A Multiple Model Approach to Time-Series Prediction Using an Online Sequential Learning Algorithm

Koshy George, *Senior Member, IEEE*, and Prabhanjan Mutalik

***Abstract*—Time-series prediction is important in diverse fields. Traditionally, methods for time-series prediction were based on fixed linear models because of mathematical tractability. Researchers turned their attention to artificial neural networks due to their better approximation capability. In this paper, we use feedforward neural networks with a single hidden layer, and present a rather simple online sequential learning algorithm (OSLA) together with its proof. The convergence properties of this algorithm are those of the well-known recursive least squares algorithm. We demonstrate that the prediction performance is better than other OSLAs, and show that it is statistically different from them. In addition, we also present the multiple models, switching, and tuning methodology that enhances the prediction performance of the learning algorithm.**

***Index Terms*—Prediction models, recurrent neural networks, supervised learning, time-series analysis.**

I. Introduction

The analysis of a temporally or spatially sampled collection of observations is important in a variety of fields. Examples of such collections include sunspot activity, intensity of ocean tides, and signals from an array of sensors. EEG signals are analyzed in [1]. The sampling may be periodic or aperiodic. Predicting the future trend of a temporally sampled collection is a significant component of time-series analysis. Applications range from engineering to economics, for purposes of planning, mid-course correction and proactive decision making.

Researchers have involved prediction in varied examples. These include the prediction of the life of a battery (i.e., state-of-charge) in [2], the varying demand rate in the context of economic lot scheduling problem in [3], and the performance of service-oriented systems in [4] that helps the user to optimally choose the composition of the required Web services. Gray modeling is used for prediction in [5] for the power generated by a photovoltaic system connected to a grid, the failure rate of weapon spare parts, electricity consumption, and tuberculosis incidence. Applications to financial markets were considered in [6] and [7]. Time-series forecasting of the future behavior of the sensors is integrated in [8] along with data fusion, consensus method and fuzzy-logic inference system in a monitoring system that uses information from a wireless sensor network. Predictive noise detection is used in [9] to improve the experience of listening to music from gramophone records. Prediction is not restricted to time-series alone. In [10], the behavior of a driver is predicted using a nested Pitman–Yor language model. From these examples it is abundantly clear that prediction plays an important role in many applications.

Over the past several decades a number of techniques have been proposed for time-series prediction [11]–[14]. Initial attempts for prediction were based on fitting a trend-curve on the time-domain data [14]. Yule's [15] autoregressive (AR) technique led to a paradigm shift in the analysis of time-series [16], [17]. The use of such models provided better prediction performance relative to mere extrapolation via a trend-curve, leading to the age of modern time-series prediction. Accordingly, the objective shifted to determining the rules governing the evolution of the time-series that best explained the behavior of the observed phenomenon, and to subsequently use these rules to predict its future values. Equivalently, the goal was to model the observed phenomenon as a dynamical system with specified input and output spaces. The output of such a system corresponds to the observed time-series, and it may be a function of one or more variables or inputs.

For mathematical tractability, the dynamical system models were chosen to be linear. One such comprehensive characterization is the AR integrated moving average (ARIMA) model [12], [14]. (The forecasting procedure based on this mathematical representation is known as the Box–Jenkins approach [14], [18].) Variations of this include the AR, the moving average (MA) and the ARMA models. These models are characterized by a finite set of parameters. The principal idea is to determine this set of parameters using the available data. These models are then subsequently used to extrapolate the data and hence provide an estimate of the future values of the time-series.

A fundamental limitation is that not all observed phenomena can be represented using linear models as they do not

Manuscript received March 22, 2017; accepted May 6, 2017. Date of publication June 27, 2017; date of current version April 15, 2019. This paper was recommended by Associate Editor K. T. Seow. *(Corresponding author: Koshy George.)*

K. George is with the Department of Electronics and Communication Engineering, PES University, Bengaluru 560085, India, and also with the PES Centre for Intelligent Systems, PES University, Bengaluru 560085, India (e-mail: kgeorge@pes.edu).

P. Mutalik was with the PES Centre for Intelligent Systems, PES University, Bengaluru 560085, India. He is now with the Department of Computational Sciences, KTH Royal Institute of Technology, 114 28 Stockholm, Sweden.

Digital Object Identifier 10.1109/TSMC.2017.2712184

approximate with sufficient accuracy the complexities and intricacies of time-series data. For example, the logistic map cannot be approximated by a linear model [16]. Indeed, a number of papers in the literature related to economy have shown that nonlinearities in the model is important for prediction or forecasting; refer [19] and the references cited therein.

Dynamical systems maps input spaces to output spaces. Modeling the time-series using dynamical systems leads to the following important issue. In nearly all applications, the exogenous inputs that affect the time-series are not measurable, and only the past observations of the time-series are available for providing estimates of the future values. Thus, it is imperative that it is possible to determine the hidden patterns in the collections of observations to predict better the subsequent values. Over the past several decades artificial neural networks (ANNs) have emerged as a practical tool for pattern recognition [20]. Accordingly, ANNs have the ability to detect hidden patterns in a time-series. Moreover, feedforward neural networks (FNNs) are universal approximators [21]–[23] in that they can approximate a sufficiently smooth function to any desired accuracy. This implies that they approximate better dynamical systems that are nonlinear [24]. Therefore, it is quite natural to expect that FNNs can provide better prediction performance compared to modeling the time-series with an ARIMA model as networks can fit more complicated functions [16]. Indeed, several researchers have established that ANNs provide better estimates of the subsequent values when compared to the Box–Jenkins method, e.g., [25] and [26]. Accordingly, we use FNNs for time-series prediction in this paper.

The available data are used to adapt the synaptic weights of the FNN until its output approximates the time-series with substantial veracity; i.e., the error in the approximation is within some *a priori* specified bounds. This idea is based on the universal approximation property of FNNs that have a single hidden layer with sufficient number of neurons. A number of approaches have been proposed to adapt the weights. Although batch processing techniques have the ability to analyze the richness of data, the focus of this paper is on online sequential processing. Whilst the former requires the availability of data *a priori*, the latter extracts the necessary information from the incoming data one-by-one which are subsequently discarded. It has been observed in [27] that an online updating of the synaptic weights is faster and consumes less memory. Moreover, our emphasis is on the prediction problem in real-time wherein the one-step ahead prediction is based solely on the information that is available at the time. Accordingly, we use online sequential processing of data in this paper.

The principal contributions of this paper are threefold. First, to present the online sequential learning algorithm (OSLA) for FNNs with a single hidden layer together with its proof of convergence. Second, to propose the multiple model approach to improve performance in the context of time-series prediction. Third, to demonstrate that such an approach provides better time-series prediction performance relative to several existing sequential training techniques. We demonstrate this with two metrics as well as statistical tests.

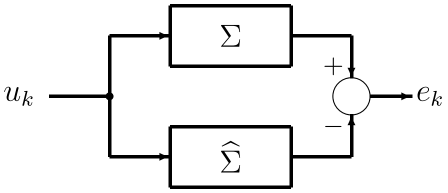


Fig. 1. Block diagram for system identification.

The rest of this paper is organized as follows. In Section II, we discuss the mathematical systems perspective to time-series prediction. We present our learning algorithm in Section III and contrast it with several existing sequential training methods. The multiple model approach is treated in Section IV, and the results of numerical experiments on a variety of time-series data are presented in Section V. Statistical comparisons are also provided in this section. The conclusions are provided in Section VI followed by the proof of convergence of the algorithm in the Appendix.

II. Time-Series Prediction: Mathematical Perspective

Mathematical systems theory deals with functions of time, and maps that relate these functions [28]. That is, a dynamical system Σ maps the space of exogenous inputs $\mathcal{U}$ to the space of all outputs $\mathcal{Y}$. (In what follows, both spaces are restricted to functions of time.) A fundamental problem in mathematical systems theory is that of *system identification* [24]. As illustrated in Fig. 1, it deals with determining an approximate $\widehat{\Sigma}$ to the given dynamical system Σ. That is, given $\epsilon > 0$, determine $\widehat{\Sigma}$ such that

$$\|\widehat{\Sigma}u - \Sigma u\| \leq \epsilon \quad \forall\, u \in \mathcal{U} \tag{1}$$

where $\|\cdot\|$ is an appropriately defined norm on the output space $\mathcal{Y}$.

A second important problem of mathematical systems theory is that of *system characterization* which deals with mathematical representation of a system [24]. The determination of $\widehat{\Sigma}$ that satisfies (1) is mathematically tractable only if the representation of Σ is linear. Accordingly, an oft-used representation or characterization is the ARIMA model [12], [14] described as follows:

$$\sum_{i=0}^{p_1} \alpha_i q^{-i}\left(1 - q^{-1}\right)^{p_2} y_k = \sum_{j=0}^{p_3} \beta_j q^{-j} u_k \tag{2}$$

where y_k and u_k are, respectively, the observations of the phenomenon and the exogenous input at instant k, q^{-1} is the unit delay operator, p_1, p_2, and p_3 are integers, α_i and β_j are real constants where the typical value of α_0 is unity. This model reduces to the ARMA model when $p_2 = 0$, the MA model when $p_1 = p_2 = 0$, and the AR model when $p_2 = p_3 = 0$. In the context of time-series prediction, the aim is to determine the parameters α_i and β_j at least ϵ-suboptimal in the sense of (1).

There are three issues with this approach. First, the model parameters are determined based on off-line data and (1). Consequently, novel information contained in new data cannot

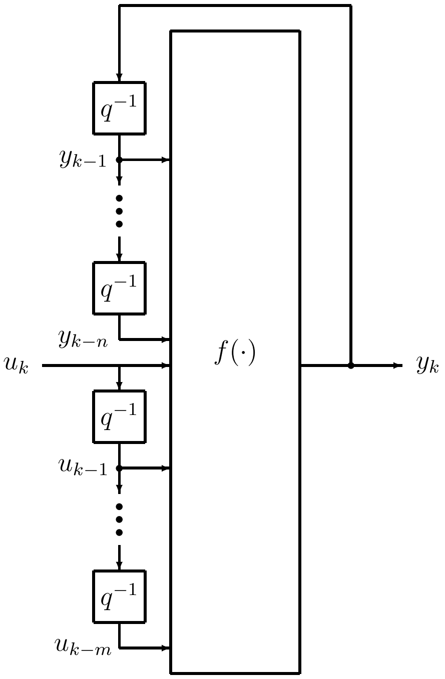


Fig. 2. General nonlinear system that maps the present and previous values of the input u_k, and the previous values of the output, to the present value of the output y_k. Here, q^{-1} is the delay operator.

directly and easily be assimilated into the model without recomputing the model parameters based on (1). This is computationally intensive. Alternatively, adaptive laws for updating the parameters in a stable manner are available for linear time-invariant systems [29], thereby allowing continuous learning. Despite these possibilities, the assumed model is still linear. This leads us to the second issue.

It has been reported (see [16]) that linear models are reasonably good choices as long as the power spectrum of the time-series characterizes its relevant features. However, it is also well-known that even simple nonlinearities cannot be approximated by a linear map; for example, the logistic map. It is, however, possible to model differently the different regions of the state-space of the dynamical system corresponding to the time-series [30], [31]. Evidently, estimating the number of linear models required to approximate the nonlinear dynamical system is an open problem.

Third, the approach assumes the knowledge of the exogenous input u_k. Even though the physical phenomenon underlying the given time-series is influenced by exogenous inputs, they are typically not measurable. The focus of this paper is on an online sequential time-series prediction technique that primarily deals with the first two issues. As shown in the sequel, the effect of the third issue is resolved to some extent.

One general characterization for nonlinear dynamical systems is depicted in Fig. 2, and is described by the difference equation

$$y_k = f(y_{k-1}, y_{k-2}, \ldots, y_{k-n}, u_k, u_{k-1}, \ldots, u_{k-m}) \tag{3}$$

where m and n are constants that have a direct bearing on the input–output behavior of the dynamical system. Suppose that we define the regression vectors

$$\begin{aligned}\phi_{y_k} &= \begin{pmatrix} y_k & y_{k-1} & \cdots & y_{k-n+1}\end{pmatrix}^T\\ \phi_{u_k} &= \begin{pmatrix} u_k & u_{k-1} & \cdots & u_{k-m}\end{pmatrix}^T\end{aligned}$$

where A^T denotes the transpose of a matrix A, then (3) can compactly be written as

$$y_k = f\big(\phi_{y_{k-1}}, \phi_{u_k}\big). \tag{4}$$

Special cases of this include when the nonlinearity $f(\cdot)$ can be expressed as a sum of two separable maps, one a function of a regression on the output and the other a function of a regression on the input [i.e., $g(\phi_{y_{k-1}}) + h(\phi_{u_k})$]; and, when the maps $g(\cdot)$ and $h(\cdot)$ are independently linear. (These are described in [24].)

ANNs are natural candidates for system characterization of the underlying dynamics of a time-series: it is nonlinear and it admits continuous learning. In particular, FNNs have been mathematically proved to be universal approximators, e.g., [21]–[23]. Thus, when the map $f(\cdot)$ [or, $g(\cdot)$ and $h(\cdot)$] is sufficiently smooth, there exists a single-hidden-layer FNN $\mathcal{N}(\cdot)$, with a sufficient number of neurons, such that

$$\|f(x) - \mathcal{N}(x)\| < \epsilon \tag{5}$$

for all x in a compact subset of the domain of definition for some *a priori* given $\epsilon > 0$. Accordingly, given the input–output pairs $\{(u_k, y_k)\}$, there exists such a neural network that can approximate the behavior of the underlying dynamics of the time-series. Consequently, the first two issues mentioned earlier are addressed if such FNNs are used.

The two choices of the identification models that have been proposed in the past are referred to as the "parallel" structure and the "series–parallel" structure (see [24]), and are, respectively, described as follows:

$$\hat{y}_k = \mathcal{N}\big(\phi_{\hat{y}_{k-1}}, \phi_{u_k}\big) \tag{6}$$

$$\hat{y}_k = \mathcal{N}\big(\phi_{y_{k-1}}, \phi_{u_k}\big) \tag{7}$$

where $\hat{y}_k$ is the estimate of y_k, and $\mathcal{N}(\cdot)$ represents the FNN that approximates the nonlinear function $f(\cdot)$ to the *a priori* chosen accuracy. Whilst the former strictly mimics (4), the latter is a function of the actual observations rather than their estimates as shown in Fig. 3. It has been suggested that the series–parallel structure provides better estimates [24]. It may be noted that an FNN together with a tapped delay line—with Fig. 3 as an example—forms a recurrent neural network [24].

As mentioned earlier, the third issue is that the exogenous input u_k is typically not measurable, and hence cannot be used in either (6) or (7). Thus, for the series–parallel structure, the system characterization reduces to the following:

$$\hat{y}_k = \mathcal{N}\big(\phi_{y_{k-1}}\big). \tag{8}$$

(This approach to time-series prediction appears not to be emphasized in the literature. One exception is [32] where the network is trained offline with batch processing. This is in contrast to our online sequential training process presented later.) Evidently, as shown in Fig. 4, the prediction is open-loop. As is well-known, feedback has the ability to stabilize the

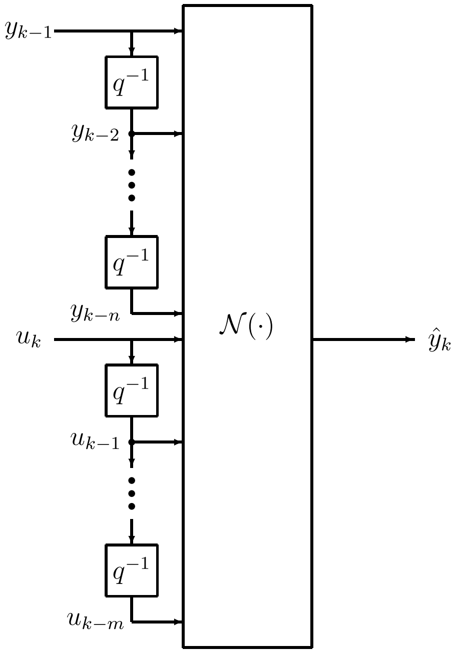


Fig. 3. Identification using series–parallel structure. The structure is same as that in Fig. 2 with the map $f(\cdot)$ replaced by the neural network $\mathcal{N}(\cdot)$. The output $\hat{y}_k$ is an estimate of the actual output.

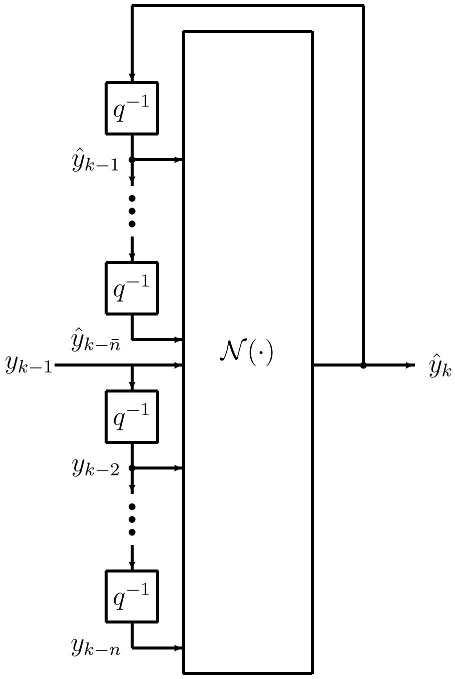


Fig. 5. Closed-loop structure for one-step ahead prediction.

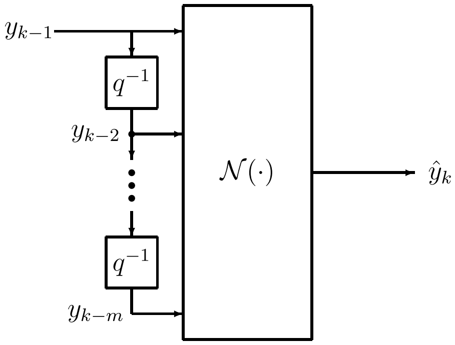


Fig. 4. Open-loop architecture for time-series prediction.

output of a system. For example, Black [33] deliberately introduced feedback in amplifiers as early as 1912. In the context of deterministic chaotic systems, iterated predictions have been shown in [34] to be better than direct predictions. Accordingly, the characterization chosen here for iterated predictions is as follows:

$$\hat{y}_k = \mathcal{N}\big(\phi_{\hat{y}_{k-1}}, \phi_{y_{k-1}}\big). \tag{9}$$

The input to the FNN now includes a regression on the estimates; the architecture is depicted in Fig. 5. In the context of time-series prediction, this was first introduced in [35] and found to improve prediction performance relative to methods based on an open-loop characterization.

III. Sequential Learning

FNNs, due to their ability as universal approximators, have been used in time-series prediction by many researchers; examples include [16], [25], and [36]–[38]. The classical algorithm to adapt the weights is the backpropagation algorithm (BPA). As is well-known this is a stochastic gradient-based algorithm. Whilst this algorithm is powerful and is applicable to FNNs with arbitrary number of hidden layers and number of neurons in each hidden layer, it is notorious for its slow convergence and is often unable to reach the global minimum of the performance surface [39], [40]. In contrast, our objective here is to quickly determine a satisfactory estimate of the future values of the time-series. Such an approach is necessary in applications that require decisions to be made on-the-fly.

Consider an FNN with a single hidden layer. Let the number of nodes in the input layer be m_0. Further, let m_1 and m_2, respectively, denote the number of neurons in the hidden and output layers. We denote such a network as $\mathcal{N}_{m_0:m_1:m_2}$. The neurons in the hidden layer have a nonlinear activation function $\varphi(\cdot)$, and the output layer is linear. Suppose that at instant k, the input–output pair is (x_k, y_k), where $x_k \in \mathbb{R}^{m_0}$ and $y_k \in \mathbb{R}^{m_2}$. Let $W_1 \in \mathbb{R}^{m_1\times(m_0+1)}$ and $W_2 \in \mathbb{R}^{m_2\times(m_1+1)}$, respectively, be the matrix of synaptic weights corresponding to the hidden and output layers. Evidently, the forward pass may be summarized as follows:

$$\begin{aligned}\bar{v}_{1,k} &= \varphi\big(W_1\, v_{0,k}\big)\\ v_{2,k} &= W_2\, v_{1,k}\end{aligned}$$

where

$$v_{0,k} \triangleq \begin{pmatrix} 1 \\ x_k \end{pmatrix}, \quad v_{1,k} \triangleq \begin{pmatrix} 1 \\ \bar{v}_{1,k} \end{pmatrix}.$$

Here $\bar{v}_{1,k} \in \mathbb{R}^{m_1}$. It may be noted that bias has been taken into account.

A. Online Sequential Learning Algorithm

The OSLA is described in this section. Let W_1 be initialized randomly and $W_{2,0}$ be the zero matrix. Let $P_0 = (1/\lambda) I_{m_1+1}$

with $\lambda > 0$ and I_{m_1+1} is an identity matrix of dimensions $(m_1+1)\times(m_1+1)$. The weight matrix W_1 remains unchanged. In contrast, the weight matrix corresponding to the output layer W_2 is adapted according to the following equations:

$$P_{k+1} = P_k - \frac{P_k v_{1,k+1} v_{1,k+1}^T P_k}{1 + v_{1,k+1}^T P_k v_{1,k+1}}, \quad k \geq 0 \tag{10}$$

$$W_{2,k+1} = W_{2,k} + e_{k+1} v_{1,k+1}^T P_{k+1}, \quad k \geq 0 \tag{11}$$

where the *a priori* prediction estimate is

$$e_{k+1} \overset{\Delta}{=} y_{k+1} - W_{2,k} v_{1,k+1}. \tag{12}$$

We have the following result with its proof provided in the Appendix.

Theorem 1: Suppose that a finite collection of input–output pairs $\{(x_k, y_k)\}_{k=1}^N$ is given, W_1 is initialized randomly and $W_{2,0}$ is the zero matrix of appropriate dimensions. Let the weight matrix $W_{2,k}$ be adapted in accordance to (10) and (11) with $P_0 = (1/\lambda)I_{m_1+1}$. Then, the weight matrix

$$\lim_{k \longrightarrow N} W_{2,k} = W_{2,*}$$

where $W_{2,*}$ is the minimizer of the cost function

$$W_{2,*} = \arg\min_W \frac{1}{2}\left\{\|Y - W V_1\|_F^2 + \lambda \|W\|_F^2\right\}. \tag{13}$$

Here

$$V_1 = \begin{pmatrix} v_{1,1} & v_{1,2} & \cdots & v_{1,N} \end{pmatrix} \tag{14}$$

$$Y = \begin{pmatrix} y_1 & y_2 & \cdots & y_N \end{pmatrix} \tag{15}$$

and $\|\cdot\|_F$ is the Frobenius norm.

The cost function (13) is the Tikhonov–Phillips functional, where λ is the regularization parameter [41]–[43]. The parameter λ provides a tradeoff between minimization of the prediction error and the strength of the synaptic weights of the output layer. The minimizer to this functional is the following least-squares solution:

$$W_{2,*} = Y V_1^T \left(V_1 V_1^T + \lambda I\right)^{-1}. \tag{16}$$

The result easily follows by minimizing the following equivalent cost function:

$$J = \frac{1}{2}\mathrm{tr}\left\{(Y - W V_1)^T (Y - W V_1)\right\} + \frac{\lambda}{2}\mathrm{tr}\left\{W^T W\right\}$$

where $\mathrm{tr}\{\cdot\}$ is the trace operator. Clearly, the first derivative of the functional is $-Y V_1^T + W V_1 V_1^T + \lambda W$. Moreover, the second derivative $V_1 V_1^T + \lambda I > 0$; accordingly, $W_{2,*}$ achieves a minimum which is global and unique. The computational steps of OSLA are depicted in Table I.

Comments: The following remarks are in order.

1) Clearly, OSLA seeks the global minimizer of the Tikhonov–Phillips functional. Moreover, it is a recursive least squares (RLSs) solution and possesses all its convergence properties [44].
2) As evident from Table I, the algorithm is sequential from $k \geq 1$.
3) It has been argued that randomization of W_1 can lead to performance that is inconsistent. However, it has been our experience that such a randomization has little effect on the performance. In this paper, we suggest that using multiple FNNs each trained with OSLA can overcome this effect to an extent. This is treated in Section IV.

TABLE I
OSLA

Initialization: Choose $W_1 \in \mathbb{R}^{m_1 \times (m_0+1)}$ randomly and set $W_{2,0} \in \mathbb{R}^{m_2 \times (m_1+1)}$ a zero matrix. Let $P_0 = \frac{1}{\lambda} I_{m_1+1}$ and $k = 1$.

Sequential computations: For all $k \geq 1$:

1) Given the input-output pair, (x_k, y_k), compute the output of the hidden layer $v_{1,k}$,

$$v_{0,k} = \begin{pmatrix} 1 \\ x_k \end{pmatrix}, \quad \bar{v}_{1,k} = \varphi\left(W_1 v_{0,k}\right), \quad v_{1,k} = \begin{pmatrix} 1 \\ \bar{v}_{1,k} \end{pmatrix},$$

and the predicted output $\hat{y}_k = W_{2,k-1} v_{1,k}$.

2) Update P_k and $W_{2,k}$ as follows:

$$\begin{aligned} P_k &= P_{k-1} - \frac{P_{k-1} v_{1,k} v_{1,k}^T P_{k-1}}{1 + v_{1,k}^T P_{k-1} v_{1,k}}, \\ W_{2,k} &= W_{2,k-1} + e_k v_{1,k}^T P_k, \end{aligned}$$

where $e_k = y_k - \hat{y}_k$.

4) OSLA was initially used in the context of system identification and control [45], [46]. It was observed in these papers that the transient performance improved considerably relative to other sequential learning algorithms including BPA and the online sequential extreme learning machine (OSELM) proposed in [36], and some of the variants of the latter. This is due to the convergence properties of OSLA which is similar to the RLS algorithm. Subsequently, similar performance improvements were observed in [47] in the context of time-series prediction.

In this paper, we formally present for the first time the algorithm and its proof of convergence.

B. Comparison of OSLA With Other Sequential Learning Algorithms

As indicated earlier, BPA converges slowly, and not necessarily to the global minimum [39], [40]. For this reason, researchers have sought to improve its convergence properties. The extreme learning machine (ELM) proposed by Huang *et al.* [48] was a step in this direction. Applicable to FNNs with a single hidden layer, the output weight matrix is determined by

$$W_{2,\mathrm{ELM}} = Y V_1^T \left(V_1 V_1^T\right)^{-1} \tag{17}$$

where the matrices Y and V_1 are as defined earlier in (14) and (15), and the weight matrix W_1 is randomly initialized. Clearly, ELM learns batch-wise.

The OSELM is the first sequential version of ELM, and it was clearly demonstrated that its performance and convergence properties are superior to BPA [36]. This algorithm comprises two phases. In the initialization phase, a chunk of data, say N_0, is collected and the weight matrix corresponding to the output layer computed by treating this as a batch learning problem (i.e., an ELM) using this initial chunk of data

$$W_{2,0,\mathrm{OSELM}} = Y_0 V_{1,0}^T \left(V_{1,0} V_{1,0}^T\right)^{-1} \overset{\Delta}{=} Y_0 V_{1,0}^T P_{0,\mathrm{OSELM}} \tag{18}$$

where $V_{1,0}$ and Y_0 are, respectively, the concatenations of the outputs of the hidden layer to the initial chunk of data and the desired outputs, similar to (14) and (15). In the weight-update phase, let (x_1, y_1) be the first new input–output pair beyond the initial chunk of data of size N_0. (For ease of representation, the input–output pairs beyond N_0 have been relabeled so that (x_{N_0+1}, y_{N_0+1}) is (x_1, y_1), (x_{N_0+2}, y_{N_0+2}) is (x_2, y_2), and so on.) Let $v_{1,1}$ be the output of the hidden layer computed as mentioned earlier

$$v_{1,1} = \begin{pmatrix} 1 \\ \varphi(W_1 v_{0,1}) \end{pmatrix}, \quad v_{0,1} = \begin{pmatrix} 1 \\ x_1 \end{pmatrix}.$$

The weight matrix is then updated as follows:

$$W_{2,1,\text{OSELM}} = Y_1 V_{1,1}^T \big(V_{1,1} V_{1,1}^T\big)^{-1} \triangleq Y_1 V_{1,1}^T P_{1,\text{OSELM}}$$

where $V_{1,1} = (\, V_{1,0} \quad v_{1,1}\,)$ and $Y_1 = (\, Y_0 \quad y_1\,)$. Similarly, with (x_2, y_2)

$$W_{2,2,\text{OSELM}} = Y_2 V_{1,2}^T (V_{1,2} V_{1,2}^T)^{-1} \triangleq Y_2 V_{1,2}^T P_{2,\text{OSELM}}$$

where $V_{1,2} = (\, V_{1,1} \quad v_{1,2}\,)$ and $Y_2 = (\, Y_1 \quad y_2\,)$. This is repeated for all $k \geq 1$ in the update phase of OSELM. Observe that during this stage, for every iteration step, an ELM is solved with the available data. After some algebra, the weight update process may be written as follows:

$$\begin{aligned} P_{k+1,\text{OSELM}} &= P_{k,\text{OSELM}} \\ &\quad - \frac{P_{k,\text{OSELM}}\, v_{1,k+1} v_{1,k+1}^T\, P_{k,\text{OSELM}}}{1 + v_{1,k+1}^T\, P_{k,\text{OSELM}}\, v_{1,k+1}} \\ W_{2,k+1,\text{OSELM}} &= W_{2,k,\text{OSELM}} + e_{k+1} v_{1,k+1}^T P_{k+1,\text{OSELM}} \end{aligned}$$

where $e_{k+1} = y_{k+1} - W_{2,k,\text{OSELM}}\, v_{1,k+1}$. These update equations appear to be similar to (10) and (11). However, it is evident from the above analysis that the sequences of matrices $\{P_{k,\text{OSELM}}\}$ and $\{W_{2,k,\text{OSELM}}\}$, are very different from the sequences of matrices $\{P_k\}$ and $\{W_{2,k}\}$ associated with OSLA by virtue of the different initializations and definitions. Thus, the similarity between OSELM and OSLA is restricted to the structure of the update equations.

The OSELM is improved in [37] and [49]. A step-size for the incremental change—referred to as a penalizing factor—is introduced in [37] to reduce the effect of perhaps meaningless new data. The variance of the weight matrix is adjusted in [49] using a Kalman filter. An approach similar to [36] is used in [27] but with more output basis functions to address nonstationary data.

The initialization phase requires a matrix related to the available data to be inverted: $V_{1,0} V_{1,0}^T$. Since the condition number of this matrix depends on the quality of data, it is not clear *a priori* how large a chunk of data is required to ensure stable numerical inversion. Huynh and Won [50] addressed this problem by introducing a diagonal loading factor thereby improving the numerical properties of OSELM. Essentially, in this regularized OSELM, the definition of $P_{k,\text{OSELM}}$ is modified to $P_{k,\text{Re-OSELM}} = (V_{1,k} V_{1,k}^T + \lambda I)^{-1}$ for all k. Recursive update equations similar to the ones for OSELM are obtained with the initializations suitably modified. A kernel-based approach with a similar initialization is used in [38] and [51]–[52].

Evidently, OSELM and all of the aforementioned variants have an initialization phase wherein data of size N_0 must be collected before the algorithm becomes truly sequential in nature. In contrast to all these variants, the initializations of the recursions (10) and (11) in OSLA are independent of the data. In particular, P_0 results from λI_{m_1+1} which is guaranteed to be nonsingular for our choice of λ. Moreover, OSLA is sequential for all input–output pairs, $k \geq 1$. Further, the convergence properties of OSLA are similar to the RLS algorithm.

Since OSLA is different from OSELM and its variants, its application to system identification and control [45], [46] resulted in transient performance that was considerably better than BPA, OSELM, and some of the variants of the latter. With OSELM and its variants, identification and control is open-loop during the initialization phase. For linear or nonlinear systems, open-loop control during the initialization phase of OSELM and its variants can lead to unbounded growth in the outputs from which the overall system is unable to recover. Performance improvements in the context of time-series prediction were similarly observed in [47]. In all these applications, OSLA performs better as the adaptation of the output synaptic weights takes place for $k \geq 1$, the initializations of the recursions are independent of the data, and it has satisfactory convergence properties.

IV. Multiple Models, Switching, and Tuning for Time-Series Prediction

Improving approximation accuracy by using multiple models is not new. Indeed, as mentioned earlier, multiple linear models were used in [30] for time-series prediction. An optimal way to combine multiple offline trained neural networks to improve model accuracy was proposed in [53]. The weights were tuned in a probabilistic approach to arrive at an optimal linear combination of available trained models. Various techniques to combine structurally different models to improve the solutions to time-series prediction problems have been reviewed in [54]. The author also discusses the possibilities and effects of combining ARIMA and ANN models on the quality of predictions. Combining several ELMs to achieve better generalizations have been discussed in [55] and [56]. These are offline techniques which aim at building an optimal ensemble of ELMs.

Diversity in the models used in an ensemble can be achieved in three different ways—by initializing different models, by deriving a unique subset for each model that is rich enough to provide good generalization, or by randomly selecting different subsets of data for different models. Since our focus is on online learning, the first approach is the only possible way to create diversity among the chosen models.

An adaptive ensemble approach with multiple ELMs was proposed in [57] where a weighted sum of the outputs of multiple offline trained ELMs are considered with the weights being adapted based on the observed prediction error. (ELMs have linear output layers which are trained offline.) Clearly, this approach does not allow online learning. Moreover, it does not select the best among the offline trained ELMs.

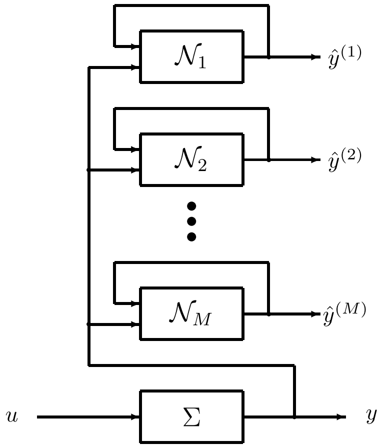


Fig. 6. MMST for time-series prediction.

In the approach adopted in this paper, the predictor consists of M FNNs, $\mathcal{N}_1, \ldots, \mathcal{N}_M$, operating in parallel as shown in Fig. 6. Each such network has an overall structure shown in Fig. 5, where the inputs to each network are the regression vectors $\phi_{\hat{y}_{k-1}}$ and $\phi_{y_{k-1}}$. However, the actual structure of these networks can randomly be chosen; that is, the length of these regression vectors and the number of hidden neurons of each network can differ from others. The networks are trained independently with OSLA described in Section IV after being initialized independently. Accordingly, there are multiple models to provide different one-step ahead predictions, and each of these multiple models are adapted or tuned at every time step.

In contrast to previous approaches where the outputs of the networks are combined, we choose the best estimate of $\hat{y}_k$ at every instant k, as each network is expected to provide an independent analysis of the time-series. The question remains which output is to be chosen as the best estimate amongst the set $\{\hat{y}_{k+1}^{(1)}, \hat{y}_{k+1}^{(2)}, \ldots, \hat{y}_{k+1}^{(M)}\}$. In this paper, we propose to select the best estimate based on the performance at the previous instant. That is, at instant $k-1$, the estimates $\hat{y}_k^{(1)}, \hat{y}_k^{(2)}, \ldots, \hat{y}_k^{(M)}$ are available. When the new observation y_k is available, it is compared with these estimates and the best predictor is chosen

$$j = \arg \min_{1 \leq i \leq M} \left\| y_k - \hat{y}_k^{(i)} \right\|^2. \tag{19}$$

Thus, the predictor $\mathcal{N}_j$ is chosen at instant k as the best one in the sense of (19). Accordingly, the estimate $\hat{y}_{k+1}^{(j)}$ is taken as the best predicted value of y_{k+1}. Thus, at each instant of time, the overall algorithm switches to the best available estimate. The algorithm for multiple models, switching, and tuning (MMST) is given in Table II.

Clearly, there are a number of models for prediction with each of them being adapted continuously, and the best predictor chosen by switching. In the past, the individual concepts of MMST, have been independently introduced and developed by several researchers. These notions were combined for the first time in [58] for the purpose of improving performance in the field of adaptive control of linear systems with unknown parameters. Subsequently, global asymptotic stability of the overall system was, respectively, proved for deterministic and stochastic systems in [59] and [60]. It was shown in [61] that different parametric estimation techniques can be combined. The methodology was extended in [62] to time-varying systems, and a class of nonlinear systems in [63]. A combination of linear and nonlinear systems was considered in [64]. It was shown by several researchers that the MMST methodology was essential for effective adaptive control in several applications (see [62] and the references cited therein.) These include fault-tolerant flight control and control of a chemical reactor. Applications in signal processing include interference cancellation and blind source separation when all the sources of the signals are moving, and active noise control [65]. In the context of time-series prediction, it was first proposed in [35] using multiple FNNs, each trained as an OSELM.

TABLE II
MMST

```
Initialization: Set-up $M$ FNNs. Each network has different $W_1$, and can have different number of neurons, and different number of inputs.
Sequential computations: For all $k \geq 1$:
1) Given the input-output pair, $(x_k, y_k)$, compute the outputs of the $M$ networks: $\hat{y}_k^{(1)}, \hat{y}_k^{(2)}, \ldots, \hat{y}_k^{(M)}$.
2) If $k = 1$, randomly choose any model as the best predictor, say $j$. Set $\hat{y}_1 = \hat{y}_1^{(j)}$. Otherwise, choose the best predictor as follows:
       $j = \arg \min_{1 \leq i \leq M} \|y_k - \hat{y}_k^{(i)}\|^2.$
   Set $\hat{y}_{k+1} = \hat{y}_{k+1}^{(j)}$.
```

V. Results

The contributions in this section are twofold. First, we compare the predictive performance of OSLA with other methods. Specifically, we show through numerical experiments that OSLA performs better. We achieve this by comparing the performances with two metrics. Further, we show that OSLA is statistically different from other methods. Second, we demonstrate that the performance of a method is enhanced when augmented with MMST.

The following six methods for time-series prediction are compared here.

1) *Method A:* OSELM [36].
2) *Method B:* MMST methodology with OSELM (OSELM-MMST) [35].
3) *Method C:* Regularized OSELM (Re-OSELM) [50].
4) *Method D:* OSELM with kernels (OSELM-K) [52].
5) *Method E:* OSLA.
6) *Method F:* MMST methodology with OSLA (OSLA-MMST).

In all methods, feedback is incorporated; i.e., the inputs to the networks are the regression vectors $\phi_{\hat{y}_{k-1}}$ and $\phi_{y_{k-1}}$. Thus the manner in which Re-OSELM and OSELM-K is applied to time-series prediction is novel in this paper as the input to the FNN includes $\phi_{\hat{y}_{k-1}}$. Further, the combination of MMST and OSLA is also new. To compare these methods, datasets are chosen from various sources. These are described below.

1) *Data I:* This is referred to as the gas-furnace data [12]. Air and methane were combined in a furnace resulting in a mixture of gases. Here, the concentration of carbon

dioxide is monitored. The data consists of 296 input–output pairs with a sampling interval of 9 s. In this paper, we do not use information about the input which is the methane gas feed-rate.

2) *Data II:* This data reflects the annual record of the number of Canadian lynx trappings between 1821 and 1934 in the Mackenzie River district of Northwest Canada. A number of researchers have attempted to provide a model for the time-series data with the first one attributed to Moran in 1953 [66].
3) *Data III:* The multivariate dataset is a collection of currency exchange rates between U.S. dollar and Swiss franc for a period of eight months between 1990 and 1991. This data was part of the Sante Fe Time-Series Prediction and Analysis Competition held by the Sante Fe Institute in 1991 [67]. Automatic processing was essential due to the enormous volume of data [67].
4) *Data IV:* Changes in periodicity caused by acute or chronic diseases like the irregular breathing patterns in adults with Cheynes–Stokes syndrome have been described as bifurcations in the differential delay systems by Mackey–Glass [68]

$$\frac{dx(t)}{dt} = \frac{\alpha_1 \alpha_2^n x(t-\tau)}{\alpha_2^n + x^n(t-\tau)} - \alpha_3 x(t) \tag{20}$$

where α_1, α_2, α_3, and n are constants. The fluctuations in the peripheral white blood cells with chronic granulocytic leukemia are another example. We use the benchmark problem suggested in [36] where the parameters have been chosen as follows: $\alpha_1 = 0.2$, $\alpha_2 = 1$, $\alpha_3 = 0.1$, $n = 10$, and the delay $\tau = 17$. Specifically, the time-series is generated using (20) by integrating the equation over the time interval $[t, t+T]$ with the trapezoidal rule

$$x(t+T) = \frac{2-\alpha_3 T}{2+T} + \frac{\alpha_1 T}{2+T} \times \left(\frac{x(t+T-\tau)}{1+x^n(t+T-\tau)} + \frac{x(t-\tau)}{1+x^n(t-\tau)} \right).$$

This dataset was considered by several other researchers. Functional link neural networks and neural fuzzy systems are, respectively, considered in [69]–[71].
5) *Data V:* The data corresponds to fluctuations of a laser operating in the far infrared region. The laser is in a chaotic state and the physical phenomenon follows a Lorenz model. This can be modeled by three simultaneous nonlinear ordinary differential equations. It was part of the Santa Fe competition [67]. This data was also considered in [72].
6) *Data VI:* This is a multivariate physiological dataset of a patient with sleep apnea [67]. There are no premature beats implying that the sudden changes in the heart rate are not artifacts. Data on the heart rate, chest volume, blood oxygen concentration and EEG state of the patient are provided. Only the heart rate was to be predicted as its variations provide information about the onset of sleep apnea.

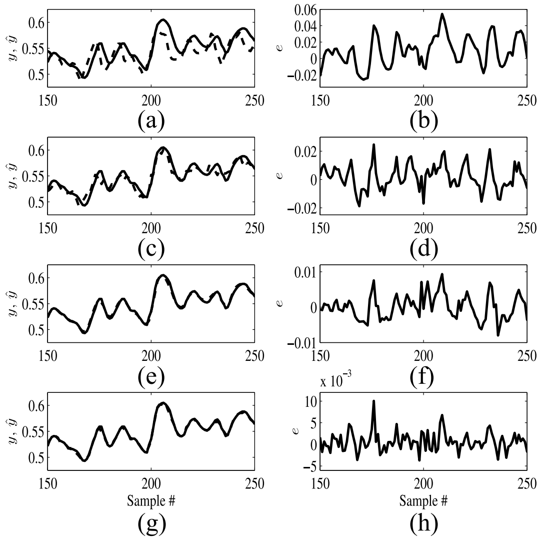


Fig. 7. Performance comparison for data I (gas-furnace). (a) and (b) OSELM. (c) and (d) OSELM-MMST. (e) and (f) OSLA. (g) and (h) OSLA-MMST. (a), (c), (e), and (g) compare the predicted outputs ($\hat{y}$: – –) with the actual values (y: —). (b), (d), (f), and (h) corresponding errors: $e = y - \hat{y}$.

These datasets vary in the complexity of the underlying dynamical systems. Whilst, data III, V, and VI have a large number of samples in the time-series, data II and IV are relatively small.

A. Comparison Using Two Metrics

The performance of four methods (OSELM, OSELM-MMST, OSLA, and OSLA-MMST) are compared in Fig. 7 for data I. The predicted output with OSELM is compared with the actual value in Fig. 7(a). It is clear that the prediction errors are significant using OSELM. The corresponding prediction error is depicted in Fig. 7(b). These errors are substantially reduced when MMST methodology is introduced where the FNNs are trained as OSELMs. This is evident from Fig. 7(c) and (d) relative to the corresponding plots in Fig. 7(a) and (b). For this data, and in this particular experiment, the algorithm proposed here (OSLA) does appear to have only a little effect as can be observed from Fig. 7(e) and (f). The MMST approach with FNNs trained with OSLA (OSLA-MMST) does appear to yield significant improvement for this dataset. This is evident from Fig. 7(g) and (h).

However, OSLA does have an impact as the RMSE averaged over 50 trials reduces from 28.7339 for OSELM to 0.5238 for OSLA, two orders lower. Indeed, OSLA provides the least RMSE when compared to OSELM, Re-OSELM, and OSELM-K, as indicated in the first row of Table III. Further, it is evident that MMST enhances the performance of a method. Specifically, the averaged RMSE for OSELM-MMST is an order lesser than OSELM, and there is a reduction of 17% with OSLA-MMST when compared to OSLA. The performance of OSLA is also relatively more consistent across experiments, when compared to OSELM, Re-OSELM, and OSELM-K. This is evident from the standard deviation σ provided in Table IV. Amongst the methods without MMST,

TABLE III
Root Mean Square Error Averaged Over 50 Trials

Data	OSELM	OSELM-MMST	Re-OSELM	OSELM-K	OSLA	OSLA-MMST
I: Gas-furnace	28.7339	1.1771	2.0418	0.7580	0.5238	0.4315
II: Lynx trappings	1544.8	1124.4	1527.9	1153.0	996.9697	658.0932
III: Currency Exchange	0.5621	0.0046	5.8449	0.0083	0.0013	9.80×10^{-4}
IV: Mackey-Glass	0.0639	0.0179	0.2324	0.0261	0.003	0.0016
V: Laser	129.0031	45.6651	19.3153	59.5094	7.6694	5.6178
VI: Physiological	672.4214	4.8364	90.2795	3.288	2.1791	1.6938

TABLE IV
Standard Deviation of the RMSE Across 50 Trials

Data	OSELM	OSELM-MMST	Re-OSELM	OSELM-K	OSLA	OSLA-MMST
I: Gas-furnace	25.6592	0.7164	1.2348	0.4283	0.3863	0.3341
II: Lynx trappings	960.049	719.8293	956.7307	843.475	711.0845	554.7065
III: Currency Exchange	0.5619	0.0037	6.8596	0.0073	1.1×10^{-3}	7.07×10^{-4}
IV: Mackey-Glass	0.0638	0.0092	0.2324	0.004	1.16×10^{-3}	2.32×10^{-6}
V: Laser	49.9134	32.9614	15.3396	35.7877	6.8484	4.6982
VI: Physiological	664.8271	3.2856	3.222	2.3397	1.561	1.0995

OSLA yielded the least σ, and is two orders less when compared to OSELM. There is a considerable reduction in σ with OSELM-MMST relative to OSELM, and a 13% reduction with OSLA-MMST relative to OSLA. Indeed, for data I, the maximum RMSE obtained amongst the 50 trials were 85.3637, 2.6263, 1.6972, and 0.4134 for the methods OSELM, Re-OSELM, OSELM-K, and OSLA, respectively. This is further indicative of the consistency in the performance of OSLA. For the same dataset, the maximum RMSE for OSELM-MMST and OSLA-MMST are 3.1448 and 0.2905. This confirms the fact that MMST enhances the performance at the cost of additional computational complexity.

We use a second metric to compare the six methods: *variance accounted for* (VAF). This measure compares two signals s_1 and s_2, and is defined as follows (e.g., [73]):

$$\text{VAF}(s_1, s_2) = \left(1 - \frac{\text{variance}(s_1 - s_2)}{\text{variance}(s_1)}\right) 100\%.$$

The smaller the ratio of the variance of the error between s_1 and s_2 to the variance of s_1, the larger the value of the metric, and the closer is the signal s_2 to s_1. The two signals can be considered to be equal if VAF is 100, and very different if the VAF is very small or negative.

The performances of the six methods are compared with this metric in Table V. As indicated by the first row, OSLA provides much better performance when compared to OSELM, Re-OSELM, and OSELM-K. The averaged VAF for OSLA is 97.3691% when compared to −1000% for OSELM. Whenever there is a possibility of improving the performance, MMST is able to achieve this. Thus, OSELM-MMST provides an averaged VAF of 88.2822%. A marginal improvement is observed with OSLA-MMST relative to OSLA.

Similar observations can be made for the other datasets. The performances of the methods are compared in Figs. 8–12, respectively, for data II–VI, and summarized in Tables III–V. (For each dataset, these figures depict only one of the 50 experiments.) We note that the actual improvement varies with the nature of data. Visually, the improvement with OSLA is more evident with all other datasets. This is obvious when the error plot in Figs. 8(f)–12(f) is compared to the error plots in Figs. 8(b) and (d)–12(b) and (d) in each of these figures. The improvement with OSELM-MMST over OSELM is visually evident in Figs. 8–10 and 12, and the improvement with OSLA-MMST over OSLA can be observed in Fig. 10. Clearly, OSLA provides the best performance amongst OSELM, Re-OSELM, and OSELM-K in terms of the average RMSE, the standard deviation of these values, and the average VAF, respectively, shown in Tables III–V. For all datasets, OSELM-MMST improves OSELM, and OSLA-MMST improves OSLA, with respect to these metrics.

Thus, OSLA provides significantly better and consistent performance in that the RMSE errors are acceptably low, and the VAF values are high. The standard deviations corresponding to OSLA are smaller than the best performance of an OSELM variant for all datasets. Our simpler approach provides better performance for data IV when compared to the results in [69]–[71], and for data V when compared to the results in [72]. In all cases, the MMST methodology improves the prediction performance compared to prediction with a single model. Further, the convergence of OSLA is observed to be fast for all datasets; these are evident from the error plots corresponding to Figs. 7(h)–12(h) relative to the error plots corresponding to other methods. Furthermore, since this algorithm yields reasonably low values of RMSE across datasets it can be concluded that the effect of randomization of W_1 and subsequently not adapting these weights has very little effect on the prediction performance. The computational complexity with multiple models is higher when compared to the methods that use a single model. Accordingly, unless the performance improvement is expected to be significant it is sufficient to use a single model.

TABLE V
VARIANCE ACCOUNTED FOR AVERAGED OVER 50 TRIALS

Data	OSELM	OSELM-MMST	Re-OSELM	OSELM-K	OSLA	OSLA-MMST
I: Gas-furnace	-1000	88.2822	63.3894	94.5574	97.3691	98.2367
II: Lynx trappings	-4.1842	46.4363	-0.7457	46.3763	57.845	82.9689
III: Currency Exchange	-1000.0	70.9551	-1000	69.2759	97.7639	98.65
IV: Mackey-Glass	-152.778	20.1484	-1000	79.3662	99.3238	99.5138
V: Laser	-1000	5.0449	83.0023	32.9633	97.3224	98.926
VI: Physiological	-1000	13.4083	-1000	65.7787	80.4155	88.1368

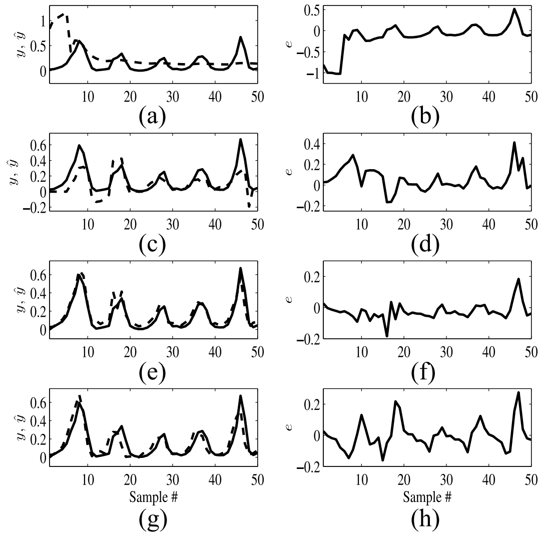


Fig. 8. Performance comparison for data II (lynx trappings). (a) and (b) OSELM. (c) and (d) OSELM-MMST. (e) and (f) OSLA. (g) and (h) OSLA-MMST. (a), (c), (e), and (g) compare the predicted outputs ($\hat{y}$: – –) with the actual values (y: —). (b), (d), (f) and (h) corresponding errors: $e = y - \hat{y}$.

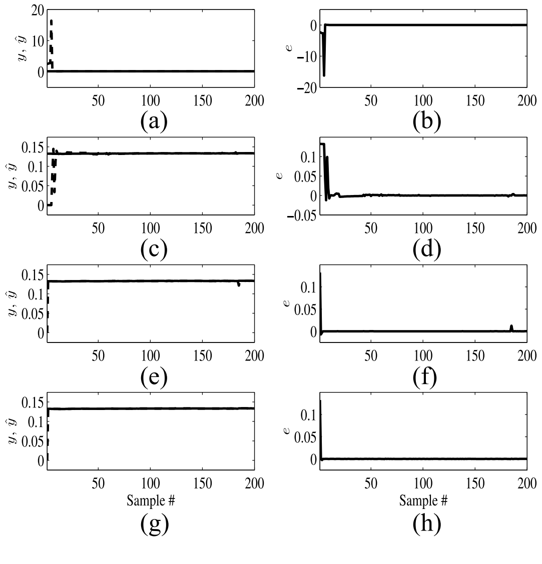


Fig. 9. Performance comparison for data III (currency exchange). (a) and (b) OSELM. (c) and (d) OSELM-MMST. (e) and (f) OSLA. (g) and (h) OSLA-MMST. (a), (c), (e), and (g) compare the predicted outputs ($\hat{y}$: – –) with the actual values (y: —). (b), (d), (f), and (h) corresponding errors: $e = y - \hat{y}$.

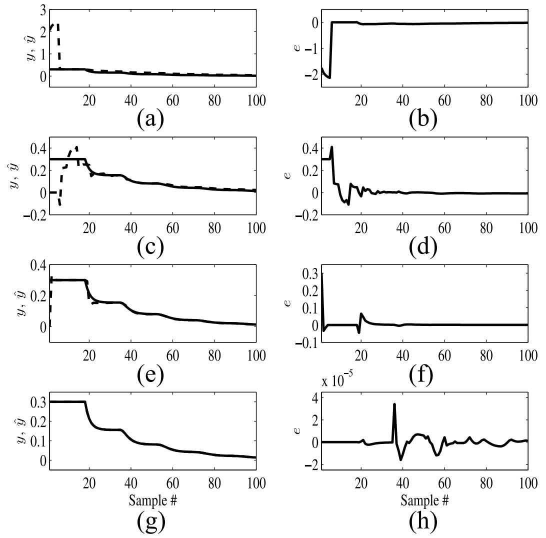


Fig. 10. Performance comparison for data IV (Mackey–Glass). (a) and (b) OSELM. (c) and (d) OSELM-MMST. (e) and (f) OSLA. (g) and (h) OSLA-MMST. (a), (c), (e), and (g) compare the predicted outputs ($\hat{y}$: – –) with the actual values (y: —). (b), (d), (f), and (h) corresponding errors: $e = y - \hat{y}$.

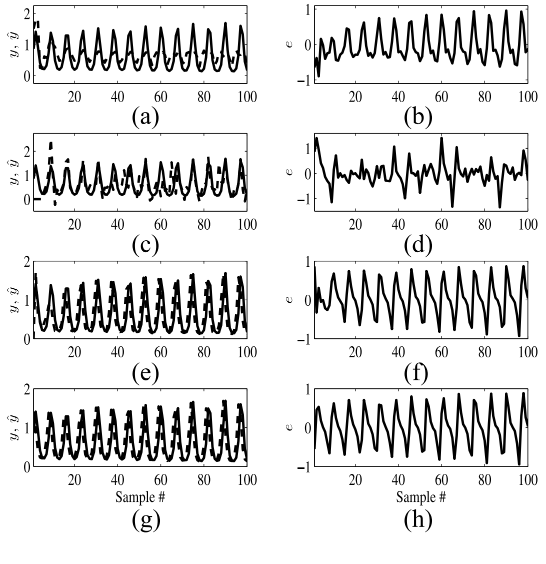


Fig. 11. Performance comparison for data V (laser). (a) and (b) OSELM. (c) and (d) OSELM-MMST. (e) and (f) OSLA. (g) and (h) OSLA-MMST. (a), (c), (e), and (g) compare the predicted outputs ($\hat{y}$: – –) with the actual values (y: —). (b), (d), (f), and (h) corresponding errors: $e = y - \hat{y}$.

For OSLA, the choices of the regularization parameter λ are given in Table VI. These choices provide the best prediction performance and have been arrived at after several numerical experiments. Clearly, the choice depends on the dataset. For all methods, we use the same activation function, $\varphi(v) = \tanh(v)$. Moreover, the number and choice of inputs to the FNNs are

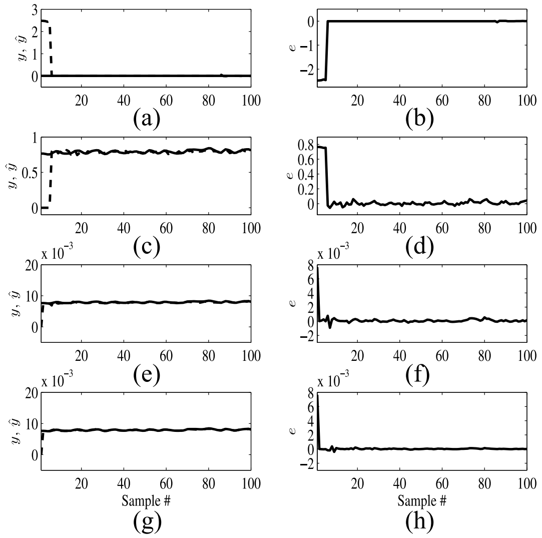


Fig. 12. Performance comparison for data VI (physiological). (a) and (b) OSELM. (c) and (d) OSELM-MMST. (e) and (f) OSLA. (g) and (h) OSLA-MMST. (a), (c), (e), and (g) compare the predicted outputs ($\hat{y}$: – –) with the actual values (y: —). (b), (d), (f), and (h) corresponding errors: $e = y - \hat{y}$.

TABLE VI
CHOICE OF PARAMETER λ

Data	λ
I: Gas-furnace	0.0001
II: Lynx trappings	0.01
III: Currency Exchange	1.00×10^{-08}
IV: Mackey-Glass	1.00×10^{-05}
V: Laser	0.01
VI: Physiological	1.00×10^{-05}

TABLE VII
PROCESSING TIME PER ITERATION

Method	Time (secs.)
A: OSELM	0.01570
B: OSELM-MMST	0.07264
C: Re-OSELM	0.01170
D: OSELM-K	0.02649
E: OSLA	0.00260
F: OSLA-MMST	0.02200

the same. Specifically, in Fig. 5, $n = 4$ and $\bar{n} = 2$. Further, W_1 is randomized for each of the methods. These synaptic weights are uniformly distributed in the range [0, 1]. For OSLA, after several experiments, the numbers of hidden neurons are fixed at 55, 50, 55, 20, 15, and 75, respectively, for the datasets I–VI for optimal prediction performance. The number of multiple models M for OSELM-MMST and OSLA-MMST are five. Any further increase in the number of models did not provide an appreciable improvement in the performance when compared across datasets. A Gaussian kernel was used in OSELM-K.

We emphasize that in our approach to time-series prediction there is no separate training phase. The network parameters are updated with every new piece of information and the one-step ahead prediction determined. Therefore, only the time taken per iteration step is provided in Table VII, and is independent of the dataset. It is evident from this table that the processing times for OSELM and Re-OSELM are comparable which is due to the similarity in the computations. OSLA takes relatively lesser time as it does not require a matrix to be inverted during the initialization phase. Due to the presence of multiple models, the methods OSELM-MMST and OSLA-MMST take more time compared to OSELM and OSLA, respectively. However, these are still lesser when compared to OSELM-K. Clearly, with the addition of each model in the MMST augmented approaches, the processing time increases; e.g., the values for OSLA-MMST are 0.0149, 0.0160, and 0.0220 s, respectively, with 3, 4, and 5 models. The aforementioned experiments were conducted on a machine with an i5-4200 CPU at 250 GHz on a MATLAB version 7.13.0.564 platform.

B. Comparison Using Statistical Analysis

In this section, we compare the expected accuracies of two one-step ahead predictions, say, $\{\hat{y}_{1,k}\}$ and $\{\hat{y}_{2,k}\}$ with the corresponding errors $\{e_{1,k}\}$ and $\{e_{2,k}\}$, where $e_{i,k} \overset{\Delta}{=} y_k - \hat{y}_{1,k}$. When $\mathcal{E}\{L(e_{1,k})\} = \mathcal{E}\{L(e_{2,k})\}$, we say that the two predictors satisfy the null hypothesis of equal prediction accuracy [74]; here, $\mathcal{E}\{\cdot\}$ is the mathematical expectation operator and $L(\cdot)$ is some *a priori* chosen loss function.

Pattern classifiers are generally compared using statistical methods recommended in [75] and extended in [76]. In contrast, a number of other tests have been used to compare the accuracy of predictions or forecasts. In this paper, we consider the following tests.

1) Wilcoxon's signed-rank (WSR) test [77].
2) Diebold–Mariano (DM) test [74].
3) Kolmogorov–Smirnov (KS) test [78].

The first and the third tests are nonparametric.

The DM test is better than a number of older competitors including the WSR test and the Morgan–Granger–Newbold test [79], and is quite popular for comparing forecasts [80]. The KS test was applied to compare prediction accuracies in [81]. It complements the DM test and overcomes some of its issues. These include finite sample properties, stationarity of the covariance, and dependence on the student's t distribution.

OSLA is statistically compared with the other techniques in Table VIII using the WSR test. Here $h = 1$ indicates the rejection of the null hypothesis and $h = 0$ indicates a failure to reject the null hypothesis, both at a chosen significance value. The p-value is the probability of observing a test statistic greater than or equal to the observed value under the null hypothesis, and k is the test statistic. As is evident from Table VIII, for a 5% significance value, the null hypothesis was rejected for data I, III, IV, and V. That is, for these datasets OSLA performed differently when compared to the other methods. For data II and VI, the WSR test is rather inconclusive. For all datasets other than III and VI, OSLA is statistically different from OSLA-MMST.

The results of comparison of OSLA with other methods using the DM test are shown in Table IX, where the statistic

TABLE VIII
WSR Test: OSLA Versus Other Methods

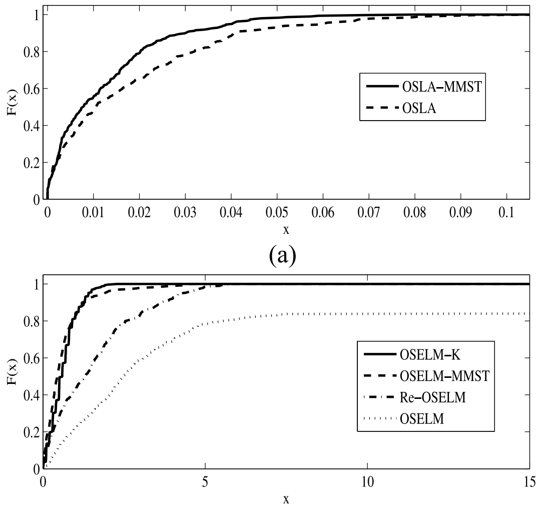
Data		OSELM	OSELM-MMST	Re-OSELM	OSELM-K	OSLA-MMST
I: Gas-furnace	h	1	0	1	1	1
	p	1.46×10^{-4}	0.0873	9.36×10^{-4}	0.0117	0.0171
	k	3.7974	1.7099	3.3092	2.5217	2.3853
II: Lynx trapping	h	0	1	0	1	1
	p	0.1146	6.58×10^{-4}	0.0522	0.005	0.0035
	k	1.578	3.4065	1.9418	2.8056	2.9176
III: Currency exchange	h	1	1	1	1	0
	p	0	2.22×10^{-4}	0	0	0.7227
	k	24.2156	3.6924	38.4872	29.6336	0.3549
IV: Mackey-Glass	h	1	1	1	1	1
	p	0	0	0	0	0
	k	82.2719	49.5125	48.0935	78.9157	35.1339
V: Laser	h	1	0	1	1	1
	p	0	0.191	0	0	2.57×10^{-5}
	k	7.1652	1.307	6.5324	17.506	4.2086
VI: Physiological	h	0	1	1	1	0
	p	0.1249	0	9.68×10^{-10}	0	0.059
	k	1.5345	8.5796	6.1146	14.183	1.8881

TABLE IX
DM Test: OSLA Versus Other Methods

Data	OSELM	OSELM-MMST	Re-OSELM	OSELM-K	OSLA-MMST
I: Gas-furnace	6.3732	4.6411	11.9165	5.4837	-2.3511
II: Lynx trappings	3.8523	3.2714	3.6930	3.6293	-5.3538
III: Currency exchange	25.6965	3.6549	9.9455	34.1353	-0.7104
IV: Mackey-Glass	7.1256	1.9408	5.6452	7.1256	-1.938
V: Laser	16.2327	6.8969	6.6451	16.2327	-3.6234
VI: Physiological	7.8794	13.0313	12.8197	6.5306	-3.4402

is provided. The DM statistic is indicative of a scaled difference between the square of the errors, when the loss function is the square of the error in the prediction, $e_{i,k}^2$. For a 5% significance level, the null hypothesis—indicating that two predictions have similar accuracies—is rejected if this statistic falls outside the range $[-1.96, 1.96]$. From the table it is clear that the statistic is outside this range for nearly all considered pairwise comparisons. Evidently, these comparisons are more conclusive than the WSR test. Thus the predictive accuracy of OSLA is statistically different from that of OSELM, Re-OSELM, and OSELM-K. However, the performances vis-á-vis OSELM-MMST and OSLA-MMST are less conclusive. For four of the datasets, the DM statistic indicates that MMST improved prediction performance.

With the KS test, it can be observed from Table X that OSLA performs differently from all other methods as the test rejected the null hypothesis for all pairwise comparisons. (The quantities h and p are defined similar to the WSR test, and k is the KS statistic.) For three datasets the test statistic k shows that OSLA-MMST is nearest in performance to OSLA. Here, k indicates the maximum absolute difference between the empirical cumulative distribution functions (CDFs) of the two prediction errors; i.e., the maximum distance between the two curves. It may be noted that in these tables if the returned p-value is smaller than 10^{-15}, it is arbitrarily assigned to zero.



(a)

(b)

Fig. 13. One-sided KS test for data I. Comparison of the empirical CDFs for (a) OSLA and OSLA-MMST and (b) OSELM, OSELM-MMST, OSELM-K, and Re-OSELM.

TABLE X
KS TEST: OSLA VERSUS OTHER METHODS

Data		OSELM	OSELM-MMST	Re-OSELM	OSELM-K	OSLA-MMST
I: Gas-furnace	h	1	1	1	1	1
	p	0	0	0	0	0
	k	0.4403	0.2544	0.3808	0.2894	0.1875
II: Lynx trapping	h	1	1	1	1	1
	p	0	0	0	1.64×10^{-4}	0
	k	0.3367	0.4794	0.4794	0.3118	0.3762
III: Currency exchange	h	1	1	1	1	1
	p	0	0	0	0	0.045
	k	0.7607	0.2614	0.999	0.8189	0.0615
IV: Mackey-Glass	h	1	1	1	1	1
	p	0	0	0	0	0
	k	0.9487	0.9954	0.9814	0.9289	0.9479
V: Laser	h	1	1	1	1	1
	p	0	0	0	0	0
	k	0.408	0.4349	0.4203	0.6683	0.1573
VI: Physiological	h	1	1	1	1	1
	p	0.0273	0	0	0	0.0077
	k	0.0653	0.2997	0.2958	0.2907	0.0743

The one-sided KS test compares the empirical CDFs. If X and Y are two random variables with CDFs F_X and F_Y, then if F_X lies above and to the left of F_Y, then X is said to have a lower stochastic error than Y [81]. That is, for the null hypothesis $\mathcal{H}_0$, $F_X(x) \leq F_Y(x)$ for all x, and for $\mathcal{H}_1$, $F_x(x) > F_Y(x)$ for some x [81]. Such a comparison is shown in Fig. 13 for the considered methods in this paper when applied to data I. For clarity, only the CDFs of OSLA and OSLA-MMST is shown in Fig. 13(a), and the CDFs for the other methods in Fig. 13(b). Evidently, the stochastic errors corresponding to OSLA-MMST is the least amongst all methods, followed by OSLA. There is a significant difference between OSLA and OSELM and its variants. Further, from Fig. 13(b), the prediction performances of OSELM-MMST and OSELM-K are not distinguishable. These two are better than Re-OSELM which improves upon OSELM. A similar trend is observed for other datasets.

VI. CONCLUSION

An OSLA to train the synaptic weights corresponding to the output layer of FNNs with a single hidden layer was presented in this paper. These synaptic weights converge to the global minimum of a Tikhonov–Phillips functional, and hence is a regularized solution. Further, the properties of the learning algorithm are those of the recursive least squares algorithm. This learning algorithm was shown to proffer better performance compared to several existing algorithms in the context of time-series prediction. The algorithm provides better performance for a variety of datasets that differ in the complexity of the underlying dynamics as well as the number of available samples. Further, the performance was consistent in the sense of lower standard deviation in the root-mean-square errors, and the effect of randomization of the synaptic weights corresponding to the hidden layer was observed to be rather insignificant. The prediction performance of this algorithm was shown to be statistically different from other algorithms. In addition, the MMST methodology was shown to enhance the performance of a chosen learning algorithm. In this paper we focused on one-step ahead prediction. Work is in progress on the application of OSLA and OSLA-MMST on multiple-step ahead prediction.

APPENDIX
PROOF OF THE THEOREM 1

From the respective definitions of V_1 and Y in (14) and (15), we have

$$V_1\, V_1^T = \sum_{i=1}^{N} v_{1,i} v_{1,i}^T, \quad Y\, V_1^T = \sum_{i=1}^{N} y_i v_{1,i}^T.$$

Define the following recursions for $k \geq 1$:

$$R_k = R_{k-1} + v_{1,k} v_{1,k}^T, \quad R_0 = \lambda I \tag{21}$$

$$r_k = r_{k-1} + y_k v_{1,k}^T, \quad r_0 = 0. \tag{22}$$

Clearly, $R_N = V_1\, V_1^T + \lambda I$ and $r_N = Y\, V_1^T$.

When (x_1, y_1) is available, the optimal weight matrix for the second layer is $W_{2,1} = r_1 R_1^{-1}$. Similarly, when (x_1, y_1) and (x_2, y_2) are available, the optimal weight matrix is $W_{2,2} = r_2 R_2^{-1}$, and so on. Therefore, at the kth stage, the optimal weight matrix is given by

$$W_{2,k} = r_k R_k^{-1}, \quad k \geq 1. \tag{23}$$

But $R_k^{-1} = (R_{k-1} + v_{1,k} v_{1,k}^T)^{-1}$. Using the matrix inversion lemma

$$R_k^{-1} = R_{k-1}^{-1} - \frac{R_{k-1}^{-1} v_{1,k} v_{1,k}^T R_{k-1}^{-1}}{1 + v_{1,k}^T R_{k-1}^{-1} v_{1,k}}.$$

The recursion (10) naturally follows with $P_k \overset{\Delta}{=} R_k^{-1}$. Moreover, $W_{2,k} = r_k R_k^{-1}$. Accordingly

$$\begin{aligned} W_{2,k} &= W_{2,k-1} + e_k \frac{v_{1,k}^T R_{k-1}^{-1}}{1 + v_{1,k}^T R_{k-1}^{-1} v_{1,k}} \\ &= W_{2,k-1} + e_k \frac{v_{1,k}^T P_{k-1}}{1 + v_{1,k}^T P_{k-1} v_{1,k}} \\ &= W_{2,k-1} + e_k v_{1,k}^T P_k \end{aligned}$$

where $e_k \overset{\Delta}{=} y_k - W_{2,k-1} v_{1,k}$. The recursion (11) is evident.

Acknowledgment

The authors thank the reviewers for their comments which helped in improving the quality of this paper.

References

[1] S. Kar and A. Routray, “Effect of sleep deprivation on functional connectivity of EEG channels,” *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 43, no. 3, pp. 666–672, May 2013.

[2] H. Li, D. Pan, and C. L. P. Chen, “Intelligent prognostics for battery health monitoring using the mean entropy and relevance vector machine,” *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 7, pp. 851–862, Jul. 2014.

[3] S.-C. Horng, “Combining artificial bee colony with ordinal optimization for stochastic economic lot scheduling problem,” *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 3, pp. 373–384, Mar. 2015.

[4] Y. Zhang, Z. Zheng, and M. R. Lyu, “An online performance prediction framework for service-oriented systems,” *IEEE Trans. Syst. Man Cybern., Syst.*, vol. 44, no. 9, pp. 1169–1181, Sep. 2014.

[5] L. Wu, S. Liu, and Y. Yang, “A gray model with a time varying weighted generating operator,” *IEEE Trans. Syst. Man Cybern., Syst.*, vol. 46, no. 3, pp. 427–433, Mar. 2016.

[6] A. Sfetsos and C. Siripoulos, “Time series forecasting of averaged data with efficient use of information,” *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 35, no. 5, pp. 738–745, Sep. 2005.

[7] K.-H. Huarng, T. H.-K. Yu, and Y. W. Hsu, “A multivariate heuristic model for fuzzy time-series forecasting,” *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 4, pp. 836–846, Aug. 2007.

[8] K. Kolomvatsos, C. Anagnostopoulos, and S. Hadjiefthymiades, “Data fusion and type-2 fuzzy inference in contextual data stream monitoring,” *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published, doi: 10.1109/TSMC.2016.2560533.

[9] C. F. Stallmann and A. P. Engelbrecht, “Gramophone noise detection and reconstruction using time delay artificial neural networks,” *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 6, pp. 893–905, Jun. 2017.

[10] T. Taniguchi *et al.*, “Sequence prediction of driving behavior using double articulation analyzer,” *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 9, pp. 1300–1313, Sep. 2016.

[11] B. L. Bowerman and R. T. O’Conell, *Time Series Forecasting: Unified Concepts and Computer Implementation*, 2nd ed. Boston, MA, USA: Duxbury Press, 1987.

[12] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, *Time Series Analysis: Forecasting and Control*, 3rd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 1994.

[13] J. D. Hamilton, *Time Series Analysis*. Princeton, NJ, USA: Princeton Univ. Press, 1994.

[14] C. Chatfield, *The Analysis of Time Series: An Introduction*, 5th ed. London, U.K.: Chapman & Hall, 1995.

[15] G. U. Yule, “On a method of investigating periodicities in disturbed series, with special reference to Wolfer’s sunspot numbers,” *Philosph. Trans. Roy. Soc. A Math. Phys. Eng. Sci.*, vol. 226, pp. 267–298, Jan. 1927.

[16] N. A. Gershenfeld and A. S. Weigend, “The future of time series: Learning and understanding,” in *Time Series Prediction: Forecasting the Future and Understanding the Past*, A. S. Weigend and N. A. Gershenfeld, Eds. Reading, MA, USA: Addison-Wesley, 1993, pp. 1–70.

[17] D. S. G. Pollack, *A Handbook of Time-Series Analysis, Signal Processing and Dynamics*. London, U.K.: Academic Press, 1999.

[18] J. J. F. Commandeur and S. J. Koopman, *An Introduction to State Space Time Series Analysis*. Oxford, U.K.: Oxford Univ. Press, 2007.

[19] M. Chauvet and S. Potter, “Forecasting output,” in *Handbook of Economic Forecasting, Volume 2A*, G. Elliott and A. Timmermann, Eds. Amsterdam, The Netherlands: Elsevier, 2013, pp. 141–194.

[20] C. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

[21] G. Cybenko, “Approximation by superpositions of sigmoidal function,” *Math. Control Signal Syst.*, vol. 2, no. 4, pp. 303–314, Dec. 1989.

[22] K.-I. Funahashi, “On the approximate realization of continuous mappings by neural networks,” *Neural Netw.*, vol. 2, no. 3, pp. 183–192, 1989.

[23] K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” *Neural Netw.*, vol. 2, no. 5, pp. 359–366, 1989.

[24] K. S. Narendra and K. Parthasarathy, “Identification and control of dynamical systems using neural networks,” *IEEE Trans. Neural Netw.*, vol. 1, no. 1, pp. 4–27, Mar. 1990.

[25] Z. Tang and P. A. Fishwick, “Feed-forward neural nets as models for time series forecasting,” *ORSA J. Comput.*, vol. 5, no. 4, pp. 374–385, 1993.

[26] A. Shabri, “Comparison of time series forecasting methods using neural networks and Box–Jenkins model,” *Matematika*, vol. 17, pp. 25–32, Jun. 2001.

[27] Y. Ye, S. Squartini, and F. Piazza, “Online sequential extreme learning machine in nonstationary environments,” *Neurocomputing*, vol. 116, pp. 94–101, Sep. 2013.

[28] L. A. Zadeh and C. A. Desoer, *Linear System Theory: The State Space Approach*. New York, NY, USA: McGraw-Hill Book Company, 1963.

[29] G. C. Goodwin and K. Sin, *Adaptive Filtering, Prediction and Control*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1984.

[30] H. Tong and K. S. Lim, “Threshold autoregression, limit cycles, and cyclical data,” *J. Roy. Stat. Soc. B*, vol. 42, no. 3, pp. 245–292, 1980.

[31] H. Tong, *Nonlinear Time Series Analysis: A Dynamical Systems Approach*. Oxford, U.K.: Oxford Univ. Press, 1990.

[32] R. Sitte and J. Sitte, “Analysis of the predictive ability of time delay neural networks applied to the S&P 500 time series,” *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 30, no. 4, pp. 568–572, Nov. 2000.

[33] H. S. Black, “Inventing the negative feedback amplifier,” *IEEE Spectr.*, vol. 14, no. 12, pp. 55–60, Dec. 1977.

[34] J. D. Farmer and J. J. Sidorowich, “Exploiting chaos to predict the future and reduce noise,” in *Evolution, Learning, and Cognition*, Y. C. Lee, Ed. Singapore: World Scientific, 1988, pp. 277–330.

[35] K. George, S. Prabhu, and P. Mutalik, “An online multiple-model approach to univariate time-series prediction,” in *Proc. ELM Vol. 1*, 2015, pp. 215–225.

[36] 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.

[37] Y. Gu, J. Liu, Y. Chen, X. Jiang, and H. Yu, “TOSELM: Timeliness online sequential extreme learning machine,” *Neurocomputing*, vol. 128, pp. 119–127, Mar. 2014.

[38] W.-Y. Deng, Y.-S. Ong, P. S. Tan, and Q.-H. Zeng, “Online sequential reduced kernel extreme learning machine,” *Neurocomputing*, vol. 174, pp. 72–84, Jan. 2016.

[39] O. L. Mangasarian and M. V. Solodov, “Backpropagation convergence via deterministic nonmonotone perturbed minimization,” in *Advances in Neural Information Processing Systems 6*, J. D. Cowan, G. Tesauro, and J. Alspector, Eds. San Francisco, CA, USA: Morgan Kaufmann, 1994, pp. 383–390.

[40] G. D. Magoulas, M. N. Vrahatis, and G. S. Androulakis, “Improving the convergence of the backpropagation algorithm using learning rate adaptation methods,” *Neural Comput.*, vol. 11, no. 7, pp. 1769–1796, Oct. 1999.

[41] D. L. Phillips, “A technique for the numerical solution of certain integral equations of the first kind,” *J. Assoc. Comput. Mach.*, vol. 9, no. 1, pp. 84–97, 1962.

[42] A. N. Tikhonov, “On solving incorrectly posed problems and method of regularization,” *Doklady Akademii Nauk USSR*, vol. 151, no. 3, pp. 501–504, 1963.

[43] A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill-Posed Problems*. Washington, DC, USA: V. H. Winston, 1977.

[44] A. H. Sayed, *Adaptive Filters*. Hoboken, NJ, USA: Wiley, 2008.

[45] K. Subramanian, S. G. Krishnappa, and K. George, “Performance comparison of learning algorithms for system identification and control,” in *Proc. 12th IEEE India Int. Conf. (INDICON)*, New Delhi, India, Dec. 2015, pp. 1–6.

[46] K. George, K. Subramanian, and N. Seshadhri, “Improving transient performance in adaptive control of nonlinear systems,” in *Proc. 4th Int. Conf. Adv. Control Optim. Dyn. Syst. (ACODS)*, Tiruchirappalli, India, Feb. 2016, pp. 658–663.

[47] K. George and P. Mutalik, “Online time series prediction with metacognition,” in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Vancouver, BC, Canada, Jul. 2016, pp. 2124–2131.

[48] 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, 2006.

[49] J. P. Nobrega and A. L. I. Oliveira, “Kalman filter-based method for online sequential extreme learning machine for regression problems,” *Eng. Appl. Artif. Intell.*, vol. 44, pp. 101–110, Sep. 2015.

[50] H. T. Huynh and Y. Won, “Regularized online sequential learning algorithm for single-hidden layer feedforward neural networks,” *Pattern Recognit. Lett.*, vol. 32, no. 14, pp. 1930–1935, 2011.

[51] X. Wang and M. Han, “Online sequential extreme learning machine with kernels for nonstationary time series prediction,” *Neurocomputing*, vol. 145, pp. 90–97, Dec. 2014.

[52] S. Scardapane, D. Comminiello, M. Scarpiniti, and A. Uncini, “Online sequential extreme learning machine with kernels,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 9, pp. 2214–2220, Sep. 2015.

[53] S. Hashem and B. Schmeiser, “Improving model accuracy using optimal linear combinations of trained neural networks,” *IEEE Trans. Neural Netw.*, vol. 6, no. 3, pp. 792–794, May 1995.

[54] P. G. Zhang, “Time series forecasting using a hybrid ARIMA and neural network model,” *Neurocomputing*, vol. 50, pp. 159–175, Jan. 2003.

[55] P. J. Garcìa-Laencina, “Improving predictions using linear combinations of multiple extreme learning machines,” *Inf. Technol. Control*, vol. 42, no. 1, pp. 86–93, 2013.

[56] N. Liu and H. Wang, “Ensemble based extreme learning machine,” *IEEE Signal Process. Lett.*, vol. 17, no. 8, pp. 754–757, Aug. 2010.

[57] M. V. Heeswijk *et al.*, “Adaptive ensemble models of extreme learning machines for time series prediction,” in *Proc. 19th Int. Conf. Artif. Neural Netw. Part II*, Limassol, Cyprus, 2009, pp. 305–314.

[58] K. S. Narendra and J. Balakrishnan, “Performance improvement in adaptive control systems using multiple models and switching,” in *Proc. 7th Yale Workshop Adapt. Learn. Syst.*, New Haven, CT, USA, May 1992, pp. 27–33.

[59] K. S. Narendra and J. Balakrishnan, “Adaptive control using multiple models,” *IEEE Trans. Autom. Control*, vol. 42, no. 2, pp. 171–187, Feb. 1997.

[60] K. S. Narendra and C. Xiang, “Adaptive control of discrete-time systems using multiple models,” *IEEE Trans. Autom. Control*, vol. 45, no. 9, pp. 1669–1686, Sep. 2000.

[61] K. S. Narendra and O. A. Driollet, “Stochastic adaptive control using multiple models for improved performance in the presence of random disturbances,” *Int. J. Adapt. Control Signal Process.*, vol. 15, no. 3, pp. 287–317, May 2001.

[62] K. S. Narendra, O. A. Driollet, M. Feiler, and K. George, “Adaptive control of time-varying systems using multiple models,” *Int. J. Adapt. Control Signal Process.*, vol. 17, no. 2, pp. 87–102, Mar. 2003.

[63] K. S. Narendra and K. George, “Adaptive control of simple nonlinear systems using multiple models,” in *Proc. Amer. Control Conf.*, Anchorage, AK, USA, May 2002, pp. 1779–1784.

[64] L. Chen and K. S. Narendra, “Nonlinear adaptive control using neural networks and multiple models,” *Automatica*, vol. 37, no. 8, pp. 1245–1255, Aug. 2001.

[65] K. George, “Some applications of multiple models methodology,” in *Proc. 15th Yale Workshop Adapt. Learn. Syst.*, New Haven, CT, USA, Jun. 2011, pp. 81–86.

[66] P. A. P. Moran, “The statistical analysis of the Canadian lynx cycle–I: Structure and prediction,” *Aust. J. Zool.*, vol. 1, no. 2, pp. 163–173, 1953.

[67] A. S. Weigend and N. A. Gershenfeld, Eds., *Time Series Prediction: Forecasting the Future and Understanding the Past*. Reading, MA, USA: Addison-Wesley, 1993.

[68] M. C. Mackey and L. Glass, “Oscillation and chaos in physiological control systems,” *Science*, vol. 197, no. 4300, pp. 287–289, 1977.

[69] N. Kasabov, “Evolving fuzzy neural networks for supervised/unsupervised online knowledge-based learning,” *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 31, no. 6, pp. 902–918, Dec. 2001.

[70] C.-H. Chen and W.-H. Chen, “United-based imperialist competitive algorithm for compensatory neural fuzzy systems,” *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 9, pp. 1180–1189, Sep. 2016.

[71] M. Prasad *et al.*, “Soft-boosted self-constructing neural fuzzy inference network,” *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 3, pp. 584–588, Mar. 2017.

[72] C. L. P. Chen and J. Z. Wan, “A rapid learning and dynamic stepwise updating algorithm for flat neural networks and the application to time-series prediction,” *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 29, no. 1, pp. 62–72, Feb. 1999.

[73] K. George, M. Verhaegen, and J. M. A. Scherpen, “A systematic and numerically efficient procedure for stable dynamic model inversion of LTI systems,” in *Proc. 38th IEEE Conf. Decis. Control*, Phoenix, AZ, USA, Dec. 1999, pp. 1881–1886.

[74] F. X. Diebold and R. S. Mariano, “Comparing predictive accuracy,” *J. Bus. Econ. Stat.*, vol. 13, no. 3, pp. 134–144, Jul. 1995.

[75] J. Demšar, “Statistical comparisons of classifiers over multiple data sets,” *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Jan. 2006.

[76] S. García and F. Herrera, “An extension on ‘Statistical comparisons of classifiers over multiple data sets’ for all pairwise comparisons,” *J. Mach. Learn. Res.*, vol. 9, pp. 2677–2694, Dec. 2008.

[77] F. Wilcoxon, “Individual comparisons by ranking methods,” *Biometr. Bull.*, vol. 1, no. 6, pp. 80–83, Dec. 1945.

[78] F. J. Massey, Jr., “The Kolmogorov–Smirnov test for goodness of fit,” *J. Amer. Stat. Assoc.*, vol. 46, no. 253, pp. 68–78, Mar. 1951.

[79] D. Harvey, S. Leybourne, and P. Newbold, “Testing the equality of prediction mean squared errors,” *Int. J. Forecasting*, vol. 13, no. 2, pp. 281–291, 1997.

[80] F. X. Diebold, “Comparing predictive accuracy, twenty years later: A personal perspective on the use and abuse of Diebold–Marino tests,” *J. Bus. Econ. Stat.*, vol. 33, no. 1, pp. 1–9, Jan. 2015.

[81] H. Hassani and E. S. Silva, “A Kolmogorov–Smirnov based test for comparing the predictive accuracy of two sets of forecasts,” *Econometrics*, vol. 3, no. 3, pp. 590–609, 2015.

Koshy George (SM’05) received the B.E. degree in electrical and electronics engineering from the University of Mysore, Mysuru, India, the M.S. degree in electrical engineering from the Indian Institute of Technology Madras, Chennai, India, and the Ph.D. degree in engineering from the Indian Institute of Science, Bengaluru, India.

He is currently a Professor of Electronics and Communications Engineering with PES University, Bengaluru, where he is also the Director of the PES Centre for Intelligent Systems. His current research interests include adaptive systems and nonlinear systems.

Prabhanjan Mutalik received the B.E. degree from Visvesvaraya Technological University, Belgaum, India, in 2014. He is currently pursuing the M.S. degree in machine learning with the KTH Royal Institute of Technology, Stockholm, Sweden.

From 2015 to 2016, he was with the PES Centre for Intelligent Systems, PES University, Bengaluru, India. His current research interests include time-series prediction, deep learning, and biologically inspired computing.