Bayesian State Estimation for Unobservable Distribution Systems via Deep Learning

Kursat Rasim Mestav, *Member, IEEE,* Jaime Luengo-Rozas, *Member, IEEE,* and Lang Tong, *Fellow, IEEE*

Abstract—The problem of state estimation for unobservable distribution systems is considered. A deep learning approach to Bayesian state estimation is proposed for real-time applications. The proposed technique consists of distribution learning of stochastic power injection, a Monte Carlo technique for the training of a deep neural network for state estimation, and a Bayesian bad-data detection and filtering algorithm. Structural characteristics of the deep neural networks are investigated. Simulations illustrate the accuracy of Bayesian state estimation for unobservable systems and demonstrate the benefit of employing a deep neural network. Numerical results show the robustness of Bayesian state estimation against modeling and estimation errors and the presence of bad and missing data. Comparing with pseudo-measurement techniques, direct Bayesian state estimation via deep learning neural network outperforms existing benchmarks.

Index Terms—Distribution system state estimation, bad-data detection, Bayesian inference, deep learning, neural networks, smart distribution systems.

I. INTRODUCTION

We consider the problem of state estimation for power systems that have limited measurements. We are motivated by the need for achieving a higher degree of situation awareness in distribution systems where the growing presence of distributed energy resources (DER) creates exciting opportunities and daunting challenges for system operators. A compelling case can be made that effective state estimation is essential to optimize DER in real-time operations [2].

A major obstacle to state estimation in distribution systems is that such systems are nominally *unobservable* [3], [4]. By unobservable it means that there is a manifold of uncountably many states that correspond to the same measurement. System unobservability arises when the number of sensors is not sufficiently large—typical in distribution systems—or sensors are not well placed in the network. An observable system may become unobservable when sensors are at fault, sensor data missing, or data tampered by malicious agents [5].

A direct implication of unobservability is that the class of state estimators that assume deterministic system state cannot provide guarantees on the accuracy and consistency of their estimates. In particular, the popular weighted least-squares (WLS) estimator and its variants can no longer be used when

Kursat Rasim Mestav, Jaime Luengo-Rozas, and Lang Tong are with the School of Electrical and Computer Engineering, Cornell University, Ithaca, NY, 14850 USA e-mail: {krm264,jl3752,lt35}@cornell.edu.

the system is unobservable because a small WLS error in model fitting does not imply a small error in estimation; large estimation error may persist even in the absence of noise.

A standard remedy of unobservability is to use the so-called *pseudo measurements* based on interpolated observations or forecasts from historical data. Indeed, the use of pseudo measurements has been a dominant theme for distribution system state estimation. These techniques, however, are ad hoc and do not assure the quality of estimates. More significantly, historical data are often limited and have a poor temporal resolution for capturing real-time state dynamics.

The advent of smart meters and advanced metering infrastructure provide new sources of measurements. Attempts have been made to incorporate smart meter data for state estimation [6]–[8]. Not intended for state estimation, smart meters measure accumulative consumptions. They often arrive at a much slower timescale, *e.g.,* in 15-minute to hourly intervals, that is incompatible with the more rapid changes of DER. Unfortunately, existing techniques rarely address the mismatch of measurement resolution among the slow timescale smart meter data, the fast timescale real-time measurements (*e.g.,* current magnitudes at feeders and substations), and the need of fast timescale state estimation.

State estimation for unobservable systems must incorporate additional properties beyond the measurement model defined by the power flow equations. To this end, we pursue a *Bayesian inference* approach where the system states (voltage phasors) and measurements are modeled as random variables endowed with (unknown) joint probability distributions. Given the highly stochastic nature of the renewable injections, such a Bayesian model is both natural and appropriate.

The most important benefit of Bayesian inference is that observability is no longer required. A Bayesian estimator exploits probabilistic dependencies of the measurement variables on the system states; it improves the prior distribution of the states using available measurements, even if there are only a few such measurements. Unlike the least squares techniques that minimize *modeling error*, a Bayesian estimator minimizes directly the *estimation error*.

The advantage of Bayesian inference, however, comes with significant implementation issues. First, the underlying joint distribution of the system states and measurements is unknown, and some type of learning is necessary. Second, even if the relevant probability distribution is known or can be estimated, computing the actual state estimate is often intractable analytically and prohibitive computationally.

A. Summary of results and contributions

As a significant departure from the pseudo-measurement approach to distribution system state estimation, this paper presents a novel application of Bayesian state estimation for possibly unobservable systems where measurements may be unreliable, missing, or subject to data attack. Specifically, we develop a machine learning approach to the minimum mean-squared-error (MMSE) estimation of system states. A pre-estimation bad-data detection and filtering algorithm based on the Bayesian model is also proposed.

The main benefit of the proposed Bayesian state estimation is twofold. First, the online computation cost of the Bayesian estimate is several orders of magnitude lower than that of the WLS techniques, thanks to the neural network implementation of the MMSE estimator. Second, the Bayesian inference framework provides a level of flexibility to assemble information from a variety of data sources. For instance, smart meter data are not used directly in state estimation; they contribute to the learning of the probability distribution of network states. The issues of incompatible timescales, delayed measurements, and missing data are mitigated.

The proposed machine learning approach consists of *distribution learning* and *deep regression learning;* the former uses smart meter data to learn bus injection distributions from which training samples are drawn. A novel contribution is learning the distribution of fast timescale power injection from slow timescale smart meter data. In regression learning, a deep neural network is trained for the MMSE state estimation. A key innovation here is the way that the power system model (the power flow equations) is embedded in the regression learning.

Numerical results demonstrate several features of the proposed approach. First, we show that the proposed Bayesian state estimator performs considerably better than the benchmark pseudo-measurement techniques, including those generating pseudo measurements using neural networks. Second, we show that the proposed method is robust against inaccuracies in distribution learning*. Third, our results suggest that using deep learning seems essential. We observed that neural networks with five layers or more performed better than flatter networks, and neural networks with bulging middle sections performed better than rectangular ones. Finally, simulations show that the Bayesian bad-data detection and filtering are considerably more effective than the non-Bayesian pseudo-measurement techniques.

The proposed technique is of course not without limitations. Some of these limitations are summarized in Section VI.

B. Related Work

State estimation based on deterministic state models has been extensively studied. See [3], [4] and references therein. We henceforth highlight only a subset of the literature with techniques suitable for distribution systems.

In some of the earliest contributions [9]–[12], it was well recognized that a critical challenge for distribution system state estimation is the lack of observability. Different from the Bayesian solution considered in this paper, most existing approaches are two-step solutions that produce *pseudo measurements* to make the system observable followed by applying WLS and other well-established techniques.

From an estimation theoretic perspective, generating pseudo measurements can be viewed as one of forecasting the real-time measurements based on historical data. Thus the pseudo-measurement techniques are part of the so-called forecasting-aided state estimation [13], [14]. To this end, machine learning techniques that have played significant roles in load forecasting can be tailored to produce pseudo measurements. See, *e.g.,* [15]–[25].

Bayesian approaches to state estimation are far less explored even though the idea was already proposed in the seminal work of Schweppe [26]. Bayesian state estimation generally requires the computation of the conditional statistics of the state variables. An early contribution that modeled explicitly states as random was made in [27] where load distributions were used to compute moments of states, although real-time measurements were used as optimization constraints rather than as *conditioning variables* in Bayesian inference. One approach to calculating conditional statistics is based on a graphical model of the distribution system from which belief propagation techniques are used to generate state estimates [28], [29]. These techniques require a dependency graph of the system states and explicit forms of probability distributions. Another approach is based on a linear approximation of the AC power flow [30].

The approach presented in this paper belongs to the class of Monte Carlo techniques in which samples are generated and empirical conditional means computed. In our approach, instead of using Monte Carlo sampling to calculating the conditional mean directly as in [31], [32], Monte Carlo sampling is used to train a neural network that, in real-time, computes the MMSE estimate directly from the measurements.

Bad-data detection and identification has been studied extensively [3], [4]. Classical methods are *post-estimation* techniques where states are first estimated and used to compute the residue error. The presence of bad data is declared if the residue error exceeds a certain threshold. For such techniques, system observability is a prerequisite. To identify and remove bad data, an iterative process is often used where state estimation is performed repeatedly after each positive bad-data detection and removal. Such techniques often fail to identify bad data or mistakenly remove good data.

In contrast to post-estimation bad-data detection, the method proposed in this paper belongs to the less explored class of *pre-estimation* detection and filtering techniques. Several such techniques [33]–[36] are based on exploiting a dynamic model to predict the current measurement using past measurements, from which the prediction error becomes test statistics for bad-data detection. In [37], a neural network trained as an autoencoder[†] is used to test against bad data.

---

*In our simulation, we generated test data based on a distribution learned in a different year.

[†]The authors of [37] did not use the autoencoder concept to explain their approach.

II. System Model and Bayesian state estimation

A. Algebraic and Statistical Models

We adopt a generic static power flow model for an N-bus three-phase power system. We assume that node 1 is the slack bus that, for distribution systems, represents the point of common coupling (PCC), where the distribution network is connected to the main grid.

The three-phase voltage phasors at bus i is a complex column vector $x_i = [x_i^1, x_i^2, x_i^3]^\mathsf{T}$, where the superscripts are phase indices, and $x_i^k = V_i^k \angle \theta_i^k$, where V_i^k is the voltage magnitude and θ_i^k is the phase angle for the state variable at phase k of bus i. The overall system state $x = [x_1, \cdots, x_N]^\mathsf{T}$ is the column vector consisting of voltage phasors at all buses.

We adopt a static system model defined by a pair of equations that characterize the relationship among the vector of (complex) power injections $s = [s_1, \cdots, s_N]^\mathsf{T}$ at network buses, the system states x, the vector of measurements $z = [z_1, \cdots, z_M]^\mathsf{T}$, and measurement noise e:

$$x = g(s), \quad z = h(x) + e, \tag{1}$$

where $g(\cdot)$ is the mapping from net injection s to the system state x and $h(x)$ is the measurement function defined by the sensor types and the locations in the network. Vector z above includes standard types of measurements such as branch power flows, power injections, and current magnitudes. Without loss of generality, we can treat the variables in (1) as real by taking either the rectangular or the polar form of the complex variables and modifying $g(\cdot)$ and $h(\cdot)$ in (1) accordingly. The specific forms and parameters of (1) with different levels of modeling details can be found in [38]–[40].

Aside from the system model above, Bayesian estimation requires a probability model that specifies the statistical dependencies of the variables. Here we assume that the probability space is defined by the joint distribution of measurement noise e and net power injection s. We assume further that e and s are statistically independent.

B. Bayesian State Estimation

A state estimator $\hat{x}(\cdot)$ is a function of measurement z. In Bayesian estimation, the estimator $\hat{x}(z)$ is defined by the joint distribution of x and z and the adopted optimization objective. Here we focus on the minimum mean squared error (MMSE) estimator[‡] that minimizes the expected squared estimation error $\mathbb{E}(||x(z) - x||^2)$ where the (Euclidean) 2-norm is used.

From the standard estimation theory, the MMSE estimator $\hat{x}^*(z)$ is given by the mean of the state x conditional on the realization of the observation vector z:

$$\min_{\hat{x}(\cdot)} \mathbb{E}(||x - \hat{x}(z)||^2) \quad \Rightarrow \quad \hat{x}^*(z) = \mathbb{E}(x|z). \tag{2}$$

Note the difference between the MMSE and the least squares estimators:

$$\hat{x}_{\text{WLS}}(z) = \arg \min_x ||z - h(x)||^2,$$

[‡]The developed technique can be also applied to other Bayesian techniques such as the robust estimator based on the minimum absolute error (MMAE) estimator and the maximum aposteriori probability (MAP) estimator.

where the goal of least squares is to minimize the modeling error.

Simple as (2) may appear, the computation of the conditional mean can be exceedingly complex. For all practical cases, the functional form of the conditional mean is unavailable. More importantly, perhaps, the underlying joint distribution of x and z is unknown or impossible to specify, which makes the direct computation of $\hat{x}^*$ intractable.

III. Bayesian State Estimation via Deep Learning

A. Overview of Methodology

We present an overview of the proposed methodology using Fig. 1. Each functional block is explained in section(s) labeled. All variables are consistent with those used in the paper.
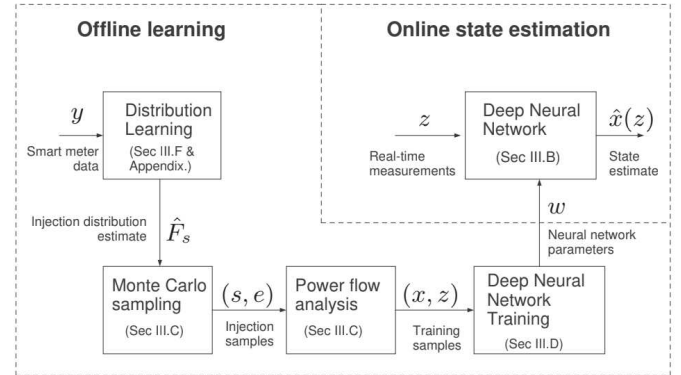

Fig. 1: Schematic of methodology.

The proposed scheme includes (i) online state estimation in the upper right of Fig. 1 and (ii) offline learning in the rest of the figure. This online-offline partition is of course not strict; the offline learning becomes online if learning continues as data arrive.

The online state estimation is through a neural-network approximation of the MMSE estimator as described in Sec III-B. The offline learning includes distribution and regression learning modules. Taking (historical) samples of the smart meter measurement y, the distribution learning module produces $\hat{F}_s$ that approximates the probability distribution of the net injection s. The three submodules at the bottom of the figure are part of the regression learning that produces parameter w of the neural network. Specifically, the Monte Carlo sampling and the power flow analysis submodules generate samples from the estimated net-injection probability distributions $\hat{F}_s$ and convert them to a set of state-measurement training samples $\{(x, z)\}$. The deep neural network training module sets the neural network parameter w via an empirical risk minimization.

B. Deep Neural Network Approximation

The MMSE state estimator in (2) can be viewed as a *nonparametric regression* with the measurement z as the regressor. Such a regression is defined on the *unknown* joint distribution of measurement z and state x. The corresponding learning problem is of infinite dimensionality and intractable.

We consider a finite dimensional approximation of (2) using a deep neural network shown in Fig 2. The neural

network consists of multiple layers of neurons. Neurons at each layer produce a vector output for the next layer using a (parameterized) nonlinear function of the output from the previous layer. The input-output relation $\mathcal{K}(\cdot; w)$ of the neural network is

$$\hat{x}(z) = \mathcal{K}(z; w), \quad (3)$$

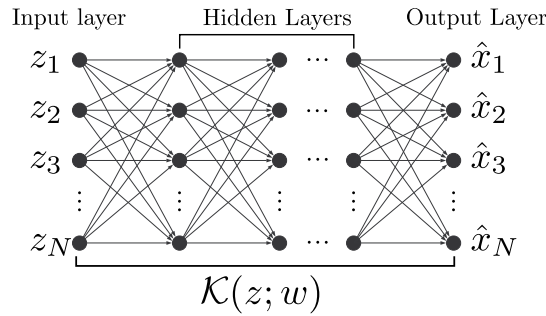where w is the parameter matrix. See [40] for the form of $\mathcal{K}$.

Fig. 2: Multi-layer Forward Neural Network.

The universal approximation theorem (see, *e.g.,* [41]) has established that a neural network with a single hidden layer is sufficient to approximate an arbitrary continuous function. This means that with a sufficiently large neural network and appropriately chosen w, a neural network can well approximate the MMSE state estimator. Under this approximation, the infinite dimensional learning problem of the conditional mean becomes a *finite dimensional* learning problem with dimensionality being the number of parameters in w.

C. Regression Learning: Training Samples

We now focus on setting the neural network parameter w to approximate the MMSE state estimator $\hat{x}^*(z)$ in (2). Standard deep learning algorithms apply to cases where there is a set of training samples made of the input-output pairs $\{(z, \hat{x}^*(z))\}$. Such pairs, unfortunately, are not available directly. Nor do we have samples from which the underlying joint distribution $F_{x,z}$ of state x and measurement z can be learned directly.

The key to obtaining a training set is to incorporate the underlying physical model characterized by model equation (1). If we can learn the (marginal) distribution F_s of the net power injection vector s, drawing an injection sample s determines the state sample $x = g(s)$ from (1). Assuming that independent measurement noise e has distribution F_e, $s \sim F_s$ and $e \sim F_e$ produce a state sample $z = h(g(s)) + e$. Conceptually,

$$(s, e) \sim F_s \times F_e \xrightarrow{(1)} (x, z). \quad (4)$$

Measurement noise distribution F_e is assumed to be, say, zero-mean Gaussian $\mathcal{N}(0, \sigma^2)$. The distribution of the net power injection, however, depends on a combination of load and possibly renewable generations. We defer the discussion of learning F_s using data from the smart meter and other measurement devices to Section III-F.

D. Regression Learning: Training Algorithm

Given the set of training samples

$$\mathcal{S} = \{(x[k], z[k]) \mid k = 1, \cdots, |\mathcal{S}|\}$$

generated according to (4), the weight matrix w is chosen to minimize the *empirical risk* defined by

$$
\begin{aligned}
L(w; \mathcal{S}) &= \frac{1}{|\mathcal{S}|} \sum_{k:(x[k], z[k]) \in \mathcal{S}} ||x[k] - \mathcal{K}(z[k]; w)||^2, \\
w^* &= \arg\min_w L(w; \mathcal{S}).
\end{aligned}
$$

The empirical risk minimization problem above is well studied for deep learning problems, and an extensive literature exists. See, *e.g.,* [42]. For the state estimation problem at hand, the class of stochastic gradient descent algorithms is considered. The Adam algorithm [43] designed for non-stationary objectives and noisy measurements is particularly suitable.

A characteristic of deep learning is over-fitting, which means that the number of neural network parameters tends to be large relative to the training data set. A general approach to overcoming over-fitting is regularization that constraints in some way the search process in neural network training. Standard techniques include L_1 regularization, dropout, and early stopping [42]. In the next section we present a regularization technique based on structural constraints on the neural network.

E. Regression Learning: Neural Network Structure

The performance of state estimation can be affected by the structure of the neural network. The "shape" of the network can vary from shallow-and-wide to deep-and-narrow, and it does not have to be rectangular. Indeed, the shape of the network plays a role in regularization.

We propose a technique that transforms a rectangular network to a non-rectangular one based on the statistical clustering of the output variables at each layer. The intuition is that, for a feedforward network, if the outputs of two neurons at the same layer are strongly correlated statistically, they can be combined. Generalizing beyond two neurons, if the outputs of a group of neurons are highly correlated, this group of neurons may be well represented by a single neuron.

As shown in Fig. 3, if neurons at a particular layer are clustered into three groups for the network in the left panel, the network on the left panel is reduced to one on the right, replacing each subgroup of neurons by its single-neuron representative.

Clustering requires a similarity measure. To this end, we considered a standard similarity measure $\rho_{X,Y}$ between random variables X and Y defined by

$$\rho_{X,Y} = 1 - \frac{\mathbb{E}(XY)}{\sqrt{\mathbb{E}(X^2)\mathbb{E}(Y^2)}}.$$

A variety of clustering techniques can be applied once a similarity measure is chosen. In particular, hierarchical agglomerative clustering [44] allows us to control the number and the sizes of clusters.

In Section V, we present numerical results on the structure of the neural network using the above clustering analysis. The
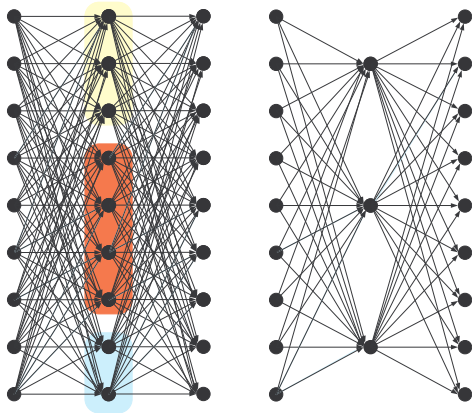
Fig. 3: A reduction of neurons in a layer through clustering. Left panel: the right three, the middle four, and right three form three clusters. Right panel: each cluster is represented by a single neuron to form a reduced network.

results from these experiments suggest that a network with a bulging middle section seems to perform better.

F. Learning Net Injection Distributions

The distribution of net injection needs to be learned to generate training samples. Distribution learning can be parametric or non-parametric [45], [46]. By restricting the distribution class, parametric techniques are well developed. The assumption on the parametric class can be wrong, however, in which case very little can be said about the performance. This makes it highly desirable that the Bayesian estimator is robust against errors in distribution learning.

Because of the historical data for injections are limited, we estimate the injection distribution based on the parametric model of Gaussian mixtures commonly used to model load and renewable generations [47]–[49]. The maximum likelihood (EM) method is used in the estimation.

If power injection can be measured locally at injection points, the injection distribution can be estimated locally without communicating the data to the central location; only the distribution parameters need to be transmitted. However, if measurements of net injection are not available, we propose an alternative technique that uses smart meter data to estimate the power injection distributions. This is nontrivial because smart meters typically measure accumulative consumptions, and smart meter data are collected at a much slower timescale. In the appendix, we present a time-series based technique that exploits the underlying structure of the Gaussian mixture distributions. This technique converts the Gaussian mixture parameters of the smart meter data distributions to distribution parameters for the fast timescale power injections.

G. Computation complexity

The computation costs of the proposed Bayesian estimator include the cost of online calculation of state estimates and that of offline learning. Specifically, to estimate an N-dimensional state vector with a neural network of fixed depth requires roughly $O(N^2)$ computations. Special hardware that exploits

massive parallelism can greatly speed up the computation [50]. In contrast, standard second-order techniques for the WLS estimator have the cost of $O(N^3)$ per iteration.

The computation cost of offline training is more difficult to quantify; it depends on the size of training data set, the algorithm used for training, and the number of iterations required to achieve some level of accuracy. Note that, for the WLS method, there is a negligible cost in obtaining a measurement sample whereas, for the Bayesian state estimation, the cost of Monte Carlo sampling is nontrivial.

For the proposed algorithm, generating a single sample requires solving the power flow equation. A standard implementation of Newton Ralphson technique requires the evaluation of the inverse of the Jacobian matrix with the (per sample) cost roughly of the order $O(N^3)$. It seems necessary that at least $O(N)$ training samples are generated, the cost of generating samples is roughly $O(N^4)$. For the stochastic gradient descent algorithm used for training, the computation cost of a neural network with fixed depth is the cost of evaluating the gradient, which is roughly $O(N^2)$.

In summary, ignoring the cost associated with iteration, we see that the online computation cost of the proposed Bayesian state estimation is considerably lower than the WLS methods, $O(N^2)$ vs. $O(N^3)$. On the other hand, if the Bayesian approach is implemented online, the cost is considerably greater than that of the WLS algorithms, $O(N^4)$ vs. $O(N^3)$.

IV. BAD-DATA DETECTION AND MITIGATION

State estimation relies on data, and the quality of data depends on the types of sensors, the quality of the collection and communications, and the possibility of cyber data attacks where measurement data are manipulated by an attacker [5], [51]. Bad-data detection and mitigation are an integral part of power system state estimation.

We first consider the simpler case of missing data that may be results of packet drops, communication delays, or that the data are deemed corrupted and removed. While no detection is needed in this case, we need to determine what to use in place of the missing data. Let the measurement vector be partitioned as $z = (\tilde{z}, z')$ where z' is the missing data. The Bayesian estimator with missing z' is given by $\hat{x}'(z) = \mathbb{E}(x|\tilde{z})$. Given the originally trained neural network $\mathcal{K}(z; w)$ that approximates $x^*(z) = \mathbb{E}(x|z)$, we have

$$\hat{x}'(z) = \mathbb{E}_{z'}(\mathbb{E}(x|\tilde{z}, z')) \approx \int \mathcal{K}((\tilde{z}, z'); w) dF_{z'},$$

where $F_{z'}$ is the cumulative distribution of z'. Thus the Bayesian state estimator with missing z' is the state estimates (without missing data) averaged over the missing data, which can be implemented by resampling missing measurements and computing the averaged state estimates. The re-sampling, however, is costly. A simple heuristic is replacing z' with estimated $\mathbb{E}(z')$ obtained in the training process (4).

Next, we consider the problem of detecting and identifying bad data that may be results of sensor malfunctions or cyberattacks. The detected bad data can then be removed, and the above solution to missing data problem can be applied.

We treat bad data as outliers. To this end, the Bayesian formulation offers a direct way to detect and identify bad data. In contrast to the conventional bad-data detection methods that are based on residue errors computed after state estimation, we propose a *pre-estimation* bad-data detection technique by exploiting the learned (prior) distributions of the measurements.

We formulate a binary hypothesis testing problem where hypothesis $\mathcal{H}_0$ models measurements without bad data and $\mathcal{H}_1$ for measurements with bad data. Consider first the simpler case when, under $\mathcal{H}_0$, measurement Y is Gaussian with mean μ_0 and variance σ_0^2. Under $\mathcal{H}_1$, Y has a different distribution with mean or variance unequal to those under $\mathcal{H}_0$. This is a composite hypothesis testing problem for which the uniformly most powerful test may not exist.

A widely used practical scheme is the Wald test [45] that evaluate the normalized deviation of the measurement away from μ_0; $\mathcal{H}_1$ (bad data) is declared when the deviation exceeds a certain threshold. Specifically, given $Y = y$, the size α Wald test is given by

$$\left| \frac{y - \mu_0}{\sigma_0} \right| \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} z_{\alpha/2} := Q^{-1}(\alpha/2),$$

where $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty \exp(-u^2/2)du$ is the Gaussian tail probability function. Typically, the size parameter α is set at $\alpha = 0.05$ to ensure that the false alarm (false positive) probability is no greater than 5%.

For the application at hand, the mean μ_0 and variance σ_0^2 of the measurement under $\mathcal{H}_0$ (no bad data) used in the Wald test are learned as a by-product of the training process described in Sec III. Specifically, the solutions of the power flow equations give directly samples of the measurements, from which the mean and variance can be estimated.

Strictly speaking, however, the measurement distributions are not Gaussian, and an alternative to the Wald test can be derived by using explicitly the learned distribution. This appears to be unnecessary from our simulation examples shown in Section V.

V. SIMULATIONS RESULTS AND DISCUSSIONS

A. Simulation Settings

a) Systems simulated: The simulations were performed on two systems defined in the MATPOWER toolbox [52]. One is an 85-bus low voltage system, the other one is the 3120-bus "Polish network". The simulation of the larger network was to demonstrate the scalability of the proposed technique and its application to mesh networks.

Two types of measurement devices were assumed: (i) Current magnitude meters were placed in 20% of the distribution system branches. (ii) One SCADA meter was placed at the slack bus to measure complex power to/from the transmission grid. The additive measurement noise was assumed to be independent and identically distributed Gaussian with zero mean and variance set at 1% of the average net consumption value.

b) Performance measure: The performance of the tested algorithms was measured by the per-node average squared error (ASE) defined by

$$\text{ASE} = \frac{1}{MN} \sum_k ||\hat{x}[k] - x[k]||^2, \tag{5}$$

where M is the number of Monte Carlo runs, k the index of the Monte Carlo run, N the number of nodes, $\hat{x}[k]$ and $x[k]$ the estimated and the state vectors, respectively.

c) Neural network specification and training: The inputs of the neural network were current magnitudes and slack bus measurements; the outputs were the state estimates. The ReLU (Rectified Linear Units) activation function was used for neurons in the hidden layers and linear activation functions in the output layer.

The Adam algorithm [43] was used to train the neural network with mini batches of 32 samples. Early stopping was applied by monitoring validation errors. To select an initial point for the optimization, He's normal method [53] was used.

B. Distribution learning

We used data sets from the Pecan Street collection[§] for distribution learning. Specifically, a data set covering May 21st to September 21st of 2015 was used for training, and a data set covering the same period in 2016 was used for testing. The data sets included power measurements from 82 households, from which 55 had solar PV installed. Power consumptions and solar PV generations were measured separately. Distributions of active power consumptions and solar generations were learned from data separately.
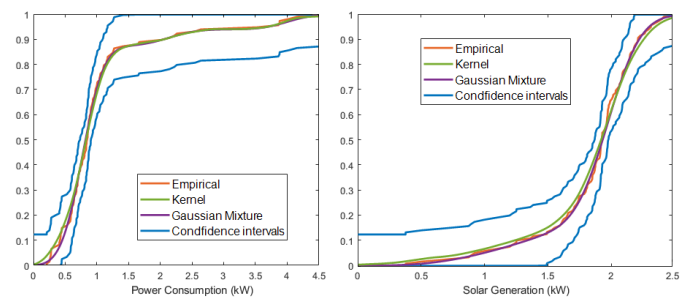

Fig. 4: Estimated cumulative distribution and 95% confidence interval.

We considered non-parametric (histogram and Kernel) and parametric distribution estimation techniques. The latter included Gaussian, Gaussian mixture, and Weibull models. Fig. 4 shows the cumulative distribution estimates of consumption and solar generation at 3 p.m. on bus 12 based on the histogram, 3-component Gaussian mixture, Kernel Epanechnikov estimators, and the 95% confidence bounds. All these estimators produced results well within the non-parametric confidence bounds obtained using the Dvoretzky-Kiefer-Wolfowitz inequality [45]. For all cases, 100% of the 3-component Gaussian mixture estimates were within the 96% level confidence bound, and 99.8% of all estimates were within

the 85% level confidence bound. We observed that Gaussian and Weibull distributions exceeded the 95% confidence interval that captures the true distribution. Our data analysis led to the adoption of the Gaussian mixture model with three components in simulations.

C. Simulation Results for an 85-bus radial network

For the 85-bus network, 82 buses were assumed to have consumptions, and 55 arbitrarily chosen buses had renewable generations. See [1] for details of network configuration. Renewable injections and load were generated separately to produce net injections. We evaluated the ASE performance as defined in (5), CPU time, and the performance of bad-data detection/filtering.

a) ASE performance: We compared the proposed *Bayesian state estimation with deep neural network* (herein abbreviated as BSEdnn) with two WLS-based pseudo-measurement methods in the literature:

1) WLSp: referred to as *WLS with pseudo measurements*, WLSp generates injection pseudo measurements by averaging the energy consumption measurement over several past samples [32];

2) WLSnnp: referred to as *WLS with neural-network generated pseudo measurements*, WLSnnp uses a neural network to generates pseudo-measurements of net injection based on a regression on the historical energy consumption vectors [18].

Separate neural networks were implemented and trained for each hour of the day. We generated 10,000 training and 10,000 validation samples. The training of the networks took on average 500 update iterations.

Fig 5 (left) shows that the ASE performance of the three state estimators on the test data for the 24 hours period. Across all hours, BSEdnn performed significantly better than the two pseudo-measurement-based WLS techniques. Specifically, the ASE of BSEdnn achieved one to two orders of magnitude lower ASE than the pseudo-measurement techniques. The performance gain was attributed to that, although measurement distributions were in some way encoded in the pseudo-measurements, WLS with pseudo measurements misused this information.

b) CPU time: The experiments were carried out on a computer Intel Core i7-8700 with 3.2 GHz processor and 64 GB RAM. For the simulation result presented in Fig. 5, the computation time of the three algorithms was measured. Estimating states using WLS with pseudo measurements took on average 22.515 seconds per Monte Carlo run. In contrast, calculating state estimate after the neural network had been trained took on average only 0.16 millisecond per Monte Carlo run, which was five orders of magnitude faster than the WLS estimator. The training algorithm was implemented in Python 3.6.5 using Keras v2.2.4 with Tensorflow v 1.12 as the backend [54]. Training the network with five layers and 2000 neurons took 23 minutes and 7 seconds.

c) Bad-data detection and filtering: The performance of bad data-detection and filtering was tested. In a Monte Carlo simulation with 1,000 runs, bad data were injected to
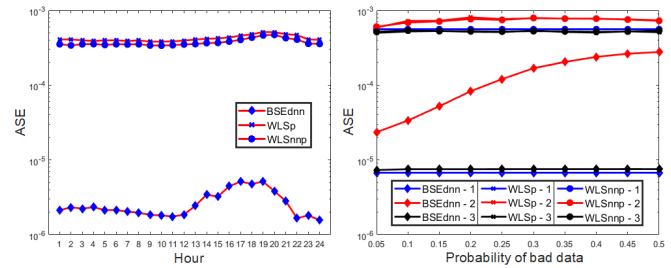

Fig. 5: ASE performance for three state estimators. Left: ASE (in absence of bad data) of estimators in different hours. Right: ASE at hour 17 in the presence of bad data of different strength; Line 1 shows ASE without bad data, Line 2 shows ASE with bad data and Line 3 shows the ASE with bad-data detection and filtering for estimators.

measurements probabilistically; each measurement had probability η to contain bad data. Different strengths of the bad data were evaluated by varying η. Note that the way bad data were introduced deviated from and was significantly more challenging than the conventional settings where a fixed number of bad data at specific locations was typically assumed.

For the bad-data model considered in Sec IV, we assumed that, under hypothesis $\mathcal{H}_1$ (bad data), the additive noise was Gaussian with zero mean and considerably larger standard deviation σ_1 than σ_0 under $\mathcal{H}_0$ (no bad data)[¶]. Once bad data were detected, BSEdnn performed a data filtering procedure that replaced the bad data with the mean of the measurements as described in Sec IV. For WLS-p and WLS-nnp, detected bad data were removed from the WLS procedure.

Fig 5 (right) shows the ASE performance vs. the probability η of bad-data occurrence under the bad-data model $\sigma_1 = 10\sigma_0$. The bad-data detection and filtering algorithms were compared in three cases. Case 1 (in blue): baseline performance without bad data; Case 2 (in red): the performance when bad data were present but not filtered; and Case 3 (in black): performance with bad-data filtering. It was evident that bad-data detection and filtering improved the performance of state estimation considerably, driving the ASE performances closer to those for the clean data case. It should also be observed that, even without bad-data filtering, BSEdnn performed better than pseudo-measurement based WLS estimators.

Additional simulations were conducted to evaluate the performance of bad-data detection. Recall that, in deriving the threshold used in the Wald test for bad-data detection in Sec IV, a Gaussian distribution of the measurement under $\mathcal{H}_0$ was assumed, which at best could only be an approximation given the Gaussian mixture nature of the net injection. To evaluate the performance of bad detection derived from the Gaussian model, we provided a comparison with the case when noise distributions were indeed Gaussian, for which the theoretical value of false alarm and miss detection probabilities could be computed analytically.

[¶]Note that neither the mean and the standard deviation nor the distribution of the bad data were assumed known in the bad-data detection and filtering algorithm.

Table I shows bad-data detection and filtering performance. The top two blocks of the table show that the threshold obtained based on the Gaussian model approximated well the false alarm and detection probabilities under the Gaussian mixture models. Also it is shown here that the pseudo-measurement schemes with WLS did not perform well. The third, fourth and fifth blocks show the effects of bad data on the ASE of state estimators. The bad-data filtering algorithm mitigated the effects of bad data satisfactorily.

	$\sigma_1 = 5\sigma_0$	$\sigma_1 = 10\sigma_0$	$\sigma_1 = 20\sigma_0$
False alarm prob. (Gaussian theoretical)	5.00%	5.00%	5.00%
False alarm rate of Wald test	4.60%	4.70%	4.70%
False alarm rate of WLSp+J(x)	4.94%	4.78%	4.91%
False alarm rate of WLSnnp+J(x)	4.94%	4.78%	4.91%
Detection prob. (Gaussian theoretical)	69.5%	84.5%	92.1%
Detection rate of Wald test	68.14%	84.16%	91.73%
Detection rate of WLSp+J(x)	36.45%	41.75%	57.33%
Detection rate of WLSnnp+J(x)	36.75%	43.37%	58.96%
ASE of BSEdnn without bad data	6.50E-06	6.50E-06	6.50E-06
ASE of WLSp without bad data	5.87E-04	5.87E-04	5.87E-04
ASE of WLSnnp MSE without bad data	5.85E-04	5.85E-04	5.85E-04
ASE of BSEdnn with bad data	1.84E-05	9.01E-05	7.31E-04
ASE of WLSp with bad data	6.97E-04	7.06E-04	7.75E-04
ASE of WLSnnp with bad data	6.98E-04	6.90E-04	7.70E-03
ASE of BSEdnn with bad-data filtering	6.91E-06	6.91E-06	6.86E-06
ASE of WLSp with bad-data filtering	5.82E-04	5.89E-04	5.79E-04
ASE of WLSnnp MSE with bad-data filtering	5.76E-04	5.82E-04	5.79E-04

TABLE I: Bad-data detection and filtering with varying strength of bad data. $\eta = 0.3$

D. Optimizing the Structure of Deep Neural Network

We examined the effects of choosing different structures of the neural network on the performance of BSEdnn. To this end, we considered the two questions: (i) Given a fixed number of neurons, is there an optimal depth of the neural network that offers the best performance? (ii) Is there a particular "shape" of the network that should be favored? We attempted to address these issues through simulations, knowing that interpretations presented here apply to the system studied and may not be conclusive.

a) The depth of deep learning: For total 1000, 2000, and 3000 neurons, we tested exhaustively different rectangular neural networks of different depths until a trend was observed. The ASE performance of BSEdnn using validation data sets is shown in Fig 6. The results suggested that a neural network of 4 to 6 layers perform the best.

b) The shape of the deep neural network: Next we examined whether using an irregular shaped neural network has an advantage. To this end, we considered a pruning approach that, starting from a rectangular network, progressively combining neurons that were highly correlated based on the clustering technique developed in Sec III. The idea was that, by combining neurons whose outputs were highly correlated, the resulting irregular neural network would be regularized within a particular structure that, potentially, could lead to faster convergence and better performance.

Table II summarizes the simulation results. First, we evaluated the rectangular structure with 400 neurons in each layer. Second, we used the clustering algorithm to prune the network, which resulted in a reduction of the number
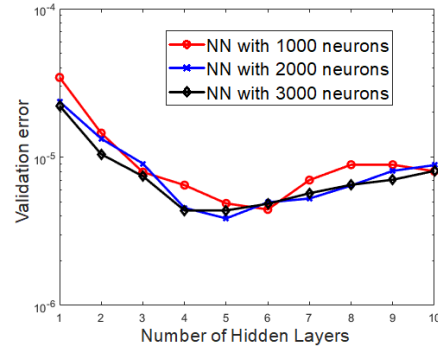

Fig. 6: NN Architecture - Total number of neurons fixed.

of neurons at the two ends of the network and improved test error. The second round of clustering-pruning resulted in further reduction training, test, and validation errors, and a shape of the network with the second and third layers having considerably more neurons than the first and last two layers. The third round of clustering-pruning resulted in increased errors, indicating the locally best clustering-pruning procedure should end at the second iteration.

	Rectangular	1st Pruning	2nd Pruning	3rd Pruning
# of Neurons 1st layer	400	399	395	391
# of Neurons 2nd layer	400	400	400	400
# of Neurons 3rd layer	400	400	399	380
# of Neurons 4th layer	400	346	278	173
# of Neurons 5th layer	400	154	64	22
Test error	6.23e-06	5.77e-06	4.45e-06	5.11e-06
Validation error	4.73e-06	4.64e-06	3.72e-06	3.85e-06
Training error	4.27e-06	4.13e-06	3.41e-06	3.73e-06

TABLE II: Pruning simulation results with the threshold = 0.0005

E. Simulation results for a 3120-bus mesh network

To demonstrate that the proposed algorithm can scale computationally for a larger network, and that the developed scheme also applies to a general mesh network, we applied our approach on the 3120-bus mesh network. For this "Polish" network, 2664 of the busses were assumed to have stochastic consumption, and 348 of the busses having stochastic generations.

In general, transmission systems are observable. To illustrate the idea of state estimation in an unobservable system, we assumed that only 20% of the power injection measurements are available. A 3-layer network with 1000 neurons in each layer is trained. Again, the neural network was trained with data generated from one power injection distribution and tested with a different power injection distribution. The injection distributions were the same used in the simulations for the 85-bus network.

We considered three test cases:

1) Normal data: power injection measurements had nominal measurement noise with standard deviation of 1% of the average net consumption value.

2) Bad data: 30% of the measurements contained bad data modeled as having measurement noise ten times of the nominal standard deviation.

3) Missing data: 30% of the data were missing.

	Training	Normal	Missing data	Bad data
ASE	2.569e-07	8.962e-06	1.082e-05	1.470e-05
Comp. Time	2 hours and 33 minutes	0.377 ms	0.358 ms	0.350 ms

TABLE III: 3120 bus system simulation results. 40,000 samples were used in training and 20,000 Monte Carlo runs were used in testing.

Table III shows the training and testing errors for the three scenarios. While standard nonlinear programming routines in MATLAB failed for the network of this size, the Bayesian state estimation implemented by a deep neural network performed well. The training and testing errors for the three cases achieved similar performance as the 85-bus system. This experiment showed that deep neural network required only millisecond level time to estimate states of the 3120 bus system, however, the training took more than 2 hours, which means that the network needed to be trained offline several hours ahead of the time of operation.

VI. CONCLUSION

This paper presents a deep learning approach to Bayesian state estimation for unobservable distribution systems under a stochastic generation and demand model. The proposed approach employs two machine learning techniques: distribution learning of power injection and regression learning of MMSE estimator. A bad-data detection algorithm developed that detects and remove bad data prior to state estimation. The use of deep neural network plays a crucial role in overcoming computation complexity in Bayesian estimation, making the online computation significantly lower than the traditional WLS solutions. Simulation results demonstrate the potential of the Bayesian state estimation for cases that are otherwise intractable for conventional WLS-based techniques.

The Bayesian approach presented here has its limitations; further research and evaluations outside the scope of this paper are needed. First, because prior distribution plays such an important role in Bayesian techniques, the estimator is less capable of adapting to changes in the network such as line and generation outages. Second, the training of deep neural networks is an area of research undergoing extensive investigation. Techniques available are largely ad hoc with little guarantee of performance. For this reason, while the Bayesian approach is sound and promising, the performance reported here are results of careful tuning of some of the design parameters for the specific systems studied.

Several extensions are possible to improve further the proposed state estimation and bad-data detection techniques. Although we focus on SCADA measurements, the approach developed here also applies to micro-PMU measurements that are not sufficient to achieve observability. It is also possible to exploit temporal dependencies and employing convolutional or recurrent neural networks.

APPENDIX

We present a technique that estimates the distribution of fast timescale power injections using slow timescale smart meter data. The main technical challenge is that smart meters measure the *cumulative energy consumption (kWh)* at a timescale slower than that of the SCADA measurements.

For convenience, assume that the fast timescale data (SCADA) are produced every interval of unit duration whereas the smart meter measurements are produced once every T intervals. Let the sequence of the smart meter measurements be Y_t. The tth interval has T fast-time measurements $X_{t1}, \cdots, X_{tT}$. We have

$$Y_t = \sum_{n=1}^{T} X_{tn}. \qquad (6)$$

Here we make the assumption that (X_{ti}) is a stationary process which makes Y_t also stationary.

Empirical studies have suggested that a good model for the probability density function (PDF) f_Y of Y_t is a Gaussian mixture, *i.e.*,

$$f_Y = \sum_{i=1}^{K} \pi_i \mathcal{N}(m_i, V_i^2),$$

where $\mathcal{N}(m, V^2)$ denotes the Gaussian distribution with mean m and covariance V^2. Such a model has the interpretation that there are K operating patterns (such as weather or the time-of-use), each can be modeled by a Gaussian process. With probability α_i, the ith pattern shows up in the interval of measurement. Under this interpretation, it is reasonable to assume that X_{ti} also follows the same underlying physical model: the distribution f_X of X_{ti} is also a Gaussian mixture with the same number of components and the same probability of appearance, *i.e.*, $f_X = \sum_{i=1}^{K} \pi_i \mathcal{N}(\mu_i, \sigma_i^2)$. Therefore, we only need to find μ_i from m_i and σ_i^2 from V_i^2.

We now fix t and a particular Gaussian mode i to derive a procedure to compute σ_i^2 from V_i^2. To avoid cumbersome notations, we drop the component index i in the Gaussian mixture and the subscript t, writing X_{tn} as X_n, V_i^2 as V^2, and σ_i^2 as σ^2.

From (6), we have immediately $\mu_i = \frac{1}{T} m_i$. Let $C_x(k)$ be the auto-covariance of X_n with lag k. Then, the variance V^2 of Y_t satisfies

$$V^2 = \sum_{i=1}^{T} \sum_{j=1}^{T} C_x(i-j) = \gamma^{\mathsf{T}} c, \qquad (7)$$

where

$$
\begin{aligned}
c &:= [C_x(0), \cdots, C_x(T-1)]^{\mathsf{T}}, \\
\gamma &:= [T, 2(T-1), 2(T-2), \cdots, 2]^{\mathsf{T}}.
\end{aligned}
$$

Our goal is to obtain $\sigma^2 = C_x(0)$ from V^2.

We now make the assumption that X_n is a stationary AR-K process defined by

$$X_n = \alpha_1 X_{n-1} + \cdots + \alpha_K X_{n-K} + \epsilon_n, \qquad (8)$$

where $\epsilon_n \in \mathcal{N}(\mu_\epsilon, \sigma_\epsilon^2)$ is the innovation sequence that is IID Gaussian. With one or more traces of X_n, the AR parameters

$\alpha = (\alpha_k)$ can be easily estimated using, *e.g.*, the least squares method.

Knowing α, the auto-covariance functions can be computed directly from a variation of the Yule-Walker equation:

$$c = A_\alpha c + \sigma_\epsilon^2 e_1 \quad \Rightarrow \quad c = \sigma_\epsilon^2 (I - A_\alpha)^{-1} e_1,$$

where $e_1 = [1, 0, \cdots, 0]^{\mathrm{T}}$, and A_α is a matrix made of entries of α. Substituting c into (7), we obtain

$$\sigma^2 = \frac{e_1^{\mathrm{T}}(I - A_\alpha)^{-1} e_1}{\gamma^{\mathrm{T}}(I - A_\alpha)^{-1} e_1} V^2.$$

To summarize, from the smart meter data Y_t, we obtain first the Gaussian mixture coefficients $\{(\pi_i, m_i, V_i^2)\}$, which gives the coefficients $\{(\pi, \mu_i, \sigma_i^2)\}$ of the Gaussian mixture of X_{tn}. Specifically,

$$\mu_i = \frac{1}{T} m_i, \quad \sigma_i^2 = \frac{e_1^{\mathrm{T}}(I - A_\alpha^{(i)})^{-1} e_1}{\gamma^{\mathrm{T}}(I - A_{\alpha^{(i)}})^{-1} e_1} V_i^2, \qquad (9)$$

where $\alpha^{(i)}$ is the coefficients of the AR process associated with component i.

REFERENCES

[1] K. R. Mestav, J. Luengo-Rozas, and L. Tong, "State estimation for unobservable distribution systems via deep neural networks," in *2018 IEEE Power Energy Society General Meeting*, July 2018.

[2] S. Lefebvre, J. Prévost, and L. Lenoir, "Distribution state estimation: A necessary requirement for the smart grid," in *2014 IEEE PES General Meeting — Conference Exposition*, July 2014, pp. 1–5.

[3] A. Abur and A. G. Expósito, *Power System State Estimation: Theory and Implementation*. CRC Press, 2004.

[4] A. Monticelli, *State Estimation in Electric Power Systems: A Generalized Approach*, ser. Power Electronics and Power Systems. Springer, 1999.

[5] O. Kosut, L. Jia, R. J. Thomas, and L. Tong, "Malicious data attacks on the smart grid," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 645–658, Dec 2011.

[6] A. Alimardani, F. Therrien, D. Atanackovic, J. Jatskevich, and E. Vaahedi, "Distribution system state estimation based on nonsynchronized smart meters," *IEEE Transactions on Smart Grid*, vol. 6, no. 6, pp. 2919–2928, Nov 2015.

[7] A. Al-Wakeel, J. Wu, and N. Jenkins, "State estimation of medium voltage distribution networks using smart meter measurements," *Applied Energy*, vol. 184, pp. 207 – 218, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306261916314349

[8] Y. Gao and N. Yu, "State estimation for unbalanced electric power distribution systems using ami data," in *2017 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, April 2017, pp. 1–5.

[9] I. Roytelman and S. M. Shahidehpour, "State estimation for electric power distribution systems in quasi real-time conditions," *IEEE Transactions on Power Delivery*, vol. 8, no. 4, pp. 2009–2015, Oct 1993.

[10] M. E. Baran and A. W. Kelley, "State estimation for real-time monitoring of distribution systems," *IEEE Transactions on Power Systems*, vol. 9, no. 3, pp. 1601–1609, Aug 1994.

[11] C. N. Lu, J. H. Teng, and W. . E. Liu, "Distribution system state estimation," *IEEE Transactions on Power Systems*, vol. 10, no. 1, pp. 229–240, Feb 1995.

[12] A. K. Ghosh, D. L. Lubkeman, M. J. Downey, and R. H. Jones, "Distribution circuit state estimation using a probabilistic approach," *IEEE Transactions on Power Systems*, vol. 12, no. 1, pp. 45–51, Feb 1997.

[13] M. B. D. C. Filho and J. C. S. de Souza, "Forecasting-aided state estimation—part i: Panorama," *IEEE Transactions on Power Systems*, vol. 24, no. 4, pp. 1667–1677, Nov 2009.

[14] M. B. D. C. Filho, J. C. S. de Souza, and R. S. Freund, "Forecasting-aided state estimation—part ii: Implementation," *IEEE Transactions on Power Systems*, vol. 24, no. 4, pp. 1678–1685, Nov 2009.

[15] A. Bernieri, C. Liguori, and A. Losi, "Neural networks and pseudo-measurements for real-time monitoring of distribution systems," in *Proceedings of 1995 IEEE Instrumentation and Measurement Technology Conference - IMTC '95*, April 1995, pp. 112–119.

[16] H. S. Hippert, C. E. Pedreira, and R. C. Souza, "Neural networks for short-term load forecasting: a review and evaluation," *IEEE Transactions on Power Systems*, vol. 16, no. 1, pp. 44–55, Feb 2001.

[17] R. Singh, B. C. Pal, and R. A. Jabr, "Distribution system state estimation through gaussian mixture model of the load as pseudo-measurement," *IET Generation, Transmission Distribution*, vol. 4, no. 1, pp. 50–59, January 2010.

[18] E. Manitsas, R. Singh, B. C. Pal, and G. Strbac, "Distribution system state estimation using an artificial neural network approach for pseudo measurement modeling," *IEEE Transactions on Power Systems*, vol. 27, no. 4, pp. 1888–1896, Nov 2012.

[19] J. Wu, Y. He, and N. Jenkins, "A robust state estimator for medium voltage distribution networks," *IEEE Transactions on Power Systems*, 2013.

[20] A. Onwuachumba, Y. Wu, and M. Musavi, "Reduced model for power system state estimation using artificial neural networks," *IEEE Green Technologies Conference*, 2013.

[21] F. Adinolfi, F. D'Agostino, A. Morini, M. Saviozzi, and F. Silvestro, "Pseudo-measurements modeling using neural network and fourier decomposition for distribution state estimation," in *IEEE PES Innovative Smart Grid Technologies, Europe*, Oct 2014, pp. 1–6.

[22] A. Abdel-Majeed, C. Kattmann, S. Tenbohlen, and R. Saur, "Usage of artificial neural networks for pseudo measurement modeling in low voltage distribution systems," in *2014 IEEE PES General Meeting — Conference Exposition*, July 2014, pp. 1–5.

[23] H. Wang, J. Ruan, G. Wang, B. Zhou, Y. Liu, X. Fu, and J. Peng, "Deep learning-based interval state estimation of ac smart grids against sparse cyber attacks," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4766–4778, Nov 2018.

[24] A. S. Zamzam, X. Fu, and N. D. Sidiropoulos, "Data-driven learning-based optimization for distribution system state estimation," *ArXiv e-prints*, 2018. [Online]. Available: https://arxiv.org/abs/1807.01671

[25] L. Zhang, G. Wang, and G. B. Giannakis, "Real-time power system state estimation and forecasting via deep neural networks," *CoRR*, vol. abs/1811.06146, 2018. [Online]. Available: http://arxiv.org/abs/1811.06146

[26] F. C. Schweppe, J. Wildes, and D. P. Rom, "Power system static state estimation, Parts I, II, III," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-89, pp. 120–135, 1970.

[27] A. K. Ghosh, D. L. Lubkeman, and R. H. Jones, "Load modeling for distribution circuit state estimation," *IEEE Transactions on Power Delivery*, vol. 12, no. 2, pp. 999–1005, April 1997.

[28] Y. Hu, A. Kuh, T. Yang, and A. Kavcic, "A belief propagation based power distribution system state estimator," *IEEE Computational Intelligence Magazine*, vol. 6, no. 3, pp. 36–46, Aug 2011.

[29] P. Chavali and A. Nehorai, "Distributed power system state estimation using factor graphs," *IEEE Transactions on Signal Processing*, vol. 63, no. 11, pp. 2864–2876, June 2015.

[30] L. Schenato, G. Barchi, D. Macii, R. Arghandeh, K. Poolla, and A. V. Meier, "Bayesian linear state estimation using smart meters and pmus measurements in distribution grids," in *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Nov 2014, pp. 572–577.

[31] K. Emami, T. Fernando, H. H. Iu, H. Trinh, and K. P. Wong, "Particle filter approach to dynamic state estimation of generators in power systems," *IEEE Transactions on Power Systems*, vol. 30, no. 5, pp. 2665–2675, Sep 2015.

[32] A. Angioni, T. Schlösser, F. Ponci, and A. Monti, "Impact of pseudo-measurements from new power profiles on state estimation in low-voltage grids," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 1, pp. 70–77, Jan 2016.

[33] D. M. Falcao, P. A. Cooke, and A. Brameller, "Power system tracking state estimation and bad data processing," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-101, no. 2, pp. 325–333, Feb 1982.

[34] K. Nishiya, J. Hasegawa, and T. Koike, "Dynamic state estimation including anomaly detection and identification for power systems," *IEE Proceedings C - Generation, Transmission and Distribution*, vol. 129, no. 5, pp. 192–198, Sep. 1982.

[35] A. Abur, A. Keyhani, and H. Bakhtiari, "Autoregressive filters for the identification and replacement of bad data in power system state estimation," *IEEE Transactions on Power Systems*, vol. 2, no. 3, pp. 552–558, Aug 1987.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPWRS.2019.2919157, IEEE Transactions on Power Systems

11

[36] M. Pignati, L. Zanni, S. Sarri, R. Cherkaoui, J. . L. Boudec, and M. Paolone, "A pre-estimation filtering process of bad data for linear power systems state estimators using pmus," in *2014 Power Systems Computation Conference*, Aug 2014, pp. 1–8.

[37] H. Salehfar and R. Zhao, "A neural network preestimation filter for bad-data detection and identification in power system state estimation," *Electric Power Systems Research*, vol. 34, no. 2, pp. 127 – 134, 1995. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0378779695009667

[38] W. H. Kersting, *Distribution System Modeling and Analysis*. CRC Press, 2007.

[39] A. Majumdar and B. C. Pal, "A three-phase state estimation in unbalanced distribution networks with switch modelling," in *2016 IEEE First International Conference on Control, Measurement and Instrumentation (CMI)*, Jan 2016, pp. 474–478.

[40] K. R. Mestav, J. Luengo-Rozas, and L. Tong, "Bayesian state estimation for unobservable distribution systems via deep neural networks," *ArXiv e-prints*, 2018. [Online]. Available: http://arxiv.org/abs/1811.02756

[41] S. Sonoda and N. Murata, "Neural network with unbounded activation functions is universal approximator," *Applied and Computational Harmonic Analysis*, vol. 43, no. 2, pp. 233 – 268, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1063520315001748

[42] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[43] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *International Conference for Learning Representations*, 2015.

[44] W. H. E. Day and H. Edelsbrunner, "Efficient algorithms for agglomerative hierarchical clustering methods," *Journal of Classification*, vol. 1, no. 1, pp. 7–24, Dec 1984. [Online]. Available: https://doi.org/10.1007/BF01890115

[45] L. Wasserman, *All of Statistics*. Springer-Verlag New York, 2004.

[46] ——, *All of Nonparametric Statistics*. Springer-Verlag New York, 2006.

[47] E. Manitsas, R. Singh, B. Pal, and G. Strbac, "Modelling of pseudo-measurements for distribution system state estimation," in *CIRED Seminar 2008: SmartGrids for Distribution*, June 2008, pp. 1–4.

[48] R. Singh, B. C. Pal, and R. A. Jabr, "Statistical representation of distribution system loads using gaussian mixture model," *IEEE Transactions on Power Systems*, vol. 25, no. 1, pp. 29–37, Feb 2010.

[49] M. J. Sanjari and H. B. Gooi, "Probabilistic forecast of pv power generation based on higher order markov chain," *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 2942–2952, July 2017.

[50] V. Sze, Y. Chen, T. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295–2329, Dec 2017.

[51] J. Kim, L. Tong, and R. J. Thomas, "Subspace methods for data attack on state estimation: A data driven approach," *IEEE Transactions on Signal Processing*, vol. 63, no. 5, pp. 1102–1114, March 2015.

[52] R. D. Zimmerman, C. E. Murillo-Sanchez, and R. J. Thomas, "Matpower: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 12–19, Feb 2011.

[53] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," *CoRR*, 2015. [Online]. Available: http://arxiv.org/abs/1502.01852

Kursat R. Mestav received the B.Sc. degree in electrical and electronics engineering from Bilkent University, Ankara, Turkey, in 2016. He is currently working toward the Ph.D. degree at Cornell University, NY, USA. His research interests include statistical inference, machine learning, and optimization, with application to energy systems and the smart grid.

Jaime Luengo-Rozas received the B.Sc. degree in electrical engineering from Comillas University ICAI, Madrid, Spain, in 2017, and the M.Eng. degree in electrical and computer engineering from Cornell University, NY, USA, in 2018, where he is currently working toward his Ph.D. degree. His research interests include machine learning and algorithm design.

Lang Tong (S'87,M'91,SM'01,F'05) is the Irwin and Joan Jacobs Professor of Engineering at Cornell University and the Cornell site Director of Power Systems Engineering Research Center (PSERC). He received the B.E. degree from Tsinghua University and the Ph.D. degrees in electrical engineering from the University of Notre Dame. He was a Postdoctoral Research Affiliate at the Information Systems Laboratory, Stanford University held visiting positions at Stanford University, the University of California at Berkeley, the Delft University of Technology, and the Chalmers University of Technology in Sweden. Lang Tong's current research focuses on data analytics, optimization, and economic problems in energy and power systems. He received the 1993 Outstanding Young Author Award from the IEEE Circuits and Systems Society, the 2004 best paper award from IEEE Signal Processing Society, and the 2004 Leonard G. Abraham Prize Paper Award from the IEEE Communications Society. He is also a coauthor of eight student paper awards. He received the Young Investigator Award from the Office of Naval Research. He was a Distinguished Lecturer of the IEEE Signal Processing Society and the 2018 Fulbright Distinguished Chair in Alternative Energy.