Short-Term Electricity Price Forecasting With Stacked Denoising Autoencoders

Long Wang, *Student Member, IEEE*, Zijun Zhang, *Member, IEEE*, and Jieqiu Chen

***Abstract*—A short-term forecasting of the electricity price with data-driven algorithms is studied in this research. A stacked denoising autoencoder (SDA) model, a class of deep neural networks, and its extended version are utilized to forecast the electricity price hourly. Data collected in Nebraska, Arkansas, Louisiana, Texas, and Indiana hubs in U.S. are utilized. Two types of forecasting, the online hourly forecasting and day-ahead hourly forecasting, are examined. In online forecasting, SDA models are compared with data-driven approaches including the classical neural networks, support vector machine, multivariate adaptive regression splines, and least absolute shrinkage and selection operator. In the day-ahead forecasting, the effectiveness of SDA models is further validated through comparing with industrial results and a recently reported method. Computational results demonstrate that SDA models are capable to accurately forecast electricity prices and the extended SDA model further improves the forecasting performance.**

***Index Terms*—Comparative analysis, data mining, electricity price, hourly forecasting, neural networks.**

I. Introduction

AN INCREASING effort of deregulating electricity markets to form a more reliable, efficient, and cost-effective system by enhancing competitions has been witnessed in world's main economies [1], [2]. In the liberalized markets, the electricity is commoditized and hence its price is dynamic. Due to the price variation, pricing electricity appropriately becomes crucial to generate profits, schedule power productions, and plan load responses [3]–[7]. The accurate electricity price forecasting is helpful to determine the electricity price and thus is valuable. As liberalized power markets include two types, day-ahead and real-time [8], it is meaningful to discuss both of the day-ahead and online forecasting of the electricity price.

The electricity price forecasting has been vigorously studied in the literature. From the application side, the forecasting of the electricity price in different deregulated markets of main economies around the world has been reported [9]–[15].

From the methodological side, two classes of methods, statistical and data-driven methods, have been applied to study various electricity price forecasting. Times-series models including the auto-regressive (AR) model [16], [17], the AR models incorporating moving averages and exogenous variables [18]–[21], as well as the generalized auto-regressive conditional heteroskedastic (GARCH) model [4] were widely applied statistical methods in the electricity price forecasting. More sophisticated time-series approaches were also studied. To better predict spikes of the electricity price, regime switch concepts were integrated into forecasting models [22]. Semi-parametric models were built with nonparametric kernel density estimators to achieve better forecasting results [23]. In [24], time-series of the electricity price was decomposed by wavelet transformations and each sub-series is predicted by the ARIMA method. Quantile regression with generalized additive models (quantGAM) [25] and least absolute shrinkage and selection operator (Lasso) [26] were reported in recent studies of the electricity forecasting. Besides forecasting point values, probabilistic forecasting was also investigated in recent Global Energy Forecasting Competition 2014 (GEFCom2014) [27]. The majority of the classical time-series models aimed at exploring the predictability of the electricity price through examining the linear relationships among input parameters. Recent studies [25], [26] considered advanced statistical models capable to capture nonlinearities in data.

Data-driven methods were applied to enhance the forecasting accuracy of the electricity prices through studying the nonlinear relationships among input parameters. Applications of SVM models in the electricity price forecasting [28]–[30] were presented. Meanwhile, the capability of NN methods in forecasting the electricity price was studied in [10], [31]. The weighted k nearest neighbor method was also assessed in forecasting the electricity price [32]. A study [8] reviewed various electricity price forecasting methods and reported that data-driven methods, typically NN-based methods, offered decent results.

In the literature, NN-based electricity price forecasting studies mostly employed shallow networks which only include one hidden layer. Since a limited number of hidden units are usually considered, the shallow network structure might limit their capabilities of exploring higher nonlinearities. DNN fits multiple layers of nonlinear transformations [33] into data and its capability of learning more complicated relationships among

Manuscript received January 3, 2016; revised May 18, 2016, August 18, 2016, and October 12, 2016; accepted November 12, 2016. Date of publication November 15, 2016; date of current version June 16, 2017. This work was supported in part by the Early Career Scheme from the Research Grants Council of the Hong Kong Special Administrative Region under Project CityU 138313 and in part by the CityU Strategic Research Grant under Project 7004551. Paper no. TPWRS-00010-2016. *(Corresponding author: Zijun Zhang.)*

L. Wang and Z. Zhang are with the Department of Systems Engineering and Engineering Management, City University of Hong Kong, Kowloon, Hong Kong (e-mail: long.wang@my.cityu.edu.hk; zijzhang@cityu.edu.hk).

J. Chen is with Microsoft Corp., Redmond, WA 98052-6399 USA (e-mail: jieqiu0808@gmail.com).

This paper has supplementary materials available at http://ieeexplore.ieee.org. The file contains details of the Boosting Tree algorithm considered in assessing the importance of parameters for predicting the electricity price, classical data-driven methods considered in developing electricity price prediction models, and a computational experiment of the RS-SDA model. The total size of the file is 271 KB.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TPWRS.2016.2628873

features is stronger. As the electricity price forecasting may involve high-dimensional inputs, two advantages of DNNs can assist the defense of the curse of dimensionality. Firstly, DNN of a linear number of hidden units distributed into multiple hidden layers is capable to express an exponential number of patterns via appropriate configurations of activations [34]. Next, the deep architecture offers an opportunity of better organizing hidden units so that DNNs could efficiently model complicated functions.

This research studies the application of SDA models, a class of DNN, in forecasting electricity prices. A SDA with an unsupervised pre-training process [35] is firstly employed to develop the forecasting model. The unsupervised pre-training is effective to prevent the over-fitting in developing a DNN model [33]. Next, an extended SDA, RS-SDA, which incorporates concepts of the random sample consensus (RANSAC) [36] and stochastic neighbor embedding (SNE) [37] is developed to further improve the forecasting performance. Contributions of this study include: 1) A pioneer study of applying SDAs into the on-line and day-ahead hourly forecasting of electricity prices is presented; 2) An extended SDA model, RS-SDA, is proposed. In RS-SDA, the SNE is utilized to automatically determine the number of hidden units in hidden layers and the RANSAC is employed to automatically filter outliers from the training data; 3) The mini-batch gradient descent [38] and the momentum [39] are integrated to develop an improved fine-tune process of SDA models; 4) An online procedure of updating the training dataset of SDA models is introduced.

A comprehensive comparative analysis of data-driven methods in forecasting the electricity price is conducted in this paper. SDA and RS-SDA models are benchmarked against the classical data-driven methods, industrial results produced by e-ISOForecast [40], and a recently reported approach [26].

The main body of this research is structured as follows. In Section II, a detailed description of data and input parameters for forecasting is provided. The development of the SDA and RS-SDA models is illustrated in Section III. Section IV presents the forecasting results and comparative analyses of the SDA and RS-SDA models and other benchmarking models. Section V concludes findings in this research.

II. Data Description and Processing

Data of the electricity price, observed load, and forecasted load of five hubs of the Midcontinent Independent System Operator Inc. (MISO) in U.S. including the Nebraska Public Power District (NPPD), Arkansas, Louisiana, Texas, and Indiana are publicly accessible [41]. Data of first four hubs are applied into the on-line hourly forecasting while, in the day-ahead hourly forecasting, the data of Indiana hub is utilized. The length of the dataset covers a period from January 2012 to November 2014.

A set of parameters listed in Table I frequently considered by traders in forecasting the electricity price is suggested by the industrial partner. Data of GP_t^f and WG_t^f associated with the considered load and electricity price data are supplied by an industrial partner. In the online forecasting, parameters listed in Table I are directly applied and the importance analysis based on the Boosting Tree (BT) algorithm [42] is performed to verify that all considered parameters are relevant to the electricity price. Details of BT are provided in Supplementary Materials.

TABLE I
Parameters and Their Importance

Symbol	Description	Units	Importance
P_{t-24}	LMP price one day ago	\$/MWh	100
P_{t-48}	LMP price two days ago	\$/MWh	65
$\hat{L}_t$	Load at t forecasted 1 hour ago	MW	37
$\hat{L}_t^{144}$	Load at t forecasted 6 days ago	MW	24
$\hat{L}_t^{120}$	Load at t forecasted 5 days ago	MW	24
$\hat{L}_t^{96}$	Load at t forecasted 4 days ago	MW	23
$\hat{L}_t^{72}$	Load at t forecasted 3 days ago	MW	23
$\hat{L}_t^{48}$	Load at t forecasted 2 days ago	MW	20
$\hat{L}_t^{24}$	Load at t forecasted 1 day ago	MW	41
WG_t^f	Forecasted wind power generation of the node	MW	17
GP_t^f	Forecasted weighted natural gas price index	\$/mmBtu	13

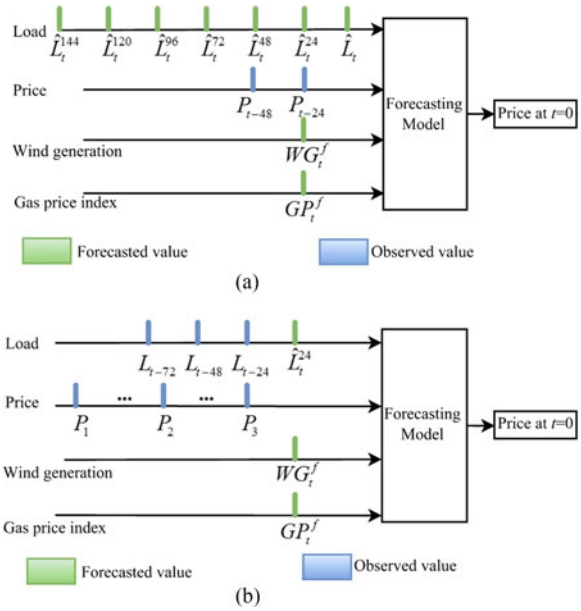


Fig. 1. The framework of the on-line and day-ahead hourly forecasting.

In the day-ahead forecasting, since values of $\hat{L}_t^{48}$, $\hat{L}_t^{72}$, $\hat{L}_t^{96}$, $\hat{L}_t^{120}$, and $\hat{L}_t^{144}$ in the Indiana hub are not available, derived parameters are considered. Three parameters, P_1, P_2, and P_3, are derived by examining the affinity between the loads and natural gas prices because we discovered that historical and forecasted electricity prices are close with a high probability if such affinity is high after analyzing data. The affinity is described by the Euclidean distance defined in (1).

$$\|D\| = \sqrt{(L_h - \hat{L}_t^{24})^2 + (GP_h - GP_t^f)^2} \tag{1}$$

In (1), L_h is the historical actual load, and GP_h is the weighted natural gas price index. Data of two months before t are utilized to extract P_1, P_2, and P_3 which result in three lowest Euclidean distances. Other input parameters in the day-ahead hourly forecasting include $\hat{L}_t^{24}$, GP_t^f, WG_t^f, L_{t-24} (the actual load observed at $t-24$ hrs), L_{t-48} (the actual load observed at $t-48$ hrs), and L_{t-72} (the actual load observed at $t-72$ hrs).

Fig. 1 illustrates the framework of the on-line and day-ahead hourly forecasting. The electricity price in 1-hr is forecasted in the on-line hourly forecasting. In day-ahead hourly forecasting, the model offers the forecast of the hourly electricity price of

the next day. Data-driven models of two types of forecasting are depicted in (2) and (3) separately.

$$\hat{P}_t = f_{on}(P_{t-24}, P_{t-48}, \hat{L}_t, \hat{L}_t^{144}, \hat{L}_t^{120}, \hat{L}_t^{96}, \hat{L}_t^{72}, \hat{L}_t^{48}, \hat{L}_t^{24}, WG_t^f, GP_t^f) \tag{2}$$

$$\hat{P}_t = f_{day}(P_1, P_2, P_3, L_{t-24}, L_{t-48}, L_{t-72}, \hat{L}_t^{24}, WG_t^f, GP_t^f) \tag{3}$$

where $\hat{P}_t$ is the electricity price forecasted at t, $f_{\text{on}}(\cdot)$ is the on-line hourly forecasting model, and $f_{\text{day}}(\cdot)$ is the day-ahead hourly forecasting model.

III. Forecasting Models

In this section, the SDA and RS-SDA for conducting the on-line and day-ahead hourly forecasting of the electricity price are described. To better initialize parameters of SDAs, an unsupervised technique for pre-training SDAs is introduced. Details of benchmarking models, the classical NN [43], SVM [44], MARS [45], and Lasso [26], [46], considered in the comparative analysis are depicted in supplementary materials. Metrics for evaluating the forecasting performance are developed.

A. Stacked Denoising Autoencoders

The SDA and RS-SDA are applied to forecast electricity prices in this work. Classical NNs have been widely considered in modelling nonlinearities in data. However, the three-layer architecture, an input layer, a hidden layer, and an output layer, constrains the modeling capability of a classical NN. A previous study [34] proved that a classical NN desired an exponential number of hidden units to approximate considered functions which could be efficiently modelled by DNNs with far less hidden units well organized in multiple hidden layers. Due to the "diminishing error problem" [47], it is challenging to directly train high quality DNNs by simply applying a standard backpropagation algorithm. More powerful training algorithms were proposed to successfully tackle such challenge [47], [48]. The pre-training of DNNs with DAs [49], Contractive Autoencoders [50], and mean-covariance restricted Boltzmann machines (mcRBMs) [51] were also discussed.

To overcome the over-fitting and have a better generalization [52], the greedy layer-wise unsupervised training is applied to initialize parameters of the SDA and RS-SDA in this research. The training process includes two phases, pre-training and fine-tuning. In pre-training, a network of autoencoders excluding the output layer is iteratively trained based on input parameters. Following autoencoders reconstruct the encoding based on the outputs of hidden layer of previous autoencoders. Once all layers have been pre-trained, the fine-tuning phase is conducted by incorporating the output layer. The complete network is further trained to minimize the square errors based on the actual and forecasted prices. Compared with SDA originally designed for classification, the inputs and output of the SDA and RS-SDA in this study directly take real values.

1) Autoencoders: Autoencoder is a type of neural networks which reconstruct inputs through encoding and decoding original inputs. As presented in (4), an autoencoder encodes an input $\mathbf{x}$ to a hidden representation $\mathbf{y}$ through its hidden layer activation function [53]

$$\mathbf{y} = s(\mathbf{W}\mathbf{x} + \mathbf{b}) \tag{4}$$

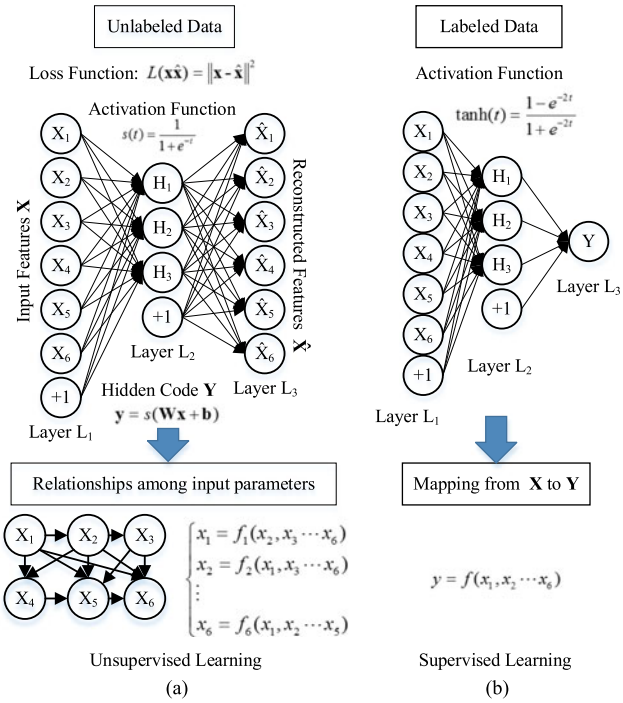


Fig. 2. The schematic diagram of an autoencoder and classical NN.

where s is a sigmoid function, $\mathbf{x}$ is the input, $\mathbf{W}$ is the weights and $\mathbf{b}$ is the bias. The code $\mathbf{y}$ is then mapped back to reconstructed features $\hat{\mathbf{x}}$. The mapping is performed through a similar way shown in (5)

$$\hat{\mathbf{x}} = s(\mathbf{W}'\mathbf{y} + \mathbf{b}') \tag{5}$$

where $\mathbf{y}$ is the hidden code, $\mathbf{W}'$ is the reconstruction weights and $\mathbf{b}'$ is the reconstruction bias.

Fig. 2(a) illustrates the structure of an autoencoder with 6 input units as an example. There is a bottleneck layer between the input and output layer. Outputs of hidden units are transformed to the code of inputs. Fig. 2(b) demonstrates a classical NN structure. When an autoencoder is trained, labels are not given and regularities among data are learnt. Thus, training autoencoders is an unsupervised learning task aiming to learn the representation of original data. However, training the classical NN is a supervised learning and labels are provided to minimize the fitting error of predicting labels.

The reconstruction error is measured by the squared error in (6).

$$\mathcal{L}(\mathbf{x}, \hat{\mathbf{x}}) = \|\mathbf{x} - \hat{\mathbf{x}}\|^2 \tag{6}$$

The training process of an autoencoder aims at minimizing the summation of squared errors over all training samples. The back-propagation algorithm can be directly applied to train the autoencoder and it has been proved that the autoencoder has the ability to capture multi-modal aspects of the input distribution [54].

2) Denoising Autoencoders: Autoencoders might encounter noise corruptions. The DA is built to recover corrupted inputs through the reconstruction. This allows the autoencoder to learn more robust representation of the original input rather than the simple identity [35]. A DA offers two functions, encoding the input and eliminating the effect of corruptions. In many cases, empirical evidence confirms that deploying DAs in the

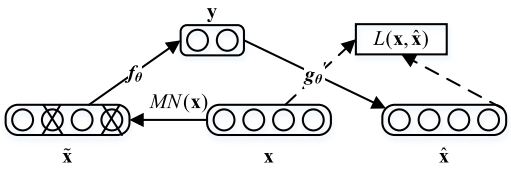


Fig. 3. Denoising autoencoder structure.

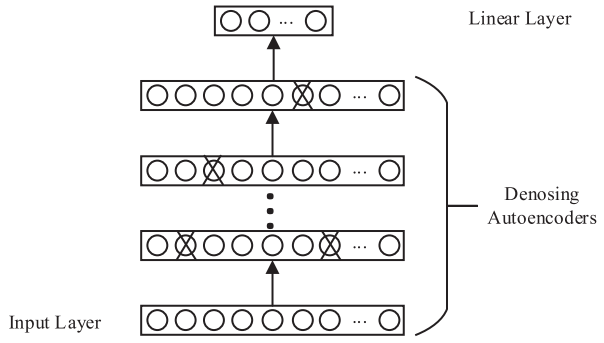


Fig. 4. Stacked denoising autoencoders.

pre-training of the deep neural network offers better performance than RBMs [35].

To transform the autoencoder into a DA, a stochastic corruption needs to be incorporated into the raw input. This can be achieved through masking entries of the input by making them zeros as shown in (7).

$$\tilde{\mathbf{x}} = MN(\mathbf{x}) \tag{7}$$

where $\tilde{\mathbf{x}}$ is the corrupted version of $\mathbf{x}$ and $MN(\cdot)$ makes randomly chosen entries of $\mathbf{x}$ be 0.

Fig. 3 demonstrates the procedure of building a DA. A proportion (20%) of entries of $\mathbf{x}$ is corrupted with an equal chance by masking them as zeros. Next, the corrupted input,$\tilde{\mathbf{x}}$, is encoded through the activation function f_θ and the code, $\mathbf{y}$, is decoded through another activation function, g'_θ. Through minimizing the loss function, DAs are trained.

3) Layer-Wise Pre-Training: A SDA is a set of stacked DAs with a linear output layer as shown in Fig. 4.

In Fig. 4, the input layer of each autoencoder is the previous hidden layer. Thus, the upper layer always utilizes the hidden representation of the lower layer as its inputs. The unsupervised pre-training is applied to initialize the weights and biases of autoencoders. The procedure is described by the Pseudo Code, TrainUnsupervisedSDA():

```
Train UnsupervisedSDA(D, N, W, b, e)
{
    initialize b_0 = 0
    for ℓ = 1 to N:
        while e_ℓ not minimum:
            update W_ℓ and b_{ℓ−1}
        end while
    end for
}
```

In TrainUnsupervisedSDA(), $\mathbf{D}$ is the input data for training the network, N is the number of hidden layers to train, $\mathbf{W}_\ell$ is the weight for layer ℓ, $\ell = 1, 2, \cdots, N$, $\mathbf{b}_\ell$ is the bias for layer ℓ, $\ell = 1, 2, \cdots, N$ and e_i is the reconstruction error for layer ℓ, $\ell = 1, 2, \cdots, N$. The $\mathbf{W}$ and $\mathbf{b}$ are updated with the stochastic gradient descent method [55].

4) RS-SDA: The RS-SDA is a SDA incorporating the RANSAC and SNE. The RANSAC is an iterative method for eliminating influences of outliers during the construction of RS-SDA models. A layer–wise implementation of the SNE is applied to automatically determine the number of hidden units of hidden layers in the RS-SDA. The development of the RS-SDA is described in the Pseudo Code, RSSDA().

```
RSSDA(D, N, W, b, e)
{
    D′= RANSACTD(D)
    initialize b_0 = 0
    for ℓ = 1 to N:
        v_ℓ = HNSNE (D′)
        while e_ℓ not minimum:
            update W_ℓ and b_{ℓ−1}
        end while
    end for
}
```

In RSSDA(), v describes the number of hidden units. The RANSACTD() in RSSDA() for further processing the training data, $\mathbf{D}'$, is offered in the following. The DA rather than the RS-SDA is utilized to rapidly determine $\mathbf{D}'$.

```
RANSACTD(I, h, c, d, D)
{
    let i = 0, tData = null, best = 0
    while i < I:
        subData = randomly sample h data from D
        DA = fitDAmodel(subData)
        for x in D not in sub Data:
            compute ℒ
            if ℒ < c
                tData = tData ∪ x
            end if
        end for
        if size of tData > max(d, best):
            D′ = subdata ∪ tData
            best = size of D′
        end if
        i = i + 1
    end while
    return output
}
```

In RANSACTD(), I describes iteration number for executing the RANSAC, h is the minimal number of data for fitting the DA model, and c and d are two thresholds. I is set to 100, h is set to 300, c is set to the mean of $\mathcal{L}$ over training data, and d is set to 50% of training data.

The SNE was designed to transform a high dimensional data to a lower dimension or another dimension by maximizing the similarity between the original and transformed data in terms of the perplexity. Such advantage is brought to layer-wisely determine the number of hidden units of hidden layers based on $\mathbf{D}'$. The HNSNE() in RSSDA() is presented in the following Pseudo Code.

```
HNSNE(D′)
{
    compute p_ij of D′
    for v ∈ A:
        initialize Z with N(0, 10^{−4} I_v)
```

compute q_{ij} and update $\mathbf{Z}$ iteratively
compute KL_v
end for
let v = index of min($\{KL_v\}$)
return v
}

In HNSNE(), v describes the number of hidden units in the next hidden layer. A is a pre-defined interval of v. The probabilities p_{ij} are determined based on $\mathbf{D}'$ according to (8).

$$p_{j|i} = \frac{\exp\left(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma_i^2\right)}{\sum_{k \neq i} \exp\left(-\|\mathbf{x}_i - \mathbf{x}_k\|^2 / 2\sigma_i^2\right)}, p_{\mathrm{ij}} = \frac{p_{j|i} + p_{i|j}}{2\mathcal{N}} \quad (8)$$

The σ_i is adapted to the density of data in $\mathbf{D}'$ via a binary search. The probabilities q_{ij} is computed based on (9).

$$q_{\mathrm{ij}} = \frac{\left(1 + \|\mathbf{z}_i - \mathbf{z}_j\|^2\right)^{-1}}{\sum_{k \neq l} \left(1 + \|\mathbf{z}_k - \mathbf{z}_l\|^2\right)^{-1}} \quad (9)$$

The v dimensional data $\mathbf{z}_i$ in $\mathbf{Z}$ is determined by minimizing the Kullback–Leibler defined in (10) with the gradient descent.

$$KL(P||Q) = \sum_{i \neq j} p_{\mathrm{ij}} \log \frac{p_{\mathrm{ij}}}{q_{\mathrm{ij}}} \quad (10)$$

Although the objective of SNE is to minimize the layer-wise KL divergence and training RS-SDA aims at minimizing e, a computational experiment presented in the Supplementary Materials shows that a better minimization of KL divergence in SNE is beneficial to improve the RS-SDA performance trained based on $\mathbf{D}'$.

5) Fine Tuning: After initializing parameters of the stacked autoencoders, their values will be further fine-tuned through a standard backpropagation algorithm. During the fine-tuning process, the electricity price is provided as the response parameter. To facilitate the supervised fine-tuning process, a mini-batch gradient descent training process [38] is utilized. Compared with the batch gradient descent which uses the full training sample, the mini-batch gradient descent only employs b samples which indicate the mini-batch size. As the SDA and RS-SDA have multiple hidden layers and the targeted dataset contains redundant information, the application of the mini-batch gradient descent can gain following advantages: 1) the mini-batch gradient descent method analyses the data more efficiently since DNN parameters are updated by examining b samples; 2) it does not require the iterative update of parameters and a complete screen of the full dataset. In the implementation, data with a fixed batch size are sequentially drawn from the training dataset and the training dataset is shuffled after each epoch.

Given the mini-batch size, b, the number of training examples, m, and the n dimensional input data, the algorithm is presented in the Pseudo Code, MiniBatchUpdate():

MiniBatch Update(**x**, y, θ, α, h_θ)
{
while not stopping criterion:
for $i = 1, 1 + b, 1 + 2b, \ldots, m - b + 1$:
for $j = 0, 1, \ldots, n$

$$\theta_j = \theta_j - \alpha \frac{1}{b} \sum_{k=i}^{i+b-1} \left(h_\theta(\mathbf{x}^{(k)}) - y^{(k)}\right) \mathbf{x}_j^{(k)}$$

end for
end for
end while
}

In MiniBatchUpdate(), **x** is the input of the network, y is the output of the network, θ describes parameters to be estimated, α is the learning rate and h_θ is the activation function. The α is set to 0.01 according to [56] and the MiniBatchUpdate() stops if the sum of square errors does not decrease.

Besides the mini-batch gradient descent, the momentum is applied to further accelerate the training process. In this case, values of parameters are changed with a velocity which is a gradient iteratively. This velocity decays exponentially over time while a multiplication of the gradient and a learning rate is continuously added into it. Since the velocity is a vector, the gradient is able to change both of its direction and magnitude.

6) The Training and Forecasting Procedure: The training and forecasting procedures of the SDA and RS-SDA are summarized as follows.

Step 1): Determine the training sets: Data of past 30 days are treated as the training set for fine-tuning and such training set will be updated daily.

Step 2): Select prices with similar loads: At the forecasted hour, three historical prices with most similar loads and weight gas price indexes are selected as the input parameters.

Step 3): Pre-training of the network: The SDA or RS-SDA is pre-trained with the training dataset to initialize the weights and biases of the network.

Step 4): Fine-tuning of the network: The network is further fine-tuned with the mini-batch gradient descent.

Step 5): Forecasting using DNN: Based on the trained SDA or RS-SDA, the forecasted price is obtained.

Step 6): Forecasting all 24 hours: Steps 1–5 are repeated for all 24 forecasting hours and the SDA or RS-SDA is applied to generate the day-ahead forecasts of electricity prices.

In this study, the online hourly forecasting is achieved by implementing Steps 3–5 and the day-ahead hourly forecasting requires the implementation of all steps.

B. Model Evaluation

Four metrics, mean absolute percentage error (MAPE), hit rate (HR), daily mean absolute percentage error ($\mathrm{MAPE}_{\mathrm{day}}$) and monthly mean absolute percentage error ($\mathrm{MAPE}_{\mathrm{month}}$), defined in (11)–(14), assess the performance of developed data-driven models.

$$\mathrm{MAPE} = \frac{1}{n} \sum_{i=1}^{n} \left(\left| \frac{P_i^{\mathrm{true}} - P_i^{\mathrm{est}}}{P_i^{\mathrm{true}}} \right| \right) \quad (11)$$

$$\mathrm{HR} = \frac{1}{n} \sum_{i=1}^{n} I(i),$$

$$I(i) = \begin{cases} 1 & \text{if } \left| \frac{P_i^{\mathrm{true}} - P_i^{\mathrm{est}}}{P_i^{\mathrm{true}}} \right| \leq 0.07 \\ 0 & \text{if } \left| \frac{P_i^{\mathrm{true}} - P_i^{\mathrm{est}}}{P_i^{\mathrm{true}}} \right| > 0.07 \end{cases} \quad (12)$$

$$\mathrm{MAPE}_{\mathrm{day}} = \frac{1}{24} \sum_{i=1}^{24} \left(\frac{|P_i^{\mathrm{true}} - P_i^{\mathrm{est}}|}{\overline{P}_{\mathrm{day}}^{\mathrm{true}}} \right),$$

$$\bar{P}^{\text{true}}_{\text{day}} = \frac{1}{24}\sum_{i=1}^{24} P_i^{\text{true}} \tag{13}$$

$$\text{MAPE}_{\text{month}} = \frac{1}{m}\sum_{i=1}^{m}\left(\frac{|P_i^{\text{true}} - P_i^{\text{est}}|}{\overline{P}^{\text{true}}_{\text{month}}}\right),$$

$$\bar{P}^{\text{true}}_{\text{month}} = \frac{1}{m}\sum_{i=1}^{m} P_i^{\text{true}} \tag{14}$$

where P_i^{true} is the actual price in the *i*-th hour,P_i^{est} is the forecasted price, *n* is the number of samples in the testing dataset, and *m* denotes the number of days in the forecasted month.

In the on-line electricity price forecasting, overall accuracy of forecasting, such as MAPE, and the frequency of achieving such accuracy, such as HR, are both important. Thus, the MAPE and HR are applied to evaluate models. In the day-ahead hourly forecasting, the MAPE_{day} and $\text{MAPE}_{\text{month}}$ [9], [57], are considered.

IV. Computational Studies

The SDA and RS-SDA models are applied to forecast electricity prices of considered hubs and their performance is evaluated with metrics in Section III.B. In the on-line hourly forecasting, the MAPE and HR obtained from the SDA and RS-SDA models are compared with classical data-driven approaches including NN, MARS, SVM and Lasso. All models utilized same input parameters stated in (2). In the day-ahead hourly forecasting, the SDA and RS-SDA models are further compared with forecasting results generated by e-ISOForecast [40] and a newly reported Lasso-based method [26] besides classical data-driven approaches. As preliminary computational results present that the Lasso model using inputs described in [26] (average $\text{MAPE}_{\text{day}} = 20.38\%$ based on data of 30 randomly selected days from Indiana hub) rather than inputs stated in (3) (average $\text{MAPE}_{\text{day}} = 25.67\%$ based on the same data) performs better, the Lasso model development in day-ahead forecasting exactly follows the procedure and inputs described in [26]. All other data-driven models use inputs stated in (3).

A. On-Line Hourly Forecasting

Hourly electricity prices of NPPD, Arkansas, Louisiana, and Texas hubs are forecasted. The proposed SDA and RS-SDA are compared with NN, SVM, MARS, and Lasso based on data collected from January, 2012 to November, 2014. The whole dataset is split into the training and test datasets with the 70% and 30% rule. Models are trained with the training dataset and evaluated with the test dataset. In SDA, number of hidden units is set to 30 for all hidden layers while *A* is set to [20, 40] in the RS-SDA. The hidden layers of SDA and RS-SDA are increased until the testing error does not decrease. Finally, SDA and RS-SDA with three hidden layers are chosen in the comparison. A grid search is applied to optimize parameters of other considered modeling approaches and details are provided in Supplementary Materials. All considered algorithms were implemented with Python 2.7 and based on a computer with the CORE I5 CPU and the 8 GB memory. Meanwhile, the GPU was utilized for training the NN, SDA and RS-SDA.

Data of two days (Monday, September 1st, 2014, and Sunday, October 12th, 2014) from the NPPD hub are selected as the illustrative examples. The MAPE and HR of two representative days are computed and reported in Table II.

TABLE II
MAPE and HR for Two Representative Days

Model	MAPE for Monday	HR for Monday	MAPE for Sunday	HR for Sunday
NN	4.67	83.33	6.20	70.83
MARS	4.28	83.33	6.22	70.83
SVM	5.98	66.67	6.25	70.83
Lasso	6.19	62.50	6.78	66.67
SDA	4.28	87.50	5.91	75.00
RS-SDA	4.02	87.50	5.17	79.17

TABLE III
MAPE and HR (%) of Different Model for NPPD Hub

Model	MAPE	HR
NN	6.42	70.29
MARS	6.51	69.95
SVM	7.52	61.61
Lasso	7.51	61.14
SDA	6.39	66.64
RS-SDA	6.28	71.91

TABLE IV
MAPE and HR (%) for Arkansas, Louisiana and Texas Hubs

	Arkansas		Louisiana		Texas	
Model	MAPE	HR	MAPE	HR	MAPE	HR
NN	5.52	75.86	6.19	70.91	6.19	72.29
MARS	5.79	71.25	6.28	70.83	6.54	70.16
SVM	6.22	68.53	6.14	71.53	7.34	62.50
Lasso	7.56	52.77	7.70	52.73	7.88	61.83
SDA	4.45	76.83	4.66	75.17	5.42	76.39
RS-SDA	4.36	79.95	4.51	79.34	5.16	78.32

Table II indicates that SDA and RS-SDA are generally better than other models in forecasting the electricity prices of the specific weekday and weekend. Meanwhile, RS-SDA has more accurate results than SDA. To further compare different models, the MAPE and HR based on all test data of the NPPD are computed and presented in Table III.

In Table III, it is observable that RS-SDA has the best performance with the lowest MAPE and highest HR while SDA has the second best performance. The Lasso model offers the worst performance. Table IV shows the MAPE and HR of NN, MARS, SVM, Lasso, SDA, and RS-SDA in the electricity price forecasting of Arkansas, Louisiana and Texas hubs.

According to the MAPE and HR, we can observe that the SDA and RS-SDA outperforms other models in forecasting the electricity price for all three hubs. In addition, RS-SDA can improve the performance of SDA. The overall computational results prove that the SDA and RS-SDA model are more effective than classical data-driven models in the online hourly electricity price forecasting.

The computational time of developing data-driven models with optimized settings of the algorithm parameters is provided in Table V. It is observable that SDA with GPU implementation requires less time than SVM based on CPU. Although the

TABLE V
COMPUTATIONAL TIME FOR EACH ALGORITHM

Algorithms	Time
MARS	0.524 s
SVM	1.959 s
SDA	1.582 s
Lasso	0.033 s
RS-SDA	14.386 s
NN	1.034 s

TABLE VI
THE $\mathrm{MAPE}_{\mathrm{day}}$ (%) FOR THE NINE REPRESENTATIVE DAYS

Date	SDA	RS-SDA	SDA$^+$	IM	MARS	NN	SVM	Lasso
January 1	8.03	7.89	10.41	7.25	14.77	13.95	15.16	11.71
January 10	17.37	17.05	18.15	22.30	18.67	17.22	18.29	29.52
January 30	23.99	21.78	44.91	44.39	116.38	52.61	25.47	65.26
April 1	12.19	11.35	20.98	11.20	14.13	13.62	17.91	32.34
April 10	13.71	13.67	16.63	15.38	16.12	16.52	16.71	14.19
April 30	15.68	13.96	13.56	25.43	16.04	16.81	25.40	14.53
July 1	11.71	11.39	14.58	13.39	13.96	13.73	28.38	12.53
July 10	4.37	4.39	11.16	11.82	8.67	14.38	20.18	4.91
July 30	8.79	8.13	10.96	9.33	47.56	8.98	16.73	9.69
Average	12.87	12.18	17.93	17.83	25.59	18.65	20.47	21.63

RS-SDA requires more computational time, it is acceptable in the real practice.

B. Day-Ahead Hourly Forecasting

The effectiveness of SDA and RS-SDA in the day-ahead hourly forecasting are evaluated with the data (from November 2013 to November 2014) taken from the Indiana Hub. The day-ahead hourly electricity prices of 2014 are all forecasted.

The data of year 2013 are utilized to pre-train the SDA and RS-SDA. In addition, a SDA model, denoted as SDA$^+$, is developed based on input parameters considered by the Lasso model in [26] for forecasting electricity prices of 24 hrs. In order to handle 221 inputs considered in [26], a SDA with 500 hidden units in each hidden layer is employed. Parameters of NN, SVM and MARS are also optimized in the day-ahead hourly forecasting. Data of year 2013 and the latest past 30 days are utilized to train benchmarking models and a daily update is applied to maintain the effectiveness of benchmarking models.

Three days, 1st, 10th and 30th, of three months, January, April and July, are selected to demonstrate the forecasting results. The $\mathrm{MAPE}_{\mathrm{day}}$ of forecasting day-ahead hourly electricity prices with different methods for all nine days is summarized in Table VI. As shown in Table VI, the RS-SDA offers lowest errors in general and SDA is the second best. Moreover, RS-SDA and SDA yield better results than the IM. Two-sided *t*-tests are performed between absolute percentage errors (APEs) of SDA and RS-SDA, and APEs of other approaches with a significance level $= 0.05$. Results of *t*-tests conclude that APEs of SDA are significantly different from APEs of other approaches except APEs of NN at Jan 10th, APEs of IM at Apr 1st, and APEs of NN at July 30th, while APEs of RS-SDA are significantly different from those of others excluding APEs of IM at Apr 1st.

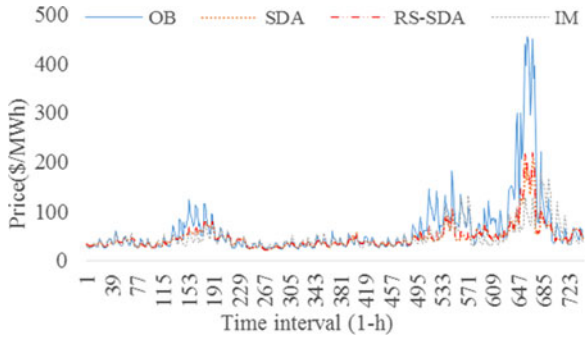


Fig. 5. Observed and forecasted day-ahead electricity prices (January, 2014).

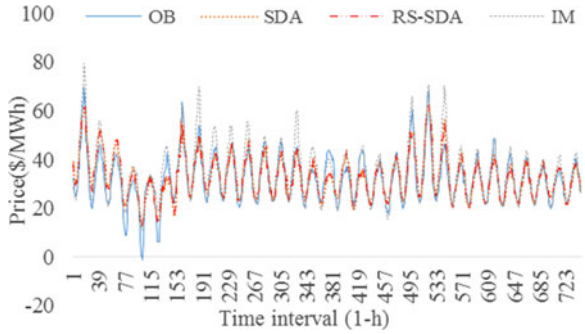


Fig. 6. Observed and forecasted day-ahead electricity prices (April, 2014).

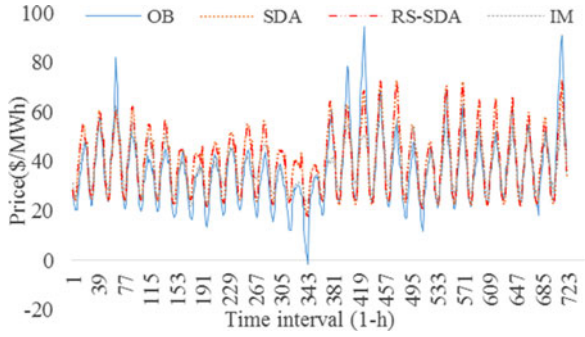


Fig. 7. Observed and forecasted day-ahead electricity prices (July, 2014).

To further validate the capability of SDA, RS-SDA and IM in forecasting electricity prices, forecasted results with SDA, RS-SDA, and IM of considered three months, January, April and July, are examined and shown in Figs. 5–7.

In Fig. 5, significant spikes were observed at the end of January. Such spikes were probably caused by the strategic behavior of market players. Both of the SDA, RS-SDA, and IM cannot accurately forecast such occasional fluctuation. Nevertheless, the performance of the SDA and RS-SDA are still better than the IM. The $\mathrm{MAPE}_{\mathrm{month}}$ of the IM over January is 44.7% while that of SDA and RS-SDA are less than 32% (31.8% and 29.78% respectively). The advantage of SDA and RS-SDA gains from the continuous update of training dataset which ensure them to learn patterns of previous spikes and offer more decent forecasts later.

The variation of hourly electricity prices in April is less significant than that in January. The SDA and RS-SDA outperform

TABLE VII
THE $\mathrm{MAPE}_{\mathrm{month}}$ (%) FOR THE THREE REPRESENTATIVE MONTHS

Month	SDA	RS-SDA	SDA$^+$	IM	MARS	NN	SVM	Lasso
January	31.80	29.78	36.91	44.70	46.35	40.47	49.64	37.46
April	2.51	2.47	9.88	7.54	11.78	10.26	18.65	9.87
July	10.04	8.97	11.09	11.45	20.79	12.00	23.20	11.91

IM in the forecasting. The $\mathrm{MAPE}_{\mathrm{month}}$ of SDA and RS-SDA are 2.51% and 2.47%, much lower than that of IM, 7.54%.

A strong fluctuation of hourly electricity price in July, 2014 is obtained again as shown in Fig. 7. Therefore, the $\mathrm{MAPE}_{\mathrm{month}}$ of all models increases slightly. The $\mathrm{MAPE}_{\mathrm{month}}$ of the SDA, RS-SDA, and IM are 10.04%, 8.97%, and 11.45% respectively. Thus, the performance of RS-SDA is the best among them. At the beginning of July 2014, negative prices are observed and this was due to the strategic behavior of the dominant player. Table VII summarizes the $\mathrm{MAPE}_{\mathrm{month}}$ of all three months for all considered methods.

Other results are consistent with those of nine days: SDA and RS-SDA have better performance than other methods while RS-SDA slightly wins SDA. Based on same input parameters considered in [26], SDA$^+$ offered similar performance to the Lasso-based approach as shown in Table VII. Meanwhile, results in Table VII indicate that considering exogenous factors in addition to time-series of electricity prices is beneficial to improve the forecasting accuracy. Same *t*-tests are performed between APEs of SDA and RS-SDA, and APEs of other approaches for all three months and we discover that APEs of SDA and RS-SDA are significantly different from APEs of other approaches. The comparative analyses prove the effectiveness of the SDA and RS-SDA models in both online and day-ahead hourly electricity price forecasting. Meanwhile, RS-SDA is able to further improve the performance of SDA.

V. CONCLUSIONS

This paper studied the application of the SDA and RS-SDA models, in both of the online and day-ahead hourly electricity price forecasting. Data collected from the Nebraska, Arkansas, Louisiana, Texas, and Indiana ISO hubs in the U.S. were utilized. A comprehensive analysis of the capability of the SDA and RS-SDA models in the electricity price forecasting was performed.

To validate the effectiveness of SDA and RS-SDA models in forecasting electricity prices, a comparative analysis of the SDA, RS-SDA, classical NN, SVM, and MARS models in online hourly forecasting was firstly conducted. Next, the SDA and RS-SDA models were compared with the IM and Lasso model in day-ahead hourly forecasting in addition to NN, SVM, and MARS. Four metrics, the MAPE, HR, $\mathrm{MAPE}_{\mathrm{day}}$ and $\mathrm{MAPE}_{\mathrm{month}}$, were utilized to assess the forecasting performance.

Results of computational studies validated the capability of the SDA and RS-SDA models in forecasting electricity prices. In online hourly forecasting, RS-SDA provided the lowest MAPE and highest HR while SDA had the second best performance. In day-ahead hourly forecasting, the SDA and RS-SDA models were generally better than other compared models. In both of these two forecasting, RS-SDA generated more accurate results than SDA and this superiority proved the effectiveness of the proposed enhanced pre-training procedures. The settings of DNN parameters could affect its performance, such as the number of considered hidden layers and learning rate. A more advanced approach for determining the optimal settings of parameters in DNN needs to be discussed in the future research.

REFERENCES

[1] T. Mathaba, X. Xia, and J. Zhang, "Analysing the economic benefit of electricity price forecast in industrial load scheduling," *Elect. Power Syst. Res.*, vol. 116, pp. 158–165, 2014.

[2] K. Sarada and V. Bapiraju, "Comparison of day-ahead price forecasting in energy market using Neural Network and Genetic Algorithm," in *Proc. Int. Conf. Smart Elect. Grid*, 2014, pp. 1–5.

[3] M. Shafie-Khah, M. P. Moghaddam, and M. Sheikh-El-Eslami, "Price forecasting of day-ahead electricity markets using a hybrid forecast method," *Energ. Convers. Manage.*, vol. 52, no. 5, pp. 2165–2169, 2011.

[4] R. C. Garcia, J. Contreras, M. Van Akkeren, and J. B. C. Garcia, "A GARCH forecasting model to predict day-ahead electricity prices," *IEEE Trans. Power Syst.*, vol. 20, no. 2, pp. 867–874, May 2005.

[5] M. Shahidehpour, H. Yamin, and Z. Li, "Market overview in electric power systems," in *Market Operations in Electric Power Systems*. New York, NY, USA: Wiley, 2002, pp. 1–20.

[6] G. Ci-wei, E. Bompard, R. Napoli, and H. Cheng, "Price forecast in the competitive electricity market by support vector machine," *Physica A: Stat. Mech. Appl.*, vol. 382, no. 1, pp. 98–113, 2007.

[7] Y. Cai, J. Lin, C. Wan, and Y. Song, "A stochastic bi-level trading model for an active distribution company with distributed generation and interruptible loads," *IET Renew. Power Gener.*, to be published.

[8] R. Weron, "Electricity price forecasting: A review of the state-of-the-art with a look into the future," *Int. J. Forecast.*, vol. 30, no. 4, pp. 1030–1081, 2014.

[9] L. Hu and G. Taylor, "A novel hybrid technique for short-term electricity price forecasting in UK electricity markets," *J. Int. Council Elect. Eng.*, vol. 4, no. 2, pp. 114–120, 2014.

[10] S. Voronin and J. Partanen, "Forecasting electricity price and demand using a hybrid approach based on wavelet transform, ARIMA and neural networks," *Int. J. Energy Res.*, vol. 38, no. 5, pp. 626–637, 2014.

[11] P. Kou, D. Liang, L. Gao, and J. Lou, "Probabilistic electricity price forecasting with variational heteroscedastic Gaussian process and active learning," *Energy Convers. Manage.*, vol. 89, pp. 298–308, 2015.

[12] N. A. Shrivastava and B. K. Panigrahi, "A hybrid wavelet-ELM based short term price forecasting for electricity markets," *Int. J. Elect. Power Energy. Syst.*, vol. 55, pp. 41–50, 2014.

[13] K. He, Y. Xu, Y. Zou, and L. Tang, "Electricity price forecasts using a curvelet denoising based approach," *Physica A: Stat. Mech. Appl.*, vol. 425, pp. 1–9, 2015.

[14] 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.

[15] C. Wan, M. Niu, Y. Song, and Z. Xu, "Pareto optimal prediction intervals of electricity price," *IEEE Trans. Power Syst.*, to be published.

[16] O. B. Fosso, A. Gjelsvik, A. Haugstad, B. Mo, and I. Wangensteen, "Generation scheduling in a deregulated system. the Norwegian case," *IEEE Trans. Power Syst.*, vol. 14, no. 1, pp. 75–81, Feb. 1999.

[17] T. Jonsson, P. Pinson, H. A. Nielsen, H. Madsen, and T. S. Nielsen, "Forecasting electricity spot prices accounting for wind power predictions," *IEEE Trans. Sustain. Energy*, vol. 4, no. 1, pp. 210–218, Jan. 2013.

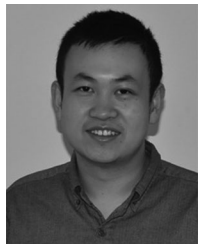
[18] J. Contreras, R. Espinola, F. J. Nogales, and A. J. Conejo, "ARIMA models to predict next-day electricity prices," *IEEE Trans. Power Syst.*, vol. 18, no. 3, pp. 1014–1020, Aug. 2003.

[19] H. Zareipour, C. A. Cañizares, K. Bhattacharya, and J. Thomson, "Application of public-domain market information to forecast Ontario's wholesale electricity prices," *IEEE Trans. Power Syst.*, vol. 21, no. 4, pp. 1707–1717, Nov. 2006.

[20] F. J. Nogales, J. Contreras, A. J. Conejo, and R. Espínola, "Forecasting next-day electricity prices by time series models," *IEEE Trans. Power Syst.*, vol. 17, no. 2, pp. 342–348, May 2002.

[21] A. J. Conejo, J. Contreras, R. Espínola, and M. A. Plazas, "Forecasting electricity prices for a day-ahead pool-based electric energy market," *Int. J. Forecast.*, vol. 21, no. 3, pp. 435–462, 2005.

[22] F. Paraschiv, S. Fleten, and M. Schürle, "A spot-forward model for electricity prices with regime shifts," *Energy Econ*., vol. 47, pp. 142–153, Jan. 2015.
[23] R. Weron and A. Misiorek, "Forecasting spot electricity prices: A comparison of parametric and semiparametric time series models," *Int. J. Forecast.*, vol. 24, no. 4, pp. 744–763, 2008.
[24] A. J. Conejo, M. A. Plazas, R. Espinola, and A. B. Molina, "Day-ahead electricity price forecasting using the wavelet transform and ARIMA models," *IEEE Trans. Power Syst.*, vol. 20, no. 2, pp. 1035–1042, May 2005.
[25] P. Gaillard, Y. Goude, and R. Nedellec, "Additive models and robust aggregation for GEFCom2014 probabilistic electric load and electricity price forecasting," *Int. J. Forecast.*, to be published.
[26] F. Ziel, "Forecasting electricity spot prices using lasso: On capturing the autoregressive intraday structure," *IEEE Trans. Power Syst.*, vol. 31, no. 6, pp. 4977–4987, Nov. 2016.
[27] T. Hong, P. Pinson, S. Fan, H. Zareipour, A. Troccoli, and R. J. Hyndman, "Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond," *Int. J. Forecast.*, to be published.
[28] Y. Yang, Y. Dong, Y. Chen, and C. Li, "Intelligent optimized combined model based on GARCH and SVM for forecasting electricity price of New South Wales, Australia," *Abstr. Appl. Anal.*, vol. 2014, 2014, Art. no. 504064.
[29] X. Yan and N. A. Chowdhury, "Mid-term electricity market clearing price forecasting: A multiple SVM approach," *Int. J. Elect. Power*, vol. 58, pp. 206–214, 2014.
[30] X. Yan and N. A. Chowdhury, "Hybrid SVM & ARMAX based mid-term electricity market clearing price forecasting," in *Proc. IEEE Conf. Elect. Power Energy Conf.*, 2013, pp. 1–5.
[31] M. Tripathi, K. Upadhyay, and S. Singh, "Electricity price forecasting using generalized regression neural network (GRNN) for PJM electricity market," *Int. J. Manage.-Theory Appl.*, vol. 2, no. 4, pp. 137–142, 2014.
[32] A. T. Lora, J. M. R. Santos, A. G. Expósito, J. L. M. Ramos, and J. C. R. Santos, "Electricity market price forecasting based on weighted nearest neighbors techniques," *IEEE Trans Power Syst*., vol. 22, no. 3, pp. 1294–1301, Aug. 2007.
[33] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
[34] Y. Bengio and O. Delalleau, "On the expressive power of deep architectures," in *Proc. 22nd Int. Conf. Algorithmic Learn. Theory*, 2011, pp. 18–36
[35] P. Vincent, H. Larochelle, Y. Bengio, and P. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. Int. Conf. Mach. Learn.*, 2008, pp. 1096–1103.
[36] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.
[37] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, 2008.
[38] T. Nakama, "Theoretical analysis of batch and on-line training for gradient descent learning in neural networks," *Neurocomputing*, vol. 73, no. 1, pp. 151–15, Nov. 2009.
[39] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1139–1147.
[40] PRT. (2016, May 12). Load and Price Forecasting for North American Electricity Markets. [Online]. Available: http://www.prt-inc.com/e-ISOForecast.aspx
[41] MISO Energy. (2016, May 8). Markets and Operations. [Online]. Available: https://www.misoenergy.org/MarketsOperations/Pages/MarketsOperations.aspx
[42] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, 2001.
[43] B. Kröse, B. Krose, P. van der Smagt, and P. Smagt, *An Introduction to Neural Networks*. Amsterdam, The Netherlands: Univ. Amsterdam Press, 1993.
[44] A. Smola and V. Vapnik, "Support vector regression machines," *Adv. Neural Inf. Process Syst.*, vol. 9, pp. 155–161, 1997.
[45] J. H. Friedman, "Multivariate adaptive regression splines," *Ann. Statist.*, vol. 19, no. 1, pp. 1–67, 1991.
[46] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Statist. Soc. Ser. B Statist. Methodol.* vol. 58, no. 1, pp. 267–288, 1996.
[47] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2010, pp. 249–256.
[48] G. E. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput*., vol. 18, no. 7, pp. 1527–1554, 2006.
[49] M. A. Ranzato, F. J. Huang, Y. Boureau, and Y. LeCun, "Unsupervised learning of invariant feature hierarchies with applications to object recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.
[50] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive auto-encoders: Explicit invariance during feature extraction," in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 833–840.
[51] G. Dahl, A. Mohamed, and G. E. Hinton, "Phone recognition with the mean-covariance restricted Boltzmann machine," in *Proc. Adv. Neural. Inf. Process. Syst.*, 2010, pp. 469–477.
[52] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
[53] C. Liou, W. Cheng, J. Liou, and D. Liou, "Autoencoder for words," *Neurocomputing*, vol. 139, no. 2, pp. 84–96, 2014.
[54] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 28, 2006.
[55] S. Amari, "Backpropagation and stochastic gradient descent method," *Neurocomputing*, vol. 5, no. 4, pp. 185–196, 1993.
[56] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Neural Networks: Tricks of the Trade*. Berlin, Germany: Springer, 2012, pp. 437–478.
[57] N. Amjady, "Day-ahead price forecasting of electricity markets by a new fuzzy neural network," *IEEE Trans. Power Syst.*, vol. 21, no. 2, pp. 887–896, May 2006.



Long Wang (S'16) received the M.Sc. degree in computer science with distinction from the University College London, London, U.K., in 2014. He is currently working toward the Ph.D. degree in the Department of Systems Engineering and Engineering Management, City University of Hong Kong, Hong Kong.

His research interests include electricity price forecasting, wind turbine condition monitoring, computer vision, and parallel computing.

Zijun Zhang (M'12) received the B.Eng. degree in systems engineering and engineering management from the Chinese University of Hong Kong, Hong Kong, China, in 2008 and the M.S. and Ph.D. degrees in industrial engineering from the University of Iowa, Iowa City, IA, USA, in 2009 and 2012, respectively.

He is currently an Assistant Professor in the Department of Systems Engineering and Engineering Management, City University of Hong Kong, Hong Kong, China. His research focuses on data mining and computational intelligence with applications in wind energy, HVAC, and wastewater processing domains.

Jieqiu Chen received the Ph.D. degree in business administration with a concentration in operations research from the University of Iowa, Iowa City, IA, USA, in 2010.

She is currently a Senior Data Scientist with the Microsoft Corporation. She was an Argonne Scholar at Argonne National Laboratory during 2010–2012. She received the Wilkinson Fellowship in Scientific Computing from the Mathematics and Computer Science Division, Argonne National Laboratory in 2010.