Hybrid Kalman Filters for Very Short-Term Load Forecasting and Prediction Interval Estimation

Che Guan, *Member, IEEE*, Peter B. Luh, *Fellow, IEEE*, Laurent D. Michel, and Zhiyi Chi

***Abstract*—Very short-term load forecasting predicts the loads in electric power system one hour into the future in 5-min steps in a moving window manner. To quantify forecasting accuracy in real-time, the prediction interval estimates should also be produced online. Effective predictions with good prediction intervals are important for resource dispatch and area generation control, and help power market participants make prudent decisions. We previously presented a two level wavelet neural network method based on back propagation without estimating prediction intervals. This paper extends the previous work by using hybrid Kalman filters to produce forecasting with prediction interval estimates online. Based on data analysis, a neural network trained by an extended Kalman filter is used for the low-low frequency component to capture the near-linear relationship between the input load component and the output measurement, while neural networks trained by unscented Kalman filters are used for low-high and high frequency components to capture their nonlinear relationships. The overall variance estimate is then derived and evaluated for prediction interval estimation. Testing results demonstrate the effectiveness of hybrid Kalman filters for capturing different features of load components, and the accuracy of the overall variance estimate derived based on a data set from ISO New England.**

***Index Terms*—Extended Kalman filter, prediction interval estimation, unscented Kalman filter, very short-term load forecasting, wavelet neural networks.**

I. Introduction

VERY short-term load forecasting (VSTLF) predicts the loads in electric power system one or several hours into the future in steps of a few minutes (e.g., 5-min) in a moving window manner. To quantify forecasting accuracy in real-time, the forecasting process should also estimate prediction intervals (PI) online. Accurate VSTLF with good PIs is important for resource dispatch and area generation control, and helps power market participants make prudent decisions. Based on data analysis, load series have multiple frequency components, and each may have its unique pattern, such as monthly, weekly, and hourly patterns. Effective VSTLF, however, is difficult in view of different characteristics of load components and the accurate derivation for online PI estimates.

Methods for VSTLF have been reviewed in our recent paper [1], including persistence [2], extrapolation [3]–[6], time series [7]–[11], Kalman filters [12], [13], fuzzy logic [7], [14]–[16], and neural networks (NN) [7], [17], [18]. Among these methods, NNs have been widely used. A standard NN trained by back propagation was used for VSTLF in [7]. To make data stationary, the load inputs to an NN were transformed by using a relative increment transformation in [18]. A single NN, however, may not be able to accurately capture complicated load features. This is because the load series has multiple frequency components, and each may have its unique pattern. To quantify VSTLF accuracy, the PI estimates should also be produced online. Since very few of these VSTLF methods have the capability of providing PI estimates online, methods of general prediction(s) with PI(s) will be reviewed in Section II-A, including maximum likelihood, distribution assumptive model, resampling, Bayesian inference, and Kalman filters.

Recently, we have developed a VSTLF method using wavelet neural networks (WNN) with data pre-filtering in [1]. This method will be briefly reviewed in Section II-B. The key idea was to use a wavelet technique to decompose filtered loads into three orthogonal components at different frequencies: low-low (LL), low-high (LH), and high (H) frequency components. All three NNs were applied to forecast individual components, and NNs' outputs were then combined to form forecasts. To perform the VSTLF in a moving manner, twelve dedicated WNNs were used to form the moving forecast. Since WNNs were trained by back propagation, the dynamic covariance cannot be produced for PI estimation. To quantify forecasting accuracy, a general resampling method was used for PI estimates [1]. The resampling, however, may not be accurate enough to estimate PIs due to the use of the back propagation algorithm for training NNs' weights. To capture complicated load features with accurate PIs, the WNN method needs to be extended, and PIs need to be further derived.

In this paper, our previous method of wavelet neural networks trained by back propagation [1] is further improved. By replacing the first-order back propagation algorithm with the second-order Kalman-type algorithms, a dynamic covariance can be produced for PI estimates. A method of wavelet neural networks trained by hybrid Kalman filters (WNNHKF) is developed. It forecasts loads one hour into the future in 5-min steps in a moving window manner with associated PI estimates in real-time. The data analysis shows that the LL frequency component has a near-linear relationship between the LL load input

Manuscript received July 15, 2012; revised July 16, 2012, December 20, 2012, March 10, 2013, and April 25, 2013; accepted May 05, 2013. Date of publication June 11, 2013; date of current version October 17, 2013. This work was supported in part by ISO New England and Alstom Grid. Paper no. TPWRS-00829-2012.

C. Guan and P. B. Luh are with the Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT 06269-4157 USA.

L. D. Michel is with the Department of Computer Science and Engineering, University of Connecticut, Storrs, CT 06269-2155 USA.

Zhiyi Chi is with the Department of Statistics, University of Connecticut, Storrs, CT 06269-4120 USA.

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.2013.2264488

and output measurement, whereas the LH and H frequency components have nonlinear relations. To capture the near-linear relationship between the LL input and output measurement, the extended Kalman filter is used to train a neural network (EKFNN) because the EKF is derived through linearizing a system and is good for the near-linear system. To capture highly nonlinear relationships for LH and H components, the unscented Kalman filter is used to train neural networks (UKFNN) because the UKF is good for highly nonlinear systems. Hybrid Kalman filters details will be presented in Section III.

Prediction intervals for VSTLF are estimated and then evaluated in Section IV. To accurately estimate online PIs, the overall variance estimate is calculated by adding up three orthogonal variance estimates from H, LH, and LL frequency NNs. The estimates for H and LH components are directly obtained. The estimate for LL component is further derived because the relative increment, a nonlinear transformation, is applied to the LL component. This relative increment is used to make the LL series stationary so that the transformed series can be easily captured by the NN. To assess the PIs, the distribution of the forecasting errors is analyzed, and then PIs are thoroughly evaluated.

In Section V, our model is configured by training, validation, and test processes in a three-way data split, as presented in [19, Ch. 2]. Example 1 uses a classroom-type problem to compare our WNNHKF to the methods of persistence, linear AR, single NN, and WNN so that our method can be verified in a simple way. Based on a data set from ISO New England (ISO-NE), Example 2 shows the values of EKFNN for the near-linear LL frequency component and UKFNNs for highly nonlinear LH and H frequency components. This example also demonstrates the accuracy of standard deviations derived for PI estimates. It is difficult to compare this method to others since the implementation details for other methods are not open, and there is no standard test data set. Nevertheless, it is clear that Kalman filters provide as a by-product dynamic covariance matrix for PI estimates, which, based on testing, are consistent with those calculated based on static historical errors.

A preliminary version of this paper was presented in [20] where a WNN trained by hybrid Kalman filters was established for VSTLF, and standard deviations from Kalman filters were derived for PI estimates. Based on the preliminary results, the relationships between input and output measurement for individual load components are thoroughly analyzed. The consistency of the dynamic innovation covariance to the static covariance for Kalman filters is discussed. Forecasting errors are further investigated, and PIs are then thoroughly evaluated. The results of other forecasting methods are added as the reference to be outperformed. For our method, model parameters are selected and justified based on a three-way data split, as presented in [19, Ch. 2].

II. Literature Review

A. Prediction Interval Estimation

Existing VSTLF methods have been reviewed in [1]. Since very few of these methods have the capability of producing the accurate PI estimate(s), methods of the general prediction(s) with PI(s) construction are reviewed in this paper. These methods mainly include the maximum likelihood method, the distribution assumptive model, the resampling method, the Bayesian approach, and Kalman-type filters.

The maximum likelihood algorithm is used to obtain a set of NN weights by minimizing an error function. As presented in [21], a traditional NN was extended with a new set of hidden neurons used for computing a variance for data noises. Based on this variance, the PI was constructed.

The distribution assumptive model assumes a certain distribution for loads or forecasting errors. A probabilistic load model in [22] and [23] assumed that load data had a multivariable probability density function, and predictions with variance estimates were obtained from the conditional distribution of the load given the weather information. A normal distribution for errors was assumed in [24], and the PI was constructed by multi-linear regression adapted to NNs. The method was further developed in [25] to consider effects of noisy data.

The resampling method derives the PI(s) by using subsets of available data (e.g., the load or the wind generation) or drawing sample errors randomly with replacement from a set of forecasting errors. Assuming that error samples are independent and identically distributed, the PI was estimated from a cumulative distribution function using ordered sample errors in [21]. An adapted resampling method was presented to provide prediction intervals for wind power generation in [26]. The method relied on a classification of recent forecast errors, a fuzzy inference model, and a multisampling resampling scheme for combining probability distributions. A modified bootstrap method was developed in [27] to estimate the distribution of short-term load forecasting. Based on this, PIs were obtained.

The Bayesian approach for an NN starts with a prior distribution of the NN's weights, and then optimized weights are determined by maximizing the posterior distribution based on historical data. Through Taylor series expansion, the prediction distribution conditioned on a new input and weights was derived and approximated as a Gaussian distribution [28], [29]. Markov Chain Monte Carlo methods were used to calculate a covariance for PI estimate in [28]. To improve computation efficiency, Quasi-Newton methods were applied, as presented in [29].

Kalman-type filters have been applied to NNs with PI estimates. Standard NNs are based on back propagation, which is a first-order gradient method and cannot produce a dynamic covariance for PI estimates. Therefore, the EKF was used to train and update a feed-forward NN by treating the NN's weights as a state vector [30]. To improve computation efficiency, the EKF was extended to the decoupled EKF by ignoring the interdependence of mutually exclusive groups of weights in [31]. The numerical stability and accuracy of the decoupled EKF were further improved by U-D factorization in [32] for short-term load price forecasting.

Among all the methods described above, NNs have been widely used, and they provide valuable information for PI estimate(s). However, few papers have presented effective and efficient ways to produce accurate online PIs for VSTLF.

B. Wavelet Neural Networks

Recently, we have developed a method of wavelet neural networks with spike pre-filtering for VSTLF [1]. The schematic of

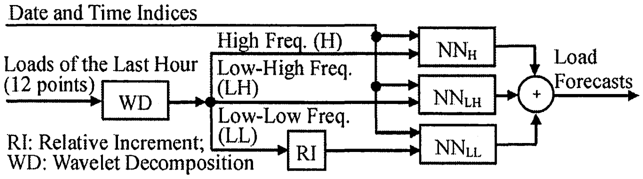

Fig. 1. Schematic of the wavelet neural networks (WNN).

WNN is highlighted in Fig. 1. The key idea for WNN was to use a wavelet technique to decompose the pre-filtered load data into three orthogonal components at different frequencies: LL, LH, and H components. The relative increment transformation in [18] was applied to the LL component to make the series stationary. The date and time indices were used to help NNs identify the periodical patterns of load data. Separate NNs were then used to predict individual components, and results of NNs were combined to form forecasts. However, it should also estimate PIs in order to quantify the forecasting accuracy in real-time. Since the WNN trained by back propagation cannot produce a dynamic covariance for PI estimates, the WNN method needs to be further improved.

III. Wavelet Neural Networks Trained by Hybrid Kalman Filters

In WNN described in Section II-B, load data have complicated features. To accurately categorize them, individual components are thoroughly analyzed. A linear autoregressive (AR) model (with a constant term added) and a standard nonlinear NN are separately used to investigate the relationship between the input load and the output measurement. Following [1], last hour's loads (12 points) are used as inputs to both models. To perform a time series of forecasts (12 points) by using AR, the input data are time-shifted. For example, data from $l(t-11)$ to $l(t)$ are used to forecast $l(t+1)$. Next, data from $l(t-10)$ to $l(t)$ plus the prediction of $l(t+1)$ are used together to forecast $l(t+2)$, and the process repeated until a prediction is made for $l(t+12)$.

To analyze individual components, take 60-min-ahead forecasting results for example. For LL component, the coefficient of determination value is 0.97 for AR, indicating a linear mapping for LL. To explore further, the scatter plot in Fig. 2(a) shows a nonlinear pattern between the prediction (x) and the residual (y) generated by the AR model, whereas the scatter plot in Fig. 2(d) does not show a clear nonlinear pattern by the NN. This indicates that the AR is incapable of capturing the residual nonlinearity, while the NN is capable of capturing both linearity and nonlinearity. It can thus be concluded that the LL component has a near-linear relationship between input and output measurement. A similar analysis is conducted on the LH component. The coefficient of determination value is 0.08 for AR. Moreover, Fig. 2(b) shows predictions from AR are concentrated around zero, whereas Fig. 2(e) shows a complex pattern in predictions by the NN. The above indicates a highly nonlinear mapping for the LH component. Similar to LH, the same conclusion is made on the H component from the coefficient of determination value as well as Fig. 2(c) and (f). Both AR and NN methods are also used for analyzing 5- to 55-min-ahead

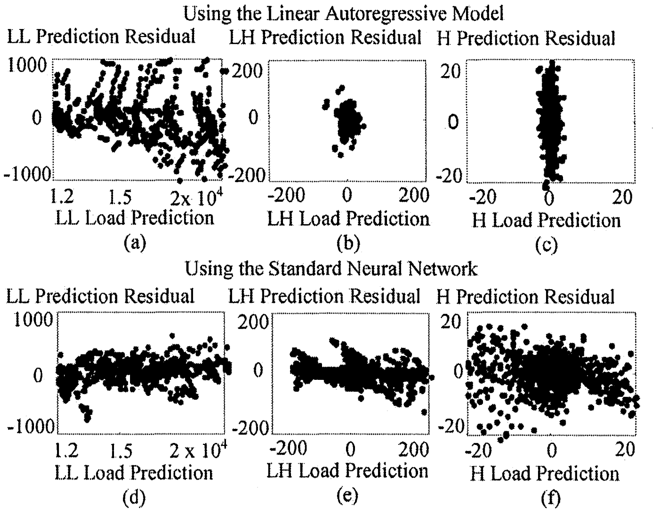

Fig. 2. Scatter plots of 60-min-ahead predictions and residuals for individual LL, LH, and H load components (based on 1000 pair data for individual plots).

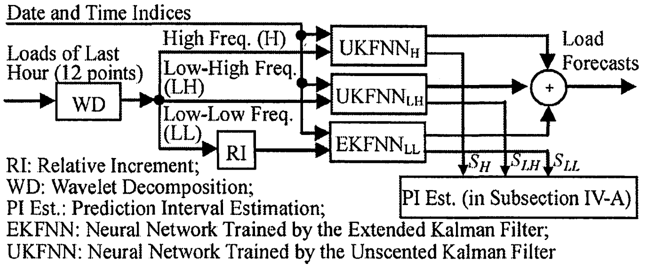
Date and Time Indices
High Freq. (H)
Loads of Last
Hour (12 points)
Low-High Freq.
(LH)
WD
Low-Low Freq.
(LL)
RI
UKFNN$_H$
UKFNN$_{LH}$
EKFNN$_{LL}$
+
Load
Forecasts
S_H
S_{LH}
S_{LL}
PI Est. (in Subsection IV-A)
RI: Relative Increment;
WD: Wavelet Decomposition;
PI Est.: Prediction Interval Estimation;
EKFNN: Neural Network Trained by the Extended Kalman Filter;
UKFNN: Neural Network Trained by the Unscented Kalman Filter

Fig. 3. Schematic of wavelet neural networks trained by hybrid Kalman filters (WNNHKF).

forecasting results. The result analysis again indicates that the LL component has the near-linear relationship between input and output measurement, whereas LH and H components separately have highly nonlinear relationships.

To forecast near-linear and highly nonlinear relationships for individual load components with accurate online PI estimates, the back propagation algorithm is replaced by Kalman-type filters for training WNN's weights. Generally, the back propagation is a first-order steepest decent method, whereas the Kalman filter is a second-order Newton method for recursive state estimation of linear dynamic systems, and is a minimum mean-square-error estimator. Through treating NN's weights as a slowly varying state and the (scaled) loads as the measurement, Kalman-type algorithms are adopted because they can produce a dynamic innovation covariance whose diagonal elements can be used for PI estimates. As shown in Fig. 3, the schematic of wavelet neural networks trained by hybrid Kalman filters is presented. To capture the near-linear relationship between the LL input and output measurement for an NN, an extended Kalman filter is used to train the neural network (EKFNN) in Section III-A, because EKF is derived through linearizing the system and is good for near-linear systems. To capture the highly nonlinear relationships for individual LH and H components, an unscented Kalman filter is used to train the neural network (UKFNN) in Section III-B, because UKF is good for highly nonlinear systems. Finally, results from these three NNs are added up to form forecasts. The overall variance will be derived and evaluated for PI estimates in Section IV.

A. EKFNN for the Low-Low Load Component

The key idea for forecasting the LL component is to use the EKFNN. The EKF trains the $\mathrm{NN_{LL}}$ by treating its weight $w(t)$ as a slowly varying state and the (scaled) load input as the measurement $z(t)$ following [30]–[32]. Training an NN can be described as a state estimation problem with state and measurement equations (the symbol LL is dropped in following equations for convenience):

$$w(t+1) = w(t) + \varepsilon(t) \tag{1}$$

$$z(t) = h(u(t), w(t)) + \nu(t) \tag{2}$$

where $w(t)$ is an $n_w \times 1$ weight vector trained by using a set of input-output measurement pairs of an NN $\{u(t), z(t), t = 1, \ldots, T\}$, the $u(t)$ is an $n_u \times 1$ input vector including loads of the last hour as well as the date and time indices following [1], the z(t) is a corresponding $n_z \times 1$ load measurement vector (n_z is equal to 12 indicates 5- to 60-min-ahead predictions), the variable T represents a forecasting horizon, and the $h(\cdot)$ represents an input-output function of an NN. Following the standard assumption for EKF, the $n_w \times 1$ process noise $\varepsilon(t)$ is assumed to be zero-mean white Gaussian with a positive covariance $Q(t)$, and the $n_z \times 1$ measurement noise $\nu(t)$ is assumed to be zero-mean white Gaussian with a positive covariance $R(t)$.

In EKF, the state and covariance propagations are implemented in time-update equations. After linearizing the underlying nonlinear system, the Bayesian rule is then implemented in measurement-update equations. Following the procedure of [33, pp. 200–210 and 382–385], key EKF steps are presented for completeness. The time-update equations are as follows:

$$\hat{w}(t+1 \,|\, t) = \hat{w}(t \,|\, t) \tag{3}$$

$$P(t+1 \,|\, t) = P(t \,|\, t) + Q(t) \tag{4}$$

$$\hat{z}(t+1 \,|\, t) = h(\hat{w}(t+1 \,|\, t), u(t)) \tag{5}$$

where the prior state (weight vector) $\hat{w}(t \,|\, t)$ and state covariance $P(t \,|\, t)$ are propagated to $\hat{w}(t+1 \,|\, t)$ and $P(t+1 \,|\, t)$, respectively. Here, the state transition matrix for the weight vector is an identity matrix. Next, the estimated weight $\hat{w}(t+1 \,|\, t)$ together with the input $u(t)$ are used to generate the prediction $\hat{z}(t+1 \,|\, t)$ which is treated as the $\hat{z}_{LL}(t+1 \,|\, t)$ for the LL component. Since the function $h(\cdot)$ is nonlinear, the Taylor series expansion is used to linearize the nonlinear system, and the $H(t+1)$ is calculated:

$$H(t+1) = (\partial h(u,w)/\partial w),\ given\ u = u(t)\ \&\ w = \hat{w}(t+1 \,|\, t). \tag{6}$$

Based on the Bayesian rule, the obtained function $H(t+1)$ is then used to produce the gain $K(t+1)$, the posterior weight $\hat{w}(t+1 \,|\, t+1)$, and the state covariance $P(t+1 \,|\, t+1)$. The measurement-update equations are as follows:

$$K(t+1) = P(t+1 \,|\, t) \cdot H(t+1)^T \cdot S(t+1)^{-1} \tag{7}$$

$$\hat{w}(t+1 \,|\, t+1) = \hat{w}(t+1 \,|\, t) + K(t+1) \cdot (z(t+1) - \hat{z}(t+1 \,|\, t)) \tag{8}$$

$$P(t+1 \,|\, t+1) = P(t+1 \,|\, t) - K(t+1) \cdot S(t+1) \cdot K(t+1)^T \tag{9}$$

$$S(t+1) = H(t+1) \cdot P(t+1 \,|\, t) \cdot H(t+1)^T + R(t+1) \tag{10}$$

where $S(t+1)$ is an $n_z \times n_z$ innovation covariance (the covariance of the load measurement) and treated as the $S_{LL}(t+1)$, to be used to derive PIs in Section IV-A.

The dynamic innovation covariance S is generally consistent with the covariance calculated based on the static historical errors. This is because the state covariance P converges to a steady-state covariance under the conditions of controllability and observability as presented in [33, pp. 211–212]. To justify these two conditions, take EKF as an example. The state transition matrix in (3) is an identity matrix, the process noise covariance Q in (4) is positive, and the measurement matrix H in (6) is believed to have a full rank given sufficient measurements. Therefore, it can be shown that the pair of state transition matrix and Cholesky factor of Q is completely controllable, and the pair of state transition matrix and H is completely observable. This yields the steady-state P and K, indicating that S is consistent with the static covariance. To demonstrate this, testing results in Example 2 of Section V show that the estimated standard deviation (derived from S) is close to the standard deviations of the sample errors. One advantage for PI estimates is that EKF can easily provide, as a by-product, an S for PI estimation. The second is that S is dynamic. Through linearizing the nonlinear system, the most recent error can be used to calculate S.

Using the EKF described above, the NN will be trained offline based on a set of input-output measurement pair data and then trained online (updated) when a new measurement is available. The EKF flowchart can be found in [33, p. 386]. For EKFNN, its load input and output are described below.

Following our previous WNN method in [1], the input LL component is transformed by using the relative increment transformation which is used to make the LL series stationary:

$$l_t^{RI} = (l_t - l_{t-1})/l_{t-1} \tag{11}$$

where l_t represents an LL load component at the time t, RI represents the relative increment transformation, and the l_t^{RI} is an element of load input vector $l_{RI}(t) = \{l_{t-n_z+1}^{RI}, \ldots, l_t^{RI}\}$. To satisfy NN's input requirement, $l_{RI}(t)$ has to be normalized:

$$u_{LL}(t) = \left(l_{RI}(t) - l_{RI}^{\min}\right) / \left(l_{RI}^{\max} - l_{RI}^{\min}\right) \tag{12}$$

where $u_{LL}(t)$ represents the normalized LL load input part at time t, and $l_{RI}^{\min}$ and $l_{RI}^{\max}$ are the minimum and maximum values of the relative increment in LL load, respectively.

After preparing NN inputs, the EKFNN performs forecasting. The forecasting output $\hat{z}_{LL}(t+1\,|\,t)$ has to be de-normalized:

$$\hat{z}^{d'}(t+1\,|\,t) = \hat{z}_{LL}(t+1\,|\,t)\cdot\left(l_{RI}^{\max} - l_{RI}^{\min}\right) + l_{RI}^{\min} \tag{13}$$

where $\hat{z}^{d}(t+1\,|\,t)$ is a de-normalized output vector and has to be inverse-transformed with respect to the relative increment transformation in an element-wise manner. For convenience, the conditioned variable t in $\hat{z}^{d}(t+1\,|\,t)$ is dropped for all the individual elements in $\{\hat{z}_{t+1}^{d},\ldots,\hat{z}_{t+n_z}^{d}\}$:

$$\hat{l}_{t+1} = \left[\hat{z}_{t+1}^{d}+1\right]\cdot l_t \tag{14a}$$

$$\hat{l}_{t+2} = \left[\hat{z}_{t+2}^{d}+1\right]\cdot \hat{l}_{t+1} = \left[\hat{z}_{t+2}^{d}+1\right]\cdot\left[\hat{z}_{t+1}^{d}+1\right]\cdot l_t,\ldots \tag{14b}$$

$$\hat{l}_{t+n_z} = \left[\hat{z}_{t+n_z}^{d}+1\right]\cdot \hat{l}_{t+n_z-1} \tag{14c}$$

where $\hat{L}(t+1\,|\,t) = \{\hat{l}_{t+1},\ldots,\hat{l}_{t+n_z}\}^T$ is the LL load prediction.

B. UKFNN for the Low-High and High Load Components

When the relationship between input and output measurement for an NN is highly nonlinear, EKF performance could be poor because the mean and covariance are propagated by linearizing an underlying nonlinear model. The key idea for forecasting LH and H frequency components is to use the UKFNN. The UKF uses an unscented transform to generate a minimal set of sample points, called sigma points, around the mean. These sigma points are then propagated through nonlinear functions. The mean and covariance of estimates are then recovered through weighting. Because the set of sigma points are symmetrically selected, the odd central moments are zero. If the distribution for the state is multiple dimensional Gaussian, the first three moments are the same as the original moments [34]. Therefore, UKF predicts the mean more accurately than EKF, and it predicts the covariance at least as accurately as EKF. It also avoids the need to calculate the Jacobian functions.

Similar to the EKF described in Section III-A, the UKF also adopts the time-update and measurement-update equations. Rather than using the Taylor series expansion to calculate the H matrix of EKF, a set of sigma points are generated, propagated through the function, and then weighted to produce predictions with variance estimates. Following the procedure of [34], key steps of UKF are presented below for completeness. The time-update equations are the same as (3)–(4), where the prior state (weight vector) $\hat{w}(t\,|\,t)$ and covariance $P(t+1\,|\,t)$ are propagated to $\hat{w}(t+1\,|\,t)$ and $P(t+1\,|\,t)$, respectively. The propagations are then performed to generate a set of $2n_w+1$ sigma points χ:

$$\begin{aligned}\chi_0(t+1\,|\,t) &= \hat{w}(t+1\,|\,t)\\ \chi_i(t+1\,|\,t) &= \hat{w}(t+1\,|\,t)\\ &\quad+\left(\sqrt{(n_w+\lambda)\cdot P(t+1\,|\,t)}\right)_i,\quad i=1,\ldots,n_w\\ \chi_i(t+1\,|\,t) &= \hat{w}(t+1\,|\,t)\\ &\quad-\left(\sqrt{(n_w+\lambda)\cdot P(t+1\,|\,t)}\right)_{i-n_w}\\ &\qquad i=n_w+1,\ldots,2n_w \end{aligned} \tag{15}$$

where n_w is the number of NN weights, λ is a scaling parameter, and $(\sqrt{(n_w+\lambda)\cdot P(t+1\,|\,t)})_i$ is the ith column of the square root of the matrix $(n_w+\lambda)\cdot P(t+1\,|\,t)$. Through the nonlinear function $h(\,\cdot\,)$, the χ points are projected to γ points which are then weighted to produce the NN output:

$$\gamma_i(t+1) = h(\chi_i(t+1\,|\,t),u(t)), i=0,\ldots,2n_w \tag{16}$$

$$\hat{z}(t+1\,|\,t) = \sum_{i=0}^{2N} W_i\cdot\gamma_i(t+1) \tag{17}$$

where W_{i} is the weight for the ith γ point, and its definition and default value can be found in [35, (15)], and $\hat{z}(t+1\,|\,t)$ is the UKFNN's prediction which is treated as $\hat{z}_H(t+1\,|\,t)$ for the H component, and $\hat{z}_{LH}(t+1\,|\,t)$ for LH.

Similar to the steps for EKF, the UKFNN prediction $\hat{z}(t+1\,|\,t)$ together with χ and γ points are used to calculate the posterior weight state and covariance based on the Bayesian rule. The measurement-update equations are as follows:

$$\begin{aligned}K(t+1) = \Bigg\{&\sum_{i=0}^{2n_w} W_i\cdot[\chi_i(t+1\,|\,t)-\hat{w}(t+1\,|\,t)]\\ &\cdot[\gamma_i(t+1)-\hat{z}(t+1\,|\,t)]\Bigg\}\cdot S(t+1)^{-1}\end{aligned} \tag{18}$$

$$\begin{aligned}S(t+1) = &\sum_{i=0}^{2n_w} W_i\cdot[\gamma_i(t+1)-\hat{z}(t+1\,|\,t)]\\ &\cdot[\gamma_i(t+1)-\hat{z}(t+1\,|\,t)]+R(t+1)\end{aligned} \tag{19}$$

where the posterior weight $\hat{w}(t+1\,|\,t+1)$ and the state covariance $P(t+1\,|\,t+1)$ are as same as (8)–(9). The $n_z\times n_z$ innovation covariance $S(t+1)$ is treated as the $S_H(t+1)$ for the H component, and $S_{LH}(t+1)$ for the LH component. They will be used for PI estimates in Section IV-A.

Following our WNN method in [1], the H input is normalized without applying the relative increment transformation:

$$u_H(t) = \left(h_H(t)-h_H^{\min}\right)/\left(\left(h_H^{\max}-h_H^{\min}\right)\right) \tag{20}$$

where $u_H(t)$ is the normalized load component input part at time t, $h_H(t)$ represents the H load component at time t, and $h_H^{\min}$ and $h_H^{\max}$ are the minimum and maximum values of the H component series, respectively.

After input preparation, UKFNN performs the prediction which has to be de-normalized:

$$\hat{h}_H(t+1) = \hat{z}_H(t+1\,|\,t)\cdot\left(h_H^{\max}-h_H^{\min}\right)+h_H^{\min}. \tag{21}$$

Similar to the H component, the prediction $\hat{h}_{LH}(t+1)$ can be obtained for the LH component.

IV. Prediction Interval Estimation and Evaluation

To estimate prediction intervals online for VSTLF, the overall variance estimate is derived in Section IV-A. As shown in Fig. 4, the key idea is to use an overall variance estimate obtained by adding up three estimates from EKFNN$_{\mathrm{LL}}$, UKFNN$_{\mathrm{LH}}$, and UKFNN$_{\mathrm{H}}$. This is because these components are orthogonal based on the wavelet theory. To obtain individual variance estimates, the diagonal elements of the innovation covariance for H, LH, and LL components should be de-normalized individually. The de-normalized estimate for LL is further approximated due to the relative increment transformation. To assess the PI estimates, In Section IV-B, the Kolmogorov-Smirmov

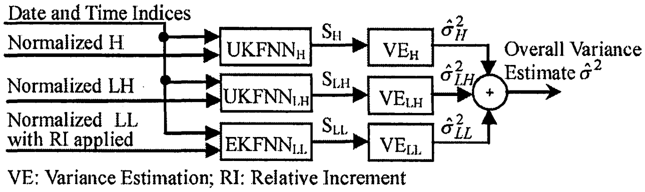

Fig. 4. Schematic of the prediction interval estimation.

test and Quantile-Quantile plot show that the forecasting errors have heavier tails than a Gaussian distribution. Based on this, the estimated PIs are thoroughly evaluated.

A. Prediction Interval Estimation

To obtain an overall variance estimate, three variance estimates derived from individual NNs are added together:

$$\hat{\sigma}^2(t+1) = \hat{\sigma}_H^2(t+1) + \hat{\sigma}_{LH}^2(t+1) + \hat{\sigma}_{LL}^2(t+1) \quad (22)$$

where $\hat{\sigma}^2(t+1)$ is the overall variance estimate used for online PI estimates, and $\hat{\sigma}_H^2(t+1), \hat{\sigma}_{LH}^2(t+1)$, and $\hat{\sigma}_{LL}^2(t+1)$ are the individual variance estimates calculated based on $S_H(t+1)$, $S_{LH}(t+1)$, and $S_{LL}(t+1)$, respectively. To obtain the variance estimates for H and LH components, diagonal elements of $S_H(t+1)$ and $S_{LH}(t+1)$ should be de-normalized:

$$\begin{aligned}\hat{\sigma}_H^2(t+1) &= \left(h_H^{\max} - h_H^{\min}\right)^2 \cdot diag(S_H(t+1))\\ \hat{\sigma}_{LH}^2(t+1) &= \left(h_{LH}^{\max} - h_{LH}^{\min}\right)^2 \cdot diag(S_{LH}(t+1)). \quad (23)\end{aligned}$$

Similarly, the diagonal ones of $S_{LL}(t+1)$ are de-normalized:

$$\hat{\sigma}_{LL}^{d2}(t+1) = \left(l_{RI}^{\max} - l_{RI}^{\min}\right) \cdot diag(S_{LL}(t+1)) \quad (24)$$

where $\hat{\sigma}_{LL}^{d2}(t+1)$ is a de-normalized variance estimate with elements $\{\hat{\sigma}_{t+1}^{d2}, \ldots, \hat{\sigma}_{t+n_z}^{d2}\}$. For convenience, the symbol LL is omitted for individual elements here as well as in the following equations. This de-normalized variance estimate then has to be further processed because the relative increment transformation is applied to the LL load input. Since the transformation is nonlinear, the derivation is difficult in view of the complicated cross-correlations for individual elements of $z^d(t+1\,|\,t)$.

The key idea for deriving the LL variance is to ignore the cross-correlations. This is because numerical testing shows that cross-correlations of the dependent elements $\{z_{t+1}^d, \ldots, z_{t+n_z}^d\}$ in the vector $z^d(t+1\,|\,t)$ have values at 10^{-8}, whereas individual variances have values at 10^{-6}. The variance estimate is then approximated in an element-wise manner. Following (14a), the estimate $\hat{\sigma}_{t+1}^2$ for l_{t+1} is derived:

$$\hat{\sigma}_{t+1}^2 = Var\left[\left(z_{t+1}^d + 1\right) \cdot l_t\right] = \sigma_{t+1}^{d2} \cdot l_t^2. \quad (25)$$

Following (14b), the $\hat{l}_{t+2}$ is calculated based on $\hat{l}_{t+1}$. By omitting their covariance, the estimate $\hat{\sigma}_{t+2}^2$ is approximated:

$$\begin{aligned}\hat{\sigma}_{t+2}^2 &= Var\left[\left(z_{t+1}^d + 1\right) \cdot \left(z_{t+2}^d + 1\right) \cdot l_t\right]\\ &\approx \left[Var\left(z_{t+1}^d\right) + Var\left(z_{t+2}^d\right) + Var\left(z_{t+1}^d \cdot z_{t+2}^d\right)\right] \cdot l_t^2. \quad (26)\end{aligned}$$

In the equation above, the numerical testing shows that elements z_{t+1}^{d2} and σ_{t+2}^{d2} have values at 10^{-4} and 10^{-6}, respectively. Since the term $\sigma_{t+1}^{d2} \cdot \sigma_{t+2}^{d2}$ is relatively small, it is ignored. The estimate is further approximated:

$$\begin{aligned}\hat{\sigma}_{t+2}^2 \approx& \left\{\sigma_{t+1}^{d2} + \sigma_{t+2}^{d2} + E\left[\left(z_{t+1}^d \cdot z_{t+2}^d\right)^2\right]\right.\\ &\left. - \left[E\left(z_{t+1}^d \cdot z_{t+2}^d\right)\right]^2\right\} \cdot l_t^2\\ =& \left\{\sigma_{t+1}^{d2} + \sigma_{t+2}^{d2} + E\left[z_{t+1}^{d2}\right] \cdot E\left[z_{t+2}^{d2}\right] - \left[E\left(z_{t+1}^d\right)\right.\right.\\ &\left.\left. \cdot E\left(z_{t+2}^d\right)\right]^2\right\} \cdot l_t^2\\ =& \left\{\sigma_{t+1}^{d2} + \sigma_{t+2}^{d2} + \left(\hat{z}_{t+1}^{d2} + \sigma_{t+1}^{d2}\right) \cdot \left(\hat{z}_{t+2}^{d2} + \sigma_{t+2}^{d2}\right)\right.\\ &\left. - \left(\hat{z}_{t+1}^{d2} \cdot \hat{z}_{t+2}^{d2}\right)\right\} \cdot l_t^2\\ =& \left[\sigma_{t+1}^{d2} + \sigma_{t+2}^{d2} + \hat{z}_{t+1}^{d2} \cdot \sigma_{t+2}^{d2}\right.\\ &\left. + \hat{z}_{t+2}^{d2} \cdot \sigma_{t+1}^{d2} + \sigma_{t+1}^{d2} \cdot \sigma_{t+2}^{d2}\right] \cdot l_t^2. \quad (27)\end{aligned}$$

In the second equality above, $\hat{z}_{t+1}^d = E[z_{t+1}^d]$ and $\hat{z}_{t+2}^d = E[z_{t+2}^d]$ are based on [33, p. 203]:

$$\hat{z}^d(t+1\,|\,t) = E[z^d(t+1)\,|\,Z^t] \quad (28)$$

where $\hat{z}^d(t+1\,|\,t)$ has elements $\{\hat{z}_{t+1}^d, \hat{z}_{t+2}^d, \ldots, \hat{z}_{t+n_z}^d\}$, $z^d(t+1\,|\,Z^t)$ has elements $\{z_{t+1}^d, z_{t+2}^d, \ldots, z_{t+n_z}^d\}$, and Z^t represents the past observations up to t. This is because under the Markov assumption, the predicted measurement given the immediately previous one is conditionally independent of the other earlier measurements.

To estimate other variances, i.e., $\hat{\sigma}_{t+3}^2, \ldots, \hat{\sigma}_{t+n_w}^2$, the process will be repeated until the last element is calculated. Finally, a general equation is obtained:

$$\begin{aligned}\hat{\sigma}_{t+J}^2 \approx& \sum_{j=1}^{J}\left\{\left(1 + \sum_{i=1}^{J} \hat{z}_{t+i}^{d2} - \hat{z}_{t+j}^{d2}\right) \cdot \sigma_{t+j}^{d2}\right\} \cdot l_t^2\\ =& \sum_{j=1}^{J}\left\{\left(1 + \sum_{i=1}^{J}\left(\left(l_{RI}^{\max} - l_{RI}^{\min}\right) \cdot \hat{z}_{t+i} + l_{RI}^{\min}\right)^2\right.\right.\\ &\left. - \left(\left(l_{RI}^{\max} - l_{RI}^{\min}\right) \cdot \hat{z}_{t+j} + l_{RI}^{\min}\right)^2\right) \cdot \left(l_{RI}^{\max} - l_{RI}^{\min}\right)^2\\ &\left. \cdot diag\left(S_{LL}(t+1)\right)_{t+j}\right\} \cdot l_t^2, \quad J = 1, \ldots, n_z \quad (29)\end{aligned}$$

where $\hat{\sigma}_{LL}^2(t+1) = \left\{\hat{\sigma}_{t+1}^2, \cdots, \hat{\sigma}_{t+n_z}^2\right\}^T$ is an approximated variance estimate vector for LL load component.

B. Evaluation of Prediction Interval Estimates

To help evaluate PI estimates, the distribution of forecasting errors for individual 5- to 60-min outs is analyzed. The Kolmogorov-Smirmov test and Quantile-Quantile plot of the errors show that the errors have heavier tails than a Gaussian distribution. However, the Kolmogorov-Smirmov test shows that after removing the bottom and top tails of the errors (e.g., 5-min errors that are either below the 0.7th percentile or above the 99.3th percentile), the remaining errors follow a zero mean Gaussian distribution. This test is performed in two ways. First, the remaining errors are standardized without centering, and the empirical distribution of the resulting values is compared with a standard Gaussian distribution. Second, the empirical distribution of the remaining errors is compared with that of simulated data sampled from a Gaussian distribution with zero mean and the same standard deviation. Numerical details and results of these two ways of the test are given in Case 3 of Example 2 in Section V, demonstrating near Gaussian distribution of the forecasting errors except for heavy tails.

Based on the above analysis, the PI estimates are then evaluated in three ways. First, the estimated standard deviations for 5- to 60-min errors are compared with the sample ones, respectively. Second, the one sigma coverage rates based on the estimated standard deviations are compared with 68%, i.e., one sigma coverage rate of the standard Gaussian distribution. Third, for each of the coverage rates 10%, 20%,. . ., 90%, we calculate how many estimated standard deviations are needed to achieve the coverage rate for the errors, and then compare the result with how many standard deviations are needed to achieve the same rate for a Gaussian random variable. As shown from the numerical results in Case 3, the comparisons indicate that the PI estimates are reasonably accurate and conservative.

V. Numerical Testing Results

The method was implemented in MATLAB. The open source code and the part of the test data and results are open, and can be obtained from http://github.com/ldmbouge/vstlf. For this section, the software was run on a server with dual Xeon quad core Intel E5620 2.4-GHz processors and a 36-GB memory. The performance measures include mean absolute error (MAE), mean average percentage error (MAPE), standard deviation of sample errors (SD), estimated standard deviation (ESD) which is the square root of the variance estimate derived in Section IV-A, and one sigma coverage.

Two examples are presented to demonstrate our method. Example 1 uses a classroom-type problem to compare the WNNHKF to the methods of persistence, linear AR, single NN, and WNN so that our method can be verified in a simple way. Example 2 shows the values of $\mathrm{EKFNN_{LL}}$ for capturing the near-linear relationship between the LL input and output measurement, as well as $\mathrm{UKFNN_{LH}}$ and $\mathrm{UKFNN_{H}}$ for capturing highly nonlinear relationships. This example also demonstrates the accuracy of the derived PI estimates. In both examples, the training, validation, and test processes in a three-way data split are used to determine the parameters in WNNHKF. All NNs (trained by Kalman filters) are trained off-line by using training data with weights randomly initialized, and the training terminates when a fixed number of iterations are reached.

Example 1: Consider the signal:

$$\mathrm{y(t)} = 200\sin(2\pi 10\mathrm{t}/f_s) + 10\sin(2\pi 110\mathrm{t}/f_s) + \sin(2\pi 250\mathrm{t}/f_s) \quad (30)$$

where the sample rate f_s equals 1000, $y(t)$ is composed of a low frequency component $200sin(2\pi 10t/f_s)$, a medium component $10sin(2\pi 110t/f_s)$, and a high component $sin(2\pi 250t/f_s)$. This signal is similar to the actual load in terms of the relative amplitude and frequency. A total of 3600 noisy data points $(t, \tilde{y}(t))$ are randomly generated:

$$\tilde{\mathrm{y}}(\mathrm{t}) = \mathrm{y(t)} + \varepsilon(\mathrm{t}) \quad (31)$$

TABLE I
Number of Hidden Neurons, Averaged MAEs, and Averaged SDs Comparing the Results of WNNHKF to the Results of Persistence, Linear AR, Single NN, and WNN

	Persistence	Linear AR	Single NN	WNN	WNNHKF
No. of Neurons			16	15, 10, & 10 for LL, LH, & H	13, 10, & 10 for LL, LH, & H
Ave. MAE	51.58	2.30	2.06	1.68	1.46
Ave. SD	58.14	2.89	2.68	2.13	1.83

where $t \in [1, \ldots, 3600]$, and $\{\varepsilon(t)\}$ are independent and identically distributed Gaussian noises with zero mean and unit variance $N(0,1)$. The first one-third of data points are used for training, the second one-third of data points for validation, and the last one-third of data points for test.

The WNNHKF is compared to the methods of persistence, linear AR, single NN without wavelet decomposition, and WNN. For all the methods, the relative increment transformation is not used for this example because $y(t)$ consists of three periodical sine functions, and there is no need to use the transformation to make $\{y(t)\}$ stationary. As shown in Table I, the numbers of hidden neurons of NNs are separately given, and these numbers are determined based on training, validation, and test processes in the three-way data split. To evaluate the accuracy, MAEs and SDs are calculated for 1- to 12-step-ahead predictions, and then they are separately averaged. The averaged MAE and averaged SD in Table I indicate that our method is better than the single NN. These results also indicate that the WNNHKF improves the WNN. For this example, MAPE is not used since $\{y(t)\}$ may have zero values.

Example 2: Wavelet neural networks trained by hybrid Kalman filters are tested with ISO-NE's data. The training period is from January 1, 2007 to December 31, 2007, the validation is from January 1, 2008 to June 30, 2008, and the test is from July 1, 2008 to December 31, 2008. Five cases are presented. Cases 1–2 are for training and validation: Case 1 for the combination of $\mathrm{EKFNN_{LL}}$ and $\mathrm{UKFNN_{LH,H}}$ when compared to other combinations; and Case 2 for predictions with PIs. Cases 3–4 are for test: Case 3 for test results and PI evaluation; Case 4 for comparing the results of WNNHKF to the results of persistence, linear AR, ISO-NE's method, and WNN.

Case 1: The combination of EKFNN and UKFNN are examined with ISO-NE's load data. There are totally eight combinations of using EKFNN and UKFNNN for predicting three load components. To identify different strategies, the symbols LL, LH, and H are marked in subscripts. The validation results from 5- to 60-min-ahead predictions in Table II show that the combination of $\mathrm{EKFNN_{LL}}$ and $\mathrm{UKFNN_{LH,H}}$ produces the overall smallest MAPEs and SDs when compared to other seven strategies. This also supports the analysis in the beginning of Section III that the LL component has a near-linear relationship between input and output measurement, whereas LH and H components have highly nonlinear relationships. Here, the combination of EKFNN $_{\mathrm{LL}}$, and UKFNN $_{\mathrm{H,LH}}$ are treated as a nominal one and will be used for the rest of the testing.

Case 2: The MAPEs, MAEs, SDs, ESDs, and one sigma coverage values as shown in Table III are calculated based on the

TABLE II
MAPEs (%) AND SDs (MW) FOR DIFFERENT COMBINATIONS OF NNs TRAINED BY KALMAN FILTER(S) FOR INDIVIDUAL LOAD COMPONENTS

	EKFNN$_{H,LH,LL}$		UKFNN$_{H,LH,LL}$		EKFNN$_{H, LH}$ UKFNN$_{LL}$		EKFNN$_{H, LL}$ UKFNN$_{LH}$	
Min	MAPE	SD	MAPE	SD	MAPE	SD	MAPE	SD
5	0.12	24.45	0.12	24.84	0.13	23.73	0.12	24.02
10	0.17	36.31	0.18	37.52	0.18	38.00	0.17	36.37
15	0.22	45.35	0.23	47.17	0.23	47.36	0.22	45.18
20	0.26	54.33	0.27	57.28	0.27	57.08	0.26	54.63
25	0.29	62.76	0.31	65.94	0.31	66.17	0.29	62.84
30	0.34	72.29	0.36	76.50	0.36	77.03	0.34	71.88
35	0.36	80.06	0.39	84.00	0.39	84.14	0.36	79.82
40	0.41	90.39	0.43	94.65	0.43	94.76	0.41	90.59
45	0.44	98.63	0.47	103.93	0.47	103.95	0.44	98.65
50	0.48	108.25	0.51	114.15	0.51	114.35	0.48	108.13
55	0.51	114.81	0.54	120.84	0.54	120.94	0.51	114.38
60	0.55	124.15	0.59	130.37	0.59	130.80	0.55	123.81
	EKFNN$_{LH, LL}$ UKFNN$_{H}$		EKFNN$_{H}$ UKFNN$_{LH, LL}$		EKFNN$_{LH}$ UKFNN$_{H, LL}$		EKFNN$_{LL}$ UKFNN$_{H, LH}$	
Min	MAPE	SD	MAPE	SD	MAPE	SD	MAPE	SD
5	0.12	24.11	0.13	25.37	0.12	25.37	0.12	23.52
10	0.17	36.04	0.18	37.89	0.18	37.89	0.17	35.84
15	0.21	44.98	0.23	47.32	0.23	47.03	0.21	44.99
20	0.26	54.29	0.27	57.23	0.27	56.98	0.26	54.74
25	0.29	62.24	0.31	66.30	0.31	65.79	0.29	62.35
30	0.34	72.30	0.36	76.61	0.36	76.96	0.33	71.84
35	0.36	80.11	0.39	83.97	0.39	84.21	0.36	79.81
40	0.40	90.21	0.43	94.93	0.43	94.50	0.40	90.39
45	0.44	98.62	0.47	103.96	0.47	103.93	0.44	98.63
50	0.48	108.09	0.51	114.27	0.51	114.22	0.48	107.98
55	0.51	114.99	0.54	120.59	0.55	121.18	0.51	114.58
60	0.55	124.02	0.59	130.42	0.59	130.81	0.55	123.61

TABLE III
MAPEs (%), MAEs (MW), SDs (MW), ESDs (MW), AND ONE SIGMA COVERAGE (%) FOR WNNHKF METHOD (BASED ON VALIDATION DATA SET)

Min.	MAPE	MAE	SD	ESD	ONE SIGMA COVERAGE
5	0.12	17.22	23.52	22.79	74.27
10	0.17	25.48	35.84	35.60	77.52
15	0.21	31.64	44.99	49.29	81.16
20	0.26	37.70	54.74	55.62	79.30
25	0.29	42.98	62.35	61.19	77.93
30	0.33	50.12	71.84	75.70	80.75
35	0.36	54.16	79.81	81.43	80.84
40	0.40	60.38	90.39	97.32	82.78
45	0.44	65.98	98.63	102.60	81.85
50	0.48	72.12	107.98	107.60	80.75
55	0.51	76.52	114.58	112.53	80.40
60	0.55	82.76	123.61	130.08	82.33

TABLE IV
MAPEs (%), MAEs (MW), SDs (MW), ESDs (MW), AND ONE SIGMA COVERAGE (%) FOR WNNHKF METHOD (BASED ON TEST DATA SET)

Min.	MAPE	MAE	SD	ESD	ONE SIGMA COVERAGE
5	0.13	19.39	27.07	25.68	73.94
10	0.18	27.33	38.06	38.28	76.00
15	0.22	32.86	45.43	51.60	80.41
20	0.26	39.01	54.24	57.40	78.10
25	0.30	44.87	62.45	62.04	75.79
30	0.34	50.97	71.40	76.11	78.74
35	0.38	56.93	80.25	81.14	77.33
40	0.42	63.30	89.56	96.58	80.23
45	0.46	69.01	98.58	100.99	78.67
50	0.50	75.46	108.52	105.52	77.49
55	0.54	81.09	116.56	110.29	76.11
60	0.58	87.43	125.93	128.36	79.62

validation data set. The first four measures gradually increase from 5- to 60-min-ahead forecasting results because the uncertainty expands as the forecasting step increases. Based on the observation, ESDs have values from 22 MW to 131 MW, and ISO-NE's system load data have values around 15 000 MW. Since ESD values are much smaller than the system load magnitude, lower and upper bounds are always positive. For the case when the errors are not symmetric around estimates near zero, the bound can be truncated to a zero value if it is negative. Similar treatment can be found in Fig. 2 of [26]. For the case when forecasted values are out-of-range, the load prediction after de-normalization can be clipped into a zero value if the prediction is negative or a historical maximum if the prediction is very high. The observation also shows that ESDs are very close to SDs. This corresponds to the analysis in Section III-A that the dynamic innovation covariance is generally consistent with the covariance calculated based on static historical errors. Based on ESDs and predictions, the one sigma coverage values are calculated. Due to the heavy tails of errors (most of the large errors are related to the load predictions during peak hours), the coverage values for 5- to 60-min-ahead predictions have a range from 74% to 83% which are larger than the one sigma coverage rate of 68% under a Gaussian distribution. This indicates that PI estimates are reasonably accurate and conservative. The use of the Gaussian distribution is to be explained in Case 3.

Cases 1–2 above are for training and validation data sets, and the following Cases 3–4 are for the test data set.

Case 3: The five measures of the test data set in Table IV are very close to the measures of the validation data set in Table III. This indicates that WNNHKF parameters are properly selected. All the measures quantify forecasting accuracy in certain way, with the last two directly related to PIs. To further assess PI estimates, our standard-deviation-based PIs are evaluated and then compared to the empirical quantile-based PIs as follows.

1) Evaluation of Standard-Deviation-Based PIs: As discussed in Section IV-B, the 5- to 60-min-ahead forecasting errors have heavier tails than a Gaussian distribution. Take 5-min errors from July to December 2008 for example. The Quantile-Quantile plot of the errors in Fig. 5 clearly shows heavier tails than the Gaussian. After removing the tails below the 0.7th percentile or above the 99.3th percentile of the errors, the p-values of the Kolmogorov-Smirnov test, conducted in the two ways as described in Section IV-B, are both insignificant (> 0.1). This indicates that the remaining errors have a zero mean Gaussian distribution. Furthermore, the ESD based on the entire sample of errors is close to the SD (in columns 4 and 5 of Tables III and IV). The ESD leads to an actual coverage rate of 74%, which is slightly larger than the one sigma coverage

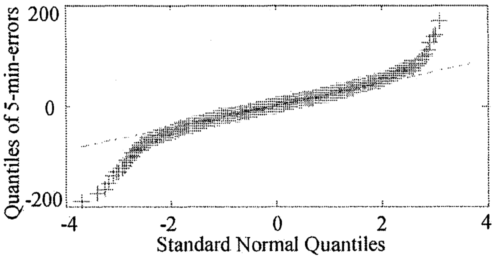

Fig. 5. Quantile-Quantile plot of the 5-min-ahead forecasting errors versus the standard normal.

TABLE V
TOTAL PROBABILITY MASS (%) OF TAILS OF ERROR REMOVED TO MAKE KOLMOGOROV-SMIRNOV TEST INSIGNIFICANT ($p > 0.1$)

Min.	5	10	15	20	25	30
Total Probability Mass of Error Tails	1.40	2.80	5.78	4.16	5.16	5.56
Min.	35	40	45	50	55	60
Total Probability Mass of Error Tails	5.78	3.80	5.10	6.82	5.74	5.82

rate of 68% under a Gaussian distribution. Therefore, the distribution of the 5-min errors has heavier tails than a Gaussian distribution, but the total probability mass of the tails is very small (1.4%). Similarly, to make the Kolmogorov-Smirnov test insignificant for each of the other look-ahead times, as shown in Table V, the total probability mass of tails is calculated as the fraction of errors that have been removed. Finally, the same conclusion is also made for 10- to 60-min forecasting results.

In view of the above distribution analysis, to evaluate PI estimates, three comparisons are conducted. First, as shown in columns 4 and 5 of Table IV, the ESDs are quite close to the SDs for 5- to 60-min outs. Second, as shown in column 6 of Table IV, the one sigma coverage rates for 5- to 60-min-ahead predictions range from 73% to 80% which are larger than 68% under the standard Gaussian distribution. Third, consider WNNHKF 5-min outs from July to December 2008 for example. To achieve the 90% coverage rate, the amount of the ESD is found to be 1.52, which is slightly smaller than 1.64 under the standard Gaussian distribution. The last two comparisons indicate the ESD is conservative. The same conclusion is also made for 10- to 60-min outs and for different coverage rates, i.e., 10%, 20%,. . ., 90%, as shown in Table VI. To further illustrate the conclusion, we graph the amount of ESDs as a function of coverage rates ranging from 10% to 90% for each look-ahead time, and compare it to the amount of sigmas under the standard Gaussian graphed in the same way. As shown in Fig. 6, the curve for the ESD is always slightly below the curve for the standard Gaussian, indicating conservative PIs. Based on these, it can be concluded that the PI estimates for coverage rates up to 90% are reasonably accurate and conservative.

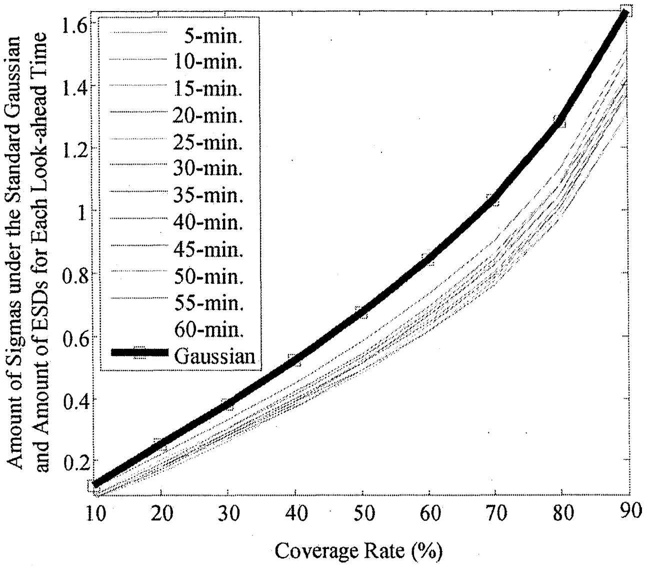

Fig. 6. Amount of ESDs as a function of coverage rates ranging from 10% to 90% for each look-ahead time when compared with the amount of sigmas under the standard Gaussian.

TABLE VI
AMOUNT OF ESD TO ACHIEVE ALMOST THE SAME COVERAGE RATES

	COVERAGE RATE								
Min.	10%	20%	30%	40%	50%	60%	70%	80%	90%
5	0.11	0.22	0.33	0.45	0.58	0.73	0.90	1.13	1.52
10	0.09	0.19	0.30	0.42	0.54	0.69	0.86	1.08	1.42
15	0.08	0.16	0.26	0.37	0.48	0.61	0.77	0.97	1.30
20	0.08	0.18	0.29	0.39	0.51	0.64	0.80	1.02	1.39
25	0.09	0.20	0.30	0.41	0.53	0.68	0.86	1.09	1.49
30	0.08	0.17	0.28	0.38	0.49	0.63	0.79	1.01	1.37
35	0.09	0.18	0.29	0.40	0.51	0.66	0.82	1.05	1.42
40	0.09	0.18	0.27	0.37	0.49	0.61	0.76	0.97	1.31
45	0.09	0.18	0.28	0.39	0.49	0.63	0.78	1.01	1.38
50	0.09	0.19	0.29	0.39	0.52	0.65	0.83	1.05	1.44
55	0.09	0.19	0.30	0.41	0.53	0.67	0.84	1.08	1.49
60	0.09	0.18	0.27	0.38	0.49	0.63	0.78	0.99	1.38
	COVERAGE RATE								
Min.	91%	92%	93%	94%	95%	96%	97%	98%	99%
5	1.57	1.63	1.70	1.77	1.86	1.94	2.12	2.25	2.45
10	1.48	1.55	1.62	1.70	1.80	1.89	2.03	2.25	2.49
15	1.35	1.39	1.47	1.56	1.65	1.73	1.85	1.99	2.20
20	1.44	1.51	1.59	1.66	1.75	1.84	1.97	2.17	2.38
25	1.54	1.60	1.67	1.73	1.84	1.98	2.13	2.34	2.62
30	1.42	1.49	1.57	1.65	1.72	1.82	1.98	2.12	2.41
35	1.48	1.56	1.64	1.73	1.81	1.92	2.11	2.30	2.59
40	1.35	1.42	1.50	1.59	1.68	1.83	2.04	2.25	2.45
45	1.43	1.49	1.56	1.68	1.78	1.92	2.16	2.38	2.64
50	1.51	1.58	1.68	1.79	1.89	2.02	2.24	2.46	2.82
55	1.55	1.63	1.70	1.77	1.94	2.09	2.30	2.53	2.89
60	1.44	1.51	1.59	1.68	1.78	1.90	2.07	2.37	2.68

To explore further, the PI estimates for coverage rates above 90% are investigated. We have seen from Fig. 5 that forecasting errors can significantly deviate from the Gaussian distribution as they become more extreme. To attain very high coverage rates, a large number of extreme errors have to be accounted for. To assess the effects of these non-Gaussian extreme errors, curves similar to those in Fig. 6 are graphed, but with coverage rates ranging from 91% to 99%, as shown in Table VI. Fig. 7 shows that for coverage rates up to 95%, the amounts of ESDs for 5- to 60-min outs are slightly lower than those derived from the Gaussian distribution, indicating the PI estimates are still accurate and conservative. The result is also consistent with the observation from Table V that the total probability mass of the tails ranges from 1.40% to 6.82% for 5- to

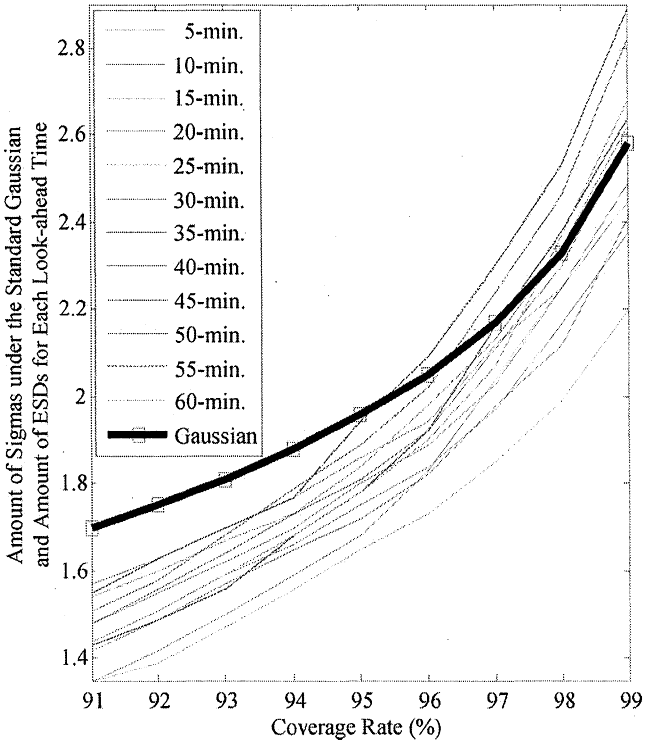

Fig. 7. Amount of ESDs as a function of coverage rates ranging from 91% to 99% for each look-ahead time when compared with the amount of sigmas under the standard Gaussian.

TABLE VII
ACTUAL COVERAGE RATES (%) OF EMPIRICAL QUANTILE-BASED PIs FOR DIFFERENT NOMINAL COVERAGE RATES

	NOMINAL COVERAGE RATE								
Min.	10%	20%	30%	40%	50%	60%	70%	80%	90%
5	10.77	18.43	27.19	38.22	48.05	57.80	67.56	78.73	90.44
10	9.56	19.52	31.36	40.65	49.80	59.62	69.04	77.93	90.44
15	11.17	19.78	29.74	37.95	47.38	57.74	66.89	77.52	88.96
20	10.23	20.86	31.22	40.78	48.32	58.82	67.43	77.52	88.69
25	11.04	21.27	30.82	39.70	49.80	57.34	66.49	75.64	87.21
30	10.23	20.73	30.82	40.65	48.99	58.28	67.97	77.66	88.29
35	9.69	19.78	29.74	39.17	47.78	57.60	68.37	76.72	87.35
40	8.34	18.57	30.01	38.22	47.78	57.87	65.95	76.04	86.94
45	9.42	20.18	29.61	38.63	48.45	57.60	67.03	76.58	87.08
50	10.36	18.44	29.21	39.03	47.91	56.39	66.76	76.58	86.68
55	9.69	19.11	27.73	36.88	48.32	57.74	67.29	76.31	86.94
60	8.88	18.44	26.92	36.47	46.84	56.39	67.03	75.91	87.08

TABLE VIII
WIDTHS (MW) OF EMPIRICAL QUANTILE-BASED PIs FOR DIFFERENT NOMINAL COVERAGE RATES AS SHOWN IN TABLE VII

	NOMINAL COVERAGE RATE								
Min.	10%	20%	30%	40%	50%	60%	70%	80%	90%
5	5.81	11.19	16.81	23.07	29.75	37.20	46.48	59.13	81.17
10	7.08	15.20	23.13	31.77	41.34	52.86	65.43	83.63	116.90
15	8.29	17.29	27.00	36.81	49.48	62.82	79.43	102.97	143.54
20	9.80	20.55	32.61	44.80	58.12	73.52	93.58	119.83	167.01
25	11.85	24.86	36.87	50.18	65.20	83.44	107.20	136.81	193.64
30	12.94	26.67	42.51	57.30	74.05	94.57	120.14	157.65	220.65
35	14.46	29.67	46.42	62.86	81.87	106.65	134.15	174.24	244.98
40	16.29	34.40	51.92	70.34	91.84	116.61	145.71	188.65	266.49
45	17.57	36.85	56.36	74.89	97.33	126.86	157.81	206.28	293.29
50	19.38	39.24	59.46	79.78	106.03	138.18	174.09	227.45	321.83
55	20.01	41.28	64.10	87.28	116.03	146.94	186.87	243.40	342.05
60	22.73	44.81	68.47	94.35	122.52	159.25	202.75	260.95	369.07

60-min outs. For coverage rates higher than 95%, the curves of the ESD for some look-ahead times (i.e., 5- to 20-min-ahead and 30- to 40-min-ahead times) are still below the curve for the Gaussian distribution, indicating conservative PIs. On the other hand, generally speaking, for larger look-ahead times, the curves of the ESD are above the curve for the Gaussian distribution, indicating large errors. This is consistent with the fact that as look-ahead time increases, data uncertainty increases as well.

2) Standard-Deviation-Based PIs Versus Quantile-Based PIs: The standard-deviation-based PIs are further evaluated by comparison to the empirical quantile-based PIs which are constructed for nominal coverage rates of 10%, 20%,..., 90%. To construct empirical quantile-based PIs, consider WNNHKF 5-min-ahead forecasting errors for example. At time t, historical 5-min errors (actual minus predicted load of 5-min-ahead) before time t are collected. For a nominal coverage rate 1-α, e.g., $\alpha = 0.1$, the 5th and 95th percentiles of the errors are calculated. The 90% prediction interval for time t is obtained by adding the 5th and 95th percentiles to the predicted load. For our testing, the errors from July 1, 2008 to November 30, 2008 are used to construct the quantile-based PI for $t = 00:05$ am on December 1, 2008. When the error at 00:05 am becomes available, the new error and previous errors are then combined to construct the prediction interval for $t = 00:10$ am, and so on. To quantify forecasting accuracy, this process is repeated for all the data collected until the end of December. The result shows that the empirical quantile-based PIs of 90% nominal coverage rate cover 90.44% of the actual load data, indicating the empirical quantile-based PIs are accurate. The same steps are taken for 10- to 60-min forecasting results and for different nominal coverage rates, ranging from 10% to 90%, and similar conclusions are obtained as shown in Table VII.

The standard-deviation-based PIs are derived from dynamic innovation covariance of Kalman filters, whereas the empirical quantile-based PIs are derived from quasistatic historical errors. To compare these two types of PIs on an equal footing, the widths of the PIs under the same actual coverage rates are compared. Again, consider WNNHKF 5-min outs for December 2008 for example. Under the 90% nominal coverage rate, the empirical quantile-based PIs have an actual coverage rate of 90.44% with an average width of 81.17 MW. To achieve the same actual coverage rate, the width of standard-deviation-based PIs is found to be 1.47 × 2 ESD with an average width of 76.28 MW. The comparison indicates that under the same actual coverage rate, the standard-deviation-based PIs are generally narrower than the empirical quantile-based PIs. This result is consistent with the dynamic nature of the innovation covariance produced by Kaman filters as explained in Section III-A. The same steps are taken for 10- to 60-min-ahead forecasting results and for different nominal coverage rates ranging from 10% to 90%, and similar results can be obtained from Tables VIII and IX.

TABLE IX
WIDTHS (MW) OF STANDARD-DEVIATION-BASED PIS ACHIEVING THE SAME ACTUAL COVERAGE RATES AS SHOWN IN TABLE VII

	NOMINAL COVERAGE RATE								
Min.	10%	20%	30%	40%	50%	60%	70%	80%	90%
5	5.71	11.42	17.12	22.83	29.58	36.33	45.67	58.12	76.28
10	6.94	15.43	23.91	33.17	42.43	52.45	64.79	79.45	110.31
15	8.32	16.63	28.07	36.38	49.90	62.37	81.09	99.80	133.06
20	9.26	18.51	31.23	40.49	55.53	69.42	90.24	111.07	148.09
25	9.99	19.98	33.72	43.71	59.95	74.94	97.42	119.90	159.87
30	12.26	24.51	41.37	53.63	73.54	91.93	119.51	147.09	196.12
35	13.05	26.11	44.06	57.11	78.32	97.90	127.27	156.64	208.86
40	15.52	31.05	52.39	67.91	93.13	116.42	141.64	184.33	242.54
45	16.23	32.48	54.81	71.05	97.43	121.79	148.18	192.84	253.73
50	16.97	33.95	57.29	74.26	101.85	127.31	154.89	201.57	265.23
55	17.71	35.43	59.78	77.49	106.28	132.85	161.63	210.34	276.76
60	20.60	41.20	66.95	90.13	123.61	154.51	187.99	244.64	321.89

TABLE X
MAPES (%) COMPARING THE RESULTS OF WNNHKF TO THE RESULTS OF PERSISTENCE, LINEAR AR MODEL, ISO-NE'S METHOD, AND WNN

Min.	Persistence	Linear AR	ISO-NE's Method	WNN	WNNHKF
5	0.38	0.16	0.26	0.08	0.12
10	0.74	0.22	0.30	0.13	0.13
15	1.10	0.32	0.34	0.16	0.15
20	1.46	0.44	0.38	0.20	0.16
25	1.82	0.57	0.43	0.23	0.18
30	2.18	0.71	0.48	0.27	0.23
35	2.53	0.85	0.53	0.31	0.26
40	2.89	1.01	0.60	0.35	0.33
45	3.24	1.17	0.64	0.38	0.37
50	3.59	1.35	0.70	0.42	0.36
55	3.94	1.54	0.75	0.45	0.40
60	4.29	1.73	0.81	0.49	0.47

Case 4: Results of our WNNHKF method are compared to the results of persistence, linear AR model, ISO-NE's method in [17], and WNN method of [1] reviewed in Section II-B, based on the ISO-NE's data set. The forecasting period for comparison is from July 1, 2008 to July 31, 2008 because ISO-NE only provided results of this period to us. MAPEs in Table X show that the results of our method are better than the results of persistence, linear AR model, and ISO-NE's method. The same conclusion is also made from MAEs. Furthermore, our WNNHKF method improves the WNN for 10- to 60-min-ahead predictions, but doesn't perform as well as the WNN for 5-min-ahead predictions. This is because the relationship between input and output measurement for an NN does not appear to be very nonlinear based on observation, and the UKFNN may not work as well as the standard NN for 5-min-ahead predictions. The same conclusion is made for winter and spring seasons (December 2008 to May 2009) when the performance of WNNHKF and WNN are compared. For the same reason, the WNNHKF does not perform as well as the WNN for fall season (September to November 2008).

VI. CONCLUSION

This paper presents a method of wavelet neural networks trained by hybrid Kalman filters. Based on data analysis, an EKFNN is used to capture the near-linear relationship between the LL input and output measurement for an NN, and two UKFNNs are used to capture the highly nonlinear relationships for LH and H load components. By replacing the first-order back propagation algorithm with the second-order Kalman-type algorithms, the dynamic innovation covariance can be obtained for PI estimates. Consequently, the estimated standard deviation, which is derived based on the nonlinear transformation of WNNHKF, is close to the sample standard deviation. To evaluate PIs, the forecasting errors are demonstrated to have heavier tails than a Gaussian distribution. For the forecasting errors, both the one sigma coverage and the amount of the estimated standard deviations needed to achieve a given coverage rate are close to the ones under the standard Gaussian distribution. Numerical testing results based on ISO-NE's data show that the WNNHKF provides the overall best predictions with accurate and conservative PI estimates.

ACKNOWLEDGMENT

The authors would like to thank Mr. P. Friedland who was at the ISO-NE and Dr. K. W. Cheung at the Alstom Grid. The authors would also like to thank anonymous editor and reviewers for their insightful and constructive comments which contributed to improve the quality of this paper.

REFERENCES

[1] C. Guan, P. B. Luh, L. D. Michel, Y. Wang, and P. B. Friedland, "Very short-term load forecasting: Wavelet neural networks with data prefiltering," *IEEE Trans. Power Syst.*, vol. 28, no. 1, pp. 30–41, Feb. 2013.

[2] B. Fox, D. Flynn, L. Bryans, N. Jenkins, D. Milborrow, M. O'Malley, R. Watson, and O. Anaya-Lara, "Wind power integration connection and system operational aspects," IET Power and Energy Series 2007.

[3] F. Wang, K. Xie, E. Yu, G. Liu, and M. Wang, "A simple and effective ultra-short term load forecasting method," *Power Syst. Technol.*, vol. 20, no. 3, pp. 41–43, Mar. 1996, 48.

[4] D. Luo and H. He, "A shape similarity criterion based curve fitting algorithm and its application in ultra-short-term load forecasting," *Power Syst. Technol.*, vol. 31, no. 21, pp. 81–84, Nov. 2007.

[5] J. Zhou, B. Zhang, J. Shang, J. Yao, and M. Cheng, "Very short-term load forecast based on multi-sample extrapolation and error analysis," *Elect. Power Automat. Equip.*, vol. 25, no. 2, pp. 15–21, Feb. 2005.

[6] Z. Yang, G. Tang, Y. Song, and R. Cao, "Improved cluster analysis based ultra-short term load forecasting method," *Automat. Elect. Power Syst.*, vol. 29, no. 24, pp. 83–86, Dec. 2005.

[7] K. Liu, S. Subbarayan, R. R. Shoults, M. T. Manry, C. Kwan, F. L. Lewis, and J. Naccarino, "Comparison of very short-term load forecasting techniques," *IEEE Trans. Power Syst.*, vol. 11, no. 2, pp. 877–882, May 1996.

[8] J. Lu, X. Zhang, and W. Sun, "A real-time adaptive forecasting algorithm for electric power load," in *Proc. IEEE PES Transmission and Distribution Conf. Expo.: Asia and Pacific*, Dalian, China, 2005.

[9] L. C. M. de Andrade and I. N. da Silva, "Using intelligent system approach for very short-term load forecasting purposes," in *Proc. IEEE Int. Energy Conf. Exhib.*, Manama, Bahrain, Dec. 2010.

[10] A. Setiawan, I. Koprinska, and V. G. Agelidis, "Very short-term electricity load demand forecasting using support vector regression," in *Proc. 2009 Int. Joint Conf. Neural Networks*, Atlanta, GA, USA, June 2009.

[11] J. W. Taylor, "An evaluation of methods for very short-term load forecasting using minute-by-minute British data," *Int. J. Forecast.*, vol. 24, pp. 645–658, 2008.

[12] D. J. Trudnowski, W. L. Mcreynolds, and J. M. Johnson, "Real-time very short-term load prediction for power system automatic generation control," *IEEE Trans. Control Syst. Technol.*, vol. 9, no. 2, pp. 254–260, Mar. 2001.

[13] K. Xie, F. Wang, and E. Yu, "Very short-term load forecasting by Kalman filter algorithm," in *Proc. Chinese Society for Electrical Engineering*, Jul. 1996, vol. 16, no. 4, pp. 245–249.

[14] H. Yang, H. Ye, G. Wang, J. Khan, and T. Hu, "Fuzzy neural very short-term load forecasting based on chaotic dynamics reconstruction," *Chaos, Solitons & Fractals*, vol. 29, pp. 462–469, 2006.

[15] S. Kawauchi, H. Sugihara, and H. Sasaki, "Development of very short-term load forecasting based on chaos theory," *Elect. Eng. Japan*, vol. 148, no. 2, pp. 55–63, 2004.

[16] L. C. M. de Andrade and I. N. da Silva, "Very short-term load forecasting using a hybrid neuro-fuzzy approach," in *Proc. 2010 11th Brazilian Symp. Neural Networks*, Sao Carlos, Brazil, Oct. 2010.

[17] P. Shamsollahi, K. W. Cheung, Q. Chen, and E. H. Germain, "A neural network based very short-term load forecaster for the interim ISO New England electricity market system," in *Proc. 22nd IEEE Power Engineering Society Int. Conf. Power Industry Computer Applications: Innovative Computing for Power-Electric Energy Meets the Market*, Sydney, Australia, May 2001.

[18] W. Charytoniuk and M. S. Chen, "Very short-term load forecasting using artificial neural networks," *IEEE Trans. Power Syst.*, vol. 15, no. 1, pp. 263–268, Feb. 2000.

[19] B. D. Ripley, *Neural Networks and Pattern Recognition.* Cambridge, U.K.: Cambridge Univ. Press, 1996.

[20] C. Guan, P. B. Luh, L. D. Michel, M. A. Coolbeth, and P. B. Friedland, "Hybrid Kalman algorithms for very short-term load forecasting and confidence interval estimation," in *Proc. IEEE Power and Energy Society 2010 General Meeting*, Minneapolis, MN, USA, 2010.

[21] G. Papadopoulos, P. J. Edwards, and A. F. Murray, "Confidence estimation methods for neural networks: A practical comparison," *IEEE Trans. Neural Netw.*, vol. 12, no. 6, pp. 1278–1287, 2001.

[22] W. Charytoniuk, M. S. Chen, P. Kotas, and P. Van Olinda, "Demand forecasting in power distribution systems using nonparametric. probability density estimation," *IEEE Trans. Power Syst.*, vol. 14, no. 4, pp. 1200–1206, Nov. 1999.

[23] C. Charytoniuk and J. Niebrzydowski, "Confidence interval construction for load forecast," *Elect. Power Syst. Res.*, vol. 48, pp. 97–103, 1998.

[24] A. P. Alves da Silva and L. S. Moulin, "Confidence intervals for neural network based short-term load forecasting," *Trans. Power Syst.*, vol. 15, no. 4, pp. 1191–1196, Nov. 2000.

[25] G. Chryssolouris, M. Lee, and A. Ramsey, "Confidence interval prediction for neural network models," *IEEE Trans. Neural Netw.*, vol. 7, no. 1, pp. 229–232, 1996.

[26] P. Pinson and G. Kariniotakis, "Conditional prediction intervals of wind power generation," *IEEE Trans. Power Syst.*, vol. 25, no. 4, pp. 1845–1856, Nov. 2010.

[27] S. Fan and R. J. Hyndman, "Short-term load forecasting based on a semi-parametric additive model," *IEEE Trans. Power Syst.*, vol. 27, no. 1, pp. 134–141, Feb. 2012.

[28] W. A. Wright, "Bayesian approach to neural network modeling with input uncertainty," *IEEE Trans. Neural Netw.*, vol. 10, no. 6, pp. 1261–1270, Nov. 1999.

[29] L. Zhang, P. B. Luh, and K. Kasiviswanathan, "Energy clearing price prediction and confidence interval estimation with cascaded neural networks," *IEEE Trans. Power Syst.*, vol. 18, no. 1, pp. 99–105, Feb. 2003.

[30] S. Singhal and L. Wu, "Training feed-forward networks with the extended Kalman algorithm," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, Morristown, NJ, USA, May 1989.

[31] G. V. Puskorius and L. A. Feldkamp, "Decoupled extended Kalman filter training of feedforward layered networks," in *Proc. Int. Joint Conf. Neural Networks*, Dearborn, MI, Jul. 1991.

[32] L. Zhang and P. B. Luh, "Neural network based market clearing price prediction and prediction interval estimation with an improved extended Kalman filter method," *IEEE Trans. Power Syst.*, vol. 20, no. 1, pp. 59–66, Feb. 2005.

[33] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation With Applications to tracking and Navigation: Algorithms and Software for Information Extraction.* New York, NY, USA: Wiley, 2001.

[34] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, "A new approach for filtering nonlinear systems," in *Proc. American Control Conf.*, 1995, pp. 1628–1632.

[35] E. A. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proc. IEEE Adaptive Systems for Signal Processing, Communications, and Control Symp.*, Lake Louise, AB, Canada, 2000.

Che Guan (M'12) is currently pursuing the Ph.D. degree in the Department of Electrical and Computer Engineering at the University of Connecticut, Storrs, CT, USA.

His research interests include power systems, control systems and optimization, intelligent systems, and signal processing.

Peter B. Luh (F'95) received the Ph.D. degree from Harvard University, Cambridge, MA, USA.

He has been with the Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT, USA, since 1980, and currently is the SNET Professor of Communications & Information Technologies. He served as the Head of the Department from 2006 to 2009.

Prof. Luh is the Senior Advisor on Automation for the Robotics and Automation Society. He was VP Publication Activities for the IEEE Robotics and Automation Society, the Editor-in-Chief of the IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, and the founding Editor-in-Chief of the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING.

Laurent D. Michel received the Ph.D. degree from Brown University, Providence, RI, USA, in 1999.

He is an Associate Professor of Computer Science and Engineering at the University of Connecticut, Storrs, CT, USA.

Prof. Michel sits on the Editorial Board of Constraints, Mathematical Programming Computation and Constraint Letters.

Zhiyi Chi received the Ph.D. degree in applied mathematics from Brown University, Providence, RI, USA, in 1998.

He is currently a Professor in the Department of Statistics at University of Connecticut, Storrs, CT, USA. His research interests include stochastic processes, large scale hypothesis testing, and their applications in biological and physical sciences.