

Online Sequential Extreme Learning Machine Algorithm for Better Predispach Electricity Price Forecasting Grids

Chixin Xiao ¹, Member, IEEE, Danny Sutanto ², Senior Member, IEEE,
 Kashem M. Muttaqi ³, Senior Member, IEEE, Minjie Zhang ⁴, Senior Member, IEEE,
 Ke Meng ⁵, Senior Member, IEEE, and Zhao Yang Dong ⁶, Fellow, IEEE

Abstract—The predispach price forecast plays a key element in the electricity market. However, such a forecast usually depends on the traditional offline batch-learning technologies, which cannot respond in time to the unexpected changes in the local power system environment. Further, the predispach local price forecast is often affected by the dynamic price changes from the neighboring regions. This article proposes a novel online learning forecast approach to overcome the above issues to provide a better predispach price forecast by using the online sequential extreme learning machine (OS-ELM) algorithm. The article proposes a novel data structure in the form of a 2-D orthogonal list and two corresponding OS-ELM modules. One module provides the rolling day-ahead price prediction and prediction intervals using the day-by-day online training update, while the other provides the rolling 30-min prediction using the 2-h-by-2-h online training update. The proposed approach can continuously perceive any unexpected events and any price fluctuations from the neighboring regions in the nonlinear patterns. The proposed approach is validated using simulation studies based on the data from the Australian electricity market, and the simulation results show that the proposed approach can help in improving the forecast accuracy,

especially when unexpected changes occur both locally and in the neighboring area.

Index Terms—Electricity market, electricity price, extreme learning machine, online training, regression analysis.

I. INTRODUCTION

THE predispach electricity price forecast, conventionally published day-ahead but now a half-hour ahead plays an important role in the electricity market to manage the generation schedules in the upcoming trading day. As the large consumers are increasingly encouraged to participate in the electricity markets, they need to be able to react intelligently to the forecasted spot price [1], [2] to gain economic benefit. Further, the power generators need to be scheduled and dispatched to match the prevailing demand every 5 minutes. An accurate and reliable electricity price forecast within shorter intervals is therefore required by the market participants.

The predispach electricity price forecasting (PDPF), also known as short-term electricity price prediction (SEPP) [3], [4], is a representative time-series regression model [5] to estimate the functional relationships between the dependent variables (often called the “target variables”) and the independent variables (often called the “input variables”) by analyzing the historical dataset using a group of statistical or machine learning [6] processes.

Many SEPP methodologies have been adopted to forecast the day-ahead electricity price, such as the use of the point, probabilistic, and threshold price forecasting. Authors in [7] and [8] proposed the use of the point price forecasting, i.e., the forecast of a single-valued electricity price in a certain fixed horizon in the future, which is dependent on the price dataset in a period T . Authors in [9] and [10] proposed the use of the probabilistic forecasting method that forecasts a probability of the electricity price instead of a single value electricity price. The threshold forecasting method utilizes a classification method to divide the future prices into several thresholds, when the task does not require an exact prediction on the future prices but requires several prespecified price thresholds for the decision-making process, e.g., for demand-side management [11]. In general, in practical engineering, not only the point price forecasting is important for risk management but the quantitative estimations

Manuscript received June 10, 2020; revised November 4, 2020; accepted December 25, 2020. Date of publication January 12, 2021; date of current version March 17, 2021. Paper 2020-IACC-0908.R1, presented at the 2019 IAS Annual Meeting, Baltimore, MD, USA, Sep. 29–Oct. 3, and approved for publication in the IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS by the Industrial Automation and Control Committee of the IEEE Industry Applications Society. This work was supported in part by the Australian Research Council linkage under Grant LP0991428, in part by the ARC Discovery Project under Grant DP140100974, in part by the ARC Research Hub for Integrated Energy Storage Solutions under Grant IH180100020, in part by Hunan province science and technology project funds under Grant 2018TP1036, China (*Corresponding author: Chixin Xiao.*)

Chixin Xiao is with the School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, NSW 2522, Australia, and also with Key Laboratory of Hunan Province for Internet of Things and Information Security, Xiangtan University, Xiangtan 411105, China (e-mail: cx472@uowmail.edu.au).

Danny Sutanto and Kashem M. Muttaqi are with the School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, NSW 2522, Australia (e-mail: soetanto@uow.edu.au; kashem@uow.edu.au).

Minjie Zhang is with the School of Computer Science and Software Engineering, University of Wollongong, Wollongong, NSW 2522, Australia (e-mail: minjie@uow.edu.au).

Ke Meng and Zhao Yang Dong are with the School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, NSW 2052, Australia (e-mail: joe.dong@unsw.edu.au; ke.meng@unsw.edu.au).

This article has supplementary downloadable material available at <https://doi.org/10.1109/TIA.2021.3051105>, provided by the authors.

Digital Object Identifier 10.1109/TIA.2021.3051105

of the uncertainties during the prediction intervals (PI) [12] are also required. Moreover, the performance quality of the SEPP methodologies is also important. Although the conventional approaches for the time-series analysis in statistics, e.g., autoregressive integrated moving average [13], mainly rely on linear models, which can fit the stationary data [14] well, they also need to have the prediction performance improved, as the linear perception can hardly be a universal estimator that can accurately fit the nonstationary data, such as electricity price series [15].

The artificial neural network (ANN) [16] is a mature machine learning methodology, that is more popular than the conventional methods, due to its unique ability to recognize the nonlinear relationship between the input and the target variables of the datasets. Deep learning or deep structured learning [17], a modern variation of ANN, has recently become more attractive because of its more universal capacity on dealing with the nonstationary time series data. However, the training of the structure of ANN, to optimize the hidden layers, the corresponding weights, and the biases, depends on the batch-learning process [18]–[20]. Such a process is often time-consuming and requires massive training data and training iterations. The batch learning process can be understood as a regression analysis consisting of two consecutive parts: 1) the training and 2) the forecasting parts. In the training part, useful patterns and information will be extracted from the historical data until a certain threshold is achieved; and in the forecasting part, the trained regression pattern can be used to calculate the long term or a short-term real-time prediction as the new data are received. In the batch-learning approaches, the trained regression pattern is not updated instantly in the forecasting part. Thus, the decision making based on the batch-learning algorithms is not adaptive and cannot react quickly to the unexpected environmental changes in the power market.

The extreme learning machine [21] (ELM), a machine learning algorithm having similar ideas as [22] and [23], relies on the weighted least squares regression [24] and uses the mathematical pseudoinverse operation [25], instead of the iterative method used in ANN, to obtain the hidden layers of the learning algorithms [24]. Thus, compared to other machine learning approaches [16], the use of the ELM approaches can obtain a faster speed in training the same massive data.

However, the conventional batch-learning mode using the ELM still cannot respond in time to any unexpected changes due to the increasing penetration of intermittent distributed energy resources [26], [27], such as the wind, solar, and wave renewable energy resources [28], [29] in the modern smart grid. Further, there is increasing use of the battery energy storage systems and an increased interregional electricity trade [30]. To this end, a real-time data-training on the recent local price is necessary to timely perceive the future uncertainties in the regional market. Besides that, the spot price can be considered as a function of many local factors, such as the local weather condition, the local fuel cost, and the price fluctuations, etc. Therefore, considering the spot price data from other connected grids in time [31] is also helpful to enhance the forecasting accuracy against uncertain factors from the global perspective.

Another member among the ELM family [21] referred to as the online sequential extreme learning machine (OS-ELM) [32], which has the one-by-one learning manner or the block-by-block with a fixed or varying data block size is helpful to provide a regular updating on the initial training. Authors in [33] and [34] suggest a forgetting factor in OS-ELM to forget the old data gradually while emphasizing the new coming data. However, in power engineering, the price forecasting often needs to be categorized seasonally, that is, the time-series data vary according to the seasons, therefore, the forgetting factor is not used in this article. Given that, in this article, the use of the quantile forecast, spot forecast, and online learning on the electricity price forecast via OS-ELM to improve the forecast accuracy and efficiency are more practical and essential.

Therefore, this article proposes a novel machine learning approach, hybridizing two parallel OS-ELM modules, one module provides the rolling day-ahead price prediction and the corresponding PI using the day-by-day online training update, while the other provides the rolling 30-min prediction using the 2-h-by-2-h online training update during the PI. The second OS-ELM considers the impact of the price information change from the other grid-connected regions, which can cause the local price to spike. A novel data-structure using a 2-D orthogonal list is developed to match the proposed OS-ELM modules. In the orthogonal list, each data cell in every 48-point group is linked symmetrically, not only to the 30-min predecessor and the 30-min successor but also to the day-ahead predecessor and day-after successor. For example, for the real-time data at 10:00 A.M., its 30-min predecessor and the 30-min successor are the data at 9:30 A.M. and 10:30 A.M., respectively; and its day-ahead predecessor and day-after successor data are at 10:00 A.M., 24-h ahead and 10:00 A.M., 24-h after, respectively. Accordingly, one of the proposed two parallel OS-ELM modules provides a rolling day-ahead price prediction during the PI, in which the trained regression pattern has a regular training update every day by using all data collected per 30-min over the past 24 h; the other OS-ELM provides a dynamic 30-min ahead prediction, in which the trained regression pattern has a regular training update every two hours by using all data collected per 30-min over the past two hours.

The novelty and contributions of this article are as follows.

- 1) A hybrid OS-ELM consisting of two types of OS-ELMs is proposed to react quickly to the unexpected changes in the power market, in which one OS-ELM with the day-by-day update provides the rolling day-ahead price prediction; the other one with the 2-h-by-2-h update provides the rolling 30-min prediction.
- 2) A novel 2-D orthogonal list is implemented to support the quantile forecast, spot forecast crossly to perceive two types of price information out of the massive historical price data to provide not only the day-ahead predictions but also the 30-min rolling predictions.
- 3) Two kinds of strategies are developed to cope with prediction uncertainties: i) PI is evaluated to quantile the prediction uncertainties in the first OS-ELM for the day-ahead prediction; ii) the impact of the price information change from the other grid-connected regions is considered in

the second OS-ELM for the 30-min rolling prediction to enhance the forecasting accuracy against uncertain factors on the global perspective;

- 4) The *Binary Search* [35] approach is utilized to refine the optimal structure of the proposed OS-ELM modules, such as the number of the hidden layers, the weights, and the biases.

The rest of the article is organized as follows. Section II describes the mathematical background of the batch learning ELM and the online sequential ELM. Section III introduces the 2-D orthogonal list structure, the grid-connected formulation for the local price prediction, and the proposed learning algorithm. Section IV describes the simulation data and provides the simulation results. Finally, Section V gives the conclusion.

II. RELEVANCE REVIEWS

This section gives a brief review of ELM and other methodologies.

A. ELM and Batch-Learning

Given a dataset consisting of N data pairs as

$$\{(x_j, t_j) | x_j \in R^n, t_j \in R^m, j \in [1, N]\} \quad (1)$$

where $x_j = (x_{j1}, \dots, x_{jn})^T \in R^n$ denotes the j th $n \times 1$ input vector, $t_j = (t_{j1}, \dots, t_{jm}) \in R^m$ denotes the j th $1 \times m$ target vector, n and m represent the input and the output numbers of the ELM, respectively, a trained ELM can be understood as a feedforward network (FN) [21] network with n -input, m -outputs, and M -layer-hidden nodes in a unified framework, which can approximately estimate the functional relationship between each x_j and t_j , often to be denoted as follows:

$$t_j \approx f_M(x_j) = \sum_{i=1}^M o_i \cdot g(a_i \cdot x_j + b_i) \quad (2)$$

where $a_i = (a_{i1}, \dots, a_{in}) \in R^n$, $i \in [1, M]$, is the weight vector linking the input layer to the i th hidden node, and $b_i \in R$ is the corresponding bias (or noise) of the i th hidden node; $o_i = (o_{i1}, \dots, o_{im}) \in R^m$ is the respective weight connecting the i th hidden node and the output nodes, $g(\cdot) : R \rightarrow R$ is the activation function, where the sigmoid function is often adapted, i.e.,

$$g(a_i \cdot x_j + b_i) = \frac{1}{1 + \exp(-(a_i \cdot x_j + b_i))}. \quad (2.1)$$

The ELM can be trained by using the batch-learning (i.e., BL-ELM) mode, in which the BL-ELM can organize a large number of data pairs [e.g., the input-target samples as shown in (1)] as an integrated training-group. According to [8], [21], given N data-pair samples, $\{(x_j, t_j)\}_{j=1}^N$, the corresponding weight vectors and biases, $a_i, b_i, i \in [1, M]$, can be assigned with random values. Thus, the training process of the BL-ELM can be understood as determining a least-squares solution of (2), i.e., to optimize the corresponding o_i to minimize the error between each t_j and its estimated value $f_M(x_j)$.

For simplicity, (2) can be expressed in the matrix notation as

$$H \cdot O = T \quad (3)$$

where

$$H = \begin{bmatrix} g(a_1 \cdot x_1 + b_1) & \cdots & g(a_M \cdot x_1 + b_M) \\ \vdots & \cdots & \vdots \\ g(a_1 \cdot x_N + b_1) & \cdots & g(a_M \cdot x_N + b_M) \end{bmatrix}_{N \times M} \quad (3.1)$$

$$O = \begin{bmatrix} o_1 \\ \vdots \\ o_M \end{bmatrix}_{M \times m} \quad (3.2)$$

$$T = \begin{bmatrix} t_1 \\ \vdots \\ t_N \end{bmatrix}_{N \times m}. \quad (3.3)$$

Equation (3.1) denotes the output matrix of the hidden layers; (3.2) denotes the matrix of the output weights; and (3.3) denotes the matrix of the target vectors.

Thus, the least-squares solution of (3), denoted by \hat{O} , can be estimated by using

$$\hat{O} = \hat{H}^\dagger \cdot T \quad (4)$$

where \hat{H}^\dagger is called the left pseudo-inverse of H (whose definition is given in (3.1)) and can be obtained by using

$$\hat{H}^\dagger = (H' \cdot H)^{-1} \cdot H' \quad (5)$$

where H' represents the transpose of H .

The quality of the least-squares solution \hat{O} can be measured by the root mean square error (RMSE) between the estimated value $H \cdot \hat{O}$ [according to (3.1) and (4)] and the target matrix T [given in (3)]. The formula is given by

$$\text{Minimize RMSE} = \sqrt{\text{mse}(H \cdot \hat{O} - T)} \quad (6)$$

where $\text{mse}(\cdot)$ denotes the mean squared error function, i.e., $\frac{1}{\text{MaxSize}} \sum_{i=1}^{\text{MaxSize}} (H \cdot \hat{O} - T)_i^2$. When RSME is small enough, such parameters as \hat{O} , a_i and b_i can be deemed to fit the BL-ELM on the N -training samples perfectly. Thus, given a new input vector, the BL-ELM with the trained \hat{O} , a_i and b_i can be utilized to predict the corresponding outputs, i.e., $H \cdot \hat{O}$, as shown in (3).

B. Online Sequential ELM (OS-ELM)

The BL-ELM demands the ELM algorithm to have all training data, e.g., N data-pair samples, prepared before training. However, in practical situations, the ELM algorithm often needs to learn from the new incoming data to detect any unexpected factors in time. OS-ELM can append the new learning parameters to the old trained results to increase the training accuracy.

If there are 2-group of distinct samples of different size: N_0 and N_1 . Based on the N_0 training data, the least-squares solution \hat{O}_0 with its corresponding weight and bias parameters $a_{i,0}$ and $b_{i,0}$ ($i \in [1, M]$), can be obtained while minimizing (6) as

$$\text{Min}(RMSE) = \sqrt{\text{mse}(H_0 \cdot \hat{O}_0 - T_0)} \quad (7)$$

where

$$H_0 = \begin{bmatrix} g(a_{10} \cdot x_1 + b_{10}) & \cdots & g(a_{M,0} \cdot x_1 + b_{M,0}) \\ \vdots & \cdots & \vdots \\ g(a_{10} \cdot x_{N_0} + b_{10}) & \cdots & g(a_{M,0} \cdot x_{N_0} + b_{M,0}) \end{bmatrix}_{N_0 \times M} \quad (7.1)$$

and

$$\hat{O}_0 = (H'_0 \cdot H_0)^{-1} \cdot H'_0 \cdot T_0. \quad (7.2)$$

Let $K_0 = (H'_0 \cdot H_0)$, then (7.2) can be rewritten as

$$\hat{O}_0 = K_0^{-1} \cdot H'_0 \cdot T_0. \quad (7.3)$$

Likewise, the new hidden layer matrix H_1 and its corresponding weight and bias parameters, $a_{i,1}$ and $b_{i,1}$ ($i \in [1, M]$), can be developed based on the N_1 training data.

Suppose the least-squares solution \hat{O} and the corresponding parameters are based on learning the 2-group samples together, i.e., N_0 and N_1 as

$$\text{Min (RMSE)} \sqrt{\text{mse} \left(\begin{bmatrix} H_0 \\ H_1 \end{bmatrix} \cdot \hat{O} - \begin{bmatrix} T_0 \\ T_1 \end{bmatrix} \right)}. \quad (8)$$

The least-squares solution \hat{O} can be expressed as an updating process based on the solution \hat{O}_0 and the trained results. As shown in (9), \hat{O} is represented in a recursive way

$$\hat{O} = K^{-1} \cdot \begin{bmatrix} H_0 \\ H_1 \end{bmatrix}' \cdot \begin{bmatrix} T_0 \\ T_1 \end{bmatrix} \quad (9)$$

where $K = \begin{bmatrix} H_0 \\ H_1 \end{bmatrix}' \cdot \begin{bmatrix} H_0 \\ H_1 \end{bmatrix} = [H'_0 \cdot H_0 + H'_1 \cdot H_1]$.

According to (7.3), the K in (9) can be rewritten as

$$K = K_0 + H'_1 \cdot H_1. \quad (9.1)$$

The product of the last two matrices in (9) can be formulated as

$$\begin{aligned} \begin{bmatrix} H_0 \\ H_1 \end{bmatrix}' \cdot \begin{bmatrix} T_0 \\ T_1 \end{bmatrix} &= H'_0 \cdot T_0 + H'_1 \cdot T_1 \\ &= K_0 \cdot K_0^{-1} \cdot H'_0 \cdot T_0 + H'_1 \cdot T_1 \\ &= K_0 \cdot \hat{O}_0 + H'_1 \cdot T_1 \\ &= (K - H'_1 \cdot H_1) \cdot \hat{O}_0 + H'_1 \cdot T_1 \\ &= K \cdot \hat{O}_0 - H'_1 \cdot H_1 \cdot \hat{O}_0 + H'_1 \cdot T_1. \end{aligned} \quad (9.2)$$

By combining (9.1) and (9.2), \hat{O} can be expressed as

$$\begin{aligned} \hat{O} &= K^{-1} \cdot \left(K \cdot \hat{O}_0 - H'_1 \cdot H_1 \cdot \hat{O}_0 + H'_1 \cdot T_1 \right) \\ &= \hat{O}_0 - K^{-1} \cdot H'_1 \cdot \left(H_1 \cdot \hat{O}_0 - T_1 \right) \end{aligned} \quad (10)$$

where K can be obtained by using (9.1).

When a set of new data are available, the least-squares solution \hat{O} can be updated on the trained result \hat{O}_0 by using (10).

In general, the updated least-squares solution \hat{O} and the new hidden layer $H = \begin{bmatrix} H_0 \\ H_1 \end{bmatrix}$ will include more recent information

than H_0 and \hat{O}_0 , which can be more accurate in a real-time prediction.

C. PIs and the Evaluation

While, so far, we are only concerned with the point price forecast, in practical engineering, it is useful to take advantage of the PI to quantify the level of the associated uncertainties and to make smart decisions.

Given a training dataset as defined in (1), the PI, which is a simple combination of the point estimate with the corresponding estimated standard error [9], is associated with a confidence level written as $(1 - \alpha)$, e.g., 90%, 95%, and 99%, where α corresponds to 0.1, 0.05, and 0.01, respectively. Based on (2), the mathematical definition of PI_α is defined by

$$\begin{aligned} PI_\alpha &= (f_M(x_j) - z_{1-\frac{\alpha}{2}} \cdot \text{std}(f_M(x_j) - t_j), f_M(x_j) \\ &\quad + z_{1-\frac{\alpha}{2}} \cdot \text{std}(f_M(x_j) - t_j)). \end{aligned} \quad (11)$$

where the term $f_M(x_j)$ represents the estimated output of the ELM as shown in (2), where $z_{1-\frac{\alpha}{2}}$ is the critical point of the standard normal distribution, and $\text{std}(\cdot)$ is the standard deviation.

The term, PI nominal confidence (PINC), indicates the confidence of quantile predictions, which specifies that the future forecasting points $f_M(x_j)$ will lie within the trained confidence interval as

$$P(f_M(x_j) \in PI_\alpha) = 1 - \alpha. \quad (11.1)$$

Given the dataset, as shown in (1), the term, PI coverage probability (PICP), shows the actual coverage probability of the corresponding PI as

$$\text{PICP} = \frac{1}{N} \sum_{j=1}^N \text{Boolean}(f_M(x_j) \in PI_\alpha) \quad (11.2)$$

where the function $\text{Boolean}(\cdot)$ returns with one when the mathematical expression in the bracket is true, otherwise, with zero.

Finally, the PI quality is often assessed by using the difference between PICP and PINC, called the average coverage error (ACE) as

$$\text{ACE} = \text{PICP} - \text{PINC}. \quad (12)$$

III. PROPOSED METHODOLOGY

In this section, the Australian National Electricity Market (ANEM) [36] is adopted to illustrate the proposed method. As shown in Fig. 1(a), the ANEM consists of five collected regions, i.e., New South Wales (NSW), Queensland (QLD), Victoria (VIC), South Australia (SA), and Tasmania (TAS). The pre-dispatch price forecasting helps us to estimate the volume of the electrical power to be supplied in the upcoming trading day via the connected grids. The generators in the interconnected grid are required to submit every 5 minutes (i.e., dispatch interval) of every day their maximum supply capacity and availability, and these are used to schedule and dispatch the generators to meet the prevailing demand. In turn, based on the bids and offers between the demand sides and the suppliers, the market-clearing price is

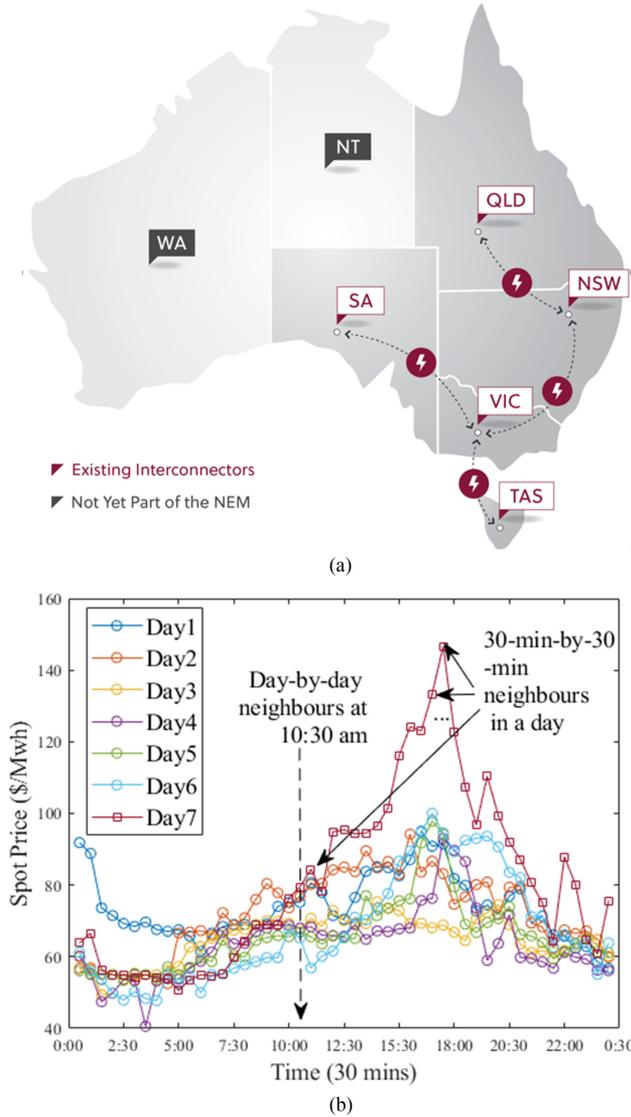


Fig. 1. (a) Grid-connected Regions in ANEM [36]. (b) Neighbors of the orthogonal list illustrated on seven successive days spot price of NSW.

obtained by averaging every six successive 5-min dispatch prices for each 30-min trading interval, where the updated predispach forecast is published. The predispach forecast for NSW is set to be the study instance.

A. 2-D Orthogonal List and Data Preparation

Fig. 1(b) illustrates the collated price data curves of seven successive days of NSW and the links based on the 2-D orthogonal list example as mentioned in Section I. Each data cell can be linked using two types of indices, i.e., day-by-day and 30-min-by-30-min.

Following the day-by-day index, each data cell links to its day-ahead predecessor and its day-after successor, and so on. As shown in Fig. 1(b), the data cell (marked by the square) at 10:30 A.M. of “Day 7” has six similar data cells along with the day-by-day axis, each of which is marked by the circle, and the day-by-day axis is indicated by the dashed arrow at the

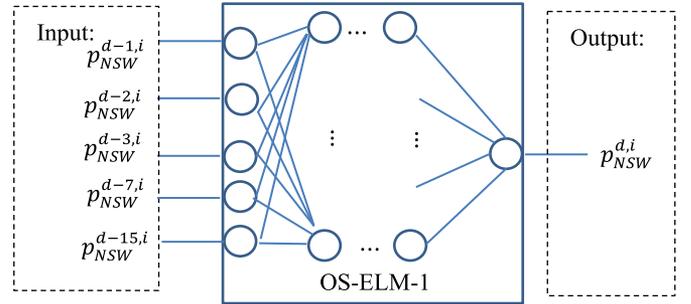


Fig. 2. Forecasting Principle of the OS-ELM-1.

10:30 A.M. Following the 30-min-by-30-min index, each data cell links to its 30-min-ahead predecessor and its 30-min-after successor, and so on. As shown in Fig. 1(b), the same data cell at 10:30 A.M. of “Day 7” has 47 similar 30-min-by-30-min data cells within the same day.

For instance, suppose a time series of the NSW price data is collected per 30 min (i.e., a trading interval) as

$$\begin{aligned} & \dots, p_{NSW}^{d-1,1}, \dots, p_{NSW}^{d-1,48}, p_{NSW}^{d,1}, \dots, p_{NSW}^{d,48}, p_{NSW}^{d+1,1}, \dots, \\ & p_{NSW}^{d+1,48}, \dots \end{aligned} \quad (13)$$

where $p_{NSW}^{d,1}$ and $p_{NSW}^{d,48}$, respectively, denote the NSW price data cells at 00:00 and 23:30 of the d th day, thus, the day-by-day neighbors of the data cell $p_{NSW}^{d,1}$ are $p_{NSW}^{d-1,1}$ and $p_{NSW}^{d+1,1}$, and the 30-min-by-30-min neighbors are $p_{NSW}^{d-1,48}$ and $p_{NSW}^{d,2}$, similarly, the day-by-day neighbors of the data cell $p_{NSW}^{d,48}$ are $p_{NSW}^{d-1,48}$ and $p_{NSW}^{d+1,48}$, and the 30-min-by-30-min neighbors are $p_{NSW}^{d,47}$ and $p_{NSW}^{d+1,1}$.

Two kinds of data preparations are developed by using two types of indices, i.e., day-by-day and 30-min-by-30-min, to support the two proposed OS-ELM modules: OS-ELM-1 is for the day-ahead prediction with PI and OS-ELM-2 is for the prediction at the half-hour level of each day.

B. OS-ELM-1 for the Day-Ahead Prediction With PI

For the day-ahead prediction, each price data cell corresponding to each 30-min trading interval of each day, e.g., $p_{NSW}^{d,i}$ ($1 \leq i \leq 48$), is assumed to have a nonlinear relationship with several direct and indirect predecessors by sampling at the day-by-day axis, such as $p_{NSW}^{d-1,i}$, $p_{NSW}^{d-2,i}$, $p_{NSW}^{d-3,i}$, $p_{NSW}^{d-7,i}$, and $p_{NSW}^{d-14,i}$. Consequently, the data pair (x_j, t_j) as given in (1) in Section II can be detailed as

$$\begin{cases} x_j = \left(p_{NSW}^{d-1,i}, p_{NSW}^{d-2,i}, p_{NSW}^{d-3,i}, p_{NSW}^{d-7,i}, p_{NSW}^{d-14,i} \right)' \\ t_j = p_{NSW}^{d,i} \end{cases}, \quad 1 \leq i \leq 48, d \in Z^+ \quad (14)$$

where Z^+ represent positive integers greater than 15.

The principle of OS-ELM-1, shown in Fig. 2, consists of five main steps.

In the first step, the input, shown in Fig. 2, can be organized in a matrix style. This can save the computational cost by using matrix operations. A sequence of the historical price data, as

shown in (13), can be organized as N_0 (usually a large number) distinct samples, $(x_1, t_1), (x_2, t_2), \dots, (x_{N_0}, t_{N_0})$, and further can form an input $5 \times N_0$ matrix $X = [x_1, x_2, \dots, x_{N_0}]$. Then, H_0 the output matrix of the hidden layers, as mentioned in (7.1) in Section II, can be obtained by using

$$H_0 = g(\text{Normalize}(X') \cdot A + B) \quad (14.1)$$

where $g(\cdot)$ and $\text{Normalize}(\cdot)$, respectively, represent the activation function as given in (2.1) in Section II and the normalizing process; A and B are given as follows:

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1,M} \\ \vdots & \ddots & \vdots \\ a_{51} & \cdots & a_{5,M} \end{bmatrix}, \quad B = I \cdot [b_1, \dots, b_M] \quad (14.2)$$

where $a_{i,j}$ and b_j $i \in [1, M]$ are uniformly distributed random numbers in domain $[0, 1]$; M is the number of hidden layers; and I is a $N_0 \times 1$ matrix of ones.

According to (7)–(7.3) in Section II, the least-squares solution \hat{O}_0 can be obtained by using a matrix operation, which is much faster than the conventional iterative process.

One of the disadvantages of neural networks, including ELM, is the weight initialization. For ELM, different initial weights and distributions would generate different results [a]. Consequently, multiple initializations should generate different optimal numbers of hidden neurons. Therefore, in this article, the initialization would keep running until the batch-training error rate can reach a satisfactory level, (in this article the RMSE is chosen to be less than 20%, or 0.2), which is significant to affect the results of the successor training, i.e., the online training process. Finally, the entire error rate should be within 10%.

Thus, in the second step, the test forecast is processed to validate the quality of the obtained least-squares solution \hat{O}_0 by organizing another group of historical data as similar as in the first step, to obtain new hidden-layer output H and calculating $H \cdot \hat{O}_0$, as shown in (3). If the forecast error RMSE as given in (6) in Section II is satisfied (e.g., less than 0.20 in this article), the PI will be determined, and the forecast process can be allowed into the next step. Otherwise, if the accuracy of the test forecast is not satisfied, which means the least-squares solution \hat{O}_0 can be singular, the above training process must be repeated by adjusting the weight matrix A , the bias matrix B , and the hidden layer number M (the details is Section III-D). This type of adjustment developed in this article is effective to promote the performance of the conventional OS-ELMs.

In the third step, the proposed OS-ELM-1 produces the future consecutive 48 spot prices with the determined PI corresponding to each trading interval of the upcoming trading day. The PI, PICP, and ACE can be obtained by using (11)–(12) in Section II, and PINC is set to 0.9.

In the fourth step, the proposed OS-ELM-1 activates the OS-ELM-2, which is explained in the next part, to provide a dynamic rolling 30-min-ahead prediction, in which a regular training update runs every two hours to update the OS-ELM-2.

In the final step, after collecting all the price data over 24 h, the new collected dataset forms 48 distinct price samples, $(x_1, t_1), (x_2, t_2), \dots, (x_{48}, t_{48})$, and the hidden layer output

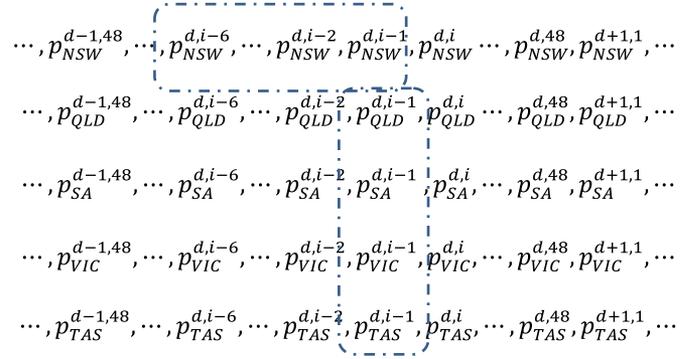


Fig. 3. Data structure of the unit sampling-pair in OS-ELM-2.

matrix H_1 is evaluated using the weight matrix A , and the bias matrix B obtained in the first step. Subsequently, the least-squares solution \hat{O}_0 obtained in the first step can be updated by using the method given in (7)–(10) in Section II (i.e., $\hat{O}_0 = \hat{O}_0 - K^{-1} \cdot H_1' \cdot (H_1 \cdot \hat{O}_0 - T_1)$, where $K = K_0 + H_1' \cdot H_1$). Then, the process returns to the third step.

Generally, using this type of updating, the result in the last step can be equal to the result obtained by using the conventional BL-ELMs calculated over the two merged hidden layer outputs (i.e., $H = \begin{bmatrix} H_0 \\ H_1 \end{bmatrix}$). However, the online-learning approach can reduce the computational cost.

C. OS-ELM-2 for the 30-min Prediction

For the 30-min-ahead prediction, the same price data cell as considered in Section III-B, e.g., $p_{NSW}^{d,i}$, ($1 \leq i \leq 48$), is also assumed to have a nonlinear relationship with several direct and indirect predecessors by sampling at not only the 30-min-by-30-min axis, such as $p_{NSW}^{d,i-1}, p_{NSW}^{d,i-2}, \dots, p_{NSW}^{d,i-6}$, but also using the recent spot prices data from the neighboring regions such as $P_{QLD}^{d,i-1}, P_{SA}^{d,i-1}, P_{VIC}^{d,i-1}, P_{TAS}^{d,i-1}$, as the dashed boxes, shown in Fig. 3.

Accordingly, the data pair (x_j, t_j) can be detailed as

$$\begin{cases} x_j = \left(p_{NSW}^{d,i-1}, \dots, p_{NSW}^{d,i-6}, P_{QLD}^{d,i-1}, \right. \\ \quad \left. P_{SA}^{d,i-1}, P_{VIC}^{d,i-1}, P_{TAS}^{d,i-1} \right)', \quad d \in Z^+ \\ t_j = p_{NSW}^{d,i} \end{cases} \quad (15)$$

where Z^+ represent positive integers. To reduce the computational cost, particularly in small case studies, the size of x_j as given in (15) can be simplified into

$$x_j = \left(p_{NSW}^{d,i-1}, \dots, p_{NSW}^{d,i-4}, P_{QLD}^{d,i-1}, P_{SA}^{d,i-1}, P_{VIC}^{d,i-1}, P_{TAS}^{d,i-1} \right)'. \quad (15.1)$$

The principle of OS-ELM-2 is shown in Fig. 4, which also consists of the similar five main steps as in OS-ELM-1:

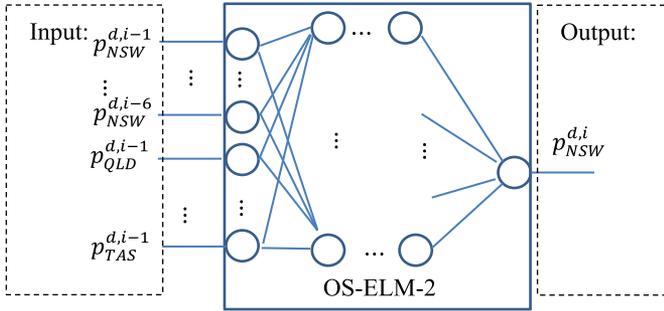


Fig. 4. Forecasting Principle of the OS-ELM-2.

Step 1: To obtain the corresponding weight matrix A , the bias matrix B , the optimal hidden layers M , and the least-squares solution \hat{O}_0 of OS-ELM-2, by training over the same historical dataset but now using the 30-min-by-30-min structure as defined in (15).

Step 2: To validate the results obtained in step 1, if the test accuracy of the forecast is greater than 10%, then step 1 is repeated, otherwise, the process goes to the next step.

Step 3: To forecast the point price for the next trading interval and to archive the new observed real price.

Step 4: To update the obtained least-squares solution \hat{O}_0 of OS-ELM-2 if the new archived data size has reached four trading intervals (i.e., all 30-min intervals over 2 h).

Step 5: To return to OS-ELM-1 if the total archived data size has reached 48, otherwise to return to step 3.

D. Main Framework of the Proposed Algorithm

Fig. 5 shows the main framework of the proposed hybrid OS-ELM. During the “Initialization” in Fig. 5, the learning performance can be affected by the weight initialization as well as by the size of the hidden layer, which can be decided by the calling function *Optimal_OS_ELM_Layer()*, as shown in Fig. 5. The detail of the definition of the function is given in Fig. 6.

E. Optimal Hidden Layers

Accordingly, the optimization of the size of the hidden layers plays an important key to promote the performance of the proposed OS-ELM modules. Given two datasets, $\{(x_j, t_j)\}_{j=1}^{N_0}$, $\{(x_k, t_k)\}_{k=1}^{N_1}$, as defined in (1) in Section II and a candidate domain $[LowB, UpperB]$, e.g., $[30, 1000]$, a fast search algorithm based on *Binary Search* [35] is developed in this section to refine the best size out of the given domain, as shown in Fig. 6(a).

Fig. 6(b) is an auxiliary function, which is used in Fig. 6(a), to access the quality of the forecast test when the size of the hidden layers is given.

IV. NUMERICAL STUDY

In this section, time-series data from the spot prices of NSW, QLD, SA, VIC, and TAS in January 2018, are collected from the Australian Energy Market Operator (AEMO) [36] to validate the proposed algorithm on its ability to forecast the day-ahead

and the real-time electricity price. Each of the datasets includes 1488 continuous-time slices (48 slices/day \times 31 days), where the time interval is 30 min between every two of the slices.

The activation functions in both OS-ELM modules are “hardlim” [32].

To validate the proposed algorithm, two case studies will be investigated, which are as follows.

- 1) *Case study 1:* to test the learning performance of the OS-ELM-1 on the day-by-day type.
- 2) *Case study 2:* to test the learning performance of the OS-ELM-2 on the 30-min-by-30-min type.

A. Case Study 1

For the OS-ELM-1, the raw time series of NSW can be organized into 768 data pairs (i.e., 16 days) by using (1) in Section II-A and (14) in Section III-B and it can be further divided into two parts: a training dataset, $\{(x_j, t_j)\}_{j=1}^{432}$, (for 9 days) and a forecasting-test dataset, $\{(x_k, t_k)\}_{k=1}^{336}$, (for one week).

Also, the optimal size of the hidden layers is set to 180, which is obtained by using the function *Optimal_OS_ELM_Layer()*, as shown in Fig. 5(a), where the input domain is $[LowB, UpperB] = [30, 1000]$.

After obtaining the training dataset $\{(x_j, t_j)\}_{j=1}^{432}$ the forecast performance of the OS-ELM-1 is investigated by using the first 144 data pairs in the forecasting-test dataset, $\{(x_k, t_k)\}_{k=1}^{336}$ of the OS-ELM-1.

First, the OS-ELM-1 is initialized by using the batch-processing over $\{(x_j, t_j)\}_{j=1}^{756}$ to simulate the setting of a conventional ELM, i.e., without forecasting the accuracy acceptable value, as shown in the second step in Section III-B, and without the online learning mechanism. Then, the obtained least-squares solutions \hat{O}_0 continues forecasting for three consecutive days without any update. The PI of electricity price, as given in Section II-C, is set with a high confidence level of 90%. The result is shown in Fig. 7.

The dashed line represents the real price series (i.e., the “Observation” in Fig. 7), the red line is the price forecast (i.e., the “Prediction” in Fig. 7), the blue line with dot marks is the upper boundary of PI, and the black line with plus marks is the lower boundary of PI.

As mentioned in Section III-B, because the weights of the hidden layers are generated randomly, it is hard to strictly guarantee that it will be an optimal nonsingular matrix for the first time. If without the RMSE threshold 0.2, the training quality would not be satisfied, as shown in Fig. 7.

The other case is processed on the same dataset, but the process strictly follows the five steps shown in Section III-B, and the results are shown in Fig. 8.

The comparison between Figs. 7 and 8 shows that both point prediction and PI prediction in Fig. 7 are not satisfied, particularly during the peak-price period each day. On the contrary, Fig. 8 shows that not only a better point prediction is obtained but also a more reliable PI can be determined.

Without loss of generality, to evaluate the corresponding interval forecast performance, which is given by (11) and (12) in

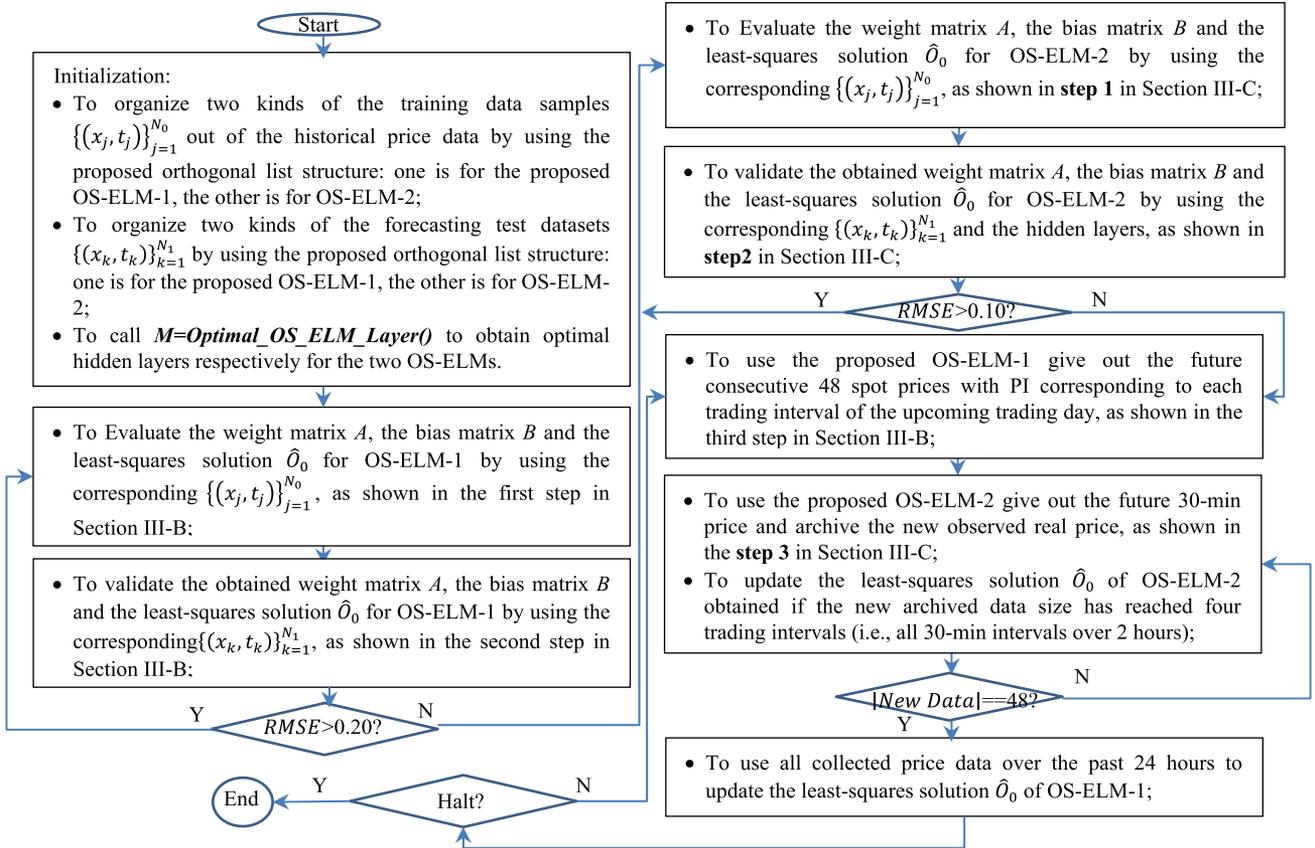


Fig. 5. Main framework of the proposed hybrid OS-ELM.

TABLE I
COMPARISON OF INTERVAL FORECAST RESULTS

Results	PINC (1- α)	Average PICP	Average ACE
Fig. 7	0.9	87.80	2.20
Fig. 8	0.9	90.18	0.18

Section II-C, the above simulations are processed 50 times independently, and the mean comparisons between the PI coverage probability and the average coverage error are listed in Table I.

According to the average PICP and the average ACE, the results for the interval forecasts shown in Fig. 8 are better than the results shown in Fig. 7.

Thus, the results in Fig. 8 and Table I show that the proposed OS-ELM-1 can achieve a better point prediction of the electricity price as well as a more reliable PI than those from the conventional ELM in the batch-learning process.

B. Case Study 2

For the OS-ELM-2, the same raw time series of NSW used in the Case study 1 is combined with the remaining four time-series, from QLD, VIC, SA, and TAS, and organized into 900 data pairs (i.e., about 18 days) by using (1) in Section II-A and (15.1) in Section III-C.

The 900 data pairs are divided into two parts: 1) a training dataset, $\{(x_j, t_j)\}_{j=1}^{756}$, (i.e., about two weeks), and 2) a forecasting-test dataset, $\{(x_k, t_k)\}_{k=1}^{144}$, (i.e., three days). The

optimal size of the hidden layers is 78 while the domain is [30, 200].

Fig. 9(a) shows that the first 756 time slice is used for training, and the remaining 144 data are treated as the forecasted price to test the proposed method.

In the first step, the OS-ELM-2 is initialized by using the batch-learning process over $\{(x_j, t_j)\}_{j=1}^{756}$, and the least-squares solutions \hat{O}_0 is obtained. Subsequently, two kinds of results are compared: 1) the trained OS-ELM-2 is used to forecast the consecutive 144-time slices without using the proposed online learning and updating actions. The results are shown in Fig. 9(b); 2) the trained OS-ELM-2 utilizes the online-learning algorithm as shown in Section III-C to update the least squares solutions \hat{O}_0 during forecasting. The results are shown in Fig. 10.

Fig. 9(a) and (b) shows that the error in the peak usage hours is relatively high and therefore this method is not satisfactory both in the training and the forecasting processes.

Fig. 10 gives the forecasted results on the same dataset as used in Fig. 9(b). The forecasting errors during the peak usage hours have decreased significantly as shown in the error curve in Fig. 10.

The comparison of the training accuracy or forecasting accuracy is given in Table II. The accuracy is measured based on the RMSE error, which represents the standard deviation of prediction errors. The results from the training stage using the trained data has the best accuracy, where the RMSE of the error

```

function  $M=Optimal\_OS\_ELM\_Layer()$ 
input: [LowB, UpperB],  $\{(x_j, t_j)\}_{j=1}^{N_0}, \{(x_k, t_k)\}_{k=1}^{N_1}$ 
output: Optimized hidden-layer size  $M$ 
1.  $Accu1 = ELMLayerTest(\{(x_j, t_j)\}_{j=1}^{N_0}, \{(x_k, t_k)\}_{k=1}^{N_1}, LowB)$ ;
2.  $Accu2 = ELMLayerTest(\{(x_j, t_j)\}_{j=1}^{N_0}, \{(x_k, t_k)\}_{k=1}^{N_1}, UpperB)$ ;
3. while(LowB < >round((LowB + UpperB)/2+0.1)
  || LowB < >round((LowB + UpperB)/2-0.1))
4. middleTemp = round((LowB + UpperB)/2);
5. if(Accu1 > Accu2)
6. LowB = middleTemp;
  Else
7. UpperB = middleTemp;
  endif
endwhile
8.  $M = LowB$ ;
End func

```

(a)

```

function Forecast_Accuracy =  $ELMLayerTest()$ 
input: trainSet, testSet, NumberHiddenLayer
output: RMSE of the test forecast
1. To generate weight matrix  $A$ , the bias matrix  $B$  based on the input
  size of the hidden layers, i.e., NumberHiddenLayer;
2. To calculate the least-squares solution  $\hat{O} = \hat{H}^+ \cdot T$  by using (3)-
  (4) given in Section II;
3. To calculate RMSE by using (6) as given in Section II;
4. To return Forecast_Accuracy = RMSE.
End func

```

(b)

Fig. 6. (a) Function to search an optimal size from a given domain (b) Function to access the quality of the forecast test when the size of the hidden layers is given.

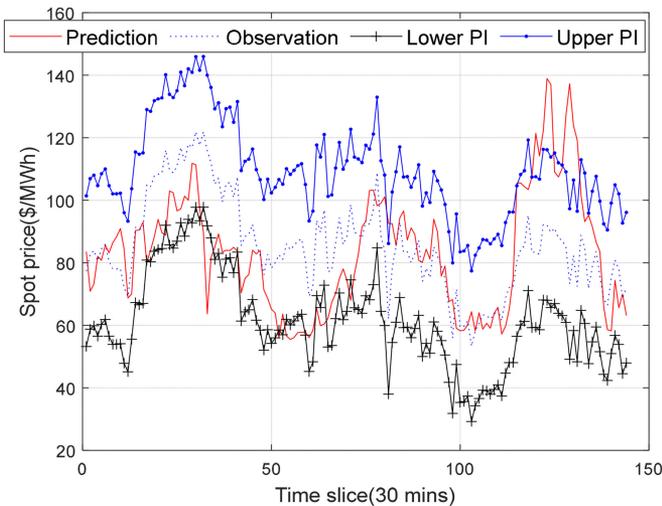


Fig. 7. 3-day forecasting performance of OS-ELM-1 without online learning.

TABLE II
SUMMARY OF FORECASTING SIMULATION ON 900 CONTINUOUS TIME SLICES OF THE NSW SPOT PRICE

Stage	Accuracy (RMSE)	Time Cost
Training	0.048873891599761	0.015625000000000
Forecasting	0.145511944617120	0
OL-Forecasting	0.068676891694583	0
P- Forecasting	82.605894485541342	0

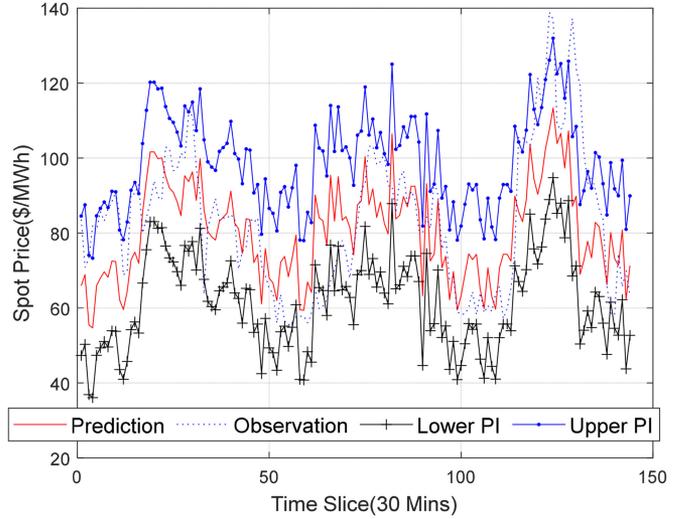


Fig. 8. 3-day forecasting performance of OS-ELM-1 with online learning.

is less than 5%; the results from the forecasting stage using the new data without the online learning (OL) process has RMSE of the error of about 14.6%; and the results from the forecasting stage using the proposed online-learning and updating processes has increased the forecasting quality significantly, for which the RMSE of the error is about 6.8%.

Furthermore, Table II shows that while batch-learning has an unignorable time cost, the online-learning only needs to update the trained results and therefore has a negligible time cost, 0.015625 s, as shown in Table II. Thus, the online-learning can be used to improve the real-time forecasting quality once the offline trained pattern has been obtained.

Besides that, to better understand the forecast performance, there is a forecast reference appended in the last row of Table II, which is based on the $t + 1$ persistence forecast, one of the simplest methods for forecasting the future trend of a time series, denoted as ‘‘P-Forecasting.’’ The detail is given in Section VI - Appendix.

Comparing to the corresponding RMSE, the batch-learning ELM has better performance, and the proposed online-learning is the best.

C. Summary

By using the proposed orthogonal list, the day-by-day forecast in Case 1 has been transferred into 48 spot forecasts in parallel, and the corresponding interval forecast such as PI is evaluated to quantile the prediction uncertainties for the day-ahead prediction; when it comes to the 30-min forecast (i.e., almost real-time), the global price information from the other grid-connected regions is considered to enhance the forecasting accuracy against uncertain factors for the 30-min rolling prediction. Such kind of design is effective to utilize the quantile forecast, spot forecast, and online learning into the practical electricity price forecast.

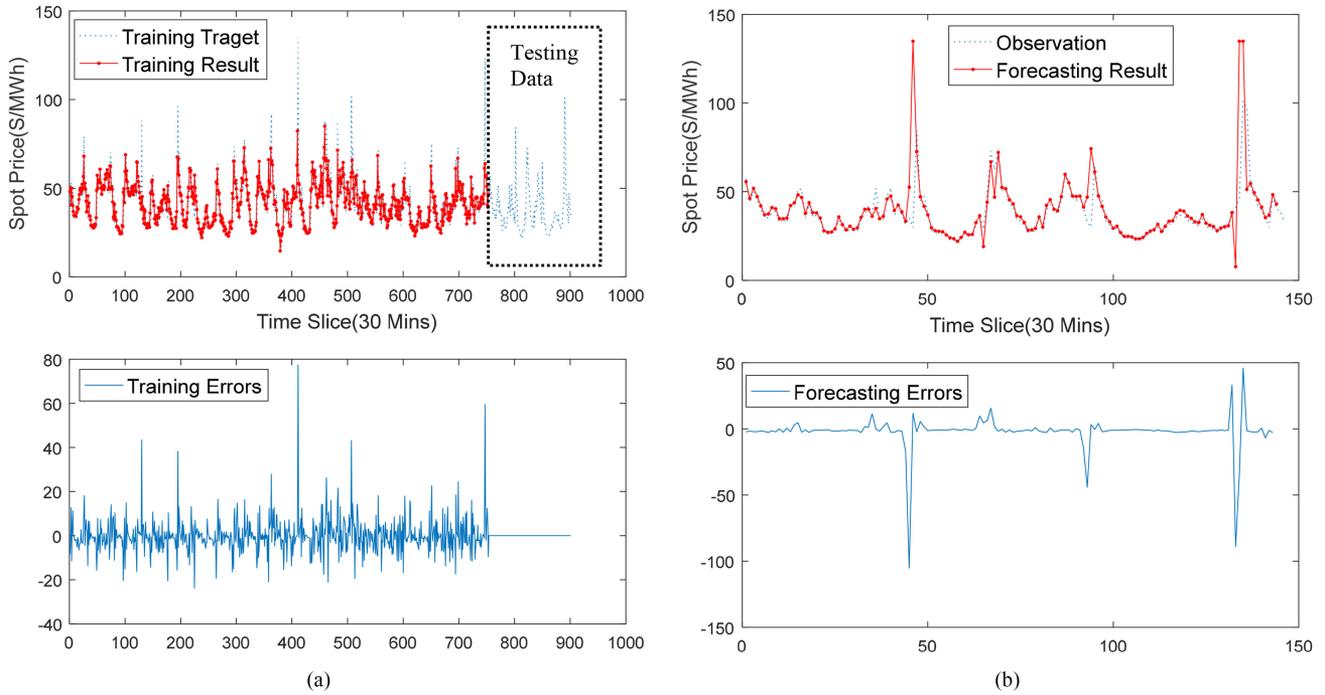


Fig. 9. OS-ELM-2 in training and forecasting (a) the training result of OS-ELM-2 and (b) the forecasting result without online sequential learning.

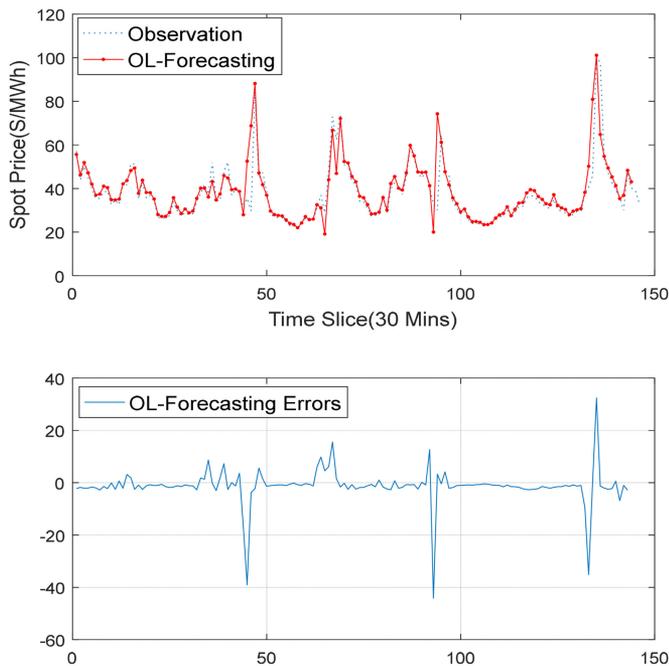


Fig. 10. OS-ELM-2 in forecasting with online-learning and updating.

All simulations in this article are processed using a computer system with Intel (R) Core (TM) i7-5500U (CPU@2.4GHz) and 16 GB RAM, using the MATLAB 2016a environment.

V. CONCLUSION

This article proposes a novel online-learning ELM algorithm to improve the conventional predispatch electricity price prediction. In the proposed algorithm, a novel data structure

using a 2-D orthogonal list is developed to help recognize the nonlinear information from two kinds of data series, i.e., the daily and the 2-hourly data series. Accordingly, the proposed algorithm consists of two OS-ELM modules that can update the two training patterns while processing the price prediction: OS-ELM-1 and OS-ELM-2. The first module provides a total day-ahead prediction with the determined PI for the next upcoming trading day; the other provides a dynamic rolling 30-min-ahead update. To increase the perception of the potential upcoming fluctuations on the local market, the recent price information from the neighboring grids is also considered in the OS-ELM-2. The article also proposes a fast search algorithm to optimize the number of hidden layers in the initialization. The results from an extensive simulation study show that the proposed method can cope with unexpected dynamic price changes better than the traditional batch-learning. Further, the impact of the unexpected price changes in the neighboring states in the studied ANEM can also be considered.

APPENDIX

Persistence forecast [37] considers that future values of the time series can be calculated on the assumption that the conditions remain unchanged between “current” time t and future time $t + \Delta t$. A straightforward implementation of the persistence model is simply as follows:

$$\hat{y}_{t+\Delta t} = y_t.$$

Also, the corresponding RMSE, i.e., root mean square error, can be calculated as follows:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{t=1}^N [\hat{y}_t - y_t]^2}$$

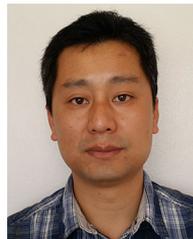
where \hat{y}_t and y_t are the forecast and the real value at time point t , respectively; N is the size of the samples.

ACKNOWLEDGMENT

The authors would like to thank Australia NEM-Watch software, which provides great help to this work, and Dr. Ahmed Saber, who is the Manager of Smart Power System Optimization Dept. (T&D) at ETAP, for his useful comments.

REFERENCES

- [1] K. Hubicka, G. Marcjasz, and R. Weron, "A note on averaging day-ahead electricity price forecasts across calibration windows," *IEEE Trans. Sustain. Energy*, vol. 10, no. 1, pp. 321–323, Jan. 2019.
- [2] H. Chitsaz, P. Zamani-Dehkordi, H. Zareipour, and P. P. Parikh, "Electricity price forecasting for operational scheduling of behind-the-meter storage systems," *IEEE Trans. Smart Grid*, vol. 9, no. 6, pp. 6612–6622, Nov. 2018.
- [3] C. González, J. Mira-McWilliams, and I. Juárez, "Important variable assessment and electricity price forecasting based on regression tree models: Classification and regression trees, bagging and random forests," *IET Gener., Transmiss. Distrib.*, vol. 9, no. 11, pp. 1120–1128, 2015.
- [4] K. Meng, Z. Y. Dong, and K. P. Wong, "Self-adaptive radial basis function neural network for short-term electricity price forecasting," *IET Gener., Transmiss. Distrib.*, vol. 3, no. 4, pp. 325–335, Apr. 2009.
- [5] G. A. Susto, A. Schirru, S. Pampuri, and S. McLoone, "Supervised aggregative feature extraction for big data time series regression," *IEEE Trans. Ind. Inform.*, vol. 12, no. 3, pp. 1243–1252, Jun. 2016.
- [6] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: MIT press, 2012.
- [7] M. Alamaniotis, D. Bargiotas, N. G. Bourbakis, and L. H. Tsoukalas, "Genetic optimal regression of relevance vector machines for electricity pricing signal forecasting in smart grids," *IEEE Trans. Smart Grid*, vol. 6, no. 6, pp. 2997–3005, Nov. 2015.
- [8] C. Xiao, Z. Dong, Y. Xu, K. Meng, X. Zhou, and X. Zhang, "Rational and self-adaptive evolutionary extreme learning machine for electricity price forecast," *Memetic Comput.*, vol. 8, no. 3, pp. 223–233, 2016.
- [9] C. Wan, M. Niu, Y. Song, and Z. Xu, "Pareto optimal prediction intervals of electricity price," *IEEE Trans. Power Syst.*, vol. 32, no. 1, pp. 817–819, Jan. 2017.
- [10] C. Wan, Z. Xu, Y. Wang, Z. Y. Dong, and K. P. Wong, "A hybrid approach for probabilistic forecasting of electricity price," *IEEE Trans. Smart Grid*, vol. 5, no. 1, pp. 463–470, Jan. 2014.
- [11] H. Zareipour, A. Janjani, H. Leung, A. Motamedi, and A. Schellenberg, "Classification of future electricity market prices," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 165–173, Feb. 2011.
- [12] F. Valencia, J. Collado, D. Sáez, and L. G. Marín, "Robust energy management system for a microgrid based on a fuzzy prediction interval model," *IEEE Trans. Smart Grid*, vol. 7, no. 3, pp. 1486–1494, May 2016.
- [13] J. P. González, A. M. S. Muñoz San Roque, and E. A. Pérez, "Forecasting functional time series with a new hilbertian ARMAX model: Application to electricity price forecasting," *IEEE Trans. Power Syst.*, vol. 33, no. 1, pp. 545–556, Jan. 2018.
- [14] D. Ryabko, "Time-series information and unsupervised learning of representations," *IEEE Trans. Inf. Theory*, vol. 66, no. 3, pp. 1702–1713, Mar. 2020.
- [15] X. Chen, Z. Y. Dong, K. Meng, Y. Xu, K. P. Wong, and H. W. Ngan, "Electricity price forecasting with extreme learning machine and bootstrapping," *IEEE Trans. Power Syst.*, vol. 27, no. 4, pp. 2055–2062, Nov. 2012.
- [16] M. Rafiei, T. Niknam, and M. Khooban, "Probabilistic forecasting of hourly electricity price by generalization of ELM for usage in improved wavelet neural network," *IEEE Trans. Ind. Inform.*, vol. 13, no. 1, pp. 71–79, Feb. 2017.
- [17] K. L. López, C. Gagné, and M. Gardner, "Demand-Side management using deep learning for smart charging of electric vehicles," *IEEE Trans. Smart Grid*, vol. 10, no. 3, pp. 2683–2691, May 2019.
- [18] S. Marsland, *Machine Learning: An algorithmic Perspective*. London, U.K.: Chapman and Hall, 2011.
- [19] A. Swaminathan and T. Joachims, "Batch learning from logged bandit feedback through counterfactual risk minimization," *J. Mach. Learn. Res.*, vol. 16, no. 1, pp. 1731–1755, 2015.
- [20] G. Hinton, N. Srivastava, and K. Swersky, "Neural networks for machine learning lecture 6a overview of mini-batch gradient descent," *Cited*, vol. 14, no. 8, pp. 8–31, 2012.
- [21] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: A new learning scheme of feedforward neural networks," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2004, vol. 2, pp. 985–990.
- [22] W. F. Schmidt, M. A. Kraaijveld, and R. P. Duin, "Feed forward neural networks with random weights," in *Proc. Int. Conf. Pattern Recognit.*, 1992, pp. 1–4.
- [23] Y.-H. Pao, G.-H. Park, and D. J. Sobajic, "Learning and generalization characteristics of the random vector functional-link net," *Neurocomputing*, vol. 6, no. 2, pp. 163–180, 1994.
- [24] A. J. Wood, B. F. Wollenberg, and G. B. Sheblé, *Power Generation, Operation, and Control*. Hoboken, NJ, USA: Wiley, 2013.
- [25] D. S. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks," *Complex Syst.*, vol. 2, pp. 321–355, 1988.
- [26] Z. Ding, W. Lee, and J. Wang, "Stochastic resource planning strategy to improve the efficiency of microgrid operation," *IEEE Trans. Ind. Appl.*, vol. 51, no. 3, pp. 1978–1986, May/Jun. 2015.
- [27] K. Rahimi, S. Omran, M. Dilek, and R. Broadwater, "Computation of voltage flicker with cloud motion simulator," *IEEE Trans. Ind. Appl.*, vol. 54, no. 3, pp. 2628–2636, May/Jun. 2018.
- [28] A. A. Almezhia, H. M. K. Al-Masri, and M. Ehsani, "Integration of renewable energy sources by load shifting and utilizing value storage," *IEEE Trans. Smart Grid*, vol. 10, no. 5, pp. 4974–4984, Sep. 2019.
- [29] A. Moreira, D. Pozo, A. Street, and E. Sauma, "Reliable renewable generation and transmission expansion planning: Co-optimizing system's resources for meeting renewable targets," *IEEE Trans. Power Syst.*, vol. 32, no. 4, pp. 3246–3257, Jul. 2017.
- [30] J. A. Aguado, V. H. Quintana, M. Madrigal, and W. D. Rosehart, "Coordinated spot market for congestion management of inter-regional electricity markets," *IEEE Trans. Power Syst.*, vol. 19, no. 1, pp. 180–187, Feb. 2004.
- [31] C. Xiao, D. Sutanto, K. M. Muttaqi, and M. Zhang, "Online sequential extreme learning machine algorithm for better prediction of the Real-time electricity price under dynamic environmental changes," in *Proc. IEEE Ind. Appl. Soc. Annu. Meeting*, Baltimore, MD, USA, 2019, pp. 1–6.
- [32] N.-Y. Liang, G.-B. Huang, P. Saratchandran, and N. Sundararajan, "A fast and accurate online sequential learning algorithm for feedforward neural networks," *IEEE Trans. Neural Netw.*, vol. 17, no. 6, pp. 1411–1423, Nov. 2006.
- [33] J. Lim, S. Lee, and H.-S. Pang, "Low complexity adaptive forgetting factor for online sequential extreme learning machine (OS-ELM) for application to nonstationary system estimations," *Neural Comput. Appl.*, vol. 22, no. 3/4, pp. 569–576, 2013.
- [34] H. Zhang, S. Zhang, and Y. Yin, "Online sequential ELM algorithm with forgetting factor for real applications," *Neurocomputing*, vol. 261, pp. 144–152, 2017.
- [35] M. T. Goodrich and R. Tamassia, *Algorithm Design: Foundation, Analysis and Internet Examples*. Hoboken, NJ, USA: Wiley, 2006.
- [36] Australian Energy Market Operator (AEMO). Accessed: Jan. 20, 2021. [Online]. Available: www.aemo.com.au
- [37] C. F. Coimbra, H. T. Pedro, and J. Kleissl, "Stochastic learning methods," in *Solar Energy Forecasting and Resource Assessment*. Amsterdam, The Netherlands: Elsevier, 2013, pp. 383–406.



Chixin Xiao (Member, IEEE) received the master's degree in computer science from Xiangtan University, Xiangtan, China, in 2004, and the first Ph.D. degree in computer application technology from Central South University, Changsha, China, in 2009. He is currently working toward the second Ph.D. degree in power engineering from the University of Wollongong, Wollongong, NSW, Australia.

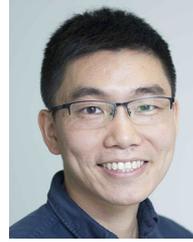
He was an Associate Professor of information science with Xiangtan University, and previously held academic positions with the University of Essex, U.K., and the University of Newcastle, Australia. His research interests include smart grid resilience, power system planning and modeling, electricity market, and computational intelligence and its application in power engineering.



Danny Sutanto (Senior Member, IEEE) received the B.Eng. (Hons.) and Ph.D. degrees in electrical engineering from the University of Western Australia, Perth, Australia, in 1978 and 1981, respectively.

He is currently a Professor of power engineering with the University of Wollongong, Wollongong, NSW, Australia. His research interests include power system planning, power system emergency, analysis and harmonics, flexible alternating current transmission system, and battery energy storage systems. He was the IEEE Industry Applications Society Area

Chair for Region 10 (Asia Pacific) from 2014 to 2017.



Ke Meng (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the University of Queensland, Brisbane, QLD, Australia, in 2009.

He is currently a Senior Lecturer with the School of Electrical Engineering and Telecommunications, The University of New South Wales, Sydney, NSW, Australia. His research interests include electric power system modeling, stability analysis, renewable energy systems, and grid integration.



Kashem M. Muttaqi (Senior Member, IEEE) received the B.Sc. degree in electrical and electronic engineering from Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, in 1993, the M.Eng.Sc. degree in electrical engineering from University of Malaya, Kuala Lumpur, Malaysia, in 1996, and the Ph.D. degree in electrical engineering from Multimedia University, Selangor, Malaysia, in 2001.

He is currently a Professor with the School of Electrical, Computer, and Telecommunications Engineering, University of Wollongong, Wollongong, NSW, Australia. He was associated with the University of Tasmania, Hobart, Australia, as a Research Fellow/Lecturer/Senior Lecturer from 2002 to 2007, and with the Queensland University of Technology, Brisbane, Australia, as a Research Fellow from 2000 to 2002. Previously, he also worked for Multimedia University as a Lecturer for three years. He has more than 22 years of academic experience and authored or coauthored 385 papers in international journals and conference proceedings. His research interests include distributed generation, renewable energy, electrical vehicles, smart-grid, and power system planning and control.



Zhao Yang Dong (Fellow, IEEE) received Ph.D. degree in electrical engineering for the University of Sydney, NSW 2006, Australia in 1999.

He is currently a Professor of Energy Systems with the School of Electrical Engineering and Telecommunications, the University of New South Wales (UNSW), Sydney, Australia. He is also the Director of UNSW Digital Grid Futures Institute. His previous role include Ausgrid Chair and Director of Ausgrid Centre for Intelligent Electricity Networks providing R&D support for the Smart Grid, Smart City national demonstration project. His research interest includes power system planning and stability, smart grid/microgrid, renewable energy grid integration, load modelling, electricity market, numerical methods and their applications for power system analysis.



Minjie Zhang (Senior Member, IEEE) received the B.Sc. degree in computer science from Fudan University, Shanghai, China, in 1982 and the Ph.D. degree in computer science from The University of New England, Armidale, NSW, Australia, in 1996.

She is currently a Professor of computer science and the Director of Centre for Big Data Analytics and Intelligent Systems, University of Wollongong, Wollongong, NSW, Australia. She is an active researcher and has published more than 200 research papers.

Her research interests include multiagent systems, agent-based modeling and simulation in complex domains, smart grid systems, and data mining, and knowledge discovery in open environments.