In this section, we give the detailed implementation of the proposed OSELM with generalized regularization and adaptive forgetting factor (AFGR-OSELM). We first theoretically analyze the inherent defect of the FR-OSELM; then an improved FGR-OSELM algorithm with generalized regularization is presented to solve this issue; next a new adaptive FF method under the condition of generalized regularization is derived to accommodate the FGR-OSELM for better tracking of the time-varying system, and finally the AFGR-OSELM is proposed and summarized.
3.1. The “Conflict” of Regularization and Forgetting Factor in FR-OSELM. The FR-OSELM is developed from the OSELM by incorporating the regularization technique in conjunction with the forgetting mechanism for time-varying system prediction. With the help of the FF, the FR-OSELM is effective in tracking the dynamic changes of the time-varying system and tends to provide good prediction results in time-varying environment. On the other hand, however, due to the impact of the FF, the regularization effect of the FR-OSELM will also be forgotten gradually as time goes on; as a result the FR-OSELM may encounter the similar ill-posed problem as the OSELM in the long run. Further theoretical analyses are as follows.

As described in Section 2.3, the derivation of the FR-OSELM is based on the intuitive analysis that in time-varying environment the new and the old samples are supposed to contribute differently to the learning model and should be weighted discriminatively. In theory, the FR-OSELM algorithm is equivalent to minimizing the following least squares cost function with FF and regularization item:

$$ J_{FR}(\beta_k) = \sum_{i=1}^{k} \lambda^{k-i} |t_i - h_i \beta_k|^2 + \delta \lambda^k \| \beta_k \|^2 $$

(19)

where $\lambda$ and $\delta$ are FF and regularization parameter, respectively. Applying the RLS method [21] to solve (19), we obtain the recursive solution of $\beta_k$:

$$ P_k = \frac{P_{k-1}}{\lambda} - \frac{P_{k-1} h_k^T h_k P_{k-1}}{\lambda (\lambda + h_k^T P_{k-1} h_k)} $$

(20)

$$ \beta_k = \beta_{k-1} + P_k h_k^T (t_k - h_k \beta_{k-1}) $$

which is completely identical to the results in FR-OSELM (see (17) and (18)).

From (19), we can find that the FR-OSELM equivalently uses a special exponential forgetting regularization item $\delta \lambda^k \| \beta_k \|^2$ in the cost function. The special scheme makes the deduced recursion formulas of FR-OSELM (see (20)) as concise as that in the original OSELM (see (9)) without extra computational burden; in other words, the computational complexity of the FR-OSELM algorithm at each iteration is the same as that of the original OSELM, that is, $O(n^2)$, where $n$ is the hidden nodes number. On the other hand, however, it also brings a very unfavorable side effect as well. We have known that in time-varying environment $\lambda$ is often assigned a value slightly less than 1 for a better tracking; in this context $\lambda^k$ is monotone decreasing and tends to zero for large $k$, which means that the regularization effect of $\delta \lambda^k \| \beta_k \|^2$ will fade gradually with time and become disabled thoroughly in the end. In conclusion, the FR-OSELM only has a short-term regularization effect within limited period, and it may still encounter the ill-posed problem in the long term and run unstably.

3.2. An Improved FR-OSELM with Generalized Regularization (FGR-OSELM). In order to effectively overcome the above-mentioned problem in the FR-OSELM and further improve the stability and practicality of the algorithm, we introduce an improvement of FR-OSELM named as FGR-OSELM by using a generalized regularization approach. Different from the FR-OSELM which uses a special exponential forgetting regularization item $\delta \lambda^k \| \beta_k \|^2$ in the cost function, the FGR-OSELM adopts a more generalized $l_2$ regularization item $\delta \| \beta_k \|^2$ instead. That is, the cost function of FGR-OSELM is written as

$$ J_{FGR}(\beta_k) = \sum_{i=1}^{k} \lambda^{k-i} |t_i - h_i \beta_k|^2 + \delta \| \beta_k \|^2 $$

(21)

It is clearly seen from (21) that the new generalized regularization scheme uses a constant $\delta$ as the regularization coefficient which is completely irrespective to the FF; thus its regularization effect will keep constant and no longer fade with time anymore, which ensures a persistent stability of the FGR-OSELM in all online learning stages.

Next, we are devoted to deriving a recursive solution of (21) under the sequential learning scenario. Because the cost function (21) is equipped with a generalized regularization item, the derivation of FGR-OSELM will be much different from the FR-OSELM algorithm. The optimal output weights for minimizing (21) can be obtained by setting the first-order partial derivative of $J_{FGR}(\beta_k)$ with respect to $\beta_k$ to zero. This yields

$$ R_k \beta_k = \Phi_k $$

(22)

where

$$ R_k = \sum_{i=1}^{k} \lambda^{k-i} h_i^T h_i + \delta I = \lambda R_{k-1} + \delta (1 - \lambda) I + h_k^T h_k $$

(23)

$$ \Phi_k = \sum_{i=1}^{k} \lambda^{k-i} h_i^T t_i = \lambda \Phi_{k-1} + h_k^T t_k $$

(24)

Define

$$ R_k^* = \lambda R_{k-1} + \delta (1 - \lambda) I $$

(25)

then (23) becomes

$$ R_k = R_k^* + h_k^T h_k $$

(26)

Applying the Sherman-Morrison-Woodbury formula and Sherman-Morrison formula [25] to (25) and (26), respectively, we get

$$ R_k^{-1} = (\lambda R_{k-1} + \delta (1 - \lambda) I)^{-1} = \lambda^{-1} R_{k-1}^{-1} $$

(27)

$$ = \frac{1}{\lambda^2} R_{k-1}^{-1} \left( I + \frac{\delta (1 - \lambda)}{\lambda} R_{k-1}^{-1} \right)^{-1} R_{k-1}^{-1} $$

(28)

$$ R_k^{-1} = \left( R_k^* + h_k^T h_k \right)^{-1} = R_k^*^{-1} - R_k^*^{-1} h_k^T h_k R_k^*^{-1} \frac{1 + h_k R_k^*^{-1} h_k}{1 + h_k R_k^*^{-1} h_k} $$

(28)
Let \( P_k = R_k^{-1} \), \( P_k^* = R_k^{-1} \); (27) and (28) can be, respectively, rewritten as

\[
P_k^* = \lambda^{-1} P_{k-1} - \frac{\delta (1 - \lambda)}{\lambda^2} P_{k-1} \left( I + \frac{\delta (1 - \lambda)}{\lambda} P_{k-1} \right)^{-1} P_{k-1}.
\]

(29)

\[
P_k = P_k^* - \frac{P_k^* h_k^T h_k P_k^*}{1 + h_k P_k^* h_k^T}.
\]

(30)

Define \( g_k^* \), \( g_k \) as follows:

\[
g_k^* = \frac{\delta (1 - \lambda)}{\lambda} P_{k-1} \left( I + \frac{\delta (1 - \lambda)}{\lambda} P_{k-1} \right)^{-1},
\]

(31)

\[
g_k = \frac{P_k^* h_k^T}{1 + h_k P_k^* h_k^T}.
\]

(32)

We can rewrite (29) and (30) as

\[
P_k^* = \lambda^{-1} P_{k-1} - \lambda^{-1} \left( I - g_k^* \right) P_{k-1},
\]

(33)

\[
P_k = P_k^* - g_k^* h_k P_k^* = \left( I - g_k h_k \right) P_k^*.
\]

(34)

Also, (32) can be transformed and rewritten as

\[
g_k = P_k^* h_k^T - g_k h_k P_k^* = \left( I - g_k h_k \right) P_k^* h_k^T.
\]

(35)

From (34), \( g_k \) can be also expressed as

\[
g_k = P_k h_k^T.
\]

(36)

From (22) and (24), we have

\[
\beta_k = R_k^{-1} \Phi_k = P_k \left( \lambda \Phi_{k-1} + h_k^T t_k \right)
\]

\[
= P_k \left( \lambda R_{k-1} \beta_{k-1} + h_k^T t_k \right).
\]

(37)

Substituting (23) into (37), obtain

\[
\beta_k = P_k \left( R_k - \delta (1 - \lambda) I - h_k^T h_k \right) \beta_{k-1} + h_k^T t_k
\]

\[
= \beta_{k-1} - \delta (1 - \lambda) P_k \beta_{k-1} - P_k h_k^T h_k \beta_{k-1}
\]

\[
+ P_k h_k^T t_k
\]

\[
= \beta_{k-1} + g_k (t_k - h_k \beta_{k-1}) - \delta (1 - \lambda) P_k \beta_{k-1}.
\]

(38)

Equations (31), (33), (32), (34), and (38) give the whole recursive formulas for updating the output weights of FGR-OSELM. When \( \lambda = 1 \), the FGR-OSELM degenerates to the general R-OSELM. When \( \lambda = 1 \) and \( \delta = 0 \), the FGR-OSELM degenerates to the original OSELM.

Since the FGR-OSELM adopts a generalized regularization item in the cost function, the obtained recursive formulas seem more complex compared with that of the FR-OSELM (see (20)). Moreover, in (31), we need to calculate the inverse of a matrix with dimension \( n \) (\( n \) is the hidden nodes number). Fortunately, the computational complexity of (31) can be much reduced with an approximation approach [26]. In the FGR-OSELM, the regularization parameter \( \delta \) is used just to avoid the ill-posed problem, and an inappropriate bigger value of \( \delta \) may equivalently introduce extra noise and negatively affect model specification; hence in practice \( \delta \) is usually set to a small value at the level of \( 10^{-4} \) or even smaller. Additionally, in time-varying environment the FF \( \lambda \) is often chosen as a positive constant slightly less than 1; then \( \delta (1 - \lambda)/\lambda^2 \) is a very small value, so we can approximately estimate that \( (\delta (1 - \lambda)/\lambda)^2 \approx 0 \), and it follows that

\[
\left( I + \frac{\delta (1 - \lambda)}{\lambda} P_{k-1} \right) \left( I - \frac{\delta (1 - \lambda)}{\lambda} P_{k-1} \right)
\]

\[
= I - \left( \frac{\delta (1 - \lambda)}{\lambda} \right)^2 P_{k-1} P_{k-1} = I.
\]

(39)

Further we get

\[
\left( I + \frac{\delta (1 - \lambda)}{\lambda} P_{k-1} \right)^{-1} = \left( I - \frac{\delta (1 - \lambda)}{\lambda} P_{k-1} \right).
\]

(40)

Therefore, (31) can be approximately expressed as

\[
g_k^* = \frac{\delta (1 - \lambda)}{\lambda} P_{k-1} \left( I - \frac{\delta (1 - \lambda)}{\lambda} P_{k-1} \right).
\]

(41)

Now, the computational complexity of updating \( g_k^* \) is much reduced by avoiding the matrix inversion.

Similar to the R-OSELM, the second term of (21) can be extended to other types of norm instead of the \( p \)-norm, such as the \( Q \)-norm. In the case of \( Q \)-norm complexity measure, a symmetric positive definite matrix \( Q \) is added in the cost function and adopted as a new machine complexity measure [16]. Using the same derivation steps as above, the learning formulas for online sequential learning process that use the \( Q \)-norm complexity measure can be obtained, which are essentially the same as those of the \( p \)-norm except that the matrix \( Q \) is added in some parts of the learning formulas. In general, the symmetric positive definite matrix \( Q \) is often defined as the identity matrix [9, 20, 24], and, in this case, the \( Q \)-norm degenerates to the \( p \)-norm.

3.3. Adaptive Forgetting Scheme for FGR-OSELM. The FF plays an important part in adapting the FR-OSELM [24] and analogous models [27, 28] to system dynamics and thus greatly improving the performances of the models for time-varying system. However, in reality it is very difficult or even impractical to compute an appropriate predetermined value for the FF without a priori knowledge [29]. Moreover, a priori selection of the FF may not guarantee global adaptation of the system dynamics in a complex time-varying environment. Hence, a variable FF (VFF) strategy is a more attractive choice in real world applications. Up to now, many VFF-RLS algorithms have been proposed and successfully applied in adaptive signal processing [29–31], but most of these algorithms are derived on the basis of some given domain knowledge in the field of signal processing like the signal-to-noise ratio and so on, which are not applicable in general
applications. Comparatively, a more universal RLS algorithm
with adaptive FF (AFF-RLS) was presented in [21], in which
the FF is adaptively tuned using a gradient based method
without need for much a priori knowledge. At present, the
AFF-RLS has been successfully applied to the OSELM for
time-varying nonlinear system identification [22] and fault
prediction [24], and it has shown good tracking performance
in time-varying environment. However, the AFF-RLS algo-

with respect to λ yields

where E[·] is the expectation operator and ε_k = t_k - h_kβ_{k-1}
is the a priori estimation error. Differentiating the cost function
J'(k) with respect to λ gives

\[ \frac{\partial J'(k)}{\partial \lambda} = -E[\psi_k^T \beta_k^* \epsilon_k], \]

(43)

Let S_k denote the derivative of the inverse correlation matrix
P_k with respect to λ,

\[ S_k = \frac{\partial P_k}{\partial \lambda}, \]

(45)

and combine (36), (38), and (45) into (44); then the update
equation of \psi_k can be expressed as

\[ \psi_k = \frac{\partial \beta_k^*}{\partial \lambda} = \frac{\partial}{\partial \lambda} \left( \beta_{k-1} + P_k h_k^T (t_k - h_k \beta_{k-1}) - \delta (1 - \lambda) P_k \beta_{k-1} \right) \]

(46)

To compute \psi_k, right now we just need to obtain S_k, and
a recursion for updating S_k can be derived as follows. Substituting (36) into (34), we get

\[ P_k = (I - P_k h_k^T h_k) P_k^*. \]

(47)

The derivative of (47) with respect to \lambda is obtained as

\[ \frac{\partial P_k}{\partial \lambda} = S_k = \left( -S_k h_k^T h_k P_k^* + (1 - P_k h_k^T h_k) \frac{\partial P_k^*}{\partial \lambda} \right). \]

(48)

By exchanging and collecting terms, (48) can be expressed as

\[ S_k = (I - g_k h_k) \frac{\partial P_k^*}{\partial \lambda} \left( I - h_k^T h_k P_k^* \right)^{-1}. \]

(49)

Applying the Sherman-Morrison formula [25] to the last term of (49), we have

\[ S_k = (I - g_k h_k) \frac{\partial P_k^*}{\partial \lambda} \left( I - h_k^T h_k P_k^* \right)^{-1}. \]

(50)

Let S_k^* denote the derivative of P_k^* with respect to \lambda and combined with (33); we get

\[ S_k = \frac{\partial P_k^*}{\partial \lambda} = \frac{\partial}{\partial \lambda} \left( \frac{1}{2} \frac{\delta^2 (1 - \lambda)^2}{\lambda^2} P_k - 1 \right) \]

(51)

where \psi_k denotes the gradient of the output weights \beta_k with respect to λ:

\[ \psi_k = \frac{\partial \beta_k^*}{\partial \lambda}. \]

(44)

Provided that the estimate of the scalar gradient \( \psi_k \) in (43) is available, in a similar manner as [21], we may
adaptively compute \lambda using the recursion

\[ \lambda_k = \lambda_{k-1} - \alpha \psi_k^* (k). \]

(54)

Using the instantaneous estimate \( \psi_k^* \) for \( \psi_k \) on the basis of (43), the complete expression for updating the FF is
given as

\[ \lambda_k = \lambda_{k-1} + \alpha \psi_k^* h_k^T h_k \lambda_k. \]

(55)
where $\alpha$ is a small, positive learning-rate parameter and the bracket denotes a truncation operator with the upper bound $\lambda_+$ and the lower bound $\lambda_-$, respectively. In practice, $\lambda_+$ may be set slightly less than 1 to ensure that some degree of forgetting is always present, while the setting of $\lambda_-$ is problem specific, and it is advisable not to set $\lambda_-$ a value too small, as it may produce numerical instability [21, 32].

3.4. Summary of the AFGR-OSELM Algorithm. Incorporating the generalized regularization approach and the adaptive forgetting scheme into the OSELM simultaneously, the new AFGR-OSELM is proposed. Similar to the original OSELM algorithm, the proposed AFGR-OSELM algorithm consists of an initial batch learning phase and a following sequential learning phase, which are summarized as follows.

(1) Initialization Phase. Given an initial training subset $\mathcal{O}_{k-1} = \{(x_j, t_j) \mid j = 1, \ldots, k-1\}$, activation function $G(a, b, x)$, hidden nodes number $n$, and regularization parameter $\delta$:

(a) Randomly assign hidden nodes parameters $(a_i, b_i)$, $i = 1, 2, \ldots, n$.

(b) Calculate the initial hidden layer output matrix $H_{k-1}$:

$$H_{k-1} = \begin{bmatrix} G(a_1, b_1, x_1) & \cdots & G(a_1, b_n, x_1) \\ \vdots & \ddots & \vdots \\ G(a_1, b_1, x_{k-1}) & \cdots & G(a_1, b_n, x_{k-1}) \end{bmatrix}_{(k-1) \times n}$$

(c) Calculate the initial output weights $\beta_{k-1} = P_{k-1}H_{k-1}^{T}T_{k-1}$, $P_{k-1} = (H_{k-1}^{T}H_{k-1} + \delta I)^{-1}$, $T_{k-1} = [t_1 \ t_2 \ \cdots \ \ t_{k-1}]^{T}$.

(2) Sequential Learning Phase. First set the initial values for $S_{k-1}$, $\psi_{k-1}$, $\lambda_{k-1}$, $\lambda_+$, $\lambda_-$ and $\alpha$; then for each sequentially arriving observation $(x_k, t_k)$:

(a) Calculate the hidden layer output vector $h_k$

$$h_k = \begin{bmatrix} G(a_1, b_1, x_k) & \cdots & G(a_1, b_n, x_k) \end{bmatrix}.$$  

(b) Calculate the prior error $\varepsilon_k$

$$\varepsilon_k = t_k - h_k \beta_{k-1}.$$  

(c) Calculate the current output weights $\beta_k$

$$\beta_k = \beta_{k-1} + \frac{\delta (1 - \lambda_{k-1})}{\lambda_{k-1}} P_{k-1} \psi_{k-1} + \frac{\psi_{k-1}^{*} g_{k-1}^{*}}{\psi_{k-1}}$$

$$g_k = \frac{h_k P_{k-1}^{*}}{1 + h_k P_{k-1}^{*} h_k^{T}},$$

$$P_k = (I - g_k h_k) P_{k-1}^{*},$$

$$\beta_k = \beta_{k-1} + \varepsilon_k g_k - \delta (1 - \lambda_{k-1}) P_k \beta_{k-1}.$$  

(d) Calculate the current FF $\lambda_k$ and update $S_k$, $\psi_k$ recursively:

$$\lambda_k = \left[ \lambda_{k-1} + \alpha \psi_{k-1}^{T} h_k^{T} \frac{\varepsilon_k}{\lambda_{k-1}} \right]^{+}$$

$$S_k^* = -\left( \frac{1 - \lambda_k}{\lambda_k^{2}} \right)^{S_{k-1}} P_{k-1} (1 - \lambda_k) P_{k-1} + S_{k-1} g_k$$

$$+ \frac{\delta^2 (1 - \lambda_k)}{\lambda_k^{3}} P_{k-1} (1 - \lambda_k) S_{k-1} P_{k-1}$$

$$- (1 - \lambda_k) \beta_{k-1}) + \delta P_k \beta_{k-1}.$$  

(e) Set $k = k + 1$. Go to Step (2).

3.5. Computational Complexity Analysis. In the sequential learning process of the AFGR-OSELM algorithm, when a new observation arrives, the most costly steps are the calculations of the current output weights and the forgetting factor with (60) and (61), respectively. Assuming that arithmetic with individual elements have complexity $O(1)$, the complexity of addition or subtraction of an $m$-by-$n$-order matrix is $O(mn)$, the complexity of multiplication of an $m$-by-$n$-order matrix and an $n$-by-$p$-order matrix is $O(mnp)$ [25]. In (60), since $P_k$, $P_k^{*}$, $g_k^{*}$ are $n \times n$ matrices and $g_k$, $h_k^{T}$, $\beta_k$ are $n \times 1$ vectors, the computational complexity of updating $\beta_k$ could be estimated as $O(n^2)$. Similarly, in (61), $P_k$, $P_k^{*}$, $g_k^{*}$, $S_k$, $\psi_k^{*}$ are $n \times n$ matrices and $g_k$, $h_k^{T}$, $\beta_k$, $\psi_k$ are $n \times 1$ vectors; then the computational complexity required to calculate $\lambda_k$ and update $S_k$, $\psi_k$ is also $O(n^2)$. Summarizing the above analysis, it can be concluded that the total computational complexity of the AFGR-OSELM algorithm for online learning is $O(n^2)$, which is a little higher than the complexity of the original OSELM ($O(n^2)$).

Since the computational complexity of the AFGR-OSELM algorithm is mainly decided by matrix multiplication, by using Strassen’s matrix multiplication method [25], the number of multiplications can be reduced to $O(n^{2.807})$; that is, the computational complexity of the AFGR-OSELM algorithm can also be reduced to $O(n^{2.807})$. Moreover, with a proper approximation approach [22], the computational complexity of AFGR-OSELM may be further reduced, and we take it as future work.
4. Simulation Experiments and Performance Evaluation

In this section, the performance of the proposed AFGR-OSELM is evaluated on a simulative time-varying system, a parameter-varying chaotic time series system, two stock price data sets, and two real world industrial data sets with dynamic behaviors. The experimental results of the proposed algorithm are also compared with those of the OSELM [6], R-OSELM [16], FR-OSELM [24], and DFF-OSELM [23]. All of the five algorithms use the same sigmoidal additive activation function \( G(a, b, x) = 1/(1 + \exp(-(a \cdot x + b))) \), where the input weights \( a \) and the biases \( b \) are randomly selected from the range \([-1, 1]\). In our experiments, all the models are evaluated for predicting time-varying systems in an online scenario; that is, the samples are arranged to feed the predictors incrementally one-by-one after the initialization phase; meanwhile the predictors update continually during the process of online prediction. For each case, the reported experimental results are the average values of 30 independent experimental runs on the online data and the prediction performance is measured by the root mean squared error (RMSE) defined as

\[
RMSE = \sqrt{\frac{1}{N} \sum_{j=1}^{N} (t_j - \hat{t}_j)^2}, \quad (62)
\]

where \( t_j \) and \( \hat{t}_j \) are the real value and predicted value, respectively, and \( N \) is the number of the testing samples. The experiments have been carried out in MATLAB R2010b environment running on an ordinary PC with 3.4 GHZ CPU and 4 GB RAM.

4.1. Prediction of a Simulative Time-Varying System. In the first experiment, an artificial data set that presents a gradual evolution in each sample is considered to simulate the dynamic behaviors of the time-varying system. The data set has a total of 2200 samples, and, for each sample, the input is constituted by four random variables which are normally distributed with mean zero and standard deviation 0.1; the output is the inner product of the input vector and the corresponding coefficient vector. In order to construct a time-varying system, the coefficients for each sample change continuously over time [1]. To be specific, we first fix the initial coefficient vector as

\[
a(0) = [0.5 \quad 0.2 \quad 0.7 \quad 0.8] \quad (63)
\]

and subsequently these values evolve over time according to the following equation:

\[
a_j(i) = a_j(i - 1) + \frac{2(i + 1000)}{10^8} \logsig(a_j(i - 1)), \quad (64)
\]

\[i = 1, 2, \ldots, 2200,
\]

where \( i \) indicates the index of the sample and \( j \) indicates the component of the coefficient vector, \( j = 1, 2, 3, 4 \). To make the prediction task real, some normal white noises with a level of 0.01 are added to the output.

All the five algorithms, OSELM, R-OSELM, FR-OSELM, DFF-OSELM, and AFGR-OSELM, are performed to model and predict the simulated time-varying system in an online mode. Since the hidden nodes number \( n \) is an important parameter which may greatly affect the learning accuracy and generalization performance of the neural network, for each algorithm we set \( n = 20, n = 50, \) and \( n = 100 \), respectively, for experiments, and the corresponding numbers of training data for initialization are taken as 50, 100, and 200, respectively. Besides, other parameters for each algorithm are set as follows: for R-OSELM, \( \delta = 10^{-8} \); for FR-OSELM, \( \delta = 10^{-8}, \lambda = 0.98 \); for DFF-OSELM, all the parameters are set the same as in [23], and an extra regularization parameter, \( \delta = 10^{-8} \), is added to stabilize the algorithm; for our AFGR-OSELM, \( \delta = 10^{-8}, \alpha = 0.1, \lambda_s = 0.999, \lambda_c = 0.8 \), the initial value for \( \lambda \) is set as 0.995, and the \( S \) and \( \psi \) can be simply initialized to identity matrix and zero vector, respectively.

Table 1 presents the mean and standard deviations (SD) of the prediction RMSE over 30 independent trials of the five models for performing different prediction steps on the simulated time-varying system. We can see from Table 1 that the performance of the OSELM is very unstable and highly sensitive to the hidden node number \( n \); when \( n = 20 \), the OSELM is able to provide effective prediction results, while in the case of \( n = 50 \) and \( n = 100 \) the OSELM is likely to produce very large prediction RMSE beyond reasonable range, and which are denoted as “\( x \)” in the table, similarly hereinafter. The reason for this is that in the sequential learning process of OSELM the autocorrelation matrix \( H^T H \) may be singular or ill-conditioning at certain instances; then the recursive calculation of \( (H^T H)^{-1} \) is nonsense and consequently the algorithm is apt to produce an unreliable result. By using the regularization technique the R-OSELM successfully avoids the potential ill-posed problem, so it obtains stable and effective prediction results in all situations. Though the FR-OSELM adopts the similar regularization technique to overcome the ill-posed problem, its regularization function decays gradually over time due to the effect of the FF, and the potential ill-posed problem may still occur in the long run, so we can see the FR-OSELM is workable only within short prediction steps and out of service in longer-term prediction steps in the cases of \( n = 50 \) and \( n = 100 \). In contrast, our AFGR-OSELM can maintain a persistent stability by using a generalized regularization approach and achieve stable and reliable performances all the time. Moreover, with the help of the adaptive forgetting mechanism our AFGR-OSELM model can effectively track the dynamic changes of the time-varying system and abandon the negative influences of the outdated samples in time, so it obtains the lowest prediction RMSE among the five models almost in all cases.

To intuitively compare the stability and the prediction accuracy of the five OSELM-related models, the typical absolute error (the difference between real value and predicted value) graphs of each model for performing 2000 steps online prediction in the case of \( n = 100 \) are demonstrated in Figure 1. Since the obtained prediction errors of FR-OSELM are very large and completely meaningless when the prediction step exceeds a certain limit, we only present the
Table 1: Mean (SD) of the prediction RMSE of OSELM, R-OSELM, FR-OSELM, DFF-OSELM, and AFGR-OSELM on simulative time-varying system.

<table>
<thead>
<tr>
<th>Hidden node (n)</th>
<th>Model</th>
<th>Prediction steps</th>
<th>200</th>
<th>500</th>
<th>1000</th>
<th>1500</th>
<th>2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>OSELM</td>
<td></td>
<td>1.56E–03 (7.35E–05)</td>
<td>1.51E–03 (6.01E–05)</td>
<td>1.82E–03 (2.75E–05)</td>
<td>2.42E–03 (2.12E–05)</td>
<td>3.38E–03 (1.50E–05)</td>
</tr>
<tr>
<td></td>
<td>R-OSELM</td>
<td></td>
<td>1.51E–03 (6.93E–05)</td>
<td>1.49E–03 (2.15E–05)</td>
<td>1.81E–03 (2.07E–05)</td>
<td>2.42E–03 (1.41E–05)</td>
<td>3.38E–03 (1.32E–05)</td>
</tr>
<tr>
<td></td>
<td>FR-OSELM</td>
<td></td>
<td>1.56E–03 (1.01E–04)</td>
<td>1.55E–03 (3.92E–05)</td>
<td>1.51E–03 (3.67E–05)</td>
<td>1.50E–03 (2.69E–05)</td>
<td>1.49E–03 (2.19E–05)</td>
</tr>
<tr>
<td></td>
<td>DFF-OSELM</td>
<td></td>
<td>1.65E–03 (7.39E–05)</td>
<td>1.62E–03 (3.30E–05)</td>
<td>1.62E–03 (2.81E–05)</td>
<td>1.66E–03 (2.66E–05)</td>
<td>1.74E–03 (5.59E–05)</td>
</tr>
<tr>
<td></td>
<td>AFGR-OSELM</td>
<td></td>
<td>1.48E–03 (3.91E–05)</td>
<td>1.46E–03 (3.80E–05)</td>
<td>1.48E–03 (1.41E–05)</td>
<td>1.50E–03 (2.13E–05)</td>
<td>1.50E–03 (1.75E–05)</td>
</tr>
<tr>
<td>50</td>
<td>OSELM</td>
<td></td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>R-OSELM</td>
<td></td>
<td>1.49E–03 (9.99E–06)</td>
<td>1.53E–03 (5.52E–06)</td>
<td>1.87E–03 (3.80E–06)</td>
<td>2.51E–03 (3.54E–06)</td>
<td>3.50E–03 (5.83E–06)</td>
</tr>
<tr>
<td></td>
<td>FR-OSELM</td>
<td></td>
<td>1.60E–03 (1.28E–05)</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>DFF-OSELM</td>
<td></td>
<td>1.60E–03 (2.10E–05)</td>
<td>1.65E–03 (1.50E–05)</td>
<td>1.73E–03 (1.50E–05)</td>
<td>1.87E–03 (1.61E–05)</td>
<td>2.14E–03 (2.83E–05)</td>
</tr>
<tr>
<td></td>
<td>AFGR-OSELM</td>
<td></td>
<td>1.49E–03 (1.42E–05)</td>
<td>1.50E–03 (5.70E–06)</td>
<td>1.53E–03 (5.53E–06)</td>
<td>1.54E–03 (5.63E–06)</td>
<td>1.54E–03 (5.21E–06)</td>
</tr>
<tr>
<td>100</td>
<td>OSELM</td>
<td></td>
<td>7.37E–01 (4.01E+00)</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>R-OSELM</td>
<td></td>
<td>1.48E–03 (9.31E–06)</td>
<td>1.55E–03 (5.49E–06)</td>
<td>2.07E–03 (5.30E–06)</td>
<td>2.70E–03 (3.77E–06)</td>
<td>3.81E–03 (8.55E–06)</td>
</tr>
<tr>
<td></td>
<td>FR-OSELM</td>
<td></td>
<td>1.55E–03 (2.27E–05)</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>DFF-OSELM</td>
<td></td>
<td>1.52E–03 (1.33E–05)</td>
<td>1.65E–03 (1.68E–05)</td>
<td>1.84E–03 (2.28E–05)</td>
<td>1.97E–03 (3.34E–05)</td>
<td>2.44E–03 (6.89E–05)</td>
</tr>
<tr>
<td></td>
<td>AFGR-OSELM</td>
<td></td>
<td>1.47E–03 (6.20E–06)</td>
<td>1.49E–03 (4.30E–06)</td>
<td>1.56E–03 (2.84E–06)</td>
<td>1.55E–03 (1.95E–06)</td>
<td>1.56E–03 (2.34E–06)</td>
</tr>
</tbody>
</table>

"×" denotes nullification owing to the too large RMSE.
Figure 1: The graphs of absolute error on simulative time-varying system. (a) OSELM. (b) R-OSELM. (c) FR-OSELM. (d) DFF-OSELM. (e) AFGR-OSELM.
partial reasonable results within the first 300 steps. It is clear to see from Figure 1(a) that there are a few large peaks at certain instances, which reveals the instability of the OSELM algorithm. Relatively, the R-OSELM is a more robust algorithm and the corresponding error graph (Figure 1(b)) is quite stable without large fluctuation. Nevertheless, the R-OSELM is essentially not a targeted method to track the dynamic behaviors of the time-varying system, so its prediction errors increase gradually with time. Figure 1(c) shows that the FR- OSELM is workable only within short prediction steps and then fails rapidly when the prediction step exceeds 270, and the obtained experimental results are completely consistent with the theoretical analysis given in Section 3.1. Comparing Figure 1(b), Figure 1(d), and Figure 1(e), we can see that the DFF-OSELM and the AFGR-OSELM with forgetting mechanism have better tracking abilities and achieve smaller prediction errors than the R-OSELM without forgetting; also we can see that the AFGR-OSELM behaves better than the DFF-OSELM. In short, our AFGR-OSELM is able to provide more stable and accurate results compared with the state-of-the-art models for predicting the time-varying system.

4.2. Prediction of the Logistics System with Time-Varying Parameter. In actual engineering, the system parameters may change over time; consequently the corresponding system exhibits complex and changing characteristics as time goes on. Thus predicting the system with time-varying parameter is a practical and important task. Here we take the parameter time-varying Logistics system as an example to verify the performance of the AFGR-OSELM. Consider the following Logistic map with parameter $\mu$ changing gradually [33]:

$$x(t+1) = \mu(t)x(t)(1-x(t)), \quad t = 1, 2, \ldots$$

$$\mu(t+1) = \mu(t) + 10^{-4}(1 - 0.2\sin(t)), \quad t = 1, 2, \ldots$$

(65)

In this simulation, a Logistic chaotic time series with a length of 4000 is generated by (65) with initial value $x(1) = 0.512$, $\mu(1) = 3.6$. In order to reduce the transient effect, the initial 1000 values are omitted and the last 3000 values are kept for experiment. To make the prediction task real, some normal white noises with a level of 0.01 are added to the generated time series. The embedding dimension and time delay for phase space reconstruction are chosen as 4 and 1, respectively.

With the generated Logistic chaotic time series, the performances of the AFGR-OSELM for performing different prediction steps are evaluated and compared with those of the OSELM, R-OSELM, FR-OSELM, and DFF-OSELM. In this simulation, the experimental design and the parameters setting for each algorithm are the same as that in the first simulation in Section 4.1 except that a different value of regularization parameter $(10^{-4})$ is used in the four regularized models, R-OSELM, FR-OSELM, DFF-OSELM, and AFGR-OSELM, for stability.

Tables 3 and 4 present the mean and SD of the prediction RMSE over 30 independent trials of the five models for performing different prediction steps on the two stock price data sets. From Tables 3 and 4, we can see that the OSELM is very unstable and the obtained mean and SD of the prediction RMSE are much larger than that of other models. Similarly, since the regularization function fades gradually with time, the FR-OSELM is workable only within short prediction steps and out of service in longer-term prediction steps. In contrast, the proposed AFGR-OSELM is very stable by using a generalized regulation method and can always provide reliable prediction results. In addition, compared with R-OSELM and DFF-OSELM, our AFGR-OSELM achieves better prediction performances in almost all cases.

4.3. Stock Price Prediction. Stock price prediction is an important and challenging task in economic field. Since many factors that affect the stock price often change with time, the stock price data often have the time-varying characteristic. Two stock price data sets are used here for experiment (obtained from https://www.pmel.noaa.gov/tao/drupal/ds-del/), and each has 3000 stock prices of sequential time points. For the two stock price time series, the embedding dimension and time delay for phase space reconstruction are chosen as 15 and 1, respectively [17].

All the five algorithms, OSELM, R-OSELM, FR-OSELM, DFF-OSELM, and AFGR-OSELM, are performed to model and predict the stock price in an online mode. In this simulation, the experimental design and the parameters setting for each algorithm are the same as that in the first simulation in Section 4.1 except that a different value of regularization parameter $(10^{-4})$ is used in the four regularized models, R-OSELM, FR-OSELM, DFF-OSELM, and AFGR-OSELM, for better stability.

4.4. Prediction of the Industrial Systems with Time-Varying Behaviors. To demonstrate the effectiveness and practicability of AFGR-OSELM in real applications, two real world industrial data sets are also employed and tested. Most
Table 2: Mean (SD) of the prediction RMSE of OSELM, R-OSELM, FR-OSELM, DFF-OSELM, and AFGR-OSELM on Logistic system with time-varying parameter.

<table>
<thead>
<tr>
<th>Hidden node (n)</th>
<th>Model</th>
<th>200</th>
<th>500</th>
<th>1000</th>
<th>1500</th>
<th>2000</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OSELM</td>
<td>6.74E-03 (5.39E-04)</td>
<td>6.59E-03 (4.60E-04)</td>
<td>7.67E-03 (7.01E-04)</td>
<td>1.14E-02 (1.46E-02)</td>
<td>1.06E-02 (1.65E-03)</td>
</tr>
<tr>
<td>20</td>
<td>R-OSELM</td>
<td>6.71E-03 (4.06E-04)</td>
<td>6.71E-03 (8.23E-04)</td>
<td>7.64E-03 (5.22E-04)</td>
<td>8.51E-03 (1.06E-03)</td>
<td>1.03E-02 (1.64E-03)</td>
</tr>
<tr>
<td></td>
<td>FR-OSELM</td>
<td>6.73E-03 (3.77E-04)</td>
<td>6.74E-03 (6.33E-04)</td>
<td>7.32E-03 (3.62E-04)</td>
<td>8.08E-03 (2.88E-03)</td>
<td>9.01E-03 (2.77E-03)</td>
</tr>
<tr>
<td></td>
<td>DFF-OSELM</td>
<td>6.89E-03 (3.70E-04)</td>
<td>7.14E-03 (4.34E-04)</td>
<td>7.97E-02 (3.62E-01)</td>
<td>8.08E-03 (2.88E-03)</td>
<td>9.01E-03 (2.77E-03)</td>
</tr>
<tr>
<td></td>
<td>AFGR-OSELM</td>
<td>6.73E-03 (3.76E-04)</td>
<td>6.55E-03 (6.05E-04)</td>
<td>7.09E-03 (4.93E-04)</td>
<td>7.42E-03 (6.77E-04)</td>
<td>9.28E-03 (1.24E-03)</td>
</tr>
<tr>
<td></td>
<td>OSELM</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>50</td>
<td>R-OSELM</td>
<td>5.47E-03 (6.40E-05)</td>
<td>5.85E-03 (4.39E-05)</td>
<td>6.46E-03 (6.84E-05)</td>
<td>6.75E-03 (7.34E-05)</td>
<td>7.56E-03 (1.35E-04)</td>
</tr>
<tr>
<td></td>
<td>FR-OSELM</td>
<td>5.60E-03 (5.51E-05)</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>DFF-OSELM</td>
<td>5.64E-03 (8.48E-05)</td>
<td>6.16E-03 (8.39E-05)</td>
<td>6.66E-03 (1.50E-04)</td>
<td>6.82E-03 (1.13E-04)</td>
<td>7.32E-03 (1.02E-04)</td>
</tr>
<tr>
<td></td>
<td>AFGR-OSELM</td>
<td>5.50E-03 (7.52E-05)</td>
<td>5.80E-03 (7.27E-05)</td>
<td>6.10E-03 (7.15E-05)</td>
<td>6.20E-03 (1.06E-04)</td>
<td>6.65E-03 (8.95E-05)</td>
</tr>
<tr>
<td></td>
<td>OSELM</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>100</td>
<td>R-OSELM</td>
<td>5.34E-03 (3.04E-05)</td>
<td>6.02E-03 (2.39E-05)</td>
<td>6.53E-03 (3.02E-05)</td>
<td>6.86E-03 (3.33E-05)</td>
<td>7.36E-03 (6.21E-05)</td>
</tr>
<tr>
<td></td>
<td>FR-OSELM</td>
<td>5.44E-03 (5.82E-05)</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>DFF-OSELM</td>
<td>5.34E-03 (3.49E-05)</td>
<td>6.31E-03 (4.21E-05)</td>
<td>6.80E-03 (3.72E-05)</td>
<td>6.87E-03 (3.77E-05)</td>
<td>7.38E-03 (4.29E-05)</td>
</tr>
<tr>
<td></td>
<td>AFGR-OSELM</td>
<td>5.31E-03 (3.05E-05)</td>
<td>5.83E-03 (3.67E-05)</td>
<td>6.07E-03 (3.52E-05)</td>
<td>6.16E-03 (4.35E-05)</td>
<td>6.54E-03 (4.21E-05)</td>
</tr>
</tbody>
</table>

"×" denotes nullification owing to the too large RMSE.
Figure 2: The graphs of absolute error on Logistics system with time-varying parameter. (a) OSELM. (b) R-OSELM. (c) FR-OSELM. (d) DFF-OSELM. (e) AFGR-OSELM.
Table 3: Mean (SD) of the prediction RMSE of OSELM, R-OSELM, FR-OSELM, DFF-OSELM, and AFGR-OSELM on stock price data set 1.

<table>
<thead>
<tr>
<th>Hidden node (ni)</th>
<th>Model</th>
<th>Prediction steps</th>
<th>200</th>
<th>500</th>
<th>1000</th>
<th>1500</th>
<th>2800</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OSELM</td>
<td></td>
<td>6.68 (11.26)</td>
<td>×</td>
<td>×</td>
<td>101.73 (314.98)</td>
<td>119.76 (157.99)</td>
</tr>
<tr>
<td></td>
<td>R-OSELM</td>
<td></td>
<td>3.03 (0.41)</td>
<td>4.47 (2.06)</td>
<td>42.99 (15.08)</td>
<td>38.13 (13.74)</td>
<td>29.93 (9.74)</td>
</tr>
<tr>
<td></td>
<td>FR-OSELM</td>
<td></td>
<td>2.72 (0.75)</td>
<td>2.75 (0.39)</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>DFF-OSELM</td>
<td></td>
<td><strong>2.40 (0.33)</strong></td>
<td><strong>2.12 (0.48)</strong></td>
<td>22.48 (20.53)</td>
<td>27.82 (29.71)</td>
<td>25.04 (29.05)</td>
</tr>
<tr>
<td></td>
<td>AFGR-OSELM</td>
<td></td>
<td>2.65 (0.43)</td>
<td>2.27 (0.42)</td>
<td><strong>11.02 (6.73)</strong></td>
<td><strong>9.82 (4.51)</strong></td>
<td><strong>8.56 (4.48)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>39.64 (120.58)</td>
<td>74.50 (270.05)</td>
<td>205.33 (283.24)</td>
<td>197.47 (284.68)</td>
<td>134.78 (187.42)</td>
</tr>
<tr>
<td></td>
<td>OSELM</td>
<td></td>
<td>2.70 (0.53)</td>
<td>4.11 (1.12)</td>
<td>55.57 (23.56)</td>
<td>51.82 (17.22)</td>
<td>33.73 (8.55)</td>
</tr>
<tr>
<td></td>
<td>R-OSELM</td>
<td></td>
<td><strong>2.34 (0.50)</strong></td>
<td>3.61 (1.73)</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>FR-OSELM</td>
<td></td>
<td>2.78 (0.88)</td>
<td>2.47 (0.46)</td>
<td>43.14 (34.42)</td>
<td>27.03 (26.67)</td>
<td>19.59 (11.01)</td>
</tr>
<tr>
<td></td>
<td>DFF-OSELM</td>
<td></td>
<td><strong>2.49 (0.58)</strong></td>
<td><strong>2.25 (0.38)</strong></td>
<td><strong>15.14 (7.19)</strong></td>
<td><strong>11.74 (4.22)</strong></td>
<td><strong>9.62 (3.52)</strong></td>
</tr>
<tr>
<td></td>
<td>AFGR-OSELM</td>
<td></td>
<td>18.19 (26.38)</td>
<td>11.71 (12.02)</td>
<td>467.43 (1429.08)</td>
<td>×</td>
<td>79.98 (81.37)</td>
</tr>
<tr>
<td></td>
<td>OSELM</td>
<td></td>
<td>2.93 (0.95)</td>
<td>3.72 (0.62)</td>
<td>69.46 (21.16)</td>
<td>67.37 (23.85)</td>
<td>50.86 (13.59)</td>
</tr>
<tr>
<td></td>
<td>R-OSELM</td>
<td></td>
<td><strong>2.69 (0.95)</strong></td>
<td><strong>3.45 (1.18)</strong></td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>FR-OSELM</td>
<td></td>
<td>2.73 (0.50)</td>
<td>3.21 (0.95)</td>
<td>38.17 (34.42)</td>
<td>32.24 (12.52)</td>
<td>33.78 (18.89)</td>
</tr>
<tr>
<td></td>
<td>DFF-OSELM</td>
<td></td>
<td><strong>2.49 (0.67)</strong></td>
<td><strong>2.25 (0.45)</strong></td>
<td><strong>15.71 (7.19)</strong></td>
<td><strong>12.98 (3.28)</strong></td>
<td><strong>10.91 (3.68)</strong></td>
</tr>
<tr>
<td></td>
<td>AFGR-OSELM</td>
<td></td>
<td>224.90 (915.90)</td>
<td>27.71 (98.35)</td>
<td>40.80 (174.70)</td>
<td>305.76 (1514.01)</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>OSELM</td>
<td></td>
<td>5.07 (1.02)</td>
<td>4.50 (0.79)</td>
<td>4.37 (0.51)</td>
<td>18.41 (18.63)</td>
<td>14.40 (9.28)</td>
</tr>
<tr>
<td></td>
<td>R-OSELM</td>
<td></td>
<td><strong>3.75 (0.56)</strong></td>
<td><strong>2.72 (0.74)</strong></td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>FR-OSELM</td>
<td></td>
<td>4.35 (1.66)</td>
<td>2.98 (0.70)</td>
<td>2.72 (0.62)</td>
<td>7.83 (5.50)</td>
<td>6.26 (4.44)</td>
</tr>
<tr>
<td></td>
<td>DFF-OSELM</td>
<td></td>
<td><strong>4.66 (0.95)</strong></td>
<td><strong>3.51 (0.72)</strong></td>
<td><strong>2.35 (0.75)</strong></td>
<td><strong>3.88 (1.52)</strong></td>
<td><strong>4.08 (2.34)</strong></td>
</tr>
<tr>
<td></td>
<td>AFGR-OSELM</td>
<td></td>
<td>182.99 (833.53)</td>
<td>11.02 (7.63)</td>
<td>7.39 (6.86)</td>
<td>×</td>
<td>21.29 (76.45)</td>
</tr>
<tr>
<td></td>
<td>OSELM</td>
<td></td>
<td>5.08 (1.07)</td>
<td>4.17 (0.80)</td>
<td>3.81 (0.70)</td>
<td>18.15 (8.90)</td>
<td>13.84 (8.81)</td>
</tr>
<tr>
<td></td>
<td>R-OSELM</td>
<td></td>
<td><strong>4.95 (1.68)</strong></td>
<td><strong>3.13 (1.16)</strong></td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>FR-OSELM</td>
<td></td>
<td><strong>4.52 (1.20)</strong></td>
<td><strong>3.40 (0.83)</strong></td>
<td><strong>3.18 (0.82)</strong></td>
<td><strong>13.04 (5.26)</strong></td>
<td><strong>9.71 (3.87)</strong></td>
</tr>
<tr>
<td></td>
<td>DFF-OSELM</td>
<td></td>
<td><strong>4.91 (1.29)</strong></td>
<td><strong>3.10 (0.82)</strong></td>
<td><strong>2.31 (1.28)</strong></td>
<td><strong>6.30 (3.29)</strong></td>
<td><strong>4.85 (2.37)</strong></td>
</tr>
<tr>
<td></td>
<td>AFGR-OSELM</td>
<td></td>
<td>51.84 (143.42)</td>
<td>104.47 (486.55)</td>
<td>29.03 (75.44)</td>
<td>99.64 (469.81)</td>
<td>28.95 (65.47)</td>
</tr>
</tbody>
</table>

“×” denotes nullification owing to the too large RMSE.

Table 4: Mean (SD) of the prediction RMSE of OSELM, R-OSELM, FR-OSELM, DFF-OSELM, and AFGR-OSELM on stock price data set 2.

<table>
<thead>
<tr>
<th>Hidden node (ni)</th>
<th>Model</th>
<th>Prediction steps</th>
<th>200</th>
<th>500</th>
<th>1000</th>
<th>1500</th>
<th>2800</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OSELM</td>
<td></td>
<td>224.90 (915.90)</td>
<td>27.71 (98.35)</td>
<td>40.80 (174.70)</td>
<td>305.76 (1514.01)</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>R-OSELM</td>
<td></td>
<td>5.07 (1.02)</td>
<td>4.50 (0.79)</td>
<td>4.37 (0.51)</td>
<td>18.41 (18.63)</td>
<td>14.40 (9.28)</td>
</tr>
<tr>
<td></td>
<td>FR-OSELM</td>
<td></td>
<td><strong>3.75 (0.56)</strong></td>
<td><strong>2.72 (0.74)</strong></td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>DFF-OSELM</td>
<td></td>
<td>4.35 (1.66)</td>
<td>2.98 (0.70)</td>
<td>2.72 (0.62)</td>
<td>7.83 (5.50)</td>
<td>6.26 (4.44)</td>
</tr>
<tr>
<td></td>
<td>AFGR-OSELM</td>
<td></td>
<td>4.66 (0.95)</td>
<td>3.51 (0.72)</td>
<td>2.35 (0.75)</td>
<td>3.88 (1.52)</td>
<td>4.08 (2.34)</td>
</tr>
<tr>
<td></td>
<td>OSELM</td>
<td></td>
<td>182.99 (833.53)</td>
<td>11.02 (7.63)</td>
<td>7.39 (6.86)</td>
<td>×</td>
<td>21.29 (76.45)</td>
</tr>
<tr>
<td></td>
<td>R-OSELM</td>
<td></td>
<td>5.08 (1.07)</td>
<td>4.17 (0.80)</td>
<td>3.81 (0.70)</td>
<td>18.15 (8.90)</td>
<td>13.84 (8.81)</td>
</tr>
<tr>
<td></td>
<td>FR-OSELM</td>
<td></td>
<td>4.95 (1.68)</td>
<td>3.13 (1.16)</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>DFF-OSELM</td>
<td></td>
<td><strong>4.52 (1.20)</strong></td>
<td><strong>3.40 (0.83)</strong></td>
<td><strong>3.18 (0.82)</strong></td>
<td><strong>13.04 (5.26)</strong></td>
<td><strong>9.71 (3.87)</strong></td>
</tr>
<tr>
<td></td>
<td>AFGR-OSELM</td>
<td></td>
<td>4.91 (1.29)</td>
<td><strong>3.10 (0.82)</strong></td>
<td><strong>2.31 (1.28)</strong></td>
<td><strong>6.30 (3.29)</strong></td>
<td><strong>4.85 (2.37)</strong></td>
</tr>
</tbody>
</table>

“×” denotes nullification owing to the too large RMSE.

Industrial processes exhibit some kind of time-varying behaviors, so these industrial data sets are crucial to evaluate the proposed methodologies. One is the Debutanizer column data set with a total of 2394 samples; the other is the Sulfur Recovery Unit (SRU) data set with a total of 10081 samples. The two data sets can be obtained from [34] and more details of them can be found there. These two data sets are well known and have been widely used to evaluate algorithms for dynamic system modeling and prediction in the literatures.

Tests are carried out by comparing AFGR-OSELM to OSELM, R-OSELM, FR-OSELM, and DFF-OSELM on the Debutanizer column and the SRU data sets. For the two
data sets, each model uses the same parameters. On the basis of the suggestions from [23], in our simulations the hidden nodes number is set as 20 and the number of training data for initialization is set as 100 uniformly for all the five models. Besides, the other parameters setting for R-OSELM, FR-OSELM, DFF-OSELM, and AFGR-OSELM are the same as that in the first simulation in Section 4.1 except that a different value of regularization parameter ($10^{-4}$) is used in all the four models for better stability and a smaller value of FF (0.9) is assigned in the FR-OSELM for better performance. According to the size of the data set, different online prediction steps are designed: they are 200, 500, 1000, 1500, and 2294, respectively, for the Debutanizer column, and 2000, 4000, 6000, 8000, and 9981, respectively, for the SRU.

Table 5 presents the mean and SD of the prediction RMSE over 30 independent trials of the five models for performing different prediction steps on the Debutanizer column; meanwhile the typical absolute error graphs of all the models for performing 2294-step online prediction are shown in Figure 3. From Table 5 we can see that the OSELM and the R-OSELM achieve similar performances; it means that the ill-posed problem is not serious in current situation. Though the FR-OSELM obtains good performances within the first 1000 prediction steps with the help of the forgetting mechanism, its prediction RMSE is the worst when the prediction step is equal to 1500 or 2294. The reason for this is the intrinsic instability of the algorithm itself, and it can be clearly verified through the corresponding absolute error graph given by Figure 3(c), in which several exceptional peaks occur after 1000 steps. From this example we may conclude that the FR-OSELM is worse than the OSELM in stability and it probably produces unreliable results even in well-conditioned environment. By comparison, the DFF-OSELM and the AFGR-OSELM equipped with adaptive forgetting scheme achieve much better performances, and our AFGR-OSELM works best on both prediction accuracy and stability.

Table 6 shows the mean and SD of the prediction RMSE over 30 independent trials of the five models for performing different prediction steps on the SRU, and the typical absolute error graphs of all the models for performing 9981-step online prediction are demonstrated in Figure 4. From Table 6 and Figure 4, it is clear to see that the OSELM is very unstable and the obtained mean and SD of the prediction RMSE are much larger than that of other models. Similarly, since the regularization function fades gradually with time, we can see from Figure 4(c) that the FR-OSELM is workable only within the first 170 prediction steps and fails rapidly after that; as a result the subsequent prediction RMSE are very large and completely invalid, which are represented as “×” in Table 6. Relatively, our AFGR-OSELM is quite stable and can always provide reliable results in all stages, and the obtained prediction RMSE are much smaller than those of the R-OSELM and the DFF-OSELM. From the two real world industrial data sets, we can conclude that the AFGR-OSELM is also very effective in providing stable and superior performances for predicting real systems with time-varying behaviors.

Finally, we give the processing times (CPU time) of the five OSELM-related models for performing online prediction with different prediction steps on Debutanizer column and SRU, which are shown in Tables 7 and 8, respectively. Since the computational complexities of OSELM, R-OSELM, and FR-OSELM are identical to each other, we can see that their prediction times are almost the same in the same cases. Relatively, there is an increase in prediction time for the DFF-OSELM and AFGR-OSELM because extra calculations are required for tuning the FF recursively in the online prediction process. Though the AFGR-OSELM is the least efficient among the five models, its stability and prediction accuracy are the best, so it is worth it.

5. Conclusions and Future Work

In this paper, a new AFGR-OSELM algorithm incorporating generalized regularization and adaptive forgetting factor has been developed for modeling and predicting the time-varying system in an online mode. By employing a generalized $I_2$ regularization item instead of the traditional exponential forgetting regularization item in the cost function, the obtained AFGR-OSELM can maintain a constant regularization effect without fading in the whole learning process and a persistent stability of the algorithm can be guaranteed. Moreover, with the help of the adaptive forgetting factor, the AFGR-OSELM can well track the dynamic changes of the time-varying system and timely reduce the adverse effects of the outdated samples; thus it is capable of producing desirable prediction results in time-varying environment. The effectiveness and practicability of the AFGR-OSELM are evaluated and compared with other four representative OSELM-related models by artificial and real world data sets. The experimental results indicate that the AFGR-OSELM behaves much better than its counterparts in terms of stability and prediction accuracy for time-varying system prediction. Although the proposed method has shown satisfactory results, it still can most probably be improved. As demonstrated in the above theoretical analyses and comparative experiments, compared with other models the AFGR-OSELM is superior in stability and prediction accuracy, yet its computation procedure is more complex and accordingly more computation times are required. Actually, the computational complexity of AFGR-OSELM may be further reduced with a proper approximation approach, and similar work has been done in [22] where a low complexity adaptive FF method was developed for the original OSELM. As future work, we will consider this issue further and try to provide a more efficient AFGR-OSELM.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grants nos. 61603326, 61379064, and 61273106), the National Key Technology Research and Development Program of the Ministry of Science and Technology of China (Grant no. 2014BAJ04B02), and the Open Project Foundation of Information Technology Research Base of
Table 5: Mean (SD) of the prediction RMSE of OSELM, R-OSLEM, FR-OSLEM, DFF-OSLEM, and AFGR-OSLEM on Debutanizer column.

<table>
<thead>
<tr>
<th>Model</th>
<th>200</th>
<th>500</th>
<th>Prediction steps</th>
<th>1500</th>
<th>2294</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSELM</td>
<td>7.90E - 02 (6.18E - 03)</td>
<td>8.16E - 02 (3.54E - 03)</td>
<td>1.18E - 01 (2.78E - 03)</td>
<td>1.30E - 01 (2.06E - 03)</td>
<td>1.37E - 01 (1.90E - 03)</td>
</tr>
<tr>
<td>R-OSLEM</td>
<td>8.53E - 02 (2.80E - 03)</td>
<td>7.96E - 02 (1.08E - 03)</td>
<td>1.19E - 01 (1.71E - 03)</td>
<td>1.29E - 01 (1.26E - 03)</td>
<td>1.36E - 01 (1.37E - 03)</td>
</tr>
<tr>
<td>FR-OSLEM</td>
<td>6.26E - 02 (2.19E - 03)</td>
<td>4.53E - 02 (1.85E - 03)</td>
<td>5.02E - 02 (2.52E - 03)</td>
<td>1.40E - 01 (3.89E - 02)</td>
<td>1.83E - 01 (5.33E - 02)</td>
</tr>
<tr>
<td>DFF-OSLEM</td>
<td>6.48E - 02 (3.40E - 03)</td>
<td>5.09E - 02 (1.75E - 03)</td>
<td>5.53E - 02 (1.73E - 03)</td>
<td>6.20E - 02 (3.54E - 03)</td>
<td>5.99E - 02 (2.79E - 03)</td>
</tr>
<tr>
<td>AFGR-OSLEM</td>
<td>6.36E - 02 (2.57E - 03)</td>
<td>4.15E - 02 (1.29E - 03)</td>
<td>3.70E - 02 (8.22E - 04)</td>
<td>3.67E - 02 (6.40E - 04)</td>
<td>3.83E - 02 (6.93E - 04)</td>
</tr>
</tbody>
</table>
Table 6: Mean (SD) of the prediction RMSE of OSELM, R-OSELM, FR-OSELM, DFF-OSELM, and AFGR-OSELM on SRU.

<table>
<thead>
<tr>
<th>Model</th>
<th>2000</th>
<th>4000</th>
<th>6000</th>
<th>8000</th>
<th>9981</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSELM</td>
<td>$1.94E-01$ (6.74E - 01)</td>
<td>$1.69E+00$ (8.92E + 00)</td>
<td>$9.61E - 02$ (1.36E - 01)</td>
<td>$1.43E - 01$ (4.41E - 01)</td>
<td>$9.46E - 02$ (1.88E - 01)</td>
</tr>
<tr>
<td>R-OSELM</td>
<td>$5.43E-02$ (8.44E - 04)</td>
<td>$5.06E-02$ (5.85E - 04)</td>
<td>$5.66E - 02$ (5.00E - 04)</td>
<td>$5.50E - 02$ (3.25E - 04)</td>
<td>$5.56E - 02$ (2.68E - 04)</td>
</tr>
<tr>
<td>FR-OSELM</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>DFF-OSELM</td>
<td>$2.94E-02$ (6.40E - 04)</td>
<td>$2.59E-02$ (5.77E - 04)</td>
<td>$3.09E - 02$ (5.49E - 04)</td>
<td>$3.03E - 02$ (8.49E - 04)</td>
<td>$3.00E - 02$ (6.19E - 04)</td>
</tr>
<tr>
<td>AFGR-OSELM</td>
<td>$2.75E-02$ (2.84E - 04)</td>
<td>$2.41E - 02$ (3.04E - 04)</td>
<td>$2.67E - 02$ (5.07E - 04)</td>
<td>$2.53E - 02$ (2.68E - 04)</td>
<td>$2.48E - 02$ (3.69E - 04)</td>
</tr>
</tbody>
</table>

“×” denotes nullification owing to the too large RMSE.
Figure 3: The graphs of absolute error on Debutanizer column. (a) OSELM. (b) R-OSELM. (c) FR-OSELM. (d) DFF-OSELM. (e) AFGR-OSELM.
Figure 4: The graphs of absolute error on SRU. (a) OSELM. (b) R-OSELM. (c) FR-OSELM. (d) DFF-OSELM. (e) AFGR-OSELM.
Table 7: Prediction time (seconds) of OSELM, R-OSELM, FR-OSELM, DFF-OSELM, and AFGR-OSELM on Debutanizer column.

<table>
<thead>
<tr>
<th>Model</th>
<th>Prediction steps</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>200</td>
</tr>
<tr>
<td>OSELM</td>
<td>2.03</td>
</tr>
<tr>
<td>R-OSELM</td>
<td>1.88</td>
</tr>
<tr>
<td>FR-OSELM</td>
<td>2.55</td>
</tr>
<tr>
<td>DFF-OSELM</td>
<td>2.71</td>
</tr>
<tr>
<td>AFGR-OSELM</td>
<td>7.45</td>
</tr>
</tbody>
</table>

*The values have been multiplied by 100.

Table 8: Prediction time (seconds) of OSELM, R-OSELM, FR-OSELM, DFF-OSELM, and AFGR-OSELM on SRU.

<table>
<thead>
<tr>
<th>Model</th>
<th>Prediction steps</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2000</td>
</tr>
<tr>
<td>OSELM</td>
<td>1.64</td>
</tr>
<tr>
<td>R-OSELM</td>
<td>1.71</td>
</tr>
<tr>
<td>FR-OSELM</td>
<td>1.87</td>
</tr>
<tr>
<td>DFF-OSELM</td>
<td>2.07</td>
</tr>
<tr>
<td>AFGR-OSELM</td>
<td>4.94</td>
</tr>
</tbody>
</table>

*The values have been multiplied by 10.

Civil Aviation Administration of China (Grant no. CAAC-ITRB-201401). All of these supports are appreciated.

References


