Using Bayesian Deep Learning to Capture Uncertainty for Residential Net Load Forecasting

Mingyang Sun, *Member, IEEE,* Tingqi Zhang, *Student Member, IEEE,* Yi Wang, *Member, IEEE,*
Goran Strbac, *Member, IEEE,* Chongqing Kang, *Fellow, IEEE,*

***Abstract*—Decarbonization of electricity systems drives significant and continued investments in distributed energy sources to support the cost-effective transition to low-carbon energy systems. However, the rapid integration of distributed photovoltaic (PV) generation presents great challenges in obtaining reliable and secure grid operations because of its limited visibility and intermittent nature. Under this reality, net load forecasting is facing unprecedented difficulty in answering the following question: *how can we accurately predict the net load while capturing the massive uncertainties arising from distributed PV generation and load, especially in the context of high PV penetration?* This paper proposes a novel probabilistic day-ahead net load forecasting method to capture both epistemic uncertainty and aleatoric uncertainty using Bayesian deep learning, which is a new field that combines Bayesian probability theory and deep learning. The proposed methodological framework employs clustering in subprofiles and considers residential rooftop PV outputs as input features to enhance the performance of aggregated net load forecasting. Numerical experiments have been carried out based on fine-grained smart meter data from the Australian grid with separately recorded measurements of rooftop PV generation and loads. The results demonstrate the superior performance of the proposed scheme compared with a series of state-of-the-art methods and indicate the importance and effectiveness of subprofile clustering and high PV visibility.**

***Index Terms*—Probabilistic net load forecasting, distributed PV generation, Bayesian deep learning, clustering, long short-term memory.**

I. Introduction

GLOBAL decarbonization is expected to be achieved by increasing the penetration of renewable energy sources (RES) and by the electrification of the heating and transport sectors. Although uncertainty at the higher system level is more likely to be traded off, in future power systems, the predictability of aggregate loads still tends to be limited by the significant uncertainties arising from climate variability, electric vehicles, distributed renewable energy generation, energy efficiency, and demand response [1]. Accurate probabilistic net load forecasting is thus of great importance to capture these massive uncertainties, contributing to the operation and planning of future smart, low-carbon energy systems.

In the literature, conventional forecasting approaches focus on point/deterministic forecasting (e.g., [2]–[5]). In particular, the pioneering work of deterministic short-term load forecasting in [4] and [5] effectively addresses the challenges of peak load estimation at system level and bus load prediction, respectively. However, in view of capturing uncertainties injected from different resources, point forecasting is becoming obsolete because it can provide only a single output per time step for the decision-making process that heavily depends on expected values. In other words, the ideal forecasting models should be capable of representing uncertainty via quantiles, intervals or probability density functions for numerous applications such as probabilistic load flow analysis, reliability planning, and optimal bidding in electricity markets. In general, according to reference [1], the probabilistic forecasts can be obtained via (i) feeding multiple scenarios to a deterministic model [6]–[9]; (ii) developing novel probabilistic forecasting models [10]–[14]; (iii) post-processing the point forecasts [8], [15]; or their combinations [16]. In particular, the novel hybrid probabilistic load forecasting model proposed in [16] was developed based on an improved wavelet neural network trained by a generalized extreme learning machine to provide the load forecast with a probabilistic interval while capturing the forecasting model and data noise uncertainties. A comprehensive review on probabilistic electric load forecasting challenges and modern probabilistic forecasting models is presented in [1].

Despite the rich literature focusing on electric load forecasting, very few studies aim to predict the net load (i.e., the load traded between the microgrid and the utility grid), which is important for smart grid management and operations as well as resource allocation and electricity market participation with respect to common coupling between interconnected grids [17]. Different from traditional load forecasting, net load refers to the total energy consumption partially supported by the distributed renewable energy, such as local PV generation, thus injecting additional uncertainty, especially when the PV generation is partially visible or completely invisible. Therefore, the researchers in [18] designed a novel method to address the invisible high PV penetration, where the net load profile is decomposed into PV output, actual load and residual, which are predicted in turn. Additionally, additive and integrated net load forecast models are compared in [17], and the results demonstrate that the forecasting errors of net load and solar are cointegrated with a common stochastic drift. Other works, such as [19], propose very short-term forecasting using a complex-valued neural network. A neural network (NN) with

This work was supported by the International (Regional) Joint Research Project of National Natural Science Foundation of China (No. 71961137004). (Corresponding Author: Tingqi Zhang)

M. Sun, T. Zhang, and G. Strbac are with the Department of Electrical & Electronic Engineering, Imperial College London, London, SW7 2AZ, UK.

Yi Wang is with the Power Systems Laboratory, ETH Zurich, 8092 Zurich, Switzerland..

C. Kang is with the State Key Lab of Power Systems, Dept. of Electrical Engineering, Tsinghua University, Beijing 100084, China.

a Levenberg-Marquardt training algorithm is used in [20] to generate the feeder net load forecast.

Beyond the aforementioned studies, which are mostly based on classical statistical or ANN methods, in recent years, deep learning, as one of the cutting-edge technologies, has received widespread attention in a range of research fields [21], [22]. Regarding energy-related time-series forecasting, researchers [23] have used deep learning methods to achieve a load forecasting task and compared the performance between a conditional restricted Boltzmann machine and a factored conditional restricted Boltzmann machine. Additionally, the authors of [24] propose a novel forecasting model for short-term power load and probability density forecasting based on deep learning, quantile regression and kernel density estimation. Furthermore, another type of network structure designed for handling sequence dependence (i.e., time-series data in this case) is recurrent neural networks. The long short-term memory (LSTM) network is one powerful type of RNN structure that includes a memory cell that can retain information for long periods of time and deal with the the problem of long-term dependencies [25]. As an example, the authors of [26] and [27] have used deep LSTM networks to tackle the challenges of high volatility and uncertainty in household-level loads, showing a verified superior performance. More recent works such as [28] proposed an improved quantile regression neural network by introducing Gaussian noise into the training process. In [29], a deep residual network is proposed based on Monte Carlo dropout to achieve probabilistic forecasting. Additionally, LSTM also has some other variations, such as dilated LSTM [30] and bidirectional LSTM [31]. When dealing with the challenges of insufficient data, the authors in [32] designed a transfer training model with shared layers to perform wind farm forecasting.

Although the existing research has successfully demonstrated the superior performance of deep learning on forecasting tasks, inherently, most of the studies are actually based on deterministic models, which lack the ability to capture uncertainty. As a new probabilistic deep learning model, the concept of Bayesian deep learning (BDL), which enables a deep learning framework to model uncertainty, is becoming increasingly prevalent in computer vision, natural language processing, medical diagnostics, and autonomous driving [33]. BDL exhibits the benefits of uncertainty representation, understanding generalization, and reliable prediction, leading to a more interpretable deep neural network through the lens of probability theory. In this paper, a novel probabilistic net load forecasting framework is proposed based on BDL, aimed at capturing both epistemic uncertainty and aleatoric uncertainty. Note that this work will focus on the net load prediction at the aggregated level, and the proposed framework implements a clustering technique to group the residential customers and employs PV outputs as parts of input features for network training. We design case studies based on real PV generation and load data from the Australian grid. Compared with other state-of-the-art methods, the proposed approach outperforms conventional approaches, and the results show the importance of clustering and high PV visibility. To summarize, this study makes the following original contributions:

(1) A clustering-forecasting-aggregation probabilistic day-ahead net load forecasting strategy is proposed to make full use of smart meter data and partially visible PV output data.

(2) Bayesian theory and deep LSTM networks are combined to generate aggregated level probabilistic net load forecasts with the target of capturing both epistemic uncertainty and aleatoric uncertainty. To the best of the authors' knowledge, this is the first paper to exploit Bayesian deep learning for net load prediction.

(3) A comprehensive comparison with a series of state-of-the-art methods is conducted. The superior performance of the proposed scheme is demonstrated with respect to both the deterministic and probabilistic forecasting results. Additionally, it is shown that the forecasting performance can be effectively enhanced in the context of high PV visibility.

The rest of this paper is organized as follows. Section II identifies the primary challenges in probabilistic net load forecasting. Section III introduces the Bayesian deep LSTM network. Section IV illustrates the proposed probabilistic short-term net load forecasting framework. Section V conducts comprehensive numerical experiments to demonstrate the superior performance of the proposed method. Section VI draws the conclusions.

II. Primary Challenges

The widespread deployment of distributed PV generation and its intermittent nature significantly diminish the predictability of the residential net load. This effect may be further intensified by the stochasticity in onsite renewable generation injected from the macrogrid [17]. Under this circumstance, the primary challenges addressed in this work are summarized as follows:

1) ***PV Visibility***: In general, distributed PV is invisible to the distribution system operators and retailers as a result of its behind-the-meter installation, which injects additional uncertainty into the net load and renders it harder to accurately predict, especially in the context of high PV penetration. However, with the development of advanced metering technologies, some residential customers have installed meters that can separately measure electricity consumption and rooftop PV output to make the distributed PV generation partially visible to stakeholders with fine-grained data. To this end, developing methods to fully exploit the partially visible or entirely visible PV to enhance the net load forecasting performance at the aggregated level will be one of the fundamental challenges that is investigated in this study.

2) ***Massive Stochastic Uncertainty***: For the net load at the aggregated level, uncertainty is composed of the load uncertainty and the distributed PV uncertainty, which is a more challenging task than either load forecasting or PV forecasting alone. In this case, we use the term *stochastic uncertainty (aleatoric uncertainty)* to represent the uncertainty within the net load injected from different sources such as climate variability, intermittent power generation, and aperiodic human activities. In recent years, although a number of probabilistic forecasting methods have been proposed to capture these massive amounts of uncertainties, in the load or the net load,

most of the existing methods can only provide the prediction interval (i.e., the upper and lower bounds), which does not give detailed information about the distribution of the forecast at each individual time step. In addition, most probabilistic forecasting models are inherently deterministic models with limited performance in explicitly capturing stochastic uncertainty. These models usually either produce a density forecast by employing the probability density function (pdf) of the residuals to the point forecast or perform post-processing to several point forecasts to generate quantiles. On the other hand, deep learning has demonstrated a fair performance in load forecasting; however, most of the existing models are not able to represent uncertainty. Consequently, it is crucial and imperative to investigate and develop a pure probabilistic deep learning model to handle the massive *stochastic uncertainty* in the net load and to provide confidence bounds for decision making.

3) ***Uncertainty in the Model***: *Model uncertainty (epistemic uncertainty)* refers to the uncertainty in the model parameters and the model structure. Beyond the aleatoric uncertainty, *model uncertainty* is also a critical part of uncertainty in the task of probabilistic net load forecasting to indicate how much uncertainty the model has about its outputs. Among a vast number of potential model structures and parameters, it is important to understand how much the selected combinations might be able to accurately predict the net load under different conditions (e.g., seasons, weekends/weekdays and social factors). In the remainder of this paper, we will illustrate in detail how the proposed Bayesian deep learning-based method can effectively handle the aforementioned challenge.

III. Bayesian Deep Learning

A. Why Bayesian Deep Learning?

Deep learning has demonstrated state-of-the-art performance in a vast number of tasks; however, as illustrated in [34], it still suffers from a series of limitations that need to be investigated and resolved, including "*i) uninterpretable black boxes; ii) being weak in its uncertainty representation; iii) being data hungry; iv) being computationally intensive; v) being finicky to optimize; and vi) being easily misled by adversarial examples*". To address the first three challenges, in this part, we will qualitatively explain the benefits and rationale of employing BDL to conduct net load forecasting.

1) Inherently probabilistic model: BDL is inherently a probabilistic model that allows a deep learning model to represent uncertainty. Unlike traditional neural networks, which have fixed parameters once trained, Bayesian network parameters (i.e., the weights and bias) are expressed as conditional probabilities. As a result, the Bayesian model generates its result by directly sampling from its parameters rather than adding noise to the output or setting up multiple input scenarios. In other words, the Bayesian model is fundamentally probabilistic rather than deterministic in nature.

2) Captures both model uncertainty and stochastic uncertainty: In the literature, most of the existing Bayesian deep learning approaches can merely capture either the model uncertainty or the stochastic uncertainty alone [33]. However, in this case, the proposed Bayesian deep LSTM network (BDLSTM) can simultaneously capture the *model uncertainty* and the stochastic uncertainty. More specifically, the model uncertainty is captured by placing a prior distribution over the model's weights; then, the posterior can be approximated via an inference algorithm. Hence, the model uncertainty is represented by the shape of the distribution of the weights. In other words, the BDLSTM attempts to capture how much those weights change based on the input data. For safety-critical applications, it is of significant importance to capture the epistemic uncertainty to understand examples that are different from the training data. Furthermore, *stochastic uncertainty* is captured by placing a distribution with small variance (usually Gaussian random noise) over the output and, therefore, the model learns the variance in the noise as a function of different inputs [35].

3) Explainable under probability theory: Traditional deep neural networks use their neurons to memorize the information inside the training data, which implies that the parameters in traditional neural networks have no physical meaning, and thus, their values can be arbitrary. Nonetheless, Bayesian networks calculate their outputs with Bayesian theory to render the parameters explainable so that the network has the ability to 'feel' certain or uncertain about its result. In particular, BDL can calibrate the model and the prediction uncertainty to obtain smart systems that know exactly what they do not know. For example, in net load forecasting, when the predictor encounters input features with extremely different or unreasonable values than it has encountered before (i.e., out-of-distribution test data), the predictor can give an answer (e.g., the quantified model uncertainty) indicating that it does not know how to handle this new dataset, rather than giving a wrong forecast like the current deep learning models. This property can help the user determine whether the current model needs to be updated or re-trained with the latest data or can inform the user that the input data may include outliers or bad data. Additionally, to make rational decisions, BDL provides a way of integrating prior knowledge into learning systems and updating that knowledge in a coherent and robust way with the influx of more data.

4) Reliable performance with small datasets: Many real-world tasks have limited amounts of data (small data) that conventional deep learning systems cannot address because the extremely high or low model complexity will lead to the issues of overfitting or poor performance, respectively. For the net load forecasting problem, although a large number of measurements can be collected through advanced smart metering systems, for classical deep learning, which usually requires millions of training samples, the performance may still be limited due to the lack of data. However, for BDL, less data are required to make accurate forecasting. By integrating prior knowledge into learning systems, BDL can effectively address the overfitting problem by imposing a prior on hidden units or neural network parameters, even with small/insufficient datasets. In other words, BDL enables the network to achieve automatic model complexity control and structure learning with the benefits of the built-in implicit regularization [36].

B. Bayesian Deep LSTM Network (BDLSTM)

The appeal of a special recurrent neural network architecture, long short-term memory networks (LSTMs) [37], has been demonstrated for short-term residential load forecasting to tackle the challenges of long-term dependencies in the literature (e.g., [26]). Beyond that, the deep architecture of LSTMs can contribute to learning highly nonlinear relationships between the input explanatory features and the output residential load data through a series of linear or nonlinear functions.

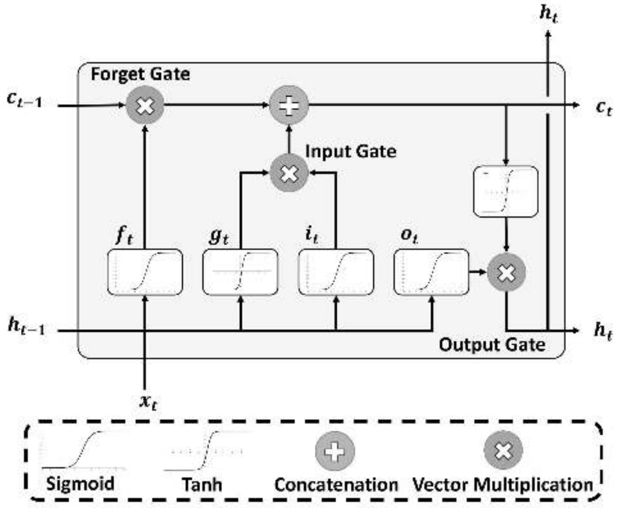


Fig. 1. The structure of one LSTM cell.

To describe the basic architecture of the proposed Bayesian deep neural network, we first briefly introduce the structure of one LSTM cell, as shown in Fig.1. The inputs of the LSTM cell at one particular time step t are the previous state h_{t-1} and the current input x_t. Through four fully connected neurons f_t, g_t, i_t, and o_t, three gates are employed to fulfill the function of memory or forget information. In particular, the forget gate decides how much previous information will be transported forward, the input gate controls the aspects of new input information, and the output gate decides what will be output at this time step. In terms of the outputs, h_t is then fed into the next time step as input, which can be considered as a short-term state, while c_t decides the longer-term dependency. The overall computation is summarized in equations (1)-(4) as follows:

$$f_t = \sigma\Big(W_{xf}^T \cdot x_t + W_{hf}^T \cdot h_{t-1} + b_f\Big) \tag{1}$$

$$i_t = \sigma\Big(W_{xi}^T \cdot x_t + W_{hi}^T \cdot h_{t-1} + b_i\Big) \tag{2}$$

$$o_t = \sigma\Big(W_{xo}^T \cdot x_t + W_{ho}^T \cdot h_{t-1} + b_o\Big) \tag{3}$$

$$g_t = \tanh\Big(W_{xg}^T \cdot x_t + W_{hg}^T \cdot h_{t-1} + b_g\Big) \tag{4}$$

Given the values of the three gates at next time step, the values of next state c_t and h_t are calculated by the equations $c_t = f_t \cdot c_{t-1} + i_t \cdot g_t$ and $y_t = h_t = o_t \cdot \tanh(c_t)$, respectively, where W_{xf}^T, W_{xi}^T, W_{xo}^T, and W_{xg}^T represent the weights of each input vector x_t, W_{hf}^T, W_{hi}^T, W_{ho}^T, W_{hg}^T are the weights of each previous short-term state h_{t-1}; b_f, b_i, b_o, and b_g are the biases for each of the four. It is notable that at the initial stage, b_f should be initialized with 1 instead of 0 to avoid forgetting everything from the beginning of training. Overall, through the above novel structure, LSTM handles time series by storing the important input information in the long-term state, preserving it for as long as required and retrieving it when necessary.

To obtain uncertainty estimates in deep learning, most of existing Bayesian deep learning approaches can capture only the epistemic uncertain or the aleatoric uncertainty alone, which are usually formalized as probability distributions over the model parameters or the model outputs [33]. To jointly capture the epistemic uncertain and the aleatoric uncertainty, a Bayesian deep LSTM network (BDLSTM), casting deep LSTMs as Bayesian models, is proposed, which retains the model architecture, placing a prior distribution upon the network weights and bias parameters of LSTMs and then inferring a posterior distribution over the given data.

Let $X_{train} = [x_1, \cdots, x_{T_{train}}]^T \in \mathbb{R}^{T_{train} \times d_x}$ and $Y_{train} = [y_1, \cdots, y_{T_{train}}] \in \mathbb{R}^{T_{train} \times d_y}$ denote the input data and output label, respectively, of the BDLSTM model that needs to be trained, where T_{train} is the total number of training data points, and d_x and d_y represent the dimensions of the input and the output, respectively. The primary target of a deep LSTM network can be formalized as identifying the optimal parameters W of a function $y = f^W(x)$ that are likely to have generated the outputs (i.e., the actual net load). In this case, $f^W(\cdot)$ represents the deep LSTM network with N_L layers and model parameters are denoted by $W = [W_1, ..., W_{N_L}]$, which is a set of random variables. An example of the Bayesian LSTM cell of the proposed BDLSTM network is given in Fig. 2 with a zoomed-in plot of the *forget gate* at time step t in the first layer. Detailed mathematical illustrations are given as follows.

1) The Epistemic Uncertainty: In general, the *epistemic uncertainty* (model uncertainty) comprises structure uncertainty and model parameter uncertainty. More specifically, structure uncertainty refers to the uncertainty in selecting the most appropriate model structure to extrapolate or interpolate the data well. Among a vast number of possible model parameters, which set of parameters should be selected to best explain the observations is uncertain, denoted by the model parameter uncertainty [33]. To capture the *epistemic uncertainty*, a prior distribution (e.g., $\mathcal{N}(0, I)$) is placed over W. In the literature, a series of studies have been carried out on prior selection (e.g., [38], [39]). In general, prior distributions can be classified into 1) non-informative prior distributions; 2) highly informative prior distributions; and 3) moderately informative hierarchical prior distributions [38].

For Bayesian deep neural networks, the prior distributions should represent the prior belief about the distribution of the neural network parameters (weights and bias), which are difficult to be identified because the physical meaning of these parameters remains unclear. In other words, selecting the prior for the Bayesian deep learning is still an open

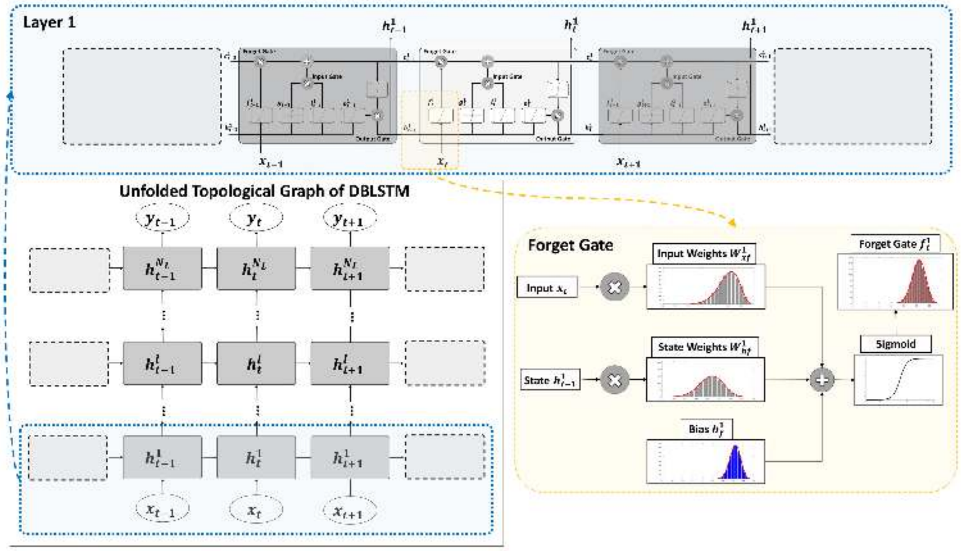


Fig. 2. An example of the proposed BDLSTM network with a zoomed-in plot of the *forget gate* at time step t in the first layer.

question that needs to be further investigated by researchers. According to references [33], [40], [41], employing standard parametric distributions has been demonstrated as one of the most effective solutions when the prior belief is difficult to be identified. Therefore, in this case, we set the standard normal distribution as our prior whose zero mean can bring about the benefit of regularization [33]. It is important to note that after training the Bayesian deep neural network, the posterior distribution will be employed to generate the samples of forecasts rather than the prior distribution.

After determining the appropriate prior, the model likelihood $p(Y_{train}|f^W(X_{train}))$ is also defined as a normal distribution $\mathcal{N}(f^W(X_{train}), \sigma^2)$ with a constant noise level σ. Based on Bayes rule, the posterior $p(W|X_{train}, Y_{train})$ is calculated by

$$p(W|X_{train}, Y_{train}) = \frac{p(Y_{train}|X_{train}, W) \cdot p(W)}{p(Y_{train}|X_{train})} \tag{5}$$

where $p(Y_{train}|X_{train})$ is the marginal probability that cannot be estimated analytically. To this end, different inference techniques such as variational inference and Markov chain Monte Carlo (MCMC) [36] are proposed to approximate it. Note that $p(W|X_{train}, Y_{train})$ represents the posterior distribution over weights given the training data $\{X_{train}, Y_{train}\}$. Given a new input point x, the new output y, which is defined as a random variable, can be predicted by integrating

$$p(y|x, X_{train}, Y_{train}) = \int p(y|x, W)p(W|X_{train}, Y_{train})\mathrm{d}W \tag{6}$$

It is notable that the true posterior is usually intractable, especially for complex models (e.g., deep LSTM networks). Therefore, an approximating variational distribution $q_\theta(W)$, parameterized by θ, is employed to ensure that the optimal distribution $\tilde{q}_\theta(W)$ can well represent $p(W|X_{train}, Y_{train})$, by minimizing the Kullback-Leibler (KL) divergence between $q_\theta(W)$ and $p(W|X_{train}, Y_{train})$ [42]:

$$KL(q_\theta(W)||p(W|X_{train}, Y_{train})) = \int q_\theta(W) \log \frac{q_\theta(W)}{p(W|X_{train}, Y_{train})}\mathrm{d}W, \tag{7}$$

using inference algorithms such as variational inference (VI), which is employed in this work. It is notable that it is intractable to analytically solve the optimization problem. Consequently, the objective is transformed from a KL divergence minimization problem to an Evidence Lower Bound (ELBO) maximization problem. More details regarding the employed VI algorithm can be found in the reference [33].

After obtaining the optimal distribution $\tilde{q}_\theta(W)$, the predictive distribution can be approximated by

$$p(y|x, X_{train}, Y_{train}) = \int p(y|x, W)\tilde{q}_\theta(W)\mathrm{d}W = \tilde{q}_\theta(y|x). \tag{8}$$

Let T_{sample} denote the number of sampled weights $\{\hat{W}_t\}_{t=1}^{T_{sample}}$, simulating the model based on the input x, the predictive mean and the predictive variance of y, which is a vector of size T_{sample}, can be approximated based on these samples. Mathematically, the predictive mean (the first raw moment) can be estimated with the unbiased estimator [33]

$$\widetilde{\mathbb{E}}[y] := \frac{1}{T_{sample}} \sum_{t=1}^{T_{sample}} f^{\hat{W}_t}(x) \tag{9}$$

where $f^{\hat{W}_t}(x)$ represents stochastic forward passes through the model (i.e., samples). On the other hand, to obtain the predictive variance, the second raw moment needs to be estimated. Similar to the estimation of the first moment, given that $\hat{W}_t \sim \tilde{q}_\theta(W)$ and $p(y|f^W(x)) = \mathcal{N}(y; f^W(x), \sigma^2)$ for some $\sigma > 0$, we have the estimator

$$\widetilde{\mathbb{E}}[y^T y] := \frac{1}{T_{sample}} \sum_{t=1}^{T} f^{\hat{W}_t}(x)^T f^{\hat{W}_t}(x) + \sigma^2 \tag{10}$$

with T_{sample} samples. Finally, the predictive variance can be approximated by $\widetilde{\mathrm{Var}}[y]$ as follows:

$$\begin{aligned}\widetilde{\mathrm{Var}}[y] &= \widetilde{\mathbb{E}}[y^T y] - \widetilde{\mathbb{E}}[y]^T \widetilde{\mathbb{E}}[y] \\ &:= MU(x, y, W) + \sigma^2 \end{aligned} \tag{11}$$

where

$$\begin{aligned} MU = &\frac{1}{T_{sample}} \sum_{t=1}^{T_{sample}} f^{\hat{W}_t}(x)^T f^{\hat{W}_t}(x) \\ &- \frac{1}{{T_{sample}}^2} \sum_{t=1}^{T_{sample}} f^{\hat{W}_t}(x)^T \sum_{t=1}^{T_{sample}} f^{\hat{W}_t}(x) \end{aligned} \tag{12}$$

represents the epistemic uncertainty (model uncertainty), which measures how much the model is uncertain about its outputs. It is important to note that in equation (11), with the increasing number of observations, the term $MU(x, y, W)$ can be reduced whereas the inherent noise measured by σ^2 cannot be vanished.

2) The Aleatoric Uncertainty: According to the dependency between the uncertainty and the inputs, the *aleatoric uncertainty* can be further divided into *homoscedastic uncertainty* and *heteroscedastic uncertainty* [35]. For *homoscedastic uncertainty*, the observation noise parameter σ is fixed whereas in this case, we need to capture the *heteroscedastic aleatoric uncertainty* because the uncertainty varies over different periods of time when dealing with the net load. To this end, σ in equation (11) needs to be adapted as a function of the input x, which means it is data-dependent. Let T_{train} denote the number of training observations, and the minimization objective of the data-dependent heteroscedastic model can be expressed as follows:

$$\mathcal{L}(\theta) = \frac{1}{T_{train}} \sum_{i=1}^{T_{train}} \frac{1}{2\sigma(x_i)^2} ||y_i - f(x_i)||^2 + \frac{1}{2}\log\sigma(x_i)^2 \tag{13}$$

In this case, maximum a posteriori (MAP) inference is carried out to locate a single parameter, θ, rather than the distribution of the weights, leading to neglect of the model uncertainty.

3) The Combined Uncertainties: To combine the *epistemic uncertainty* and the *aleatoric uncertainty* in a single BDLSTM model, the most straightforward and effective way is to transform the heteroscedastic model into a Bayesian model by placing a distribution over the weights and the bias [35]. First, we need to set up a new expression for the model to split the top layers of a deep LSTM network between the predictive mean $f(x)$ and the model precision $g(x)$ to simultaneously output $\hat{y}$ and $\hat{\sigma}^2$:

$$[\hat{y}, \hat{\sigma}^2] = f^{\hat{W}}_{BDLSTM}(x) \tag{14}$$

where f_{BDLSTM} represents the proposed Bayesian deep LSTM network parameterized by $\hat{W} \sim q_\theta(W)$. Given that a normal likelihood is chosen to model the aleatoric uncertainty, the final loss function of the BDLSTM model can be formulated as:

$$\mathcal{L}_{BDLSTM}(\theta) = \frac{1}{T_{train}} \sum_{i=1}^{T_{train}} \frac{1}{2\hat{\sigma}_i^2} ||y_i - \hat{y}_i||^2 + \frac{1}{2}\log\hat{\sigma}_i^2 \tag{15}$$

Note that the loss function can consider both the model uncertainty through $\hat{y}$ and the heteroscedastic uncertainty through $\hat{\sigma}$. Finally, the predictive uncertainty $\mathrm{Var}[y]$ of the proposed BDLSTM model, consisting of both the *aleatoric uncertainty* and the *epistemic uncertainty* can be approximated by

$$\begin{aligned}\widetilde{\mathrm{Var}}[y] := &\Big[\frac{1}{T_{sample}} \sum_{t=1}^{T_{sample}} {\hat{y}_t}^2 - \left(\frac{1}{T_{sample}} \sum_{t=1}^{T_{sample}} \hat{y}_t\right)^2 \Big] \\ &+ \frac{1}{T_{sample}} \sum_{t=1}^{T_{sample}} {\hat{\sigma}_t}^2. \end{aligned} \tag{16}$$

It is important to note that, compared with equation (13) which has a fixed σ, the second term in equation (16) is data-dependent. Detailed explanations regarding the Bayesian deep learning are presented in references [33], [35].

IV. The Proposed BDLSTM-Based Short-Term Net Load Forecasting Scheme

Based on the above-introduced BDLSTM model, a novel probabilistic short-term net load forecasting scheme is proposed to fully utilize the subprofiles of residential customers and exploit the partially visible PV output data to enhance the forecasting performance. In particular, the proposed framework includes four main stages: i) a *Clustering Stage*; ii) a *Feature Construction Stage*; iii) a *Forecasting Stage*; and iv) an *Aggregation Stage*, as shown in Fig.3.

A. Clustering Stage

In the proposed framework, the *Clustering Stage* aims to group the prosumers into different clusters based on their average daily net load patterns over the training days and to extract representative net load profiles from each cluster. This step is motivated by the fact that fine-grained subprofiles can reveal more information about the aggregated load and further assist in improving the forecasting accuracy [43]. However, it is impractical and inefficient to build a BDLSTM model for each individual customer and then aggregate them. The clustering procedure can also contribute to effectively reducing the computational complexity by balancing the number of models and the forecasting accuracy.

Let $L = [L_1, ..., L_N] \in \mathbb{R}^{T \times N}$ denote the historical load data of N residential customers where T is the total number of observations. The first step is to separate all the customers into two groups: by invisible PV generation and visible PV generation, represented by $L^{inv} \in \mathbb{R}^{T \times N^{inv}}$ and $L^{vis} \in \mathbb{R}^{T \times N^{vis}}$, respectively. Given that the numbers of clusters for each of these groups are K^{inv} and K^{vis}, respectively, as one of the most widely used and powerful methods, a hierarchical clustering method with Ward's linkage [44], [45] is applied based on the average daily net load patterns of L^{inv} and L^{vis}, defined as $RLP^{inv} \in \mathbb{R}^{N^{inv} \times 48}$ and $RLP^{vis} \in \mathbb{R}^{N^{vis} \times 48}$, respectively, to obtain the cluster label for each individual customer. In particular, hierarchical clustering has the benefits of having a deterministic nature and terminating the agglomeration procedure at any number of clusters as required [46]. A detailed explanation of the

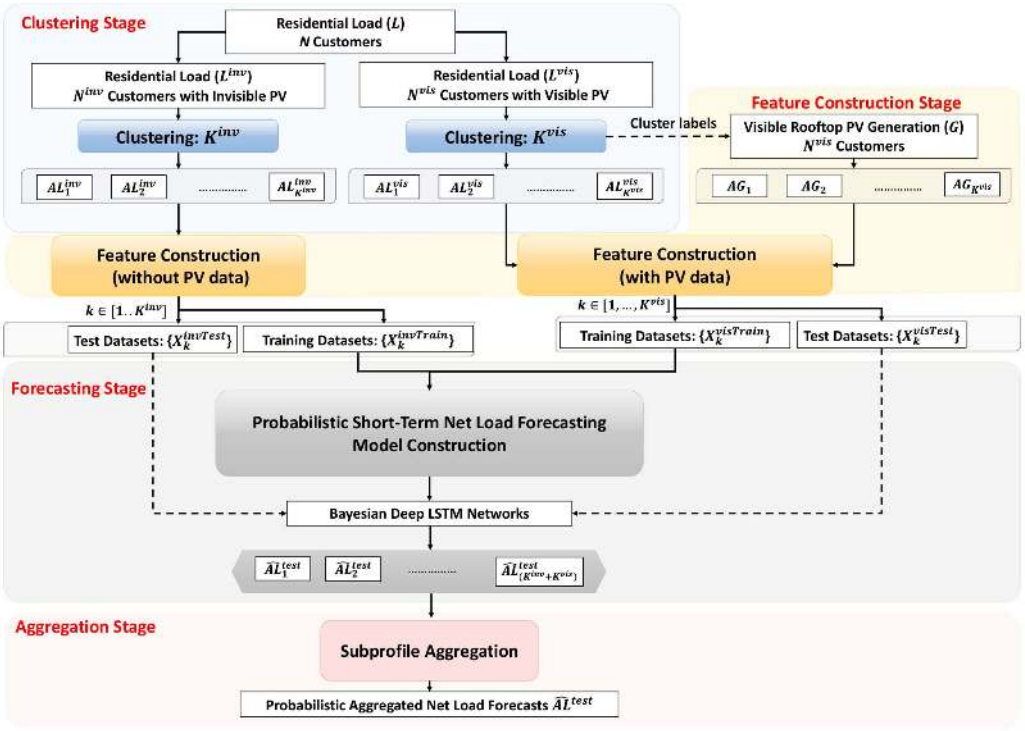

Clustering Stage
Residential Load (L)
N Customers
Residential Load (L^{inv})
N^{inv} Customers with Invisible PV
Residential Load (L^{vis})
N^{vis} Customers with Visible PV
Clustering: K^{inv}
Clustering: K^{vis}
Cluster labels
Feature Construction Stage
Visible Rooftop PV Generation (G)
N^{vis} Customers
AL_1^{inv}
AL_2^{inv}
$AL_{K^{inv}}^{inv}$
AL_1^{vis}
AL_2^{vis}
$AL_{K^{vis}}^{vis}$
AG_1
AG_2
$AG_{K^{vis}}$
Feature Construction (without PV data)
Feature Construction (with PV data)
$k \in [1..K^{inv}]$
$k \in [1,\dots,K^{vis}]$
Test Datasets: $\{X_k^{invTest}\}$
Training Datasets: $\{X_k^{invTrain}\}$
Training Datasets: $\{X_k^{visTrain}\}$
Test Datasets: $\{X_k^{visTest}\}$
Forecasting Stage
Probabilistic Short-Term Net Load Forecasting Model Construction
Bayesian Deep LSTM Networks
$\widehat{AL}_1^{test}$
$\widehat{AL}_2^{test}$
$\widehat{AL}_{(K^{inv}+K^{vis})}^{test}$
Aggregation Stage
Subprofile Aggregation
Probabilistic Aggregated Net Load Forecasts $\widehat{AL}^{test}$

Fig. 3. The overall structure of the proposed framework.

hierarchical clustering method with Ward's linkage can be found in references [44]–[46]. Subsequently, we aggregate the subprofiles in each cluster for both the *invisible* and *visible* groups to obtain the net load at a higher level:

$$AL_k^{inv(vis)} = \sum_{i \in \Omega_k^{inv(vis)}} L_i, \forall k \in 1 \cdots K^{inv(vis)} \tag{17}$$

where AL_k^{inv} and AL_k^{vis} represent the higher-level net load of the k^{th} cluster in the groups L^{inv} and L^{vis}, respectively. It is notable that the weights of each cluster, $P^{inv} = [p_1 \cdots p_{K^{inv}}]$ and $P^{vis} = [p_1 \cdots p_{K^{vis}}]$, are also saved in this stage and then will be used in the *Aggregation Stage*.

B. Feature Construction Stage

The task of the *Feature Construction Stage* is to identify the most correlated explanatory variables that contribute to forecasting and construct the training and test sets for the BDLSTM model. For short-term load forecasting, feature selection is a key procedure to obtain reliable prediction strategies by removing ineffective candidate features. Conventionally, feature selection is conducted based on either expert experience or trial-and-error procedures [47]. To automatically select the effective features, the relevance of the input features and the target variable as well as the redundancy among the candidate features are considered as the two critical information-theoretic criteria, which have been investigated in the power systems literature (e.g., [48]–[50]). Beyond that, the concept of interaction (synergy) is proposed in [47] based on the mutual information (MI) and the interaction gain (IG) to measure the interaction among candidate features of a forecast process. The effectiveness of this novel feature selection technique has been well demonstrated based on real load and price data.

Although several advanced feature selection techniques have been proposed for forecasting tasks, the development of deep learning techniques renders it possible to effectively handle raw data without the significant requirements of extensive domain expertise and careful feature design [51]. Therefore, instead of implementing or proposing novel feature selection methods, the investigation in this work focuses on the novel Bayesian deep learning technique, which has the benefit of automatically identifying the representative features based on the raw features while considering uncertainty. The integration of feature selection methods in the proposed framework will be studied in our future work to further improve the forecasting performance. As illustrated in [2], feature selection should reflect the seasonal effects, the temperature relations, and the effects of other interactions. To this end, we manually select two sets of features for the *visible* and *invisible* groups.

Given that the target is to forecast the net load at time t for cluster k in the *invisible* group, the key selected training features for $AL_{k,t}^{inv}$ include the following: 1) the net load historical data at the same time step on the previous day $AL_{k,t-24}^{inv}$; 2) $AL_{k,t-24.5}^{inv}$; 3) $AL_{k,t-25}^{inv}$; 4) the net load historical data at the same time step on the previous two days $AL_{k,t-48}^{inv}$; 5) $AL_{k,t-48.5}^{inv}$; $AL_{k,t-49}^{inv}$; 6) the hour of the day h_t; 7) the day of the week d_t; and 8) the month of the year m_t. In addition to the aforementioned features, we consider historical aggregated rooftop PV generation data $AG_{k,t-24}$, $AG_{k,t-48}$, $AG_{k,t-72}$, and $AG_{k,t-week}$ as additional features to predict the net load $AL_{k,t}^{vis}$ for cluster a k at time t in *visible* group. Afterwards, the training sets of cluster k for both the *invisible* and *visible* groups are constructed as follows:

$$X_{k,t}^{invTrain} = [AL_{k,t-24}^{invTrain}, AL_{k,t-24.5}^{invTrain}, AL_{k,t-25}^{invTrain}, AL_{k,t-48}^{invTrain}, AL_{k,t-48.5}^{invTrain}, AL_{k,t-49}^{invTrain}, h_t, d_t, m_t] \quad (18)$$

$$X_{k,t}^{visTrain} = [AL_{k,t-24}^{visTrain}, AL_{k,t-24.5}^{visTrain}, AL_{k,t-25}^{visTrain}, AL_{k,t-48}^{visTrain}, AL_{k,t-48.5}^{visTrain}, AL_{k,t-49}^{visTrain}, h_t, d_t, m_t, AG_{k,t-24}, AG_{k,t-48}, AG_{k,t-72}, AG_{k,t-week}] \quad (19)$$

where the test sets $X_k^{invTest}$ and $X_k^{visTest}$ are defined similarly to those of the training sets. In addition, the training labels, the actual net load, for a cluster k are defined as $Y_k^{invTrain}$ and $Y_k^{visTrain}$ for the invisible and visible groups, respectively.

C. Forecasting Stage

The *Forecasting Stage* is the fundamental core of the entire scheme in which a novel Bayesian deep learning method is proposed. As illustrated in Section III, BDLSTM integrates the Bayesian method with a deep LSTM network to capture both aleatoric uncertainty and epistemic uncertainty.

For each k, either in the visible group or the invisible group, the proposed BDLSTM network is trained based on the constructed features $X_k^{invTrain}$ (or $X_k^{visTrain}$) and the target labels $Y_k^{invTrain}$ (or $Y_k^{visTrain}$). When initializing the Bayesian LSTM network, the network parameters including their weights and bias values are constructed by setting up a standard normal distribution as the prior. Additionally, the hyperparameters of the deep LSTM network are optimized in this stage via grid search and cross-validation. Note that we need to construct a total of $K = K^{inv} + K^{vis}$ BDLSTM networks for each cluster. Applying the test datasets $X_k^{invTest}$ and $X_k^{visTest}$ to their corresponding models, the final outputs of this stage are the predicted aggregated net loads for each cluster $[\hat{AL}_1^{test} \cdots \hat{AL}_K^{test}]$ with a predetermined number of samples n_s for each time step.

D. Aggregation Stage

In the *Aggregation Stage*, all the individual probabilistic forecasts are aggregated through convolution with the previously saved weights to obtain the final probabilistic net load at the aggregated level, defined as $\hat{AL}^{test}$. Let $f(t)$ and $g(t)$ denote the probability density functions (PDFs) for two independent variables A and B, respectively; a convolution defined as the product of functions f and g over an infinite range, which is the probability distribution of the sum $A+B$, can be expressed as:

$$h(t) = f(t) * g(t) \triangleq \int_{-\infty}^{+\infty} f(\tau)g(t-\tau)d\tau \quad (20)$$

If A and B follow their respective Gaussian distributions

$$A \sim \mathcal{N}(\mu_A, \sigma_A^2), B \sim \mathcal{N}(\mu_B, \sigma_B^2), \quad (21)$$

then the convolution of two Gaussian distributions is another Gaussian distribution

$$C = A + B \ \sim \mathcal{N}(\mu_A + \mu_B, \sigma_A^2 + \sigma_B^2) \quad (22)$$

A detailed explanation and proof of the above equations can be found in reference [52]. In this case, the probabilistic forecast of each cluster is assumed to be independent of each other because the clustering procedure aims to differentiate the customers according to their net load patterns. Additionally, as illustrated in Section III, each individual probabilistic forecast (uncertainty component) obtained via the proposed Bayesian deep learning method follows a Gaussian distribution. Therefore, the distribution of the final aggregated net load can be directly estimated through the above convolution process, which is expressed as follows:

$$\hat{AL}^{test} \sim \mathcal{N}(\mu_1 + ... + \mu_{(K^{inv}+K^{vis})}, \sigma_1^2 + ... + \sigma_{(K^{inv}+K^{vis})}^2) \quad (23)$$

where $\hat{AL}_k^{test} \sim \mathcal{N}(\mu_k, \sigma_k^2)$ represents the sub-aggregated level net load of cluster $k \in \{1, ..., K^{inv} + K^{vis}\}$.

V. Case Study

A. Data Descriptions

The numerical experiments conducted in this study are based on real smart meter data collected from the Ausgrid distribution network, including load centers in Sydney and regional areas in NSW [53]. The Ausgrid datasets are composed of separately reported measurements of rooftop PV generation and loads at half-hour time intervals over a three-year period from 1st July 2010 to 30th June 2013. In this case, we have the training and test datasets of 21,024 observations and 480 observations, respectively, for both the load and PV generation data for all 300 customers. The target aggregated net load is directly obtained by summing the difference between customer power consumption and the PV outputs for each household. More detailed information of the Ausgrid dataset is given in the literature [53].

B. Experimental Setup

To demonstrate the superior performance of the proposed approach, a series of state-of-the-art load forecasting methods that have been widely used and firmly demonstrated with reliable performance in the literature are used for comparison. More specifically, M1 (*multiple linear regression*) [2] and M2 (*long short-term memory*) [26] are point forecasting techniques, and the rest are probabilistic models, i.e., M3 (*quantile regression*) [2], M4 (*support vector quantile regression*) [54], M5 (*gradient boosting quantile regression*) [13] and M6 (*quantile random forests*) [13]. The proposed method M7 (*BDLSTM*) is the only method that captures both the epistemic uncertainty and the aleatoric uncertainty in a single model. More specifically, the hyperparameters of the proposed BDLSTM model determined by grid searching and cross validation are given in Table I. All the tested algorithms were implemented in Python with the main packages of scikit-learn [55], Keras [56] (M1-M6) and Edward [57] (M7) and were run on an Intel Xeon PC with an NVIDIA Titan-V GPU.

TABLE I
Hyperparameters of the Proposed BDLSTM

Parameter	Value
Layer type	LSTM
Number of hidden layers	2
Number of neurons	10-20
Batch size	720
Number of epochs	150
Number of samples (T_{sample})	100
Dropout rate	0.02
Optimizer	Adam
Learning rate	0.001

C. Evaluation Metrics

Typical evaluation metrics are used to assess the forecasting performance of the examined methods (M1-M7), including the root mean square error (RMSE), the mean absolute error (MAE), the normalized root mean square deviation (NRMSD), and the mean absolute percentage error (MAPE) for point forecasting, the pinball loss function (Pinball) and the Winkler score (Winkler) for probabilistic forecasting [1], [58]. Given the actual net load AL^{test} and the predicted net load $\hat{AL}^{test}$, the aforementioned metrics are defined and formulated as below.

1) Metrics for deterministic forecasting: The RMSE measures the square root of the mean of the squares of the errors between the actual and the predicted values, which can be formulated as follows:

$$RMSE = \sqrt{\frac{\sum_{t=1}^{T}(AL_t^{test} - \hat{AL}_{p=50,t}^{test})^2}{T}} \tag{24}$$

where AL_t^{test} and $\hat{AL}_{p=50,t}^{test}$ are the actual net load and the 50th percentile value of the predicted net load, respectively, at time step t. Then, the NRMSD can be calculated as:

$$NRMSD = \frac{RMSE}{(AL_{max}^{test} - AL_{min}^{test})} \tag{25}$$

The MAE and the MAPE are calculated to quantify the absolute difference between the actual and the predicted net load in kW and percent $\%$, respectively, and are expressed as follows:

$$MAE = \frac{1}{T}\sum_{t=1}^{T}\left|AL_t^{test} - \hat{AL}_{p=50,t}^{test}\right| \tag{26}$$

$$MAPE = \frac{100\%}{T}\sum_{t=1}^{T}\left|\frac{AL_t^{test} - \hat{AL}_{p=50,t}^{test}}{AL_t^{test}}\right| \tag{27}$$

2) Metrics for probabilistic forecasting: To evaluate the performance of the probabilistic forecasting methods, the calibration, reliability, and sharpness are three main factors that indicate the consistency, the variation, and the tightness of the estimated distribution, respectively [1]. As one of the most comprehensive metrics to measure the above factors, Pinball is used in this work that can be expressed as follows:

$$Pinball = \begin{cases} (AL_t^{test} - \hat{AL}_{q,t}^{test})q & \hat{AL}_{q,t}^{test} < AL_t^{test} \\ (\hat{AL}_{q,t}^{test} - AL_t^{test})(1-q) & \hat{AL}_{q,t}^{test} > AL_t^{test} \end{cases} \tag{28}$$

Note that the average of all the Pinball values is calculated to evaluate the overall performance of the probabilistic forecasts for $q = 0.01, 0.02, ..., 0.99$, and a lower value indicates better performance.

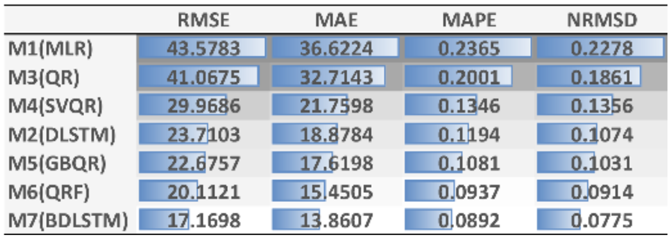
Additionally, the Winkler score is another type of comprehensive metric for probabilistic forecasting to simultaneously measure the unconditional coverage and interval width, which can be expressed as follows:

$$Winkler = \begin{cases} 2(min_t - AL_t^{test})/\alpha + \delta, & AL_t^{test} < min_t \\ 2(AL_t^{test} - max_t)/\alpha + \delta, & AL_t^{test} > max_t \\ \delta, & \text{otherwise} \end{cases} \tag{29}$$

where min_t and max_t represent the lower and upper bounds of the probabilistic forecasts at time t (i.e., $\hat{AL}_t^{test}$), respectively, and $\alpha = 0.1$ in this case. A lower score implies better probabilistic estimation results regarding the estimation interval.

D. Deterministic and Probabilistic Forecasting Results

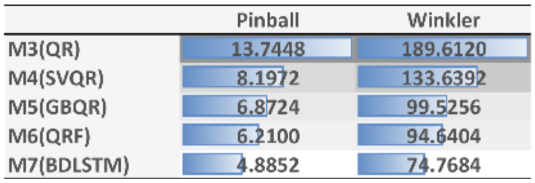
In this test, we aim to compare the forecasting performance of the proposed BDLSTM method with other popular methods in terms of both the point and probabilistic forecasting results. Note that we use the 50th percentile values for M3-M7 to evaluate their deterministic forecasting results. First, for all the considered methods, we assume that all customers belong to one cluster (i.e., K=1) and that PV data are 100% available for each individual customer. Fig. 4 presents the point forecasting results of the RMSE and the MAE in kW and the MAPE and the NRMSD in PU. The length of the bar represents the value of the evaluation metric (i.e., a higher value corresponds to a longer bar). The results show that the BDLSTM model M7 dominates with respect to the point forecasting performance, as indicated by the approximately 60.60%, 62.15%, 62.28%, and 65.98% lower RMSE, MAE, MAPE, and NRMSD, respectively, when compared with the benchmark method of multiple linear regression (M1). Moreover, the performance of the BDLSTM model also dominates when compared with the best



	RMSE	MAE	MAPE	NRMSD
M1(MLR)	43.5783	36.6224	0.2365	0.2278
M3(QR)	41.0675	32.7143	0.2001	0.1861
M4(SVQR)	29.9686	21.7598	0.1346	0.1356
M2(DLSTM)	23.7103	18.8784	0.1194	0.1074
M5(GBQR)	22.6757	17.6198	0.1081	0.1031
M6(QRF)	20.1121	15.4505	0.0937	0.0914
M7(BDLSTM)	17.1698	13.8607	0.0892	0.0775

Fig. 4. Point forecasting results for different methods.

of the state-of-the-art methods, quantile random forests (M6), with approximately 14.63 %, 10.29%, 4.8%, and 15.21% improvements in the four evaluation metrics.



	Pinball	Winkler
M3(QR)	13.7448	189.6120
M4(SVQR)	8.1972	133.6392
M5(GBQR)	6.8724	99.5256
M6(QRF)	6.2100	94.6404
M7(BDLSTM)	4.8852	74.7684

Fig. 5. Probabilistic forecasting results for different methods.

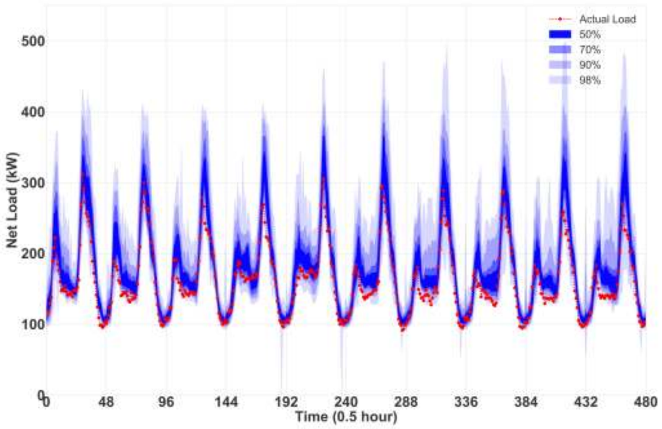


Fig. 6. Probabilistic net load forecasting results: M6 (QRF)

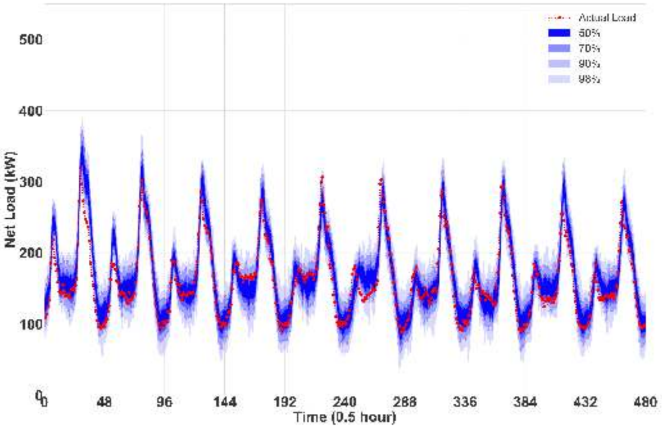


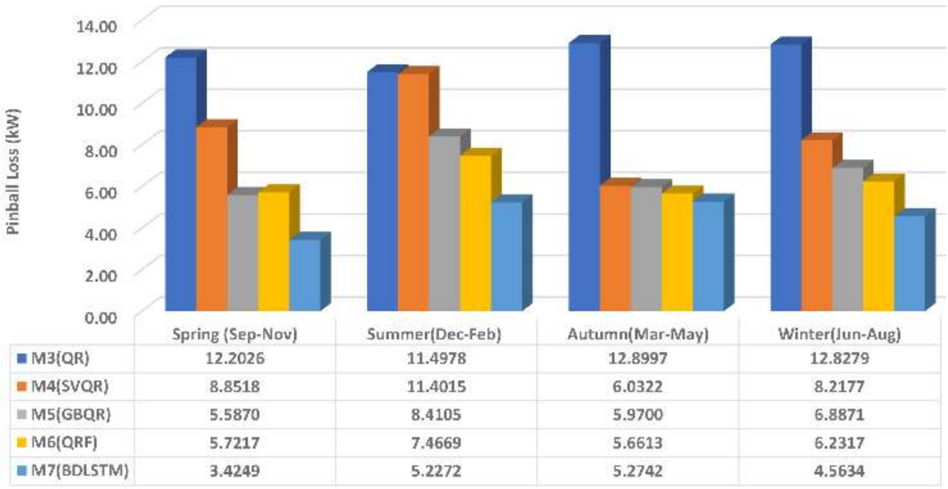
Fig. 7. Probabilistic net load forecasting results: M7 (BDLSTM)

To illustrate the effectiveness of the proposed BDLSTM method and its capability to capture uncertainty, the overall probabilistic evaluation metric values of different probabilistic methods (i.e., M3-M7) are listed in Fig. 5. The data show that the forecasting results obtained via the proposed Bayesian deep LSTM network with VI has the highest accuracy followed by the quantile random forests method (M6). The fact that M7 presents the best predictive capability indicates the significance of capturing both epistemic uncertainty and aleatoric uncertainty. As shown in Fig. 5, other methods, such as M3 and M4, perform poorly in this respect because they focus only on the uncertainty in the net load data (i.e., the aleatoric uncertainty). Another important finding is that the performance order across the different probabilistic forecasting methods is consistent with the results of point forecasting, shown in Fig. 4. For example, M7 (BDLSTM) outperforms the other tested methods, showing approximately 64.46% and 60.57% performance enhancements for the pinball loss and the Winkler score, respectively, compared with M3. Furthermore, M6 exhibits better performance than the other conventional approaches.

Additionally, Figs. 7 and 6 show the forecasting results of the 10 test days obtained via the proposed BDLSTM model and the second-best model M6 (QRF). Note that the actual net load during the tested periods is represented by the red curve with dots. The 98%, 90%, 70%, and 50% confidence intervals are indicated by an increasing color depth of the blues. In general, the probabilistic forecasting performance is evaluated in terms of three primary aspects: reliability, sharpness, and resolution [1], which have been quantified by the comprehensive evaluation criteria: the pinball loss and the Winkler score. Visually inspecting the results of M6 (QRF) and M7 (BDLSTM), the probabilistic forecasts generated using the constructed BDLSTM model present the benefits of a tighter prediction coverage interval, a lower prediction interval that varies over time, and higher unconditional coverage, corresponding to sharpness, resolution, and reliability, respectively [1]. It is constructive to highlight that the net loads during the peak hours of each day, which are crucial factors for system operation, can be well predicted with reasonable magnitudes using the proposed Bayesian deep learning method. On the other hand, it can be seen that M6 overestimates the peak demand with a misleading trend across the 10 test days.

Additionally, we expand the test datasets from 10 days to four seasons to investigate the probabilistic forecasting performance across the different seasons. Fig. 8 presents the average pinball loss values and the bar plots for all the probabilistic forecasting approaches (M3-M7). As shown, although the amount of relative improvement varies across different seasons, the proposed BDBL method (M7) consistently outperforms the other benchmark approaches, especially in spring, which exhibits a 40.14% lower average pinball loss value than that of M6 (QRF). Furthermore, compared with QRF, 29.99%, 6.83% and 26.77% improvements are obtained by using Bayesian deep learning to conduct the probabilistic net load forecasting during the periods of summer, autumn and winter, respectively.

Regarding the computational cost, the CPU times of the training process for all the examined methods are presented in Table III. The proposed BDLSTM method takes longer to train than most of the other benchmark approaches. However, it is notable that model training is an offline procedure. Given the input features, using the constructed model to conduct day-ahead forecasting only takes a few seconds in practice. Therefore, the main target in this case is to obtain an accurate forecasting result.



	Spring (Sep-Nov)	Summer(Dec-Feb)	Autumn(Mar-May)	Winter(Jun-Aug)
■ M3(QR)	12.2026	11.4978	12.8997	12.8279
■ M4(SVQR)	8.8518	11.4015	6.0322	8.2177
■ M5(GBQR)	5.5870	8.4105	5.9700	6.8871
■ M6(QRF)	5.7217	7.4669	5.6613	6.2317
■ M7(BDLSTM)	3.4249	5.2272	5.2742	4.5634

Fig. 8. The average pinball loss across different seasons.

TABLE II
Deterministic and Probabilistic Forecasting Results for BDLSTM and QLSTM

	Pinball	**Winkler**	**RMSE**	**MAE**	**MAPE**	**NRMSD**
QLSTM	6.1073	91.8621	21.9014	17.7268	0.1155	0.0993
BDLSTM	4.8852	74.7684	17.1698	13.8607	0.0892	0.0775
Relative Improvements (%)	20.01%	18.61%	21.60%	21.81%	22.77%	21.95%

TABLE III
Computational Time For Model Training

	CPU Time (s)
M1(MLR)	1.57
M2(DLSTM)	885.25
M3(QR)	53.60
M4(SVQR)	12420,67
M5(GBQR)	198.74
M6(QRF)	441.59
M7(BDLSTM)	2495.13

E. Bayesian LSTM vs Pinball Loss Guided LSTM

Recently, a series of novel deterministic deep learning-based probabilistic models have been proposed in the literature (e.g., [14], [24], [59]) to exploit deep learning to achieve state-of-the-art performance in probabilistic load forecasting. In particular, an improved wavelet neural network, a multilayer perceptron (MLP) and a deep LSTM network are considered the main networks in [59], [24] and [14], respectively. To make the comparisons based on the same type of network considered in this paper (i.e., LSTM), we implement a pinball loss guided LSTM (QLSTM) algorithm proposed in [14] in this case. More specifically, instead of using the mean square error (MSE), QLSTM employs the pinball loss as the loss function to guide the training of the parameters and thus extends traditional LSTM-based point forecasting to probabilistic forecasting in the form of quantiles. The deterministic and probabilistic forecasting results of QLSTM and BDLSTM are given in Table II. As can be seen, with the same network architecture, Bayesian deep learning exhibits a superior performance to the deterministic deep learning-based probabilistic model with approximately 20% relative improvements regarding the evaluation metric values and thus further highlighting the importance and benefit of capturing the model uncertainty.

F. Different Numbers of Clusters

After demonstrating the prominent probabilistic forecasting capability of the Bayesian deep LSTM network, this part aims to verity the effectiveness of the *Clustering Stage* in the proposed framework. In this case, we assume that all the PV data are still visible and that the number of clusters is set to $K = [1, 2, 3, 4, 5, 6]$. The point and probabilistic evaluation metrics across different Ks are shown in Fig. 9.

Most of the criteria decrease from $K = 1$ and achieve the best performance at $K = 4$ with further improvements of 3.39%, 5.99%, 8.96%, 8.77%, and 7.40% for the pinball loss, RMSE, MAE, NRMSD, and MAPE, respectively, demonstrating the importance and effectiveness of performing clustering based on the subprofiles and then aggregating to the higher-level net load. In addition, these separate clusters are all trained by the same network structure, i.e., two layers of 10 and 20 neurons in each layer, which is the best network structure for the one-cluster, 100% PV-visibility case. Therefore, further adjustment of the hyperparameters for each individual cluster may improve the forecasting performance at the aggregated level.

Furthermore, to investigate how categorizing the costumers by the invisibility of their solar power affects the prediction

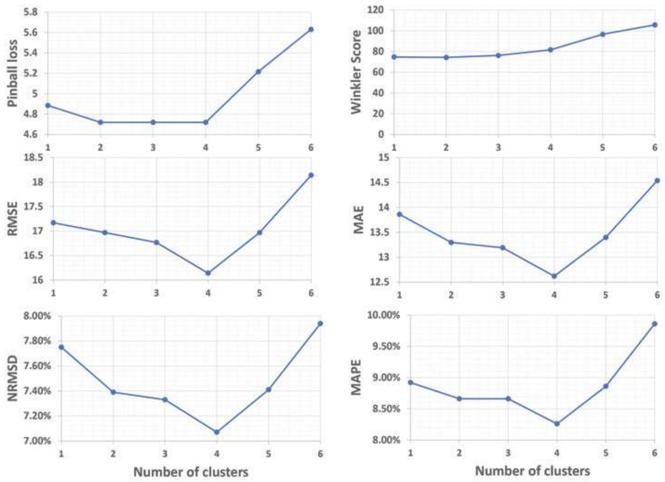


Fig. 9. Net load forecasting performance across different K.

model, an additional case study is carried out to evaluate the probabilistic forecasting performance across different combinations of K^{vis} and K^{inv} in the context of *visibility*= 50%. Table IV presents the calculated average pinball losses for the proposed BDLSTM method across different numbers of $K = K^{vis} + K^{inv}$, where $K^{inv} = 1, 2, 3$ and $K^{vis} = 1, 2, 3$. The results show that the optimal combination is $K^{inv} = 1, K^{vis} = 3$, which results in an approximately 31.14% improvement regarding the average pinball loss when compared with the no-clustering case (i.e., $K^{inv} = 1, K^{vis} = 1$). In addition, increasing the number of clusters either for the visible group or for the invisible group both lead to lower pinball losses than that of the no-clustering case, which demonstrates the effectiveness of the *Clustering Stage* in the proposed framework. Note that if the number of clusters increases to a relatively large value (e.g., $K^{inv} = 4, K^{vis} = 4$), the calculated pinball loss may become larger than that of the no-clustering case, and thus, it is imperative to select an appropriate range for K^{inv}/K^{vis} to determine the optimal combinations.

TABLE IV
The average pinball loss across different numbers of clusters (M7-BDLSTM, Visibility= 50%)

	$K^{inv} = 1$	$K^{inv} = 2$	$K^{inv} = 3$
$K^{vis} = 1$	6.2100	4.5021	4.9202
$K^{vis} = 2$	5.1478	5.4826	4.8550
$K^{vis} = 3$	4.2759	4.6060	5.8326

G. Different Levels of PV Visibility

In this part, the case study lies in investigating how and to what extent the visibility of distributed PV generation can contribute to a more accurate net load forecasting at the aggregated level. This experiment is carried out based on the assumption that $K = 1$ across various levels of PV visibility, defined by $vis = [0, 0.2, 0.4, 0.5, 0.6, 0.8, 1]$. In this case, $vis = 0$ and $vis = 1$ represent the contexts of invisible and visible PV, respectively, whereas other values indicate that PV data are partially visible. For example, $vis = 0.5$ means that 50% of the 300 households have separate meters for rooftop PV generation, and the rest of the PV outputs are not measured.

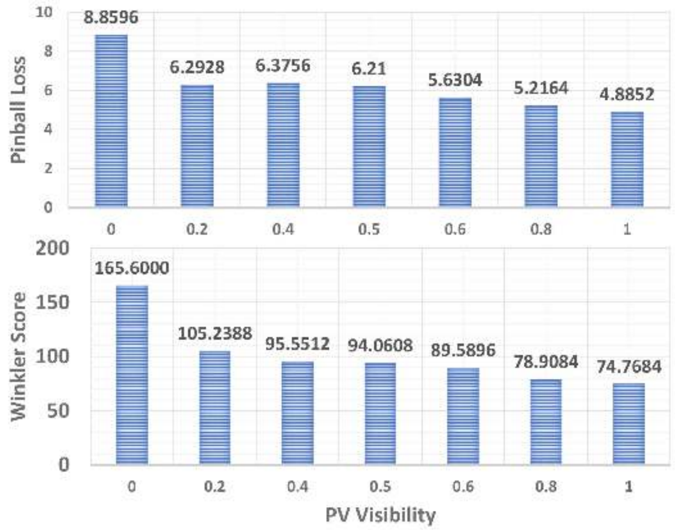
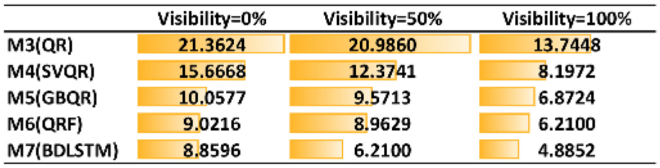


Fig. 10. Net load forecasting performance across various PV visibilities.

Fig. 10 contains the bar plots of the pinball loss and the Winkler score across different visibility levels. The primary conclusions stemming from the results are depicted as follows: i) exploiting the available PV output data from visible PV generation can enhance the forecasting performance of the net load at the aggregated level, and ii) in terms of the costs of installing meters for measuring the PV outputs separately, a trade-off between the forecasting accuracy and the PV visibility can be made based on the operator's requirements. For example, if the system operator can accept an approximately 13% lower pinball loss value ($vis = 0.6$ vs $vis = 1$), only 60% of the households need to install separate meters for rooftop PV generation, thus leading to a significant reduction in terms of the costs of the devices and their installation.



	Visibility=0%	Visibility=50%	Visibility=100%
M3(QR)	21.3624	20.9860	13.7448
M4(SVQR)	15.6668	12.3741	8.1972
M5(GBQR)	10.0577	9.5713	6.8724
M6(QRF)	9.0216	8.9629	6.2100
M7(BDLSTM)	8.8596	6.2100	4.8852

Fig. 11. The average pinball loss under different levels of PV visibility.

Finally, to demonstrate the superior performance of the proposed method under different levels of PV visibility, the pinball losses for all the tested probabilistic forecasting methods are calculated and presented in Fig 11. It can be seen that with the increasing visibility of the PV output, the probabilistic net load forecasting results of all the tested methods improve and are indicated by the reduced pinball loss values. To further enhance the performance of the proposed framework, our future work will increase the PV "visibility" by estimating the invisible PV generation using some novel invisible solar power generation estimation approaches (e.g., [60]). In addition, the results demonstrate the superiority and effectiveness of the proposed BDLSTM method across different levels of visibility.

VI. Conclusions

This paper proposes a novel probabilistic net load forecasting framework using a Bayesian deep LSTM neural network to capture epistemic uncertainty and aleatoric uncertainty simultaneously. In the proposed scheme, the *Clustering Stage* aims to enhance the forecasting performance by building a deep learning model for each individual cluster and aggregating the probabilistic forecasts of each cluster at the end to obtain the final predicted net load at the aggregated level. The effectiveness and importance of considering visible or partially visible PV output data as an input feature is investigated across different PV visibility levels. The overall performance of the proposed method is analyzed and compared with a series of state-of-the-art probabilistic forecasting models. The evaluation results demonstrate the superior performance of the proposed Bayesian deep learning-based method and highlight the improvements contributed by the *Clustering Stage* and the PV visibility.

Future work will further exploit and develop this powerful technique, Bayesian deep learning, for more challenging tasks such as net load forecasting at the household level with higher PV penetration, which exhibits higher variability and uncertainty. Additionally, with the ability to quantify both epistemic and aleatoric uncertainties, it might be helpful to use the Bayesian model as an uncertainty indicator. Hence, it can realize a self-confidence evaluation as an auxiliary safety service. In addition, selecting an appropriate prior is still an open question for Bayesian deep learning, which will also be investigated in our future work.

References

[1] T. Hong and S. Fan, "Probabilistic electric load forecasting: A tutorial review," *International Journal of Forecasting*, vol. 32, no. 3, pp. 914–938, Jul.–Sept. 2016.

[2] T. Hong, P. Wang, and H. L. Willis, "A naive multiple linear regression benchmark for short term load forecasting," in *2011 IEEE Power and Energy Society General Meeting*, Jul. 2011, pp. 1–6.

[3] P. Wang, B. Liu, and T. Hong, "Electric load forecasting with recency effect: A big data approach," *International Journal of Forecasting*, vol. 32, no. 3, pp. 585 – 597, Jul.–Sept. 2016.

[4] N. Amjady, "Short-term hourly load forecasting using time-series modeling with peak load estimation capability," *IEEE Transactions on Power Systems*, vol. 16, no. 3, pp. 498–505, Aug. 2001.

[5] N.Amjady, "Short-term bus load forecasting of power systems by a new hybrid method," *IEEE Transactions on Power Systems*, vol. 22, no. 1, pp. 333–341, Feb 2007.

[6] R. J. Hyndman and S. Fan, "Density forecasting for long-term peak electricity demand," *IEEE Transactions on Power Systems*, vol. 25, no. 2, pp. 1142–1153, May 2010.

[7] V. Dordonnat, A. Pichavant, and A. Pierrot, "Gefcom2014 probabilistic electric load forecasting using time series and semi-parametric regression models," *International Journal of Forecasting*, vol. 32, no. 3, pp. 1005 – 1011, Jul.-Sept. 2016.

[8] J. Xie, T. Hong, T. Laing, and C. Kang, "On normality assumption in residual simulation for probabilistic load forecasting," *IEEE Transactions on Smart Grid*, vol. 8, no. 3, pp. 1046–1053, May 2017.

[9] J. Xie and T. Hong, "Temperature scenario generation for probabilistic load forecasting," *IEEE Transactions on Smart Grid*, vol. 9, no. 3, pp. 1680–1687, May 2018.

[10] S. B. Taieb, R. Huser, R. J. Hyndman, and M. G. Genton, "Forecasting uncertainty in electricity smart meter data by boosting additive quantile regression," *IEEE Transactions on Smart Grid*, vol. 7, no. 5, pp. 2448–2455, Sep. 2016.

[11] B. Liu, J. Nowotarski, T. Hong, and R. Weron, "Probabilistic load forecasting via quantile regression averaging on sister forecasts," *IEEE Transactions on Smart Grid*, vol. 8, no. 2, pp. 730–737, Mar. 2017.

[12] Z. Guo, K. Zhou, X. Zhang, and S. Yang, "A deep learning model for short-term power load and probability density forecasting," *Energy*, vol. 160, pp. 1186 – 1200, Oct. 2018.

[13] Y. Wang, N. Zhang, Y. Tan, T. Hong, D. S. Kirschen, and C. Kang, "Combining probabilistic load forecasts," *IEEE Transactions on Smart Grid*, pp. 1–1, in press.

[14] Y. Wang, D. Gan, M. Sun, N. Zhang, Z. Lu, and C. Kang, "Probabilistic individual load forecasting using pinball loss guided lstm," *Applied Energy*, vol. 235, pp. 10–20, Feb. 2019.

[15] P. E. McSharry, S. Bouwman, and G. Bloemhof, "Probabilistic forecasts of the magnitude and timing of peak electricity demand," *IEEE Transactions on Power Systems*, vol. 20, no. 2, pp. 1166–1172, May 2005.

[16] M. Rafiei, T. Niknam, J. Aghaei, M. Shafie-Khah, and J. P. S. Catalao, "Probabilistic load forecasting using an improved wavelet neural network trained by generalized extreme learning machine," *IEEE Transactions on Smart Grid*, vol. 9, no. 6, pp. 6961–6971, Nov. 2018.

[17] A. Kaur, L. Nonnenmacher, and C. F. Coimbra, "Net load forecasting for high renewable energy penetration grids," *Energy*, vol. 114, pp. 1073 – 1084, Nov. 2016.

[18] Y. Wang, N. Zhang, Q. Chen, D. S. Kirschen, P. Li, and Q. Xia, "Data-driven probabilistic net load forecasting with high penetration of behind-the-meter pv," *IEEE Transactions on Power Systems*, vol. 33, no. 3, pp. 3255–3264, May 2018.

[19] S. Sepasi, E. Reihani, A. M. Howlader, L. R. Roose, and M. M. Matsuura, "Very short term load forecasting of a distribution system with high pv penetration," *Renewable Energy*, vol. 106, pp. 142 – 148, Jun. 2017.

[20] L. Jin, D. Cong, L. Guangyi, and Y. Jilai, "Short-term net feeder load forecasting of microgrid considering weather conditions," in *2014 IEEE International Energy Conference (ENERGYCON)*, May 2014, pp. 1205–1209.

[21] M. Sun, I. Konstantelos, and G. Strbac, "A deep learning-based feature extraction framework for system security assessment," *IEEE Transactions on Smart Grid*, in press.

[22] M. Sun, Y. Wang, F. Teng, Y. Ye, G. Strbac, and C. Kang, "Clustering-based residential baseline estimation: A probabilistic perspective," *IEEE Transactions on Smart Grid*, in press.

[23] E. Mocanu, P. H. Nguyen, M. Gibescu, and W. L. Kling, "Deep learning for estimating building energy consumption," *Sustainable Energy, Grids and Networks*, vol. 6, pp. 91 – 99, Jun. 2016.

[24] Z. Guo, K. Zhou, X. Zhang, and S. Yang, "A deep learning model for short-term power load and probability density forecasting," *Energy*, vol. 160, pp. 1186 – 1200, Oct. 2018.

[25] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, Dec. 1997.

[26] H. Shi, M. Xu, and R. Li, "Deep learning for household load forecasting-a novel pooling deep rnn," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 5271–5280, Sep. 2018.

[27] W. Kong, Z. Y. Dong, D. J. Hill, F. Luo, and Y. Xu, "Short-term residential load forecasting based on resident behaviour learning," *IEEE Transactions on Power Systems*, vol. 33, no. 1, pp. 1087–1088, Jan. 2018.

[28] W. Zhang, H. Quan, and D. Srinivasan, "An improved quantile regression neural network for probabilistic load forecasting," *IEEE Transactions on Smart Grid*, pp. 1–1, in press.

[29] K. Chen, K. Chen, Q. Wang, Z. He, J. Hu, and J. He, "Short-term load forecasting with deep residual networks," *IEEE Transactions on Smart Grid*, pp. 1–1, in press.

[30] S. Chang, Y. Zhang, W. Han, M. Yu, X. Guo, W. Tan, X. Cui, M. Witbrock, M. A. Hasegawa-Johnson, and T. S. Huang, "Dilated recurrent neural networks," in *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., 2017, pp. 77–87.

[31] A. Graves, N. Jaitly, and A. Mohamed, "Hybrid speech recognition with deep bidirectional lstm," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, Dec. 2013, pp. 273–278.

[32] Q. Hu, R. Zhang, and Y. Zhou, "Transfer learning for short-term wind speed prediction with deep neural networks," *Renewable Energy*, vol. 85, pp. 83 – 95, Jan. 2016.

[33] Y. Gal, "Uncertainty in deep learning," Ph.D. dissertation, University of Cambridge, 2016.

[34] Z. Ghahramani, "A history of bayesian neural networks," 2016, nIPS Workshop on Bayesian Deep Learning. [Online]. Available: http://bayesiandeeplearning.org/2016/slides/nips16bayesdeep.pdf

[35] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *Advances in neural information processing systems*, 2017, pp. 5574–5584.

[36] H. Wang and D.-Y. Yeung, "Towards bayesian deep learning: A framework and some existing methods," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 12, pp. 3395–3408, Dec. 2016.

[37] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: http://dx.doi.org/10.1162/neco.1997.9.8.1735

[38] A.Gelman, *Prior Distribution*. John Wiley Sons, 2002.

[39] A. Gelman, "Prior distributions for variance parameters in hierarchical models (Comment on Article by Browne and Draper)," *Bayesian Analysis*, vol. 1, no. 3, pp. 515–534, 2006.

[40] P. H. Garthwaite, J. B. Kadane, and A. O'Hagan, "Statistical Methods for Eliciting Probability Distributions," *Journal of the American Statistical Association*, no. 470, pp. 680–701, 2005.

[41] D. J. C. Mackay, "Bayesian Methods for Adaptive Models," Ph.D. dissertation, California Institute of Technology, 1992.

[42] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, Mar. 1951.

[43] Y. Wang, Q. Chen, M. Sun, C. Kang, and Q. Xia, "An ensemble forecasting method for the aggregated load with subprofiles," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 3906–3908, Jul. 2018.

[44] S. C. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, no. 3, pp. 241–254, Sept. 1967.

[45] J. H. Ward Jr, "Hierarchical grouping to optimize an objective function," *Journal of the American statistical association*, vol. 58, no. 301, pp. 236–244, Apr. 1963.

[46] M. Sun, F. Teng, X. Zhang, G. Strbac, and D. Pudjianto, "Data-driven representative day selection for investment decisions: A cost-oriented approach," *IEEE Transactions on Power Systems*, pp. 1–1, 2019.

[47] O. Abedinia, N. Amjady, and H. Zareipour, "A new feature selection technique for load and price forecast of electrical power systems," *IEEE Transactions on Power Systems*, vol. 32, no. 1, pp. 62–74, Jan. 2017.

[48] N. Amjady and A. Daraeepour, "Design of input vector for day-ahead price forecasting of electricity markets," *Expert Systems with Applications*, vol. 36, no. 10, pp. 12 281 – 12 294, Dec. 2009.

[49] P. A. Estevez, M. Tesmer, C. A. Perez, and J. M. Zurada, "Normalized mutual information feature selection," *IEEE Transactions on Neural Networks*, vol. 20, no. 2, pp. 189–201, Feb. 2009.

[50] N. Kwak and C.-H. Choi, "Input feature selection for classification problems," *IEEE Transactions on Neural Networks*, vol. 13, no. 1, pp. 143–159, Jan. 2002.

[51] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, May 2015.

[52] P. Bromiley, "Products and convolutions of gaussian probability density functions," *Tina-Vision Memo*, vol. 3, no. 4, p. 1, 2003.

[53] E. L. Ratnam, S. R. Weller, C. M. Kellett, and A. T. Murray, "Residential load and rooftop pv generation: an australian distribution network dataset," *International Journal of Sustainable Energy*, vol. 36, no. 8, pp. 787–806, Oct. 2017.

[54] B.-J. Chen, M.-W. Chang, and C.-J. lin, "Load forecasting using support vector machines: a study on eunite competition 2001," *IEEE Transactions on Power Systems*, vol. 19, no. 4, pp. 1821–1830, Nov. 2004.

[55] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[56] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[57] D. Tran, A. Kucukelbir, A. B. Dieng, M. Rudolph, D. Liang, and D. M. Blei, "Edward: A library for probabilistic modeling, inference, and criticism," *arXiv preprint arXiv:1610.09787*, 2016.

[58] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International Journal of Forecasting*, vol. 22, no. 4, pp. 679 – 688, Oct.-Dec. 2006.

[59] M. Rafiei, T. Niknam, J. Aghaei, M. Shafie-Khah, and J. P. S. CatalÃčo, "Probabilistic load forecasting using an improved wavelet neural network trained by generalized extreme learning machine," *IEEE Transactions on Smart Grid*, vol. 9, no. 6, pp. 6961–6971, Nov. 2018.

[60] H. Shaker, H. Zareipour, and D. Wood, "A data-driven approach for estimating the power generation of invisible solar sites," *IEEE Transactions on Smart Grid*, vol. 7, no. 5, pp. 2466–2476, Sep. 2016.

BIOGRAPHY

Mingyang Sun (M'16) received the Ph.D. degree from Imperial College London, London, U.K., in 2017. He is currently a Research Associate in this institution. His current research interests include big data analytics and artificial intelligence in energy systems.

Tingqi Zhang (S'19) received the Bachelor's degree in Electrical Engineering with Renewable Energy from University of Edinburgh, UK, and the Master's degree in Energy and Sustainability with Electrical Power Engineering from University of Southampton, U.K. He is currently working toward the Ph.D. degree in the control and power group, Imperial College London. His research focuses on probabilistic energy forecasting.

Yi Wang (S'14-M'19) received the B.S. degree from the Department of Electrical Engineering in Huazhong University of Science and Technology (HUST), Wuhan, China, in 2014 and the Ph.D. degree in Tsinghua University, Beijing, China, in 2019. He was also a visiting student researcher at the University of Washington, Seattle, WA, USA. He is currently a postdoctoral researcher in ETH Zurich. His research interests include data analytics in smart grid and multiple energy systems.

Goran Strbac (M'95) is a Professor of Electrical Energy Systems at Imperial College London, London, U.K. His research interests include electricity system operation, investment and pricing, and integration of renewable generation and distributed energy resources.

Chongqing Kang (M'01-SM'07-F'17) received the Ph.D. degree from the Department of Electrical Engineering, Tsinghua University, Beijing, China, in 1997, where he is currently a Professor.,His research interests include power system planning, power system operation, renewable energy, low-carbon electricity technology, and load forecasting.