IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 11, 2020, accepted April 10, 2020, date of publication April 14, 2020, date of current version April 30, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.2987815*

# Growing and Pruning Selective Ensemble Regression for Nonlinear and Nonstationary Systems

**TONG LIU[1,2], SHENG CHEN[3,4], (Fellow, IEEE), SHAN LIANG[1,2], (Member, IEEE), AND CHRIS J. HARRIS[3]**

[1]Key Laboratory of Dependable Service Computing in Cyber Physical Society, Ministry of Education, Chongqing University, Chongqing 400044, China
[2]School of Automation, Chongqing University, Chongqing 400044, China
[3]School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K.
[4]King Abdulaziz University, Jeddah 21589, Saudi Arabia

Corresponding authors: Tong Liu (liutong42@cqu.edu.cn) and Shan Liang (lightsun@cqu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61771077, and in part by the Key Research Program of Chongqing Science and Technology Commission under Grant CSTC2017jcyjBX0025. The work of Tong Liu was supported by the Chinese Scholarship Council for funding his research at the University of Southampton.

**ABSTRACT** For a selective ensemble regression (SER) scheme to be effective in online modeling of fast-arriving nonlinear and nonstationary data, it must not only be capable of maintaining a most up to date and diverse base model set but also be able to forget old knowledge no longer relevant. Based on these two important principles, in this paper, we propose a novel growing and pruning SER (GAP-SER) for time-varying nonlinear data. Specifically, during online operation, newly emerging process state is automatically identified and a local linear model is fitted to it. This adaptive growing strategy therefore maintains a most up to date and diverse local model set. The online prediction model is then constructed as a selective ensemble from the local linear model set based on a probability metric. Moreover, a pruning strategy is derived to remove 'unwanted' out of date local linear models in order to achieve low online computational complexity without sacrificing online modeling accuracy. A chaotic time series prediction and two real-world data sets are used to demonstrate the superior online modeling performance of the proposed GAP-SER over a range of benchmark schemes for nonlinear and nonstationary systems, in terms of online prediction accuracy and computational complexity.

**INDEX TERMS** Nonlinear and nonstationary data, local linear model, growing model, pruning model, selective ensemble.

## I. INTRODUCTION

With the growing real-world applications based on fast-arriving data streams [1]–[8], online learning has been a paramount issue for regression models. The data captured by sensor networks, industrial machinery and others alike usually exhibit both nonlinear and nonstationary characteristics. The root cause of time-varying nature may be sensor drift and/or underlying process drift. Sensor drift is a temporal shift of sensors due to ageing or environment changes [9]. Process drift is resulted from changes of operating conditions, catalyst deactivation, mechanical abrasions or external climatic variations, etc. [10]. The adverse consequence of such drifts, which can be either gradual or abrupt, degrades the performance of real-time or online predictive models.

To cope with drifting effects of data, the development of predictive models with adaptive capability is necessary. A commonly used simple method is using adaptive recursive estimators, such as the recursive least square (RLS) [11]–[13] or the online sequential extreme learning machine (OS-ELM) [14]–[16], to update the predictive model's weights in real time. Another way of online adaptation is to selectively record the important data pattern to update the predictive model's structure. One popular representative is the resource allocating network (RAN). Starting from an empty set of radial basis function (RBF) nodes, the RAN adds RBF nodes with arriving input data based on their significance [17], [18]. Hence, the RAN can only grow the model size, which usually makes

The associate editor coordinating the review of this manuscript and approving it for publication was Zhiwei Gao.

it ends up with a very large model and having high complexity in online prediction, after long online adaptive operation. By contrast, starting from an initial set of RBF nodes, the fast tunable RBF [19] adaptively replaces an 'insignificant' RBF node with a new node online. Although its model size is fixed, the fast tunable RBF can better track 'local characteristics' in a nonstationary environment. The experimental results of [19] show that this fast tunable RBF method typically outperforms the RAN and the OS-ELM.

Compared with above single global modeling approaches, ensemble learning that employs multiple models to separately modeling the data subspaces has proven to be popular in online learning [6], [20]–[22]. In the area of selective ensemble learning, most of the researches focus on adaptive classification [23]–[25] and the regression problem is rarely discussed. One important issue in ensemble learning is the diversity between models [26]. Maintaining highly diverse ensemble enables covering a wide dynamical range of data space. This is utmost important during online modeling of a nonstationary process, where the process dynamics can change significantly and often result in newly emerging multiple operating regions. The ensemble of OS-ELM (EOS-ELM) [27], however, does not have this capability, as all the neural network base models are trained on the same data set and the base model structures are not updated in real-time operation. The multiple model learning framework [28] suffers from the same drawback, as all the local RBF models are initially trained, and the local RBF model structures are not updated during online operation.

To ensure diversity, base models must be statistically different. If base models represent different local regions of the data, then they are statistically different and this guarantees the diversity of the model set. For soft sensing application, the concept of local learning is introduced in [29], where the offline training data is partitioned into multiple local regions, each covered by a local model. The works [30], [31] further extend this localization strategy to the online operation, to grow new local linear models adaptively on the newly emerging process states and, therefore, for producing the adaptive online modeling with a selective ensemble from the diverse set of local linear models. Motivated by the works [30], [31] which are for a different application of soft sensor design, our recent work [32] proposes a selective ensemble based multiple local model learning (SEMLM) for nonlinear and nonstationary systems. The SEMLM adaptively identifies newly emerging characteristics of the underlying system and grows the local linear model set online accordingly. Online modeling is then carried by a selective ensemble of subset local linear models from the model candidates. The results obtained in [32] show that this SEMLM is capable producing more accurate online predictive modeling than the fast tunable RBF of [19]. A potential drawback of the SEMLM, in comparison with the fast tunable RBF, is that it may impose higher average computation time per sample (ACTpS). This is because for a highly nonstationary system and over a long period of online adaptation, the number of local linear models can grow to be very large, which may impose high computational complexity in constructing a selective ensemble model.

The main motivation of our current work is to improve the online computational complexity of the SEMLM, while retaining its capability of maintaining highly diverse local linear model set and producing highly accurate adaptive selective ensemble modeling. Clearly, for effective online learning of fast-arriving and time-varying data, learning models not only must have the ability to capture the newly occurring knowledge as fast as possible but also are able to forget the past accumulated old concepts that are no longer relevant [7], [33]. Therefore, in order to reduce online computational complexity, it is desired to remove some 'oldest' local linear models from the model set. However, this is not as simple as it appears. Any local linear model in the model set represents some local process state or knowledge that has actually appeared. The fact that a local model is not used in the most recent selective ensemble does not imply that it will not be needed in future. Our main contribution is to derive a reliable mechanism of removing 'unwanted' local models online, without sacrificing the diversity and accuracy of selective ensemble regression.

Specifically, in this paper, a growing and pruning selective ensemble regression (GAP-SER) is proposed for time-varying nonlinear data. During online operation, newly emerging process state is automatically identified from the incoming data and a local linear model is fitted to it. This growing strategy is identical to our previous SEMLM and it maintains a most up to date and diverse local model set. However, unlike our previous work [32], which constructs the selective ensemble based on the mean square error (MSE) metric, we build the selective ensemble based on a probability metric, which is capable of achieving excellent online modeling performance with very few local models selected and hence helps to maintaining low online computational complexity. Most importantly, an ensemble pruning strategy is performed to reliably remove the 'unwanted' local linear models and, therefore, to achieve lower ACTpS without sacrificing the online modeling accuracy. The above two improvements make the proposed ensemble regression particularly suitable for online modeling of nonlinear and nonstationary systems. Three case studies, 1) chaotic time series prediction, 2) online identification of a real-world industrial system, and 3) EEG data modeling, are used to demonstrate the effectiveness of the proposed GAP-SER, in comparison with a range of benchmark schemes for modeling and identification of nonlinear and nonstationary systems.

## II. GROWING AND PRUNING SELECTIVE ENSEMBLE REGRESSION

To achieve the ultimate goal of our GAP-SER, which is to produce accurate online prediction or modeling while imposing low computational complexity, we rely on the two fundamental principles, ability to maintain most up to day and diverse local linear model set and capacity of reliably removing unwanted out of date local linear models.

Three components of our GAP-SER, namely, the growing strategy, the new pruning strategy and the new selective ensemble prediction, are now detailed.

### A. GROWING STRATEGY

Given the data sample set $\{\boldsymbol{x}(t), y(t)\}_{t=1}^{N}$, where $\boldsymbol{x}(t) \in \mathbb{R}^m$ and $y(t) \in \mathbb{R}$ are the system's input vector and output, respectively, our task is to construct the local linear models $\{f_l\}_{l=1}^{L}$ that are valid in their corresponding process states represented by their respective sub-datasets $\{\boldsymbol{X}_l, \boldsymbol{y}_l\}_{l=1}^{L}$. Without loss of generality, let a local window $\mathcal{W}_{ini} = \{\boldsymbol{X}_{ini} \in \mathbb{R}^{W_G \times m}, \boldsymbol{y}_{ini} \in \mathbb{R}^{W_G}\}$ with $W_G$ consecutive samples $\{\boldsymbol{x}(t), y(t)\}_{t=t_{ini}}^{t_{ini}+W_G}$ be initially set, and a local linear model $f_{ini}$ is built on it as

$$\widehat{\boldsymbol{y}}_{ini} = f_{ini}(\boldsymbol{X}_{ini}) = \boldsymbol{\Phi}\boldsymbol{\beta}, \tag{1}$$

where $\boldsymbol{\Phi} = [\boldsymbol{1}_{W_G} \ \boldsymbol{X}_{ini}] \in \mathbb{R}^{W_G \times (1+m)}$, $\boldsymbol{1}_{W_G}$ is the $W_G$-dimensional vector whose elements are all one, and the model parameter vector $\boldsymbol{\beta} \in \mathbb{R}^{(1+m)}$ is given by the least square (LS) estimate as

$$\boldsymbol{\beta} = (\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T\boldsymbol{y}_{ini}. \tag{2}$$

The predicted error or residual vector of this local model is

$$\boldsymbol{e}_{ini} = \boldsymbol{y}_{ini} - f_{ini}(\boldsymbol{X}_{ini}) \in \mathbb{R}^{W_G}. \tag{3}$$

By shifting the data window one sample ahead, a new window $\mathcal{W}_{sft} = \{\boldsymbol{X}_{sft}, \boldsymbol{y}_{sft}\}$ is obtained, which contains the samples $\{\boldsymbol{x}(t), y(t)\}_{t=t_{ini}+1}^{t_{ini}+1+W_G}$. If the two local data regions $\mathcal{W}_{ini}$ and $\mathcal{W}_{sft}$ are not significantly different, it can be considered that the data within $\mathcal{W}_{sft}$ follow the same distribution as in $\mathcal{W}_{ini}$ and the window continues to be shifted forward. Otherwise, $\mathcal{W}_{sft}$ is considered to represent a new process state different from the one for $\mathcal{W}_{ini}$, and a new local linear model $f_{new}$ should be developed based on $\mathcal{W}_{sft}$. Let the estimation error vector produced by $f_{ini}$ on $\mathcal{W}_{sft}$ be denoted as

$$\boldsymbol{e}_{sft} = \boldsymbol{y}_{sft} - f_{ini}(\boldsymbol{X}_{sft}). \tag{4}$$

Whether the two local data regions $\mathcal{W}_{ini}$ and $\mathcal{W}_{sft}$ are similar or not can then be turned into the equivalent testing that tests whether $\boldsymbol{e}_{ini}$ and $\boldsymbol{e}_{sft}$ are significantly different or not. Since $f_{ini}$ is a linear model, $\boldsymbol{e}_{ini}$ and $\boldsymbol{e}_{sft}$ are considered not significantly different when both their means, $\mu_{ini}$ and $\mu_{sft}$, and variances, $\sigma_{ini}^2$ and $\sigma_{sft}^2$, are the same. Therefore, the two null hypotheses can be set to

$$H_0^{\mu} : \ \mu_{ini} = \mu_{sft}, \tag{5}$$

$$H_0^{\sigma^2} : \ \sigma_{sft}^2 = \sigma_{ini}^2. \tag{6}$$

The mean $\mu_{ini}$ and variance $\sigma_{ini}^2$ are estimated based on $\boldsymbol{e}_{ini}$, while $\mu_{sft}$ and $\sigma_{sft}^2$ are estimated based on $\boldsymbol{e}_{sft}$. Since $f_{ini}$ is an unbiased estimator, $\mu_{ini} = 0$ and $\sigma_{ini}^2 = \frac{1}{W_G-1}\boldsymbol{e}_{ini}^T\boldsymbol{e}_{ini}$. Assuming that $\boldsymbol{e}_{ini}$ and $\boldsymbol{e}_{sft}$ follow normal distribution, the $T$ and $\chi^2$ statistics can be constructed as

$$T_0 = \sqrt{W_G}(\mu_{sft} - \mu_{ini})/\sigma_{sft}, \tag{7}$$

$$\chi_0^2 = (W_G - 1)\sigma_{sft}^2/\sigma_{ini}^2. \tag{8}$$

According to the statistical theory, if the hypotheses $H_0^{\mu}$ and $H_0^{\sigma^2}$ are both valid, the $T_0$ statistic (7) and $\chi_0^2$ statistic (8) follow the $t$ distribution and $\chi^2$ distribution with the degree of freedom $W_G - 1$, respectively. Thus, the $t$-test and $\chi^2$-test can be utilized to test the above two hypotheses. Specifically, the conditions of accepting $H_0^{\mu}$ and $H_0^{\sigma^2}$ are

$$|T_0| < \lambda_t \text{ and } \chi_0^2 < \lambda_\chi, \tag{9}$$

where $\lambda_t$ is the threshold of the $T$ statistic for the given significance level $\alpha_t$ which satisfies $\Pr\{|T| < \lambda_t\} = 1 - \alpha_t$, while $\lambda_\chi$ is the threshold of the $\chi^2$ statistic for the given significance level $\alpha_\chi$, which satisfies $\Pr\{\chi^2 < \lambda_\chi\} = 1 - \alpha_\chi$.

Let the local model set contain $L > 1$ independent local linear models $\{f_l\}_{l=1}^{L}$, and $f_{ini} = f_L$. When one or both conditions of (9) are violated, $\mathcal{W}_{ini}$ and $\mathcal{W}_{sft}$ are significantly different, and the new local linear model $f_{new} = f_{sft}$ is different from $f_L$. We still need to test whether $f_{new}$ differs from $\{f_l\}_{l=1}^{L-1}$. This task can also be fulfilled based on the hypothesis testing. Let the predicted errors of $\mathcal{W}_{new} = \{\boldsymbol{X}_{sft}, \boldsymbol{y}_{sft}\}$ based on $f_{new}$ and $f_l$ be defined respectively by

$$\boldsymbol{e}_{new} = \boldsymbol{y}_{sft} - f_{new}(\boldsymbol{X}_{sft}), \tag{10}$$

$$\boldsymbol{e}_l = \boldsymbol{y}_{sft} - f_l(\boldsymbol{X}_{sft}), \quad 1 \le l \le L-1. \tag{11}$$

With the assumption that $\boldsymbol{e}_{new}$ and $\boldsymbol{e}_l$ follow normal distribution, the $T$ and $\chi^2$ statistics are constructed according to

$$T_l = \sqrt{W_G}(\mu_l - \mu_{new})/\sigma_l, \tag{12}$$

$$\chi_l^2 = (W_G - 1)\sigma_l^2/\sigma_{new}^2, \tag{13}$$

where $\mu_{new}$ and $\sigma_{new}^2$ are the estimated mean and variance of $\boldsymbol{e}_{new}$, while $\mu_l$ and $\sigma_l^2$ are the estimated mean and variance of $\boldsymbol{e}_l$. If the null hypotheses

$$H_l^{\mu} : \ \mu_l = \mu_{new}, \tag{14}$$

$$H_l^{\sigma^2} : \ \sigma_l^2 = \sigma_{new}^2, \tag{15}$$

are both valid, the $T_l$ statistic (12) and $\chi_l^2$ statistic (13) follow the $t$ distribution and $\chi^2$ distribution with the degree of freedom $W_G - 1$, respectively. Therefore, if there exist an $l \in \{1, 2, \cdots, L-1\}$ such that

$$|T_l| < \lambda_t \text{ and } \chi_l^2 < \lambda_\chi, \tag{16}$$

the hypotheses (14) and (15) are both valid, and $\boldsymbol{e}_{new}$ and $\boldsymbol{e}_l$ are regarded to be identical. Hence, $f_{new}$ and $f_l$ are the same model. Since $f_l$ is 'older' than $f_{new}$, we keep $f_{new}$ and delete $f_l$. On the other hand, if one or both conditions are violated $\forall l \in \{1, 2, \cdots, L-1\}$, $f_{new}$ is different from $f_l$ for $1 \le l \le L$. Thus, we have identified a new process state, and we add $f_{new}$ to the local model set by setting $L = L+1$ and $f_L = f_{new}$.

The significance levels in the statistical testings are typically set to $\alpha_t = 0.05$ and $\alpha_\chi = 0.05$. This growing strategy is summarized in Algorithm 1. Clearly, the growing model window size $W_G$ is the only algorithmic parameter to be set.

*Remark 1:* This local learning strategy is identical to the one given in our previous work [32], and it can operate both offline and online. During online operation, when the newest

data sample $\{\boldsymbol{x}(t_{next}), y(t_{next})\}$ is available, the data widow shift one sample ahead, and the corresponding learning procedure can then be carried out. It can be seen that this growing strategy is capable of identifying every newly occurring process state and, moreover, all the local linear models in the base model set are statistically different. Therefore, our local learning strategy is capable of maintaining the maximum diversity of the base model set.

### B. NEW PRUNING STRATEGY

The essence of selective ensemble regression (SER) is to accurately capture the current local characteristics, rather than to model the overall system dynamics. The base model set $\{f_l\}_{l=1}^{L}$ identified by the growing strategy of Algorithm 1 represents all the local process states occurred so far. Some of these local linear models are likely to be far away from the current data range and they are not needed in modeling the current process dynamics. Indeed, a SER constructs a online prediction model by selecting a subset of relevant local models. For fast-arriving highly nonstationary data, over a long period of online operation, the base model set is likely to become very large, and this imposes high online computational complexity in constructing SER prediction. This issue is related to the so-called stability-plasticity dilemma [34].

**Algorithm 1** Growing Strategy

**Initialization**
Collect $\mathcal{W}_{ini}$ with $W_G$ consecutive samples from historical data, and construct LS linear model $f_{ini}$ on $\mathcal{W}_{ini}$.
Calculate $\boldsymbol{e}_{ini}$, and estimate $\mu_{ini}$ and $\sigma_{ini}^2$.
Set $L = 1$, $\{\mathcal{W}_L, f_L\} = \{\mathcal{W}_{ini}, f_{ini}\}$ and $\mathcal{W}_{sft} = \mathcal{W}_L$.
**Step 1: New local model detection**
When a new data sample is available, shift $\mathcal{W}_{sft}$ one sample ahead.
Calculate $\boldsymbol{e}_{sft}$, and estimate $\mu_{sft}$ and $\sigma_{sft}^2$.
Construct $T$ and $\chi^2$ statistics using (7) and (8).
**If** both conditions of (9) are satisfied
  Go to **Step 1**.
**End if**
Construct LS linear model $f_{sft}$ on $\mathcal{W}_{sft}$.
Set $\mathcal{W}_{new} = \mathcal{W}_{sft}$ and $f_{new} = f_{sft}$.
Calculate $\boldsymbol{e}_{new}$, and estimate $\mu_{new}$ and $\sigma_{new}^2$.
**Step 2: Redundant local model deletion**
**For** $l = 1, 2, \ldots, L-1$
  Compute $\boldsymbol{e}_l$, and estimate $\mu_l$ and $\sigma_l^2$.
  Construct $T_l$ and $\chi_l^2$ statistics using (12) and (13).
  **If** both conditions of (16) are satisfied
    Delete $f_l$, set $f_i = f_{i+1}$ for $i = l, l+1, \cdots, L-1$, set $L = L-1$, then go to **Step 3**.
  **End if**
**End for**
**Step 3: Add new local model**
Set $L = L+1$, $\mathcal{W}_L = \mathcal{W}_{new}$ and $f_L = f_{new}$.
Return to **Step 1**.

An ensemble learner should not only have the ability to retain acquired knowledge (stability) but also adapt to new concept with a fast recovery (plasticity). On one hand, stability implies that a learner retains the acquired knowledge for maintaining diversity. On the other hand, plasticity requires a learner to forget part or all previous knowledge in order to capture the new knowledge from upcoming data as fast as possible.

To achieve plasticity, it is desired to remove models that do not contribute to the selective ensemble's performance based on some ensemble pruning strategy. Two major issues in ensemble pruning are: 1) decide which model should be removed, and 2) when and how frequently to remove models. In the existing literature, various ensemble pruning strategies have been proposed. For example, the removal of models can occur when the number of models exceeds a threshold [35]–[37] or when the memory usage exceeds a threshold [38]. The excluded model can be the oldest model [35], [36] or the model with worst performance [37], [39]. None of these schemes is sufficiently reliable. In highly nonstationary environments, how to reliably perform ensemble pruning is very challenging, particularly for data with seasonality and periodicity features. Removing the oldest or the worst-performance model for example may run the risk that the removed model may actually become important in future [20].

In order to improve the reliability of ensemble pruning, we propose to remove a local linear model only if it does not contribute to the SER prediction over a pruning model window with the window size $W_P > 1$, that is, a model can be removed only it is not selected by the SER for the consecutive $W_P$ samples. If a model is not needed consistently for the current $W_P$ prediction samples, the probability of it being selected in the near future prediction samples is extremely small. Therefore, removal of such a model will not affect the prediction performance in the near future. It should be emphasized again that any pruning strategy will run the risk that the removed model may become important in future. This is because the local process state represented by the removed local linear model may re-appear in future. Fortunately, our growing strategy is capable of re-discovering it when the corresponding process state re-appears in the data stream.

Since our pruning strategy is linked to the SER construction, it is necessary to start the discussion from how the SER prediction is constructed. Assume that after the online operation at sample $t$, Algorithm 1 produces the local model set $\{f_l\}_{l=1}^{L}$. A prediction window or horizon with the $p > 1$ latest labeled samples $\{\boldsymbol{x}(t-i), y(t-i)\}_{i=0}^{p-1}$ are used in constructing the SER prediction, i.e., deciding which subset of local linear models are selected. Let $\boldsymbol{e}_l(t) = [e_l(t)\ e_l(t-1) \cdots e_l(t-p+1)]^T$ be the modeling error vector of the $l$th local linear model $f_l$ over the prediction window, which is given by

$$e_l(t-i) = y(t-i) - f_l(\boldsymbol{x}(t-i)), \qquad 0 \le i \le p-1. \tag{17}$$

The performance metric of the $l$th local model is defined as

$$J_l(t) = \|\boldsymbol{e}_l(t)\|^2. \tag{18}$$

The MSE $J_l(t)$ can be conveniently transformed into a probability metric. Specifically, $J_l(t)$ is first converted to a similarity measure [40] ranging from 0 to 1 as follows

$$\mathrm{Sm}_l(t) = \frac{1}{1 + J_l(t)}. \tag{19}$$

The probability metric $\mathrm{Pr}_l(t)$ of the $l$th model is computed as the normalized similarity measure as

$$\mathrm{Pr}_l(t) = \frac{\mathrm{Sm}_l(t)}{\sum_{i=1}^{L} \mathrm{Sm}_i(t)}. \tag{20}$$

$\mathrm{Pr}_l(t)$ can be used to quantify the contribution of the $l$th local linear model to the SER, since a large value of $\mathrm{Pr}_l(t)$ indicates that the $l$th local model is a good regressor for SER and vice versu. Arrange all the $L$ local models according to their probability values in descending order as

$$\mathrm{Pr}_{l_1} \geq \mathrm{Pr}_{l_2} \geq \cdots \geq \mathrm{Pr}_{l_M} \geq \mathrm{Pr}_{l_{M+1}} \geq \cdots \geq \mathrm{Pr}_{l_L}. \tag{21}$$

We select the first $M$ best local models for constructing the SER when the criterion

$$1 - \sum_{m=1}^{M} \mathrm{Pr}_{l_m}(t) < \varepsilon, \tag{22}$$

is met, where $0 < \varepsilon < 1$ is a desired tolerance. These selected $M$ local linear models are then used to construct the SER prediction for the next sample $t_{next} = t + 1$, which is detailed in the next subsection. Note that this SER prediction construction is based on the probability metric (20), which is different from the normalized MSE metric used in our previous SEMLM [32].

The above discussion also suggests a pruning strategy. Specifically, the local models to be removed should be the $l_L$th to $l_{M+1}$th models that are not selected to form the SER for the prediction at $t_{next}$. However, pruning a model based on its 'one-sample' prediction performance may not be sufficiently reliable. Note that it is always desired to introduce a 'memory depth' for an ensemble learner. In our growing strategy, a local linear model is constructed upon a data window with window size $W_G$. Within this data window, the process is considered to be stationary. Similarly, we introduce a data window for pruning with the window size $W_P$. If a local model is never selected over the consecutive $W_P$ prediction samples, then it can be removed with high confidence.

Our pruning strategy is listed in Algorithm 2. Since $p$ and $\varepsilon$ are the algorithmic parameters of the selective ensemble prediction, the only algorithmic parameter of our pruning strategy is $W_P$. We can conveniently set $W_P = W_G$.

*Remark 2:* Since the newest local linear model $f_L$ represents the latest data, it is highly desired to retain it. Therefore, we slightly modify the selection procedure by always retaining $f_L$. Consequently, $\mathrm{count}_L$ in Algorithm 2 is always set to zero. To ensure the diversity and hence prediction accuracy of the ensemble, a minimal number of local models $L_{\min}$ should be guaranteed. Accordingly, pruning in Algorithm 2 can be modified so that maximally only the $(L - L_{\min})$ oldest models in $\Gamma$ can be removed. More specifically, if the number of removal model candidates (given by index counter in Algorithm 2) in $\Gamma$ is larger than $L - L_{\min}$, only $(L - L_{\min})$ of them can be removed. Appropriate value of $L_{\min}$ is closely linked to the growing window size $W_G$. If the value of $W_G$ is small, each data window only represents a small region of local characteristics and, therefore, the size of the model library, i.e., $L_{\min}$, should be sufficiently large to cover the entire data range. By contrast, if $W_G$ is large, a small size of the model library is sufficient to cover all the process states. In most cases, there exists a training data set for discovering the local process characteristics and identifying the local models to represent these local process states. In this case, we can set the minimum size of the model library $L_{\min}$ to the size of the local model set identified during training.

**Algorithm 2** Pruning Strategy

**Initialization**
Give $W_P$, set counters of all local models $\mathrm{count}_l = 0$ for $1 \leq l \leq L$, set $t = t_{ini}$ and index $= 0$.
**Step 1: Pruning in pruning model window**
Perform relevant operations of SER construction.
**If** $(t - t_{ini} \leq W_P)$
  **For** $l = 1, 2, \ldots, L$
    **If** $f_l$ is not selected at current sample $t$
      $\mathrm{count}_l = \mathrm{count}_l + 1$.
    **End if**
  **End for**
  Set $t = t + 1$ and go to **Step 1**.
**Else**
  **For** $l = 1, 2, \ldots, L$
    **If** $\mathrm{count}_l = W_G$
      Add $l$ to pruning model index set $\Gamma$, and set index $=$ index $+ 1$.
    **End if**
  **End for**
  Delete $f_l$ for all $l \in \Gamma$, and set $L = L -$ index.
**End if**
**Step 2: Pruning model window update**
Clear counters for all local models, set $t_{ini} = t$ and index $= 0$, and go to **Step 1**.

### C. NEW ADAPTIVE SELECTIVE ENSEMBLE PREDICTION

After the online operations at sample $t$, the set of the local linear models $\{f_l\}_{l=1}^{L}$ have been produced. At the next sample of $t_{next} = t + 1$, the task of online modeling is to produce the model prediction $\widehat{y}(t_{next})$ for the process's true output $y(t_{next})$, given the process input $\boldsymbol{x}(t_{next})$ and the available local model set $\{f_l\}_{l=1}^{L}$. We adopt an ensemble of the selected $M$ local linear models from the model library $\{f_l\}_{l=1}^{L}$ based on the $p$ latest labeled data $\{\boldsymbol{x}(t-i), y(t-i)\}_{i=0}^{p-1}$.

Recalling (17) to (22), the $M$ selected local linear models yield the $M$ model outputs

$$\widehat{y}_{l_m}(t-i) = f_{l_m}(\boldsymbol{x}(t-i)), \quad 1 \leq m \leq M, \tag{23}$$

for $0 \le i \le p-1$. The estimate $\widehat{y}(t-i)$ of the process output $y(t-i)$ is given as the weighted summation of the $M$ selected subset models, which is computed by

$$\widehat{y}(t-i) = \sum_{m=1}^{M} \theta_m(t)\widehat{y}_{l_m}(t-i), \quad 0 \le i \le p-1, \tag{24}$$

where nonnegative $\theta_m(t)$ is the combining coefficient for the $m$th selected local model, and the combining coefficients must satisfy the constraint

$$\sum_{m=1}^{M} \theta_m(t) = 1. \tag{25}$$

The estimation errors

$$e(t-i) = y(t-i) - \widehat{y}(t-i), \quad 0 \le i \le p-1, \tag{26}$$

are utilized to determine the combining coefficients. Specifically, the optimal combining coefficients can be obtained by minimizing the LS cost function

$$V(t) = \frac{1}{2}\sum_{i=0}^{p-1} e^2(t-i), \tag{27}$$

subject to the constraint (25). Because of $\sum_{m=1}^{M} \theta_m(t) = 1$,

$$\begin{aligned} V(t) &= \frac{1}{2}\sum_{i=0}^{p-1}\left(y(t-i) - \sum_{m=1}^{M}\theta_m(t)\widehat{y}_{l_m}(t-i)\right)^2 \\ &= \frac{1}{2}\sum_{i=0}^{p-1}\left(\sum_{m=1}^{M}\theta_m(t)y(t-i) - \sum_{m=1}^{M}\theta_m(t)\widehat{y}_{l_m}(t-i)\right)^2 \\ &= \frac{1}{2}\sum_{i=0}^{p-1}\left(\sum_{m=1}^{M}\theta_m(t)e_{l_m}(t-i)\right)^2 = \frac{1}{2}\boldsymbol{\theta}^T(t)\bar{\boldsymbol{E}}(t)\boldsymbol{\theta}(t), \end{aligned} \tag{28}$$

where $\boldsymbol{\theta}(t) = \left[\theta_1(t)\cdots\boldsymbol{\theta}_M(t)\right]^T$ and $\bar{\boldsymbol{E}}(t)$ is the estimated error covariance matrix given by

$$\bar{\boldsymbol{E}}(t) = \sum_{i=0}^{p-1}\begin{bmatrix} e_{l_1}^2(t-i) & \cdots e_{l_1}(t-i)e_{l_M}(t-i) \\ \vdots & \ddots \quad \vdots \\ e_{l_1}(t-i)e_{L_M}(t-i)\cdots & e_{l_M}^2(t-i) \end{bmatrix}. \tag{29}$$

The problem of determining the optimal $\boldsymbol{\theta}(t)$ can then be formulated as the following optimization

$$\begin{aligned} &\min_{\boldsymbol{\theta}} \frac{1}{2}\boldsymbol{\theta}^T(t)\bar{\boldsymbol{E}}(t)\boldsymbol{\theta}(t), \\ &s.t. \sum_{m=1}^{M}\theta_m(t) = 1. \end{aligned} \tag{30}$$

The Lagrangian function for the optimization (30) is given by

$$L\big(\boldsymbol{\theta}(t);\gamma\big) = \frac{1}{2}\boldsymbol{\theta}^T(t)\bar{\boldsymbol{E}}(t)\boldsymbol{\theta}(t) + \gamma\big(\mathbf{1}_M^T\boldsymbol{\theta}(t) - 1\big), \tag{31}$$

where $\gamma > 0$ is a Lagrange multiplier, and $\mathbf{1}_M = [1\cdots 1]^T \in \mathbb{R}^M$. Letting $\frac{\partial}{\partial\boldsymbol{\theta}(t)}L = \mathbf{0}_M$ yields

$$\bar{\boldsymbol{E}}(t)\boldsymbol{\theta}(t) + \gamma\,\mathbf{1}_M = \mathbf{0}_M, \tag{32}$$

where $\mathbf{0}_M = [0\cdots 0]^T \in \mathbb{R}^M$. This suggests that the optimal combining vector $\widehat{\boldsymbol{\theta}}$ can be obtained as follows. First, calculate

$$\widetilde{\boldsymbol{\theta}}(t) = \bar{\boldsymbol{E}}^{-1}(t)\mathbf{1}_M, \tag{33}$$

which is followed by the normalization

$$\widehat{\theta}_m(t) = \frac{1}{\sum_{j=1}^{M}\widetilde{\theta}_j(t)}\widetilde{\theta}_m(t), \quad 1 \le m \le M. \tag{34}$$

The prediction $\widehat{y}(t_{next})$ for the process's true output $y(t_{next})$ is produced as the selected ensemble

$$\widehat{y}(t_{next}) = \sum_{m=1}^{M}\widehat{\theta}_m(t)f_{l_m}\big(\boldsymbol{x}(t_{next})\big). \tag{35}$$

Algorithm 3 summarizes the adaptive selective ensemble based prediction using our GAP-SER, where the prediction horizon $p$ and the desired threshold $\varepsilon$ are the two algorithmic parameters of selective ensemble prediction construction.

**Algorithm 3** Adaptive Prediction Using GAP-SER

1: **Initialization**
2: At beginning of online operation, local model set $\{f_l\}_{l=1}^{L_{\min}}$ has been constructed, otherwise give value of $L_{\min}$.
3: Give $W_G$, $p$ and $\varepsilon$, set $W_P = W_G$.
4: **Step 1: Online prediction**
5: Give input $\boldsymbol{x}(t_{next})$ at new sample time $t_{next} = t+1$.
6: Calculate probability $\mathrm{Pr}_l(t)$ of each local model using (20) for $1 \le l \le L$.
7: Select $M$ subset models until termination criterion (22) is satisfied.
8: Calculate error covariance matrix $\bar{\boldsymbol{E}}(t)$ using (29).
9: Calculate optimal combining coefficients $\widehat{\boldsymbol{\theta}}(t)$ using (33) and (34).
10: Predict true system output $y(t_{next})$ with selective ensemble prediction (35).
11: **Step 2: Online pruning**
12: Perform relevant pruning operations.
13: **Step 3: Online growing**
14: When $y(t_{next})$ is available, add $\{\boldsymbol{x}(t_{next}), y(t_{next})\}$ to dataset with $t = t+1$.
15: Carry out relevant growing operations to adapt local model set.
16: Set $t_{next} = t_{next} + 1$, and go to **Step 1**.

*Remark 3:* The adaptive selective ensemble prediction of Algorithm 3 is very different from the one given in our previous work [32]. First, the probability metric is used in our present work which is different from the selection metric of [32]. More importantly, unlike the scheme of [32], which only performs adaptive local modeling by growing the local linear model set, we not only perform adaptive model set growing but also carry out reliable adaptive local model set pruning. This significantly reduces the online computational complexity of the adaptive selective ensemble prediction, without sacrificing the prediction accuracy.

## III. EXPERIMENTAL RESULTS

Experiments involving a chaotic time series prediction and two real-world data sets are performed to evaluate the proposed GAP-SER, and the results are compared with existing online modeling approaches, which include single global

nonlinear modeling schemes of the RAN [17], the OS-ELM with RBF nodes [14], [15] and the fast tunable RBF [19] as well as the ensemble modeling schemes of the EOS-ELM with RBF nodes [27] and our recent SEMLM [32]. The MSE metric

$$MSE(t) = \frac{1}{t}\sum_{i=1}^{t}\big(y(i) - \widehat{y}(i)\big)^2, \tag{36}$$

is utilized to evaluate the online prediction performance, where $\widehat{y}(i)$ denotes the model prediction for $y(i)$. The online computational complexity of an adaptive modeling method is quantified by its online ACTpS. The experiments are carried out on Matlab 2017a, running on a PC with i7-3770 3.40 GHz processor of 4 cores and 16 GB of RAM.

### A. ONLINE CHAOTIC TIME SERIES PREDICTION

#### 1) DATA DESCRIPTION

Lorzen chaotic time series [41] is governed by the three differential equations as

$$\begin{cases} \dfrac{d\,x(t)}{d\,t} = a(y(t) - x(t)), \\ \dfrac{d\,y(t)}{d\,t} = cx(t) - x(t)z(t) - y(t), \\ \dfrac{d\,z(t)}{d\,t} = x(t)y(t) - bz(t), \end{cases} \tag{37}$$

where $a$, $b$ and $c$ are the parameters that control the behaviour of Lorzen system. In our experiments, three cases are considered, and they are 1) Lorenz series with fixed parameters (LSF) $a = 10$, $b = 8/3$ and $c = 28$; 2) Lorenz series with time-varying parameters (LSTV): $a = 10$ and

$$\begin{cases} b = \dfrac{4 + 3(1 + \sin(0.1t))}{3}, \\ c = 25 + 3\big(1 + \cos\big(2^{0.001t}\big)\big); \end{cases} \tag{38}$$

and 3) Lorenz series with time-based drift (LSTD): $a = 10$, $b = 8/3$ and $c = 28$ but $\{y(t)\}$ are weighted by an exponential time-based drift to obtain the new series $\{\widetilde{y}(t)\}$ according to

$$\widetilde{y}(t) = 1.1^{0.01t} y(t), \tag{39}$$

which is used in prediction, rather than the original $y(t)$. The time series $\{\widetilde{y}(t)\}$ is even more nonstationary than $\{y(t)\}$ of 1) and 2). In particular, the dynamic range of $\widetilde{y}(t)$ changes from $[-20,\ 20]$ initially to $[-2000,\ 2000]$ in the end.

The fourth-order Runge-Kutta method with a step size of 0.01 is used to generate the samples, and only $Y$-dimension samples $\{y(t)\}$ are used for time-series prediction. The 60-steps ahead prediction is considered, which predicts $y(t)$ with the past samples

$$\boldsymbol{x}(t) = \big[y(t-60)\ y(t-66)\ y(t-72)\ y(t-78)\big]^T. \tag{40}$$

In the LSTD case, $y(t)$ is replaced by $\widetilde{y}(t)$. In all the simulations, after a long initial period, 4000 samples are generated. The first 1000 samples are used for initial training and the last 3000 samples are employed for online prediction. Noted that the RAN, the fast tunable RBF, our previous SEMLM and our proposed GAP-SER do not really need such a large number of training samples but the OS-ELM and EOS-ELM need, as the ELM modeling must contain a large number of hidden nodes to cover the entire input space. For each time series, 100 independent realizations are generated. The performance of each method are presented by its mean and standard deviation (STD) of the test MSE and ACTpS over 100 realizations.

#### 2) INITIAL TRAINING

The RAN does not really need training and may be applied directly to prediction. Since we have training data, we can apply the RAN to the training data and obtain an initial RAN model. In the following, 'RAN' represents the RAN without initial training and 'RANini' denotes the RAN with initial training. Similarly, our GAP-SER may be applied directly to online prediction from scratch. Algorithm 1 will gradually build up a base model set. However, the online prediction accuracy may be poor during this built-up period, as there may exist insufficient number of base local models. In real life, a prediction model can only be applied to online prediction, if the model is known to at least match well the underlying process's past dynamics. Therefore, in practice, initial training is somewhat necessary and always desired. The same discussion also applies to the SEMLM. For the OS-ELM, we randomly select a large number of training input data points as its centers to cover the input space and determine its weights by the LS estimate. For the EOS-ELM, we use 5-model ensemble and train each base model similarly to the OS-ELM. For the fast tunable RBF, the training is done by the orthogonal least squares algorithm [42] to construct a small RBF model. For the SEMLM and GAP-SER, the initial local linear model set is obtained by Algorithm 1. Fig. 1 shows the influence of the window size $W_G$ on the number of local linear models obtained during training.

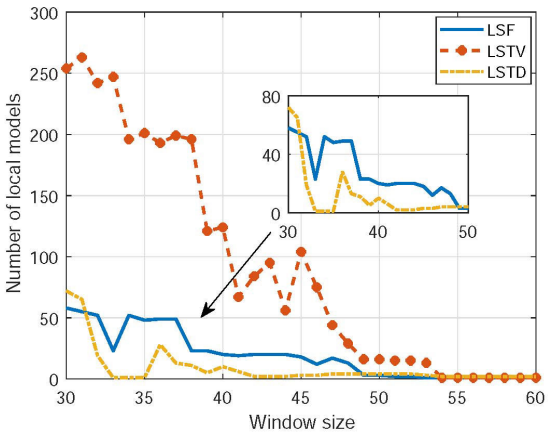



**FIGURE 1.** Influence of window size $W_G$ on number of local models obtained by Algorithm 1 for three training datasets of Lorenz time series.

#### 3) ONLINE PREDICTION PERFORMANCE COMPARISON

Three algorithmic parameters of the GAP-SER, namely, the growing window size $W_G$, the prediction horizon $p$ and the

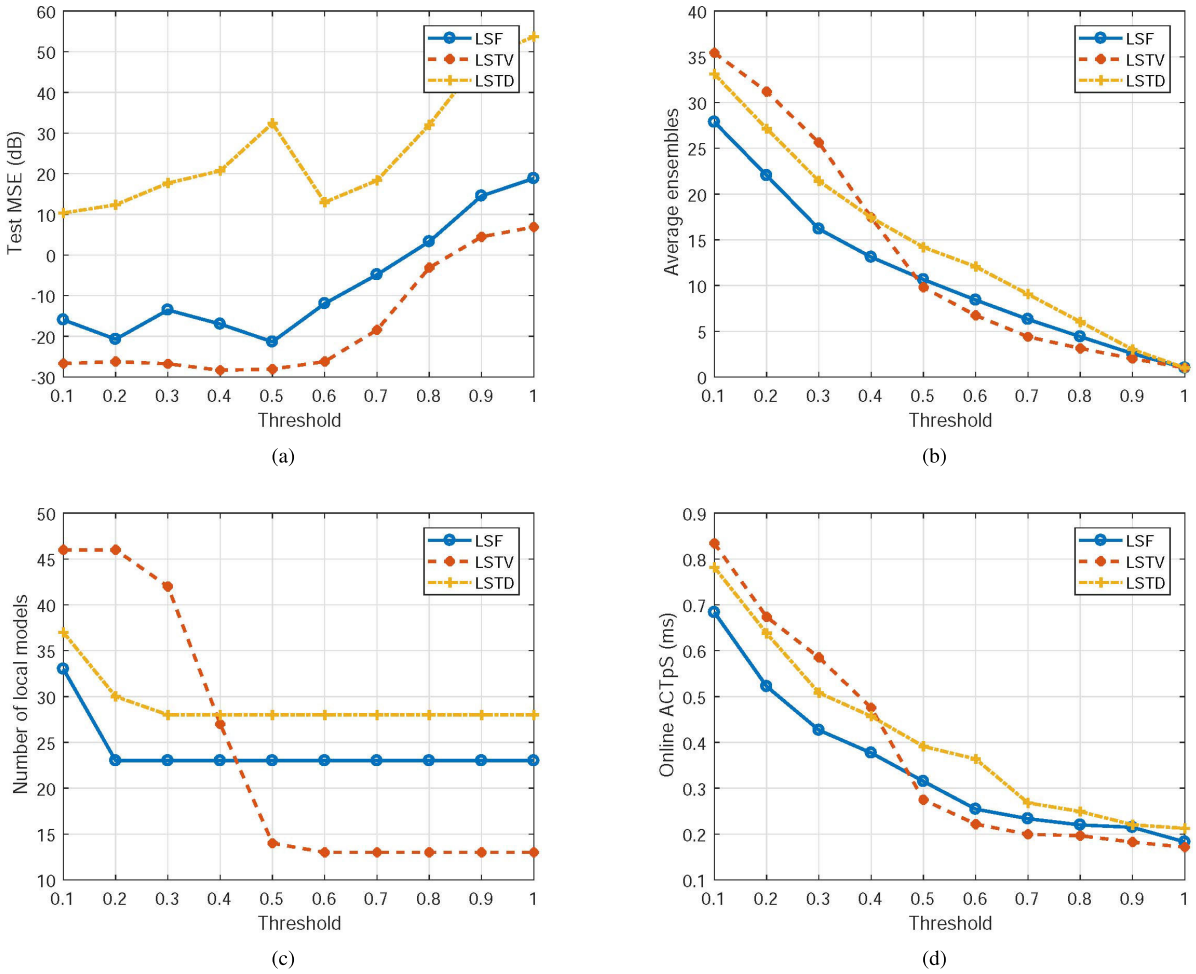



**FIGURE 2.** Influence of threshold $\varepsilon$ on: (a) test MSE, (b) size of selective ensemble, (c) number of local models, and (b) ACTpS, for online prediction of Lorenz time series.

threshold $\varepsilon$ for constructing selective ensemble, need to be set. Note that the pruning widow size is set to $W_P = W_G$. The impacts of $W_G$ and $p$ on adaptive modeling and online prediction are similar to the SEMLM, which is detailed in our previous work [32]. Generally, model built on a large window size is more stable or accurate but may not respond quickly to data changes [43]. Therefore, selecting an appropriate $W_G$ is a trade off between stability and adaptive capability. $W_G = 38$, 53 and 36 are selected empirically for the LSF, LSTV and LSTD, respectively. The choice of $p$ typically trades off the computational complexity and the robustness against noise [32]. Also since a smaller $p$ can better track local characteristics, which is beneficial in fast time-varying environments, we choose a small $p = 5$ empirically. Note that $\varepsilon$ is particularly important, as it not only influences the number of subset models for selective ensemble but also determines how many models to be removed. Given $W_G$ and $p$, the impact of $\varepsilon$ is investigated. It can be seen from Fig. 2 (a) that the test MSE increases with $\varepsilon$. The reason is twofold. 1) The number of subset models selected decreases as $\varepsilon$ increases, which can be seen from Fig. 2 (b). When $\varepsilon = 1$, only one local model is used for online prediction. 2) As $\varepsilon$ increases, more models are removed from the model set and hence the number of local models $L$ decreases, until $L$ reaches the minimum $L_{\min}$, as can be seen from Fig. 2 (c). Basically, increasing $\varepsilon$ enhances the prediction accuracy at the cost of higher online computational complexity, as clearly illustrated by Fig. 2 (a) and Fig. 2 (d). Taking into account both prediction accuracy and computational complexity, $\varepsilon = 0.5$, 0.5 and 0.6 are chosen for the LSF, LSTV and LSTD, respectively.

We set the algorithmic parameters of the SEMLM in a similar way. For the fast tunable RBF [19], the node replacement threshold and the number of latest data points for weight adaptation are empirically chosen as $10^{-6}$ and 5, respectively, while the step size and the maximum number of iterations for its iterative search procedure are empirically set as 0.01 and 5, respectively. For the RAN [18], its algorithmic parameters for online modeling, namely, the maximum and minimum center distance thresholds, the error threshold and the decay constant, are carefully tuned for each time series.

Figs. 3 to 5 show the learning curves of various schemes for the LSF, LSTV and LSTD test datasets, respectively,

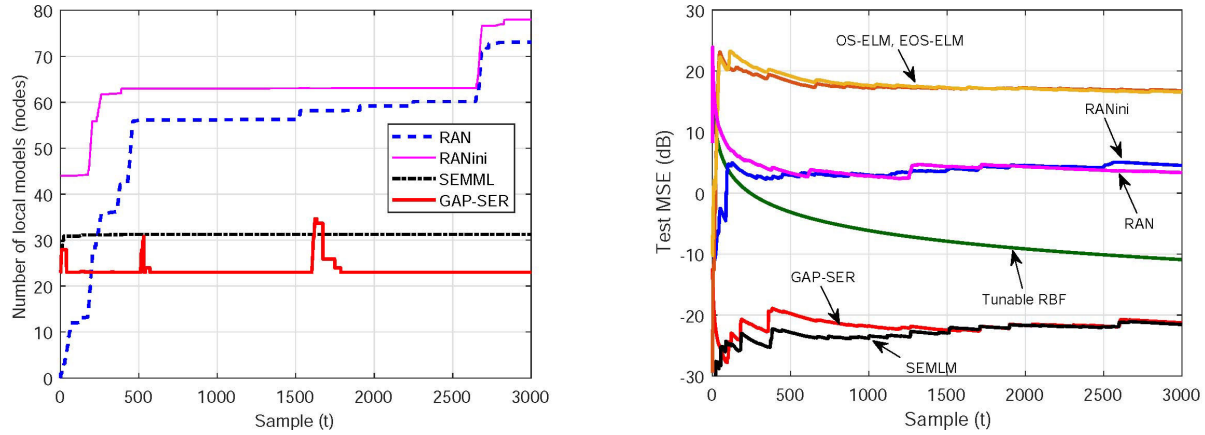



**FIGURE 3.** Online prediction of Lorenz time series with fixed parameters: (a) average model size learning curves, and (b) average MSE learning curves.

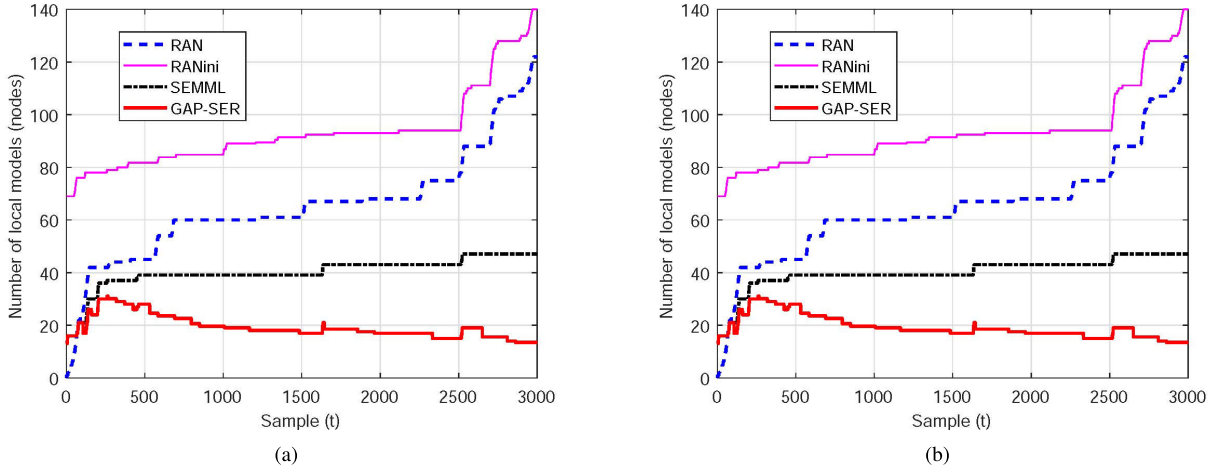



**FIGURE 4.** Online prediction of Lorenz time series with time-varying parameters: (a) average model size learning curves, and (b) average MSE learning curves.

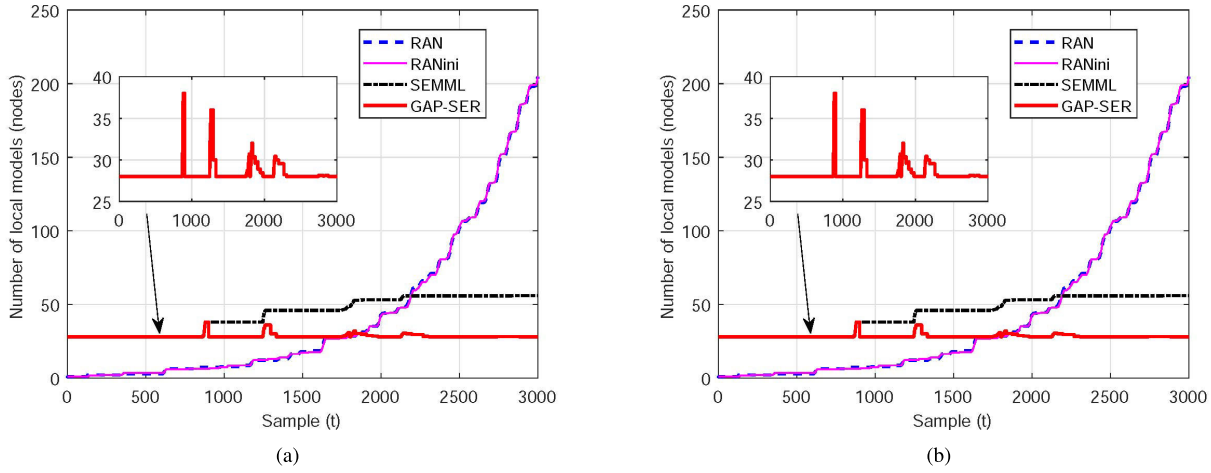



**FIGURE 5.** Online prediction of Lorenz time series with time-based drift: (a) average model size learning curves, and (b) average MSE learning curves.

in terms of average test MSE and model set size/number of RBF nodes. Since the numbers of RBF nodes are fixed for the OS-ELM, EOS-ELM and fast tunable RBF, only the learning curves of model set size/number of RBF nodes are shown for the SEMLM, GAP-SER and RAN. Table 1 further compares the performance of prediction accuracy and

**TABLE 1. Lorenz time series prediction: comparison of online prediction and adaptive modeling performance (average±STD) for the OS-ELM, EOS-ELM, RAN, fast tunable RBF, SEMLM and proposed GAP-SER.**

| Dataset | Model | MSE (dB) | Online ACTpS (ms) | Models/Nodes | | Average ensemble size |
|---|---|---|---|---|---|---|
| | | | | Initial | Final | |
| LSF | OS-ELM | 16.83±0.06 | 6.12±0.73 | 500 | 500 | - |
| | | 17.19±0.04 | 37.80±1.73 | 1000 | 1000 | - |
| | EOS-ELM | 16.76±0.03 | 58.45±2.14 | 5 × 500 | 5 × 500 | 5 |
| | RAN | 3.41±0.44 | 0.76±0.03 | 0 | 73.07±0.43 | - |
| | RANini | 4.52±0.40 | 0.74±0.03 | 44.01±0.82 | 77.96±0.28 | - |
| | Tunable RBF | **-10.90±1.68** | **0.17±0.01** | **10** | **10** | - |
| | SEMLM | **-21.54±0.17** | **0.68±0.03** | **23±0** | **31.23±1.31** | **30.18±1.20** |
| | GAP-SER | **-21.27±0.64** | **0.29±0.01** | **22.99±0.10** | **22.99±0.10** | **10.63±0.09** |
| LSTV | OS-ELM | 10.87±0.01 | 6.21±0.31 | 500 | 500 | - |
| | | 10.86±0.01 | 37.14±0.22 | 1000 | 1000 | - |
| | EOS-ELM | 11.01±0.01 | 58.12±1.45 | 5 × 500 | 5 × 500 | 5 |
| | RAN | 3.83±0.02 | 0.66±0.01 | 0 | 122±0 | - |
| | RANini | 4.21±0.04 | 1.02±0.02 | 69±0 | 139.97±0.17 | - |
| | Tunable RBF | **-13.48±0.56** | **0.17±0.01** | **10** | **10** | - |
| | SEMLM | **-27.31±0.06** | **0.83±0.06** | **13±0** | **47.06±0.23** | **38.50±0.21** |
| | GAP-SER | **-27.42±0.63** | **0.24±0.01** | **13±0** | **13.47±0.50** | **9.88±0.07** |
| LSTD | OS-ELM | 52.80±0.14 | 5.89±0.11 | 500 | 500 | - |
| | | 52.73±0.15 | 38.17±0.22 | 1000 | 1000 | - |
| | EOS-ELM | 52.72±0.14 | 57.32±2.76 | 5 × 500 | 5 × 500 | 5 |
| | RAN | 49.32±0.45 | 0.52±0.07 | 0 | 204.70±15.67 | - |
| | RANini | 40.92±1,81 | 0.40±0.05 | 1±0 | 204.78±14.20 | - |
| | Tunable RBF | **19.90±4.33** | **0.16±0.01** | **10** | **10** | - |
| | SEMLM | **17.01±0.97** | **0.82±0.14** | **28±0** | **56.01±17.51** | **38.96±6.37** |
| | GAP-SER | **13.24±1.16** | **0.32±0.02** | **28±0** | **28±0** | **12.12±0.11** |

online computational complexity achieved by the OS-ELM, EOS-ELM, RAN, fast tunable RBF, SEMLM and GAP-SER.

Not surprisingly, the OS-ELM and EOS-ELM perform poorly, in terms of both test MSE and ACTpS. As expected, the EOS-ELM imposes significantly higher ACTpS than the OS-ELM and yet it does not necessarily have better online prediction accuracy than the latter. This is because the EOS-ELM of [27] does not have necessary diversity capability, as all the base RBF models are trained on the same data. The results show that the RAN outperforms the OS-ELM and EOS-ELM considerably, in terms of both test MSE and ACTpS. Interestingly, the RANini does not always achieve better prediction accuracy than the RAN. For the LSF and LSTV prediction, for instance, the test MSE of the RANini is poorer than that of the RAN. This may due to that for learning in a nonstationary environment, a model with small memory depth is better to capture the current signal dynamics. Also the RANini does not always impose higher online computational complexity than the RAN. In the LSF and LSTD predictions, the RANini actually has lower ACTpS than the RAN.

The fast tunable RBF, the SEMLM and the proposed GAP-SER are in a complete different league, and they dramatically outperform the OS-ELM, EOS-ELM and RAN, in terms of both prediction accuracy and online computational complexity. Specifically, the fast tunable RBF of [19] with its fast adaptive capability and small model size is capable of achieving excellent prediction accuracy while imposing the lowest ACTpS. Owing to its fast adaptation capability and maximum diversity property, the SEMLM of [32] outperforms the fast tunable RBF, in terms of test MSE. The SEMLM however imposes higher ACTpS than the fast tunable RBF. The reason is that the SEMLM only grows the local linear model set online. During long online operation, the size of the local linear model set may become large and this causes high computational complexity in constructing selective ensemble prediction. By contrast, our GAP-SER with its reliable pruning strategy is capable of removing 'out-of-date' models from the local linear model set, and consequently it imposes considerably smaller computational complexity in constructing selective ensemble prediction, without sacrificing prediction accuracy, in comparison with the SEMLM. In fact, the GAP-SER may even outperform the SEMLM, in terms of test MSE, particularly for highly nonstationary data, such as the LSTV and LSTD predictions. Observe that the GAP-SER can achieve comparable ACTpS with the very efficient fast tunable RBF.

### *B. ONLINE IDENTIFICATION OF MICROWAVE HEATING SYSTEM*

#### 1) SYSTEM DESCRIPTION

Microwave heating process (MHP) is a typical nonlinear and time-varying thermal process [44]–[47]. Since MHP involves multiple physical fields coupling and its inner electromagnetic field distribution is normally unknown [44], data-driven

modeling technique offers a practical means of MHP identification [48]–[50]. Temperature is a crucial measurement during the operation of MHP, as thermal runaway often occurs due to the time-varying physicochemical properties of material [46]. With the increase of the medium temperature, its dielectric loss increases dramatically, which conversely poses a positive feedback to temperature increase. Therefore, accurate online temperature prediction is vital to detect thermal runaway in advance.

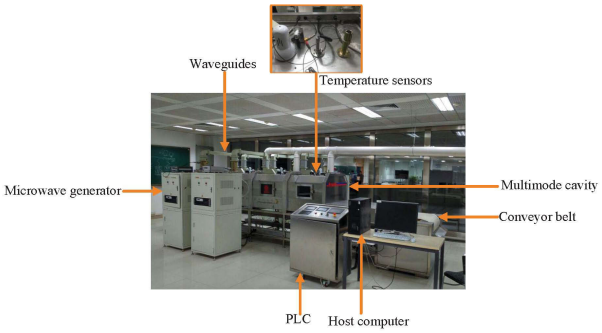



**FIGURE 6.** An industrial microwave heating system.

Fig. 6 illustrates a real-world industrial microwave heating system [48], [51], which consists of five microwave generators and waveguides, temperature measurement sensors and the control system hosted in a programmable logic controller (PLC). Microwave generated by each microwave generator is transmitted through the corresponding waveguide, which is fed into the cavity and absorbed by the heated material. Each microwave generator has a maximum power supply of 3 kW at 2.45 GHz. The material is continuously transported through cavity by the conveyor belt, whose speed can be adjusted by a motor driver. Three fiber optical sensors (FOSs), denoted as FOS1 to FOS3, are placed at three different key locations using microwave transparent taps to online record multiple-points of temperature. During the real-time operation of this MHP, the control center receives the measured temperature values from the FOSs, and sends control commends, which include the five microwave powers $u_{p_i}(t)$, $1 \le i \le 5$, for the five microwave generators as well as the conveyor speed $v(t)$ to the cavity. Thus, the control inputs to this MHP are given by

$$\boldsymbol{u}(t) = \left[u_{p_1}(t)\ u_{p_2}(t)\ u_{p_3}(t)\ u_{p_4}(t)\ u_{p_5}(t)\ v(t)\right]^T. \quad (41)$$

Each FOS measures the temperature $y_{s_j}(t)$ at the FOS's location, where $1 \le j \le 3$. Because of near instantaneous response of MHP, the temperature $y_{s_j}(t)$ at the $j$th FOS's location can be adequately represented by [48], [51]

$$y_{s_j}(t) = f_{nl-ns,j}(\boldsymbol{x}_j(t);\, t), \quad (42)$$

where $f_{nl-ns,j}(\cdot;\, t)$ represents the corresponding unknown nonlinear and time-varying system mapping with the input

$$\boldsymbol{x}_j(t) = \left[y_{s_j}(t-1)\ \boldsymbol{u}^T(t-1)\right]^T \in \mathbb{R}^7. \quad (43)$$

From large amount of data collected from this industrial microwave heating system, we use three datasets from the three FOSs, and each data set contains 3000 data samples. We first normalize the five microwave power inputs and the temperature measurements according to

$$\bar{u}_{p_i}(t) = \frac{u_{p_i}(t)}{1000}, \quad 1 \le i \le 5, \quad (44)$$

$$\bar{y}_{s_j}(t) = \frac{y_{s_j}(t) - y_{\min,s_j}}{y_{\max,s_j} - y_{\min,s_j}}, \quad 1 \le j \le 3, \quad (45)$$

where $y_{\min,s_j}$ and $y_{\max,s_j}$ are the minimum and maximum temperature measurements of the $j$th FOS, respectively. For each FOS's dataset, we use the first 1000 samples for training and and the last 2000 samples for online prediction.

#### 2) ONLINE IDENTIFICATION PERFORMANCE COMPARISON

Similar to Lorenz time series prediction, we first carry out initial training for all the schemes compared and empirically choose appropriate values for the algorithmic parameters of the SEMLM and GAP-SER. Since the system dynamics do not change as quickly as Lorenz chaotic time series, the prediction horizon $p$ can be larger to enhance model stability, and we choose $p = 25$ empirically. For the FOS$_i$, $1 \le i \le 3$, the window size and threshold are empirically set to $W_G = 22$ and $\varepsilon = 0.5$, $W_G = 22$ and $\varepsilon = 0.5$, and $W_G = 21$ and $\varepsilon = 0.8$, respectively. For the tunable RBF, the node replacement threshold and the weight update innovation length are chosen as $10^{-4}$ and 5, respectively, while the step size and iteration number are set to 0.01 and 5, respectively. The algorithmic parameters of the RAN are also careful chosen.

Figs. 7 to 9 depict the test MSE and model size learning curves for the three FOSs' data, while Table 2 compares the online identification performance of all the modeling methods, in terms of prediction accuracy and online computational complexity. From these results, the same observations as those for Lorenz chaotic time series can be drawn. In particular, our GAP-SER imposes the lowest ACTpS, while achieving the test MSE as good as the SEMLM.

### C. EEG DATA MODELING

#### 1) DATA DESCRIPTION

Analysis of electroencephalographic (EEG) time series is a practical example of nonlinear and nonstationary system identification [52], [53]. We use a real EEG time series $\{y(t)\}$ available from the University of Bonn [54]. The sampling rate is 173.61 Hz, and we construct a dataset $D_N = \{\boldsymbol{x}(t), y(t)\}_{t=1}^N$ with 10 seconds of the signal, a total of 1730 samples, where $\boldsymbol{x}(t) = [y(t-1)\ y(t-2)\ y(t-3)\ y(t-4)]^T$. The first 5 seconds of observations with 865 data pairs are used for training, and the rest 5 seconds of observations, also having 865 data pairs, are used for testing.

#### 2) ONLINE MODELING PERFORMANCE COMPARISON

Only the RAN, fast tunable RBF, SEMLM and GAP-SER are compared, as the online performance of the OS-ELM and EOS-ELM are poor. The algorithmic parameters of

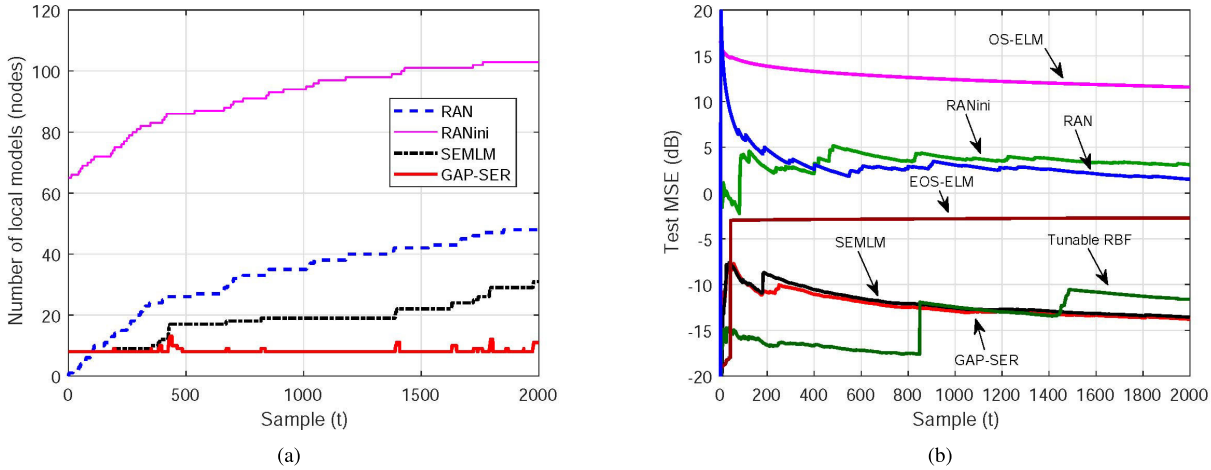



**FIGURE 7.** Online prediction of FOS1 temperature: (a) model size learning curves, and (b) MSE learning curves.

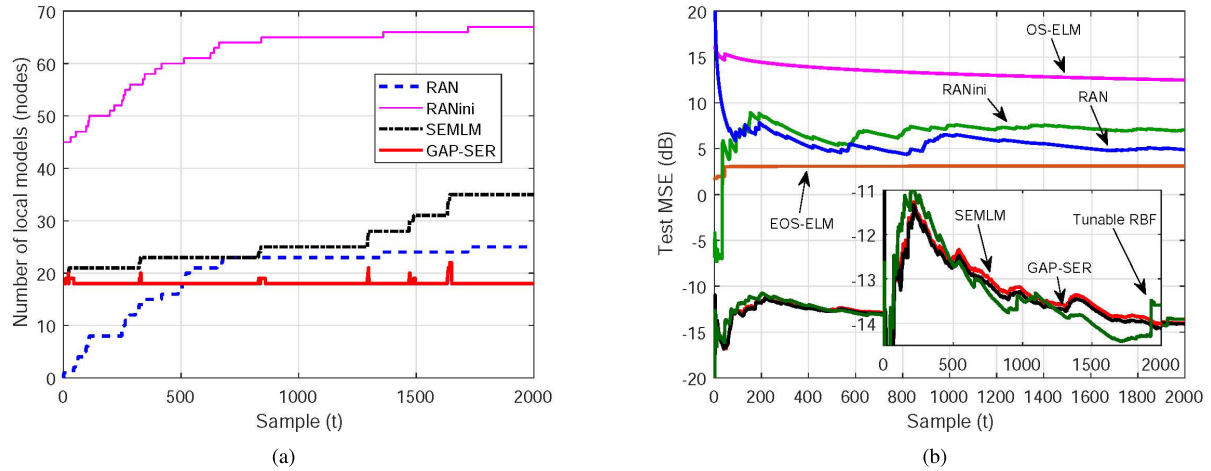



**FIGURE 8.** Online prediction of FOS2 temperature: (a) model size learning curves, and (b) MSE learning curves.

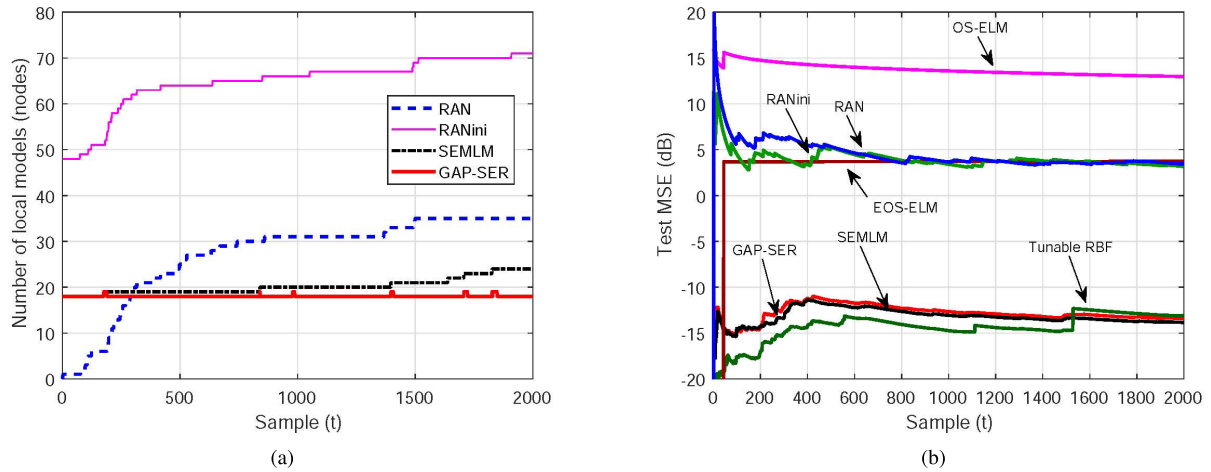



**FIGURE 9.** Online prediction of FOS3 temperature: (a) model size learning curves, and (b) MSE learning curves.

the GAP-SER are empirically chosen to be $W_G = 30$, $p = 30$ and $\varepsilon = 0.1$. The threshold of the SEMLM is $\varepsilon = 1$. For the tunable RBF, we have the node replacement threshold $10^{-5}$, the weight update innovation length 3, the step size 0.01, and the iteration number 5. For the RAN, the maximum and minimum center distances, the error threshold and the decayed constant are also set empirically.

**TABLE 2.** Online identification of MHP: comparison of online prediction and adaptive modeling performance for the OS-ELM, EOS-ELM, RAN, fast tunable RBF, SEMLM and GAP-SER.

| Dataset | Model | MSE (dB) | Online ACTpS (ms) | Models/Nodes | | Average ensemble size |
|---|---|---|---|---|---|---|
| | | | | Initial | Final | |
| FOS1 | OS-ELM | 18.4488 | 0.43 | 100 | 100 | - |
| | EOS-ELM | 6.9665 | 2.38 | $5 \times 100$ | $5 \times 100$ | 5 |
| | RAN | 1.4990 | 0.56 | 0 | 48 | - |
| | RANini | 3.1130 | 1.91 | 65 | 103 | - |
| | Tunable RBF | **-11.6108** | **0.35** | **10** | **10** | - |
| | SEMLM | **-13.5863** | **0.52** | **8** | **31** | **16.71** |
| | GAP-SER | **-13.7951** | **0.20** | **8** | **11** | **4.15** |
| FOS2 | OS-ELM | 12.6390 | 0.44 | 100 | 100 | - |
| | EOS-ELM | 7.7460 | 2.50 | $5 \times 100$ | $5 \times 100$ | 5 |
| | RAN | 4.9071 | 0.24 | 0 | 25 | - |
| | RANini | 7.0560 | 0.98 | 45 | 67 | - |
| | Tunable RBF | **-13.5971** | **0.38** | **10** | **10** | - |
| | SEMLM | **-14.1185** | **0.74** | **18** | **35** | **25.39** |
| | GAP-SER | **-14.0198** | **0.27** | **18** | **18** | **8.94** |
| FOS3 | OS-ELM | 13.5877 | 0.43 | 100 | 100 | - |
| | EOS-ELM | 8.1725 | 2.89 | $5 \times 100$ | $5 \times 100$ | 5 |
| | RAN | 3.5136 | 0.40 | 0 | 35 | - |
| | RANini | 3.1695 | 1.02 | 48 | 71 | - |
| | Tunable RBF | **-13.1200** | **0.34** | **10** | **10** | - |
| | SEMLM | **-13.8481** | **0.61** | **18** | **24** | **19.15** |
| | GAP-SER | **-13.4187** | **0.22** | **18** | **18** | **4** |

**TABLE 3.** EEG data modeling: comparison of online prediction and adaptive modeling performance for the RAN, fast tunable RBF, SEMLM and GAP-SER.

| Model | MSE (dB) | Online ACTpS (ms) | Models/Nodes | | Average ensemble size |
|---|---|---|---|---|---|
| | | | Initial | Final | |
| RAN | 24.6432 | 1.37 | 0 | 106 | - |
| RANini | 20.8746 | 1.81 | 103 | 107 | - |
| Tunable RBF | **13.6452** | **0.41** | **10** | **10** | - |
| SEMLM | **4.4942** | **0.49** | **2** | **3** | **2.97** |
| GAP-SER | **3.9840** | **0.42** | **2** | **3** | **1.97** |

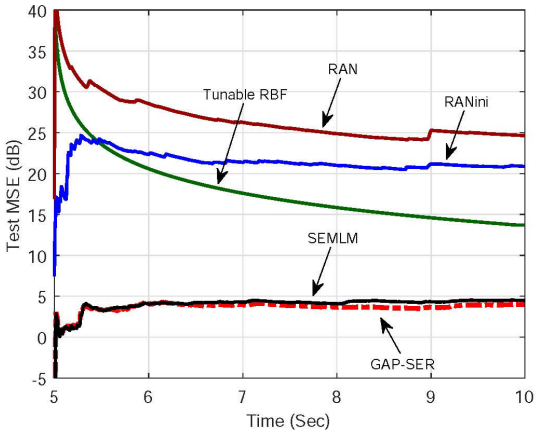



**FIGURE 10.** Online MSE learning curves for EEG data modeling.

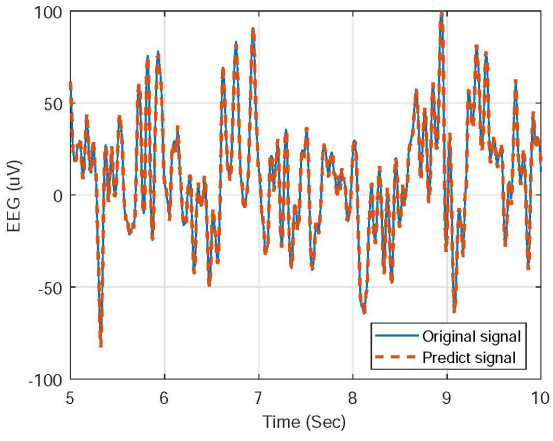



**FIGURE 11.** Comparison of the recovered signal by the GAP-SER and the original EEG observations for the EEG testing dataset.

The online MSE learning curves for the four methods are shown in Fig. 10. Additionally, the online prediction and adaptive model performance are compared in Table 3. Clearly, the results obtained demonstrate that our proposed GAP-SER is capable of capturing the nonstationary dynamics of the EEG signal accurately, while imposing very low online

computational complexity. The recovered signal calculated by the GAP-SER and the original EEG signal are shown in Fig. 11.

## IV. CONCLUSION

This work has proposed a new growing and pruning selective ensemble regression learner for online adaptive modeling of nonlinear and nonstationary data. Our growing strategy can automatically identify every newly emerging process state from fast-arriving data and fit a local linear model to it. This ensures the maximum diversity of the base model set. On the other hand, our new pruning strategy is capable of removing out-of-date local linear models reliably and, therefore, significantly enhancing the plasticity of our selective ensemble learner. A direct consequence of this reliable pruning is that online computational complexity is significantly reduced, which is vital for adaptive modeling of fast arriving data. Based on a probability metric, our new selective ensemble learner selects a small number of best subset linear models from the local linear model set and optimally combines them to produce the accurate online prediction. Extensive experiments, including a chaotic time series prediction, online identification of a real-world industrial microwave heating system and EEG data modeling, have been conducted. The results obtained have demonstrated that our GAP-SER compares very favourably with the existing state-of-the-arts adaptive modeling schemes for nonlinear and nonstationary systems. Specifically, it has been shown that our GAP-SER is capable of producing the most accurate prediction while imposing the lowest online computational complexity for highly nonlinear and nonstationary cases. Our further efforts will be devoted to apply this modeling technique in real-world industrial control systems.

## REFERENCES

[1] Z.-H. Zhou, N. V. Chawla, Y. Jin, and G. J. Williams, "Big data opportunities and challenges: Discussions from data analytics perspectives [discussion forum]," *IEEE Comput. Intell. Mag.*, vol. 9, no. 4, pp. 62–74, Nov. 2014.

[2] G. Ditzler, M. Roveri, C. Alippi, and R. Polikar, "Learning in nonstationary environments: A survey," *IEEE Comput. Intell. Mag.*, vol. 10, no. 4, pp. 12–25, Nov. 2015.

[3] L. Rutkowski, "Generalized regression neural networks in time-varying environment," *IEEE Trans. Neural Netw.*, vol. 15, no. 3, pp. 576–596, May 2004.

[4] J. Liu and D.-S. Chen, "Nonstationary fault detection and diagnosis for multimode processes," *AIChE J.*, vol. 56, no. 1, pp. 207–219, Jan. 2010.

[5] H. Ning, G. Qing, T. Tian, and X. Jing, "Online identification of nonlinear stochastic spatiotemporal system with multiplicative noise by robust optimal control-based kernel learning method," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 2, pp. 389–404, Feb. 2019.

[6] J. Shan, H. Zhang, W. Liu, and Q. Liu, "Online active learning ensemble framework for drifted data streams," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 2, pp. 486–498, Feb. 2019.

[7] J. L. Lobo, I. Laña, J. Del Ser, M. N. Bilbao, and N. Kasabov, "Evolving spiking neural networks for online learning over drifting data streams," *Neural Netw.*, vol. 108, pp. 1–19, Dec. 2018.

[8] J. Gama, R. Sebastião, and P. P. Rodrigues, "On evaluating stream learning algorithms," *Mach. Learn.*, vol. 90, no. 3, pp. 317–346, Mar. 2013.

[9] A. Vergara, S. Vembu, T. Ayhan, M. A. Ryan, M. L. Homer, and R. Huerta, "Chemical gas sensor drift compensation using classifier ensembles," *Sens. Actuators B, Chem.*, vols. 166–167, pp. 320–329, May 2012.

[10] M. Kano and M. Ogawa, "The state of the art in chemical process control in Japan: Good practice and questionnaire survey," *J. Process Control*, vol. 20, no. 9, pp. 969–982, Oct. 2010.

[11] S. Chen and S. A. Billings, "Recursive prediction error parameter estimator for non-linear models," *Int. J. Control*, vol. 49, no. 2, pp. 569–594, Feb. 1989.

[12] S. Chen, "Nonlinear time series modelling and prediction using Gaussian RBF networks with enhanced clustering and RLS learning," *Electron. Lett.*, vol. 31, no. 2, pp. 117–118, Jan. 1995.

[13] F. Ding, P. X. Liu, and G. Liu, "Multiinnovation least-squares identification for system modeling," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 3, pp. 767–778, Jun. 2010.

[14] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, Dec. 2006.

[15] N.-Y. Liang, G.-B. Huang, P. Saratchandran, and N. Sundararajan, "A fast and accurate online sequential learning algorithm for feedforward networks," *IEEE Trans. Neural Netw.*, vol. 17, no. 6, pp. 1411–1423, Nov. 2006.

[16] G.-B. Huang and L. Chen, "Enhanced random search based incremental extreme learning machine," *Neurocomputing*, vol. 71, nos. 16–18, pp. 3460–3468, Oct. 2008.

[17] J. Platt, "A resource-allocating network for function interpolation," *Neural Comput.*, vol. 3, no. 2, pp. 213–225, Jun. 1991.

[18] V. Kadirkamanathan and M. Niranjan, "A function estimation approach to sequential learning with neural networks," *Neural Comput.*, vol. 5, no. 6, pp. 954–975, Nov. 1993.

[19] H. Chen, Y. Gong, X. Hong, and S. Chen, "A fast adaptive tunable RBF network for nonstationary systems," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 2683–2692, Dec. 2016.

[20] S. G. Soares and R. Araújo, "A dynamic and on-line ensemble regression for changing environments," *Expert Syst. Appl.*, vol. 42, no. 6, pp. 2935–2948, Apr. 2015.

[21] B. Krawczyk, L. L. Minku, J. Gama, J. Stefanowski, and M. Woźniak, "Ensemble learning for data stream analysis: A survey," *Inf. Fusion*, vol. 37, pp. 132–156, Sep. 2017.

[22] A. Fern and R. Givan, "Online ensemble learning: An empirical study," *Mach. Learn.*, vol. 53, nos. 1–2, pp. 71–109, Oct. 2003.

[23] J. Z. Kolter and M. A. Maloof, "Dynamic weighted majority: An ensemble method for drifting concepts," *J. Mach. Learn. Res.*, vol. 8, pp. 2755–2790, Dec. 2007.

[24] M. Pratama, W. Pedrycz, and E. Lughofer, "Evolving ensemble fuzzy classifier," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 5, pp. 2552–2567, Oct. 2018.

[25] Y. Sun, K. Tang, L. L. Minku, S. Wang, and X. Yao, "Online ensemble learning of data streams with gradually evolved classes," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 6, pp. 1532–1545, Jun. 2016.

[26] L. L. Minku, A. P. White, and X. Yao, "The impact of diversity on online ensemble learning in the presence of concept drift," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 5, pp. 730–742, May 2010.

[27] Y. Lan, Y. C. Soh, and G.-B. Huang, "Ensemble of online sequential extreme learning machine," *Neurocomputing*, vol. 72, nos. 13–15, pp. 3391–3395, Aug. 2009.

[28] H. Chen, Y. Gong, and X. Hong, "A new adaptive multiple modelling approach for non-linear and non-stationary systems," *Int. J. Syst. Sci.*, vol. 47, no. 9, pp. 2100–2110, Jul. 2016.

[29] P. Kadlec and B. Gabrys, "Local learning-based adaptive soft sensor for catalyst activation prediction," *AIChE J.*, vol. 57, no. 5, pp. 1288–1301, May 2011.

[30] W. Shao, X. Tian, P. Wang, X. Deng, and S. Chen, "Online soft sensor design using local partial least squares models with adaptive process state partition," *Chemometric Intell. Lab. Syst.*, vol. 144, pp. 108–121, May 2015.

[31] W. Shao, S. Chen, and C. J. Harris, "Adaptive soft sensor development for multi-output industrial processes based on selective ensemble learning," *IEEE Access*, vol. 6, pp. 55628–55642, Oct. 2018.

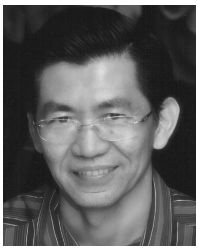
[32] T. Liu, S. Chen, S. Liang, and C. J. Harris, "Selective ensemble of multiple local model learning for nonlinear and nonstationary systems," *Neurocomputing*, vol. 378, pp. 98–111, Feb. 2020.

[33] P. Domingos and G. Hulten, "A general framework for mining massive data streams," *J. Comput. Graph. Statist.*, vol. 12, no. 4, pp. 945–949, Jan. 2003.

[34] S. Grossberg, "Nonlinear neural networks: Principles, mechanisms, and architectures," *Neural Netw.*, vol. 1, no. 1, pp. 17–61, Jan. 1988.

[35] F. Chu and C. Zaniolo, "Fast and light boosting for adaptive mining of data streams," in *Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining*, Sydney, NSW, Australia, May 2004, pp. 282–292.
[36] J. Z. Kolter and M. A. Maloof, "Using additive expert ensembles to cope with concept drift," in *Proc. 22nd Int. Conf. Mach. Learn. (ICML)*, Aug. 2005, pp. 449–456.
[37] K. Nishida and K. Yamauchi, "Adaptive classifiers-ensemble system for tracking concept drift," in *Proc. Int. Conf. Mach. Learn. Cybern.*, Hong Kong, Aug. 2007, pp. 3607–3612.
[38] D. Brzezinski and J. Stefanowski, "Combining block-based and online methods in learning ensembles from concept drifting data streams," *Inf. Sci.*, vol. 265, pp. 50–67, May 2014.
[39] M. Grbovic and S. Vucetic, "Tracking concept change with incremental boosting by minimization of the evolving exponential loss," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, Athens, Greece, Sep. 2011, pp. 516–532.
[40] S.-H. Jung, B.-C. Moon, and D. Han, "Unsupervised learning for crowdsourced indoor localization in wireless networks," *IEEE Trans. Mobile Comput.*, vol. 15, no. 11, pp. 2892–2906, Nov. 2016.
[41] E. N. Lorenz, "Deterministic nonperiodic flow," *J. Atmos. Sci.*, vol. 20, pp. 130–141, Mar. 1963.
[42] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Trans. Neural Netw.*, vol. 2, no. 2, pp. 302–309, Mar. 1991.
[43] A. Bifet and R. Gavaldà, "Learning from time-changing data with adaptive windowing," in *Proc. SIAM Int. Conf. Data Mining*, Minneapolis, USA, Apr. 2007, pp. 443–448.
[44] J. Zhong, S. Liang, Y. Yuan, and Q. Xiong, "Coupled electromagnetic and heat transfer ODE model for microwave heating with temperature-dependent permittivity," *IEEE Trans. Microw. Theory Techn.*, vol. 64, no. 8, pp. 2467–2477, Aug. 2016.
[45] J. Zhong, S. Liang, and Q. Xiong, "Improved receding horizon $H_\infty$ temperature spectrum tracking control for Debye media in microwave heating process," *J. Process Control*, vol. 71, pp. 14–24, Nov. 2018.
[46] C. A. Vriezinga, S. Sánchez-Pedreño, and J. Grasman, "Thermal runaway in microwave heating: A mathematical analysis," *Appl. Math. Model.*, vol. 26, no. 11, pp. 1029–1038, Nov. 2002.
[47] E. Akkari, S. Chevallier, and L. Boillereaux, "Global linearizing control of MIMO microwave-assisted thawing," *Control Eng. Pract.*, vol. 17, no. 1, pp. 39–47, Jan. 2009.
[48] T. Liu, S. Liang, Q. Xiong, and K. Wang, "Adaptive critic based optimal neurocontrol of a distributed microwave heating system using diagonal recurrent network," *IEEE Access*, vol. 6, pp. 68839–68849, Dec. 2018.
[49] X. Shi, J. Li, Q. Xiong, Y. Wu, and Y. Yuan, "Research of uniformity evaluation model based on entropy clustering in the microwave heating processes," *Neurocomputing*, vol. 173, pp. 562–572, Jan. 2016.
[50] K. Wang, L. Ma, Q. Xiong, S. Liang, G. Sun, X. Yu, Z. Yao, and T. Liu, "Learning to detect local overheating of the high-power microwave heating process with deep learning," *IEEE Access*, vol. 6, pp. 10288–10296, Feb. 2018.
[51] T. Liu, S. Liang, Q. Xiong, and K. Wang, "Two-stage method for diagonal recurrent neural network identification of a high-power continuous microwave heating system," *Neural Process. Lett.*, vol. 50, no. 3, pp. 2161–2182, Dec. 2019.
[52] Y. Li, W.-G. Cui, Y.-Z. Guo, T. Huang, X.-F. Yang, and H.-L. Wei, "Time-varying system identification using an ultra-orthogonal forward regression and multiwavelet basis functions with applications to EEG," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 7, pp. 2960–2972, Jul. 2018.
[53] Y. Li, H.-L. Wei, S. A. Billings, and P. G. Sarrigiannis, "Identification of nonlinear time-varying systems using an online sliding-window and common model structure selection (CMSS) approach with applications to EEG," *Int. J. Syst. Sci.*, vol. 47, no. 11, pp. 2671–2681, Aug. 2016.
[54] R. G. Andrzejak, K. Lehnertz, F. Mormann, C. Rieke, P. David, and C. E. Elger, "Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 64, no. 6, pp. 061907-1–061907-8, Nov. 2001.

**TONG LIU** received the B.Sc. degree in automation from the College of Automation, Chongqing University, Chongqing, China, in 2016, where he is currently pursuing the Ph.D. degree in control theory and control engineering. From September 2018 to September 2019, he was a Visiting Ph.D. Student with the School of Electronics and Computer Science, University of Southampton, Southampton, U.K. His current research interests include online learning, system identification, neural networks, machine learning, and intelligent control system design.

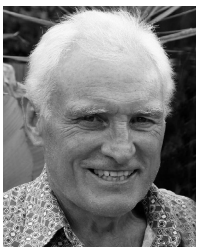


**SHENG CHEN** (Fellow, IEEE) received the B.Eng. degree in control engineering from the East China Petroleum Institute, Dongying, China, in 1982, and the Ph.D. degree in control engineering from City University, London, in 1986. In 2005, he was awarded the higher doctoral degree, Doctor of Science (D.Sc. degree), from the University of Southampton, U.K.

From 1986 to 1999, he held research and academic appointments at the Universities of Sheffield, Edinburgh and Portsmouth, all in U.K. Since 1999, he has been with the School of Electronics and Computer Science, University of Southampton, where he holds the post of Professor of intelligent systems and signal processing. He has published over 650 research articles. He has more than 14,100 Web of Science citations with H-index 53 and more than 29,500 Google Scholar citations with H-index 75. His research interests include neural networks and machine learning, wireless communications, and adaptive signal processing.

Dr. Chen is a Fellow of the United Kingdom Royal Academy of Engineering and IET, the Distinguished Adjunct Professor at King Abdulaziz University, Jeddah, Saudi Arabia, and an Original ISI Highly Cited Researcher of engineering, in March 2004.

**SHAN LIANG** (Member, IEEE) received the M.Sc. degree in control science and engineering from the College of Automation, Chongqing University, Chongqing, China, in 1995, and the Ph.D. degree from the Department of Mechanical Systems Engineering, Kumamoto University, Kumamoto, Japan, in 2004. His current research interests include numerical modeling, electromagnetic theory, nonlinear systems, adaptive control, and sensor networks.



**CHRIS J. HARRIS** received the B.Sc. degree from the University of Leicester and the M.A. degree from the University of Oxford, U.K., the Ph.D. degree from the University of Southampton, U.K., in 1972, and the D.Sc. degree from the University of Southampton, in 2001.

He is the Emeritus Research Professor with the University of Southampton, having previously held senior academic appointments at the Imperial College, Oxford and Manchester Universities, as well as Deputy Chief Scientist for the U.K. Government. He is the coauthor of over 500 scientific research articles during a 50 year research career. He was awarded the IEE Senior Achievement Medal for Data Fusion Research and the IEE Faraday Medal for Distinguished International Research in Machine Learning. He was elected to the U.K. Royal Academy of Engineering, in 1996.