

Deep Neural Network-based State Estimator for Transmission System Considering Practical Implementation Challenges

Antos Cheeramban Varghese, *Student Member, IEEE*, Hritik Shah, *Student Member, IEEE*, Behrouz Azimian, *Student Member, IEEE*, Anamitra Pal, *Senior Member, IEEE*, and Evangelos Farantatos, *Senior Member, IEEE*

Abstract—As the phasor measurement unit (PMU) placement problem involves a cost-benefit trade-off, more PMUs get placed on higher-voltage buses. However, this leads to the fact that many lower-voltage levels of the bulk power system cannot be observed by PMUs. This lack of visibility then makes time-synchronized state estimation of the full system a challenging problem. In this paper, a deep neural network-based state estimator (DeNSE) is proposed to solve this problem. The DeNSE employs a Bayesian framework to indirectly combine the inferences drawn from slow-timescale but widespread supervisory control and data acquisition (SCADA) data with fast-timescale but selected PMU data, to attain sub-second situational awareness of the full system. The practical utility of the DeNSE is demonstrated by considering topology change, non-Gaussian measurement noise, and detection and correction of bad data. The results obtained using the IEEE 118-bus system demonstrate the superiority of the DeNSE over a purely SCADA state estimator and a PMU-only linear state estimator from a techno-economic viability perspective. Lastly, the scalability of the DeNSE is proven by estimating the states of a large and realistic 2000-bus synthetic Texas system.

Index Terms—Deep neural network (DNN), phasor measurement unit (PMU), state estimation, unobservability.

I. INTRODUCTION

POWER utilities attain situational awareness of their transmission system through the process of state estimation. Particularly, state estimation provides the inputs for performing real-time contingency analysis, optimal power flow,

and even network expansion planning [1]. Traditionally, state estimation is achieved using the supervisory control and data acquisition (SCADA) system. With the introduction of phasor measurement units (PMUs), SCADA-PMU hybrid state estimators as well as PMU-only linear state estimators have been proposed. Recently, it has become necessary to perform state estimation at higher speeds (<0.1 s) to understand the impacts of rapid fluctuations in outputs of converter-interfaced resources on the security of bulk power system (BPS) [2]. However, purely SCADA state estimators and SCADA-PMU hybrid state estimators are not able to provide sub-second situational awareness, while PMU-only linear state estimators require PMUs to be optimally placed throughout the system. This paper proposes a novel Bayesian framework for transmission system state estimation (TSSE) that indirectly combines the inferences drawn from slow-timescale but widespread SCADA data with fast-timescale but selected PMU data, to attain high-speed (sub-second) situational awareness of the full BPS (69 kV and above).

Due to the asynchronous nature of their inputs, purely SCADA state estimators suffer from problems such as non-linearity, divergence, and low accuracy [3]. These problems will exacerbate with increase of the penetration level of renewable generation. Hybrid state estimators directly combine data from the SCADA system and PMUs [4]-[6]. Hence, they suffer from problems such as imperfect synchronization and time-skew errors [7]. Moreover, strategies proposed to overcome some of these problems (such as those developed in [8]-[10]) are computationally intensive, which makes the hybrid state estimators operate at slower timescales [11]. PMU-only linear state estimators provide time-synchronized outputs and are extremely fast, but they require the system to be fully observed by PMUs [12]. The unobservability issue associated with PMU-only linear state estimation (LSE) is typically relegated to solving an optimal PMU placement (OPP) problem [13]-[18]. However, many OPP formulations minimize the number of PMUs, which does not result in minimization of PMU placement cost [19]. This happens because the PMU placement cost mainly includes communication, security, and labor [20], which increases with the number of substations that are upgraded for PMU placement, and not necessarily with the number of devices. Now, as the

Manuscript received: December 17, 2023; revised: February 6, 2024; accepted: March 9, 2024. Date of CrossCheck: March 9, 2024. Date of online publication: April 9, 2024.

This work was supported in part by the U.S. Department of Energy (No. DE-EE0009355), the National Science Foundation (NSF) (No. ECCS-2145063), and the Electric Power Research Institute (EPRI) (No. 10013085). The views expressed herein do not necessarily represent the views of the U.S. Department of Energy or the U.S. Government.

This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>).

A. C. Varghese, H. Shah, B. Azimian, and A. Pal (corresponding author) are with the School of Electrical, Computer, and Energy Engineering of Arizona State University, Tempe, AZ 85281, USA (email: antos.varghese@asu.edu; hshah59@asu.edu; bazimian@asu.edu; Anamitra.pal@asu.edu).

E. Farantatos is with Electric Power Research Institute (EPRI), Palo Alto, CA 94304, USA (email: efarantatos@epri.com).

DOI: 10.35833/MPCE.2023.000997



highest-voltage buses/substations are the backbone of the BPS, and these buses are fewer in number, they become the natural choice for placing the PMUs. Conversely, placing PMUs at lower voltage levels does not yield as many benefits. This cost-benefit trade-off and law of diminishing returns prevent the lower voltage levels from being fully observed by PMUs.

We have investigated the reality of the PMU unobservability problem by collecting data from two U.S. power utilities. Table I shows the PMU coverage of a U.S. power utility in the Eastern Interconnection (EI). This power utility has more than 1400 buses, but only 129 of them are equipped with PMUs. Moreover, as the voltage levels decrease, there is a sharp drop in the number of buses with PMUs to the total number of buses at that voltage level. This confirms that PMUs are mostly placed on higher-voltage buses. Lastly, from the last column of Table I, it can be realized that none of the voltage levels are fully observed by PMUs, implying that PMU-only LSE cannot be performed at any voltage level of this power utility.

TABLE I
PMU COVERAGE OF A U.S. POWER UTILITY IN EI

Voltage level (kV)	Number of buses	Number of PMU-equipped buses	Percentage of observed buses (%)
500	52	28	79
230	15	5	53
161	1185	92	27
115	42	2	10
69	144	2	3

Table II shows the PMU coverage of a U.S. power utility of the Western Electricity Coordinating Council (WECC). A key difference compared with Table I is that the third column denotes the number of PMU devices, instead of the number of PMU-equipped buses. Furthermore, it can be realized from Table II that despite having a large number of PMUs at different voltage levels, none of these levels are completely observed by PMUs. This happens because PMUs serve other functions than state estimation [13], and the cost of adding more devices at one substation is incremental [21], [22]. Therefore, power utilities add more PMUs to the same location even if they do not aid state estimation. Thus, high-speed time-synchronized state estimation for a transmission system that is only locally observed by PMUs is a challenging practical problem. In the rest of this paper, the terms “locally observable” and “(PMU)-unobservable” will be used interchangeably.

TABLE II
PMU COVERAGE OF A U.S. POWER UTILITY IN WECC

Voltage level (kV)	Number of buses	Number of PMU devices	Percentage of observed buses (%)
500	18	53	90
230	47	89	80
115	30	23	30
69	258	207	50

To counteract the impact of unobservability on state estimation, pseudo-measurements obtained by interpolated observations or forecasts obtained using historical data can be used. However, as demonstrated in [23], such methods do not ensure quality of the estimates. Recently, machine learning (ML) has been used to address the observability issues w.r.t. high-speed state estimation [24]-[26]. Reference [24] proposes a Bayesian state estimator using deep neural networks (DNNs) that is tailored for distribution systems. An ML-based state estimator for incompletely observed transmission systems is created in [25]. A state estimator with two DNNs (one for observable part and the other for unobservable part of the system) is proposed in [26]. However, [25], [26] do not consider the practicality of PMU placement when creating the ML-based state estimators.

Motivated by the knowledge gaps outlined above, we propose a deep neural network-based state estimator (DeNSE) that estimates all the transmission system voltages in a time-synchronized manner from PMUs that are only placed at the highest-voltage buses. By performing TSSE using very few PMUs, the DeNSE also circumvents the need for a massive supporting communication infrastructure [27]. Apart from the unobservability issue, this paper addresses four other practical challenges that exist w.r.t. high-speed time-synchronized TSSE as summarized below.

The first is the scalability of the state estimator. Classical LSE formulation involves a matrix inversion step, whose computational complexity is $o(n^{2.3727})$ [28]. As such, the time consumption of this implementation increases quadratically w.r.t. the number of states. Conversely, during online implementation, the forward propagation of a neural network (NN) only involves multiplication and addition operations, whose complexity $o(n \ln n)$ is much lower [29]. The second is the presence of non-Gaussian noise in PMU measurements [30]-[33]. The LSE formulation is the solution to the maximum likelihood estimation (MLE) problem under Gaussian noise environments. This means that its performance can deteriorate in the presence of non-Gaussian noise. However, an NN-based state estimator such as the DeNSE does not have such a limitation. The third is the high-speed bad data detection and correction (BDDC) [34]. Dearth of measurements makes this challenge particularly acute for the problem to be solved here. To address this challenge, a robust BDDC algorithm based on a combination of the Wald test [35] and an extreme scenario filter is developed. The fourth is topology change. This is a major concern for NN-based state estimators because it results in the training and testing environments (of the NNs) to differ, which can then deteriorate their performance. This challenge is tackled by combining DeNSE with topology processor outputs and transfer learning [24], [36].

In summary, this paper advances the state-of-the-art for time-synchronized state estimation in transmission systems by making the following salient contributions.

- 1) A high-speed time-synchronized state estimator, i. e., DeNSE, is developed for the BPS that satisfies the need to observe the full system by PMUs.
- 2) A robust BDDC algorithm is created that ensures the

performance of DeNSE under diverse types of bad data and loading conditions.

3) The ability of the DeNSE to tackle topology changes and non-Gaussian measurement noise is demonstrated.

We also provide a logical explanation along with a numerical example in Appendix A to illustrate how DeNSE can perform state estimation for unobservable power systems.

II. PROPOSED FORMULATION OF DENSE

A. Bayesian Framework for TSSE

PMU-only LSE solves a variant of the MLE problem, with the most common being the least squares formulation. However, the least squares solution requires the system equations to have full rank, which translates to the constraint of full system observability by PMUs. One way to circumvent this constraint is to reformulate the TSSE problem within a Bayesian framework, where the states \mathbf{x} and the PMU measurements \mathbf{z} are treated as random variables. Then, the following minimum mean squared error (MMSE) estimator can be formulated:

$$\min_{\hat{\mathbf{x}}(\cdot)} \mathbb{E}(\|\mathbf{x} - \hat{\mathbf{x}}(\mathbf{z})\|^2) \Rightarrow \hat{\mathbf{x}}^*(\mathbf{z}) = \mathbb{E}(\mathbf{x}|\mathbf{z}) \quad (1)$$

where $\hat{\mathbf{x}}$ is the estimated value of the states; $\hat{\mathbf{x}}^*$ is the optimal estimate; and \mathbb{E} is the expectation operator. Equation (1) directly minimizes the estimation error without the knowledge of the physical model of the system. Note that in the classical LSE formulation $\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{e}$, the modeling error is minimized, which is embedded in the measurement matrix \mathbf{H} . By avoiding the explicit need for \mathbf{H} , the observability requirement is no longer necessary in the Bayesian framework. Furthermore, by directly minimizing the estimation error, no limitations (such as Gaussian or non-Gaussian) are imposed on the characteristics of the measurement noise \mathbf{e} .

However, there are two challenges in computing the expected conditional mean of (1). First, the conditional expectation, defined by $\mathbb{E}(\mathbf{x}|\mathbf{z}) = \int_{-\infty}^{+\infty} \mathbf{x}p(\mathbf{x}|\mathbf{z})d\mathbf{x}$, requires the knowledge of the joint probability distribution function (PDF) between \mathbf{x} and \mathbf{z} , denoted by $p(\mathbf{x}, \mathbf{z})$. When the number of PMUs is scarce, $p(\mathbf{x}, \mathbf{z})$ is unknown or impossible to specify, making the direct computation of $\hat{\mathbf{x}}^*(\mathbf{z})$ intractable. Second, even if the under-lying joint PDF is known, it can be difficult to find a closed-form solution for (1). The DNN used in DeNSE overcomes these difficulties by providing an approximation of the conditional expectation of the MMSE estimator.

B. Architecture of DNN in DeNSE

The DNN has a feed-forward architecture with m inputs and n outputs, where m is the number of measurements coming from PMUs and n is the total number of states to be estimated (i. e., $\mathbf{z} \in \mathbb{R}^m$ and $\mathbf{x} \in \mathbb{R}^n$). Due to incomplete observability of the system by PMUs, $m \ll n$. The DNN has h hidden layers, in which the input vector entering the $(i+1)^{\text{th}}$ layer is expressed in terms of the inputs from the i^{th} layer as:

$$\mathbf{c}_{i+1} = \mathbf{W}_{i+1,i}\mathbf{d}_i + \mathbf{b}_{i+1} \quad (2)$$

where \mathbf{c}_{i+1} is the input vector entering the $(i+1)^{\text{th}}$ layer; $\mathbf{W}_{i+1,i}$ is the weight between the i^{th} and the $(i+1)^{\text{th}}$ layer; \mathbf{d}_i is the output of the i^{th} layer; and \mathbf{b}_{i+1} is the bias value of the $(i+1)^{\text{th}}$ layer. Next, \mathbf{c}_{i+1} is passed through an activation function \mathbf{a}_{i+1} to yield \mathbf{d}_{i+1} :

$$\mathbf{d}_{i+1} = \mathbf{a}_{i+1}(\mathbf{c}_{i+1}) \quad (3)$$

This propagation continues through all the hidden layers and the resulting value is obtained at the output layer. The loss function compares the estimated output and corresponding true output. The error between them is represented by:

$$\varepsilon_j = \zeta(\mathbf{o}_j, \hat{\mathbf{o}}_j) \quad (4)$$

where ε_j is the error; \mathbf{o}_j is the true value of the output; $\hat{\mathbf{o}}_j$ is the estimated value of the output by the DNN in the current epoch; and ζ is an appropriate loss function that indicates how well the DNN has been trained. To improve the training accuracy, ζ is minimized by optimally tuning the weights and biases through a process called backpropagation. The process is repeated until the loss becomes acceptable.

C. Creation of Training Database

A unique feature of the DeNSE that sets it apart from other ML-based state estimators (such as [37]) is that it does not use the slow timescale measurements to directly train the DNN. Instead, the discrete power injection measurements from the SCADA system are first converted into continuous functions by fitting an appropriate distribution to them. Then, independent Monte Carlo (MC) sampling is employed to randomly sample points from the distribution to feed as inputs to a power flow solver. The power flow is solved a large number of times, providing voltage and current phasor values across all system buses under various operating conditions. Then, for training, we use voltage and current phasors (with added noise) of buses which are equipped with PMUs as inputs to the DNN, while voltage phasors of all the buses are set as outputs of the DNN. This process helps in capturing the uncertainty introduced by the load variations and makes the DNN aware of diverse loading conditions.

Training the DNN by using the above-mentioned process of indirectly combining inferences from SCADA and PMU data has two advantages: ① the problem of temporal differences and synchronization issues are completely circumvented, and ② any reasonable errors in the SCADA data do not impinge on the performance of the DeNSE. The DeNSE can be impacted by noisy as well as bad PMU data since these data are input to the trained DNN during online operation. The effects of the quality of input data are investigated analytically in Section III-B, and experimentally in Sections IV-B, IV-E, and IV-F.

III. ENHANCEMENTS TO PROPOSED DENSE FRAMEWORK AND ONLINE IMPLEMENTATION

A. Transfer Learning to Handle Topology Changes

A DNN trained using the framework proposed in Section II will perform fast and accurate time-synchronized state estimation for PMU-unobservable BPS during real-time opera-

tion as long as the topology does not change. However, if the topology used for training and testing changes, the joint PDF between the measurements and the states will change; this can deteriorate the performance of the DeNSE. A possible alternative is to train the DNN from scratch for the new topology. However, it will take a very long time to do so. Instead, we use transfer learning to update the DNN of the DeNSE when topology changes. Transfer learning refers to utilizing models learned from an old problem and leveraging them for a new problem, in order to maintain the learning performance and accuracy. In the context of TSSE, transfer learning is particularly useful because when a topology changes, the mapping between measurements and states of only a small portion of the system gets altered. This implies that the re-learning will be localized.

We employ inductive transfer learning [38] to induce knowledge transfer from the old (base) topology to the new (current) topology. Four methods have been proposed for implementing inductive transfer learning: feature-representation transfer, instance transfer, relational-knowledge transfer, and parameter transfer. We use parameter transfer to update the parameters of the DNN when topology changes. Two well-known parameter transfer methods are parameter-sharing and fine-tuning. Parameter-sharing assumes that the parameters are highly transferable due to which the parameters in the source domain (old topology) can be directly copied to the target domain (new topology), where they are kept “frozen”. Fine-tuning assumes that the parameters in the source domain are useful, but they must be trained with limited target domain data to better adapt to the target domain [39]. Since there is no guarantee that the parameters of the DNN will be highly transferable for different topologies, fine-tuning is used in this paper for transfer learning.

To determine when transfer learning via fine-tuning should be implemented, we make use of the topology processor of the BPS. After updating the DNN, the new topology is designated as the base topology to make it consistent with the DeNSE. The overall implementation of transfer learning to handle topology changes is shown in Fig. 1.

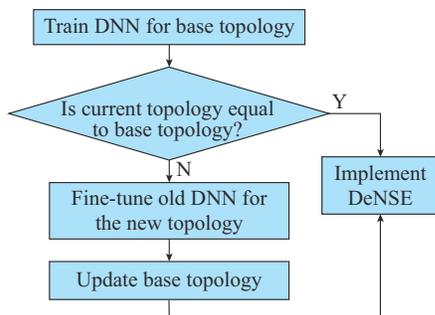


Fig. 1. Implementation of transfer learning to handle topology changes.

B. Robust BDDC

During online implementation, streaming PMU data will be fed as inputs to the proposed DeNSE framework. However, PMU data obtained from the field often suffer from bad data in the form of data dropouts and outliers [40]. This is different from measurement noise since bad data have very different

amplitudes compared with normal noisy data. To prevent such data from impacting the performance of the DeNSE, a robust BDDC algorithm capable of operating at PMU timescales (≤ 33 ms) is devised as a precursor to this state estimator.

1) BDDC Using Wald Test

A technique to detect bad data before it enters an ML-based state estimator is proposed in [23]. The technique relies on the Wald test [35] to flag incoming measurements as bad. To apply this test, two hypotheses must be defined first. ① H_0 : models the measurement without bad data and has a distribution with mean μ_0 and variance σ_0^2 , both of which are learned during training. ② H_1 : models the measurement with bad data, because of which its mean and variance are very different from those of H_0 . Mathematically, the Wald test can be expressed as:

$$\left| \frac{z - \mu_0}{\sigma_0} \right| \underset{H_0}{\overset{H_1}{>}} = Q^{-1}\left(\frac{\alpha}{2}\right) \quad (5)$$

where $Q(y) = \frac{1}{\sqrt{2\pi}} \int_y^\infty \exp\left(-\frac{u^2}{2}\right) du$ is the tail of the distribution, $y = \frac{\alpha}{2}$, and α is a tunable parameter that specifies the false positive limit. Essentially, the Wald test makes use of the fact that DNN training is done using good quality data. Hence, once the limits of good quality data become known during training, any testing data that lie outside that limit can be termed as bad. This bad data detection method based on Wald test developed in [23] is found to be compatible with the high-speed requirements of the DeNSE. However, [23] corrected the identified bad data by simply replacing them with mean value from the training database. The methodology for correcting the bad data is different, as explained below.

Since the Wald test is applied independently and simultaneously to all the m input features of a given sample of the testing dataset, it is unlikely that all the features will be bad at the same time. For a given testing dataset sample $\mathbf{z}_{\text{sample}}^{\text{test}}$, the set of indices that correspond to features flagged as bad by the Wald test are called *ibfs*. Then, if *iafs* denotes the set of indices corresponding to all the features of $\mathbf{z}_{\text{sample}}^{\text{test}}$, the difference of these two sets gives the set of indices corresponding to the good features of $\mathbf{z}_{\text{sample}}^{\text{test}}$, which is denoted by *igfs*. Now, *igfs* can be used to find that operating condition (OC) in the training database $\mathbf{Y}^{\text{train}}$ that most closely resembles the OC captured by $\mathbf{z}_{\text{sample}}^{\text{test}}$. Once that OC (called the nearest OC (NOC)) is found, its entries corresponding to *ibfs* should replace the flagged features of $\mathbf{z}_{\text{sample}}^{\text{test}}$. The overall procedure is depicted in Algorithm 1 and is performed for every sample of the testing dataset. The superiority of the proposed bad data correction method over the one where it is replaced with mean values is demonstrated in Section IV-E.

Since the Wald test is applied independently and simultaneously to all the m input features of a given sample of the testing dataset, it is unlikely that all the features will be bad at the same time. For a given testing dataset sample $\mathbf{z}_{\text{sample}}^{\text{test}}$, the set of indices that correspond to features flagged as bad by the Wald test are called *ibfs*. Then, if *iafs* denotes the set of indices corresponding to all the features of $\mathbf{z}_{\text{sample}}^{\text{test}}$, the difference of these two sets gives the set of indices corresponding to the good features of $\mathbf{z}_{\text{sample}}^{\text{test}}$, which is denoted by *igfs*. Now, *igfs* can be used to find that operating condition (OC) in the training database $\mathbf{Y}^{\text{train}}$ that most closely resembles the OC captured by $\mathbf{z}_{\text{sample}}^{\text{test}}$. Once that OC (called the nearest OC (NOC)) is found, its entries corresponding to *ibfs* should replace the flagged features of $\mathbf{z}_{\text{sample}}^{\text{test}}$. The overall procedure is depicted in Algorithm 1 and is performed for every sample of the testing dataset. The superiority of the proposed bad data correction method over the one where it is replaced with mean values is demonstrated in Section IV-E.

2) Differentiating Between Bad Data and Extreme Scenarios

The Wald test is very sensitive to the choice of α . A very small value of α may result in bad data being treated as good data, while a large value may result in an extreme scenario data being treated as bad data. This can happen because by definition, extreme scenarios are those OCs that are unlikely to occur normally. In the worst case, data corresponding to an extreme scenario will get flagged as bad data

and be replaced by normal data from the training database, making the DeNSE produce an incorrect picture of the operating state of the system. We combine our knowledge of how PMUs are placed in a power system with how extreme OCs actually manifest to design an extreme scenario filter that prevents this problem.

Algorithm 1: bad data correction using NOC in training dataset

Input: $z_{\text{sample}}^{\text{test}}, \mathbf{Y}^{\text{train}}$
Output: the corrected testing dataset sample $z_{\text{sample}}^{\text{test, crct}}$
 1: Create array of indices $iafs$ from $z_{\text{sample}}^{\text{test}}$, and set $z_{\text{sample}}^{\text{test, crct}} = z_{\text{sample}}^{\text{test}}$
 2: Conduct Wald test on $z_{\text{sample}}^{\text{test}}$ and flag the indices of bad data to create $ibfs$
 3: $\{igfs\} = \{iafs\} - \{ibfs\}$
 4: $k^* = \arg \min_k \|\mathbf{Y}^{\text{train}}[k, igfs] - z_{\text{sample}}^{\text{test}}[igfs]\|$
 5: $z_{\text{sample}}^{\text{test, crct}}[ibfs] = \mathbf{Y}^{\text{train}}[k^*, ibfs]$

Furthermore, if PMUs are placed only at the highest voltage buses (which is the premise of this paper), they will be automatically (electrically) close to each other even for PMU-unobservable BPS. This is because the highest voltage buses are connected to each other by the highest voltage lines. Thus, when an extreme scenario manifests, measurements of multiple PMUs will be simultaneously impacted. Conversely, bad data occur randomly in both space and time. This realization leads to the proposal of the following logic for designing the extreme scenario filter: if one or more features of the testing data sample are simultaneously identified as bad by the Wald test for p different PMUs, each of which is within p hops of each other, then the data sample corresponds to an extreme OC and should not be treated as bad data. This logic is implemented in the manner shown in Algorithm 2.

Note that in Algorithm 2, p indicates the severity of the extreme scenario. The higher the value of p , a greater number of hops to be considered. Lastly, the extreme scenario filter is combined with the BDDC algorithm in the following way: whenever the filter gets activated, the results of the Wald test are suppressed (i. e., no data correction occurs), and the raw PMU measurements are fed as inputs to the trained DNN of the DeNSE. The usefulness of extreme scenario filter in the DeNSE is demonstrated in Section IV-F.

Algorithm 2: implementation of extreme scenario filter

Input: features flagged as bad by Wald test, $ibfs$
Output: features passing extreme scenario filter, $ibfs_{\text{ESF}}$
 1: $ESF_{\text{ini}} = \text{PMU locations corresponding to } ibfs$
 2: $p = \text{length}(ESF_{\text{ini}})$
 3: $ibfs_{\text{ESF}} = ibfs$
 4: $ESF_p = \text{List of subsets of } ESF_{\text{ini}} \text{ with } p \text{ elements}$
 5: For $(k = 1 : \text{length}(ESF_p))$:
 If (every element of $ESF_p[k]$ is within p hops of each other):
 $Feat_{\text{ESF}} = \text{List of all features corresponding to } ESF_p[k]$
 $\{ibfs_{\text{ESF}}\} = \{ibfs_{\text{ESF}}\} - \{Feat_{\text{ESF}}\}$
 End if
 6: End for
 7: $p = p - 1$
 8: If $(ibfs_{\text{ESF}} \neq ibfs)$ or $(p < 2)$:
 9: End
 10: Else go to Step 3

C. Implementation of DeNSE

Figure 2 shows the Bayesian framework for the proposed DeNSE, where \vec{w} and \vec{b} represent the weights and bias parameters that the DNN learns during the training process, respectively. It has an offline learning phase and an online implementation phase. In the offline learning phase, appropriate distributions are fitted to historical SCADA data using Kernel density estimation (KDE). MC sampling is done from the fitted distributions and set as inputs to a power flow solver to generate training data for the DNN. The voltage and current phasors corresponding to actual PMU locations are used to train the DNN while all the voltage phasors (states) are set as outputs of the DNN. The DNN approximates the conditional expectation shown in (1). While (1) holds true for measurements in the polar or rectangular form, the DeNSE is implemented in polar form, since ① PMUs report in that form, and ② DNN is capable of approximating non-linear functions effectively (note that the relation between measurements and states in polar form is non-linear). Once the optimized DNN parameters are found, the DNN training is complete. In the online implementation phase, streaming PMU data is passed through the Wald test and a data preprocessing block (based on Section III-B), and the resulting samples are sent to the trained DNN to produce the state estimates.

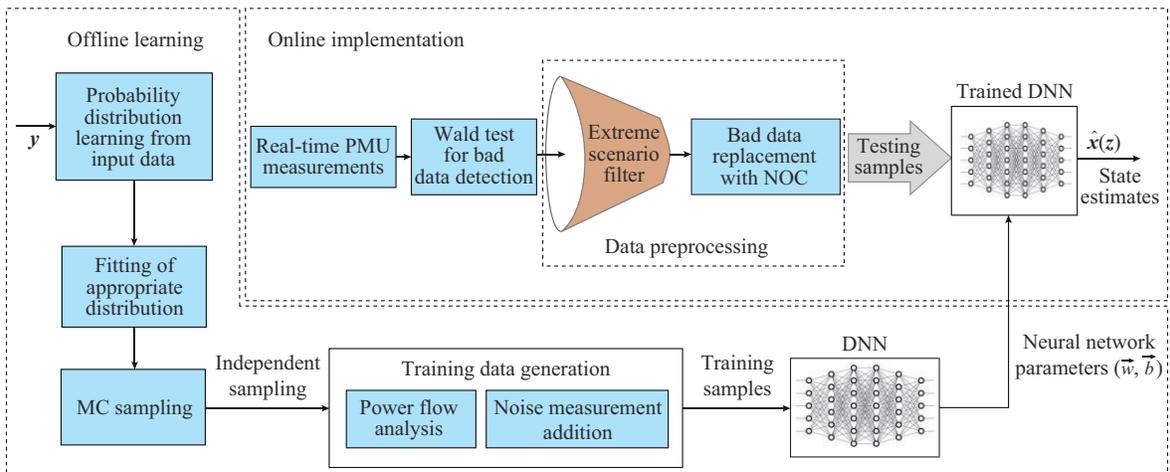


Fig. 2. Bayesian framework for proposed DeNSE.

IV. RESULTS AND DISCUSSION

A. State Estimation Results for IEEE 118-bus System

The effectiveness of the DeNSE is first illustrated using the IEEE 118-bus system. Each bus of this system is mapped to a bus in the 2000-bus synthetic Texas system [41], [42] of similar mean power rating. This is done because the Texas system has one-year of SCADA data publicly available, and this mapping helps in obtaining realistic variations in the active and reactive power for every bus of the IEEE 118-bus system. Next, the power injection distributions are found using KDE. After picking samples independently from the distributions, a power flow is solved to create the training, validation, and testing data.

It is assumed that PMUs are only placed on the highest voltage buses of this system, namely 8, 9, 10, 26, 30, 38, 63, 64, 65, 68, and 81. PMUs located at these 11 buses measure the voltage of the corresponding bus as well as the currents flowing in the lines emanating from that bus. The 41 PMU measurements (11 bus voltage phasors and 30 branch current phasors) are the inputs to the DNN. The outputs of the DNN are the 118 voltage magnitudes and angles of this system.

The training and testing of the DNN is carried out using Keras with TensorFlow as the backend library in Python [43]. Training a DNN involves finding hyperparameter values that give desired performance. The basic hyperparameters of a DNN are the number of hidden layers, the number of neurons per layer, and the activation function. The activation function used in the hidden layers is rectified linear unit (ReLU), while a linear function is used in the output layer. To overcome the problem of internal covariate shift, batch normalization is employed. Dropout regularization is used to prevent DNN overfitting. The mean squared error (MSE) loss function is used to calculate the error between the predicted and the true states. During back-propagation, the Adam optimizer is used to update the weights of the DNN. Table III summarizes the optimal values of the hyperparameters and dataset size of the DeNSE for the IEEE 118-bus system. Hyperparameter tuning is done using the ML platform WANDB [44]. All simulations are performed on a computer with 256 GB RAM, 3.40 GHz Intel Xeon 6246R CPU, Nvidia Quadro RTX 5000 GPU (16 GB). All codes for this paper can be accessed using the GitHub link provided in Appendix B.

Figure 3 shows the performance evaluation of DeNSE for the IEEE 118-bus system as a function of the distance from the buses where the PMUs are placed. The error metrics used are mean absolute percentage error (MAPE) of voltage magnitudes and mean absolute error (MAE) of voltage angles. The distance is expressed in terms of hops from the bus where the PMU is placed; i. e., a hop of zero corresponds to the 11 highest voltage buses of this system. It is clear from Fig. 3 that in comparison to conventional methods (such as LSE) that are limited to hops of zero and one (i. e., the observable regions of the system), the DeNSE is able to give reasonable state estimates even for buses that are six or seven hops away.

TABLE III
HYPERPARAMETERS AND DATASET SIZE OF DENSE FOR IEEE 118-BUS SYSTEM

Type	Name	Value
Hyperparameter	Number of hidden layers	4
	Number of neurons per hidden layer	500
	Activation functions	ReLU (hidden layers), linear (output layer)
	Loss function	MSE
	Optimizer	Adam
	Batch size	128
	Learning rate	0.0207
	Number of epochs	2,000
	Early stopping	Patience = 10
	Dropout	30%
Dataset size	Training	7500
	Validation	2500
	Testing	4000
	Total	14000

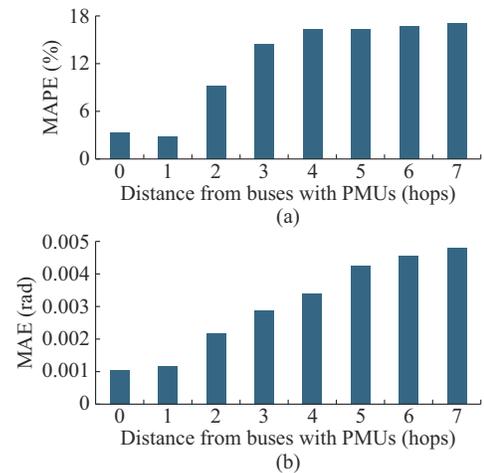


Fig. 3. Performance evaluation of DeNSE for IEEE 118-bus system as a function of distance from buses where PMUs are placed. (a) MAPE of voltage magnitude. (b) MAE of voltage angle.

B. Impact of Measurement Noise

The subplots shown in Fig. 3 are obtained under Gaussian noise environment with 1% total vector error (TVE) [45]. Now, it is important to analyze the impact that different types of noises have on the performance of a data-driven state estimator such as the DeNSE. It has recently been shown that PMU noises can have non-Gaussian characteristics [30], [31]. Keeping this in mind, three types of noise characteristics are considered in this paper, i. e., Gaussian noise, Gaussian mixture model (GMM) noise [32], and Laplacian noise [33]. The Gaussian noise has zero mean, and standard deviation of 0.0033% in magnitude and 0.0029 rad in angle. The GMM noise has two components, with mean, standard deviation, and weight vectors as [0, 0.005%], [0.0015%, 0.0015%], and [0.3, 0.7] in magnitude, and [0, 0.0043]rad, [0.0014, 0.0014]rad, and [0.3, 0.7] in angle, re-

spectively. The Laplacian noise has a location and scale of 0.001% and 0.0015% in magnitude, and 0.0009 rad and 0.0013 rad in angle, respectively. The above-mentioned noise parameters correspond to a TVE of 1%. The results obtained using the DeNSE in presence of one of these three noise types are shown in Table IV. From the table, it is observed that the DeNSE is robust enough to handle non-Gaussian measurement noise in an effective manner as there is only a very minor deterioration in performance as the noise models change.

TABLE IV
PERFORMANCE OF DENSE UNDER DIFFERENT NOISE TYPES FOR IEEE 118-BUS SYSTEM

Noise type	MAPE of voltage magnitude (%)	MAE of voltage angle (rad)
Gaussian	0.1676	0.0042
GMM	0.1667	0.0047
Laplacian	0.1678	0.0049

C. Comparison with Other State Estimators

The performance of the DeNSE is now compared with two other state estimators, namely a purely SCADA state estimator and a PMU-only linear state estimator. For fairness of comparison, 1% TVE Gaussian noise is added to all the PMU measurements. The SCADA measurements comprise all sending-end active power flows and voltage magnitudes [46], corrupted by 10% additive Gaussian noise. The linear state estimator receives PMU data from 32 buses identified from OPP studies [13]. Table V presents the average MAPE of voltage magnitudes and average MAE of voltage angles for all three state estimators. It is clear from the table that the purely SCADA-based state estimator has inferior performance compared with the DeNSE in terms of both magnitude and angle estimation. Although the PMU-only linear state estimator gives similar performance as the DeNSE, it requires almost three times the number of PMUs; moreover, these PMUs had to be placed at optimal locations in the system. Thus, considering the practical implementation challenges associated with time-synchronized TSSE, the DeNSE results are optimal from a techno-economic viability perspective.

TABLE V
COMPARISON OF DENSE WITH OTHER OPTIMIZATION-BASED STATE ESTIMATORS FOR IEEE 118-BUS SYSTEM

Type	Number of PMU locations	Average MAPE (%)	Average MAE (rad)
Purely SCADA state estimator		0.9816	0.0079
PMU-only linear state estimator	32*	0.2709	0.0026
DeNSE	11	0.1676	0.0042

Note: * means that PMUs are optimally placed to ensure complete system observability.

We have also compared the performance of DeNSE with the NN-based state estimator developed in [26]. The results are shown in Table VI. Note that in [26], PMUs are placed at 32 locations (compared with 11 locations in our case).

However, these locations are not optimally selected, resulting in five buses being unobservable in [26] for the IEEE 118-bus system. From Table VI, the following inferences are drawn. ① The proposed DeNSE has a higher root mean squared error (RMSE). This is due to the fact that the number of locations where PMUs are placed is almost one-third in our case. ② The proposed DeNSE is more robust to noise. This is because with increase in noise amplitude (standard deviation of the noise), there is a two order of magnitude increase in the RMSE values of [26], whereas there is only a 15% increase in the RMSE values of the proposed DeNSE as the noise amplitude increases.

TABLE VI
COMPARISON OF DENSE WITH NN-BASED STATE ESTIMATOR [26] FOR IEEE 118-BUS SYSTEM

Noise amplitude (standard deviation of noise)	RMSE of [26] with PMUs at 32 buses*	RMSE of DeNSE with PMUs at 11 buses
0.000	2.28×10^{-6}	6.29×10^{-3}
0.001	1.86×10^{-5}	6.60×10^{-3}
0.010	2.00×10^{-4}	6.70×10^{-3}
0.030	5.00×10^{-4}	6.94×10^{-3}
0.050	9.00×10^{-4}	7.22×10^{-3}

Note: * means that PMUs are not optimally placed (five buses left unobserved).

D. Impact of Topology Changes

Next, we investigate the ability of transfer learning in updating the DNN of DeNSE after a topology change takes place. A set of likely topologies is identified for the IEEE 118-bus system by removing one line at a time between any two buses of the system such that an island is not formed. 177 such topologies have been identified. The training data for these likely topologies are saved in the database. When a topology change is detected by the topology processor in real-time, transfer learning via fine-tuning is activated as described in Fig. 1. The results obtained are as follows.

Let the base topology be denoted by T_1 . By opening different lines, three new topologies are created from T_1 . T_2 is created by opening the line between buses 75 and 77, neither of which has a PMU. T_3 is obtained when the line between buses 38 and 37 is removed; note that bus 38 has a PMU on it. T_4 is realized by opening the line between buses 26 and 30, both of which have a PMU on them. The changes in topology and their influences on TSSE with and without transfer learning are studied, as shown in Figs. 4 and 5.

When transfer learning is used to update the DNN, fine-tuning only takes 30 s of re-training time to give similar results for the new topologies, as obtained for the base topology (the heights of the green and blue bars are similar). Note that if we had trained the DNN from scratch for every new topology, it would have taken three hours for every topology change, making the DeNSE inconsistent with the current state of the system for a much longer time period. The reason why fine-tuning is so fast is that it only needs 2000 samples and 90 epochs compared with 10000 samples and 2000 epochs that are needed to train the DNN from scratch (see Table III). Conversely, if the DNN trained for T_1 is used

throughout, the performance of DeNSE degrades significantly (shown by the heights of blue and orange bars in Figs. 4 and 5).

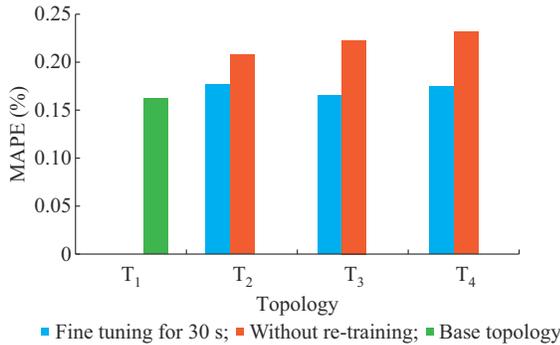


Fig. 4. Efficacy of transfer learning in terms of average MAPE of voltage magnitudes.

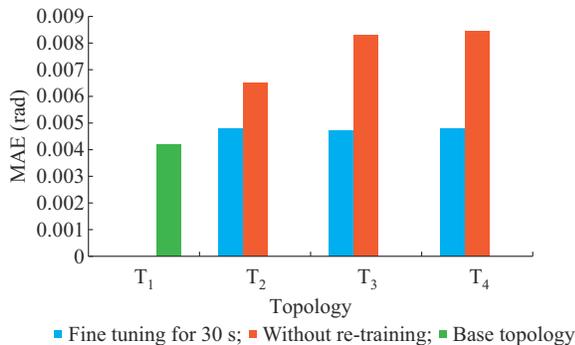


Fig. 5. Efficacy of transfer learning in terms of average MAE of voltage angles.

It can also be observed from Figs. 4 and 5 that the deterioration in estimation is more prominent for T₃ and T₄. This happened because the line that is opened for creating these two topologies has PMUs placed on one and both ends of the line, respectively. Due to this, when the line is opened, the outputs of these PMUs would become very different from what they were during the training of the DNN. This culminates in the considerable difference in the training and testing environments after the topology change occurs, causing increased deterioration in the performance of the trained DNN.

E. Mitigation of Impact of Bad Data

To investigate the performance of the proposed NOC-based BDDC algorithm, we simulate two different scenarios. In the first scenario, we increase the amount of testing samples that are bad, while fixing the severity of the bad data. To do this, the probability of bad data is randomly varied from $\eta=0\%$ to $\eta=50\%$ in steps of 10%, while the severity is kept at $\sigma=3\sigma_0$, where the standard deviation of good quality data σ_0 is computed from the training dataset. The value of α is set to be 0.05 to ensure that the false alarm (false positive) probability does not exceed 5%. The results are shown in Fig. 6 when the proposed algorithm is compared with a case where the bad data are not replaced and a case where the bad data are replaced with the mean value from

the training dataset (as done in [23]), i. e., no-replacement and replaced-by-mean cases. It is clear from Fig. 6 that in the absence of BDDC, the results become progressively worse as the amount of bad data increases (red line). Moreover, it can be observed that the bad data correction based on the NOC consistently outperforms the one that based on the mean value for both magnitude and angle estimation (the green line always lays below the blue line).

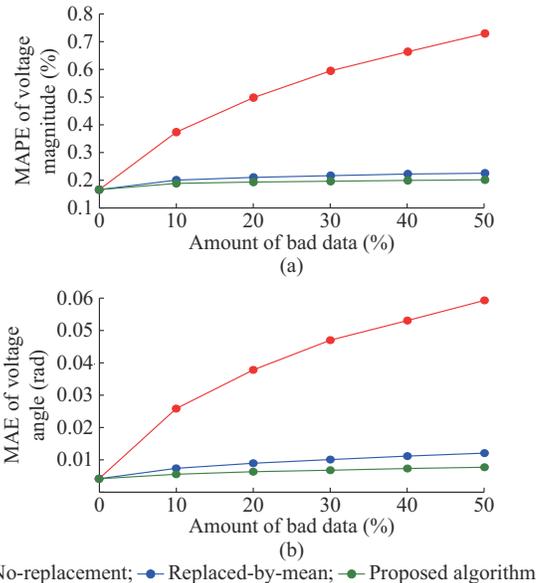


Fig. 6. Bad data replacement with increasing amount of bad data. (a) Average MAPE of voltage magnitude. (b) Average MAE of voltage angle.

In the second scenario, we increase the severity of the bad data while fixing the amount of testing samples that are bad. To do this, the severity is increased from $\sigma=3\sigma_0$ to $\sigma=7\sigma_0$, while setting $\eta=30\%$. The results are shown in Fig. 7 when the proposed algorithm is compared with the no-replacement and replaced-by-mean cases. It is clear from Fig. 6 that the proposed algorithm for correcting bad data (green line) performs much better than the no-replacement case (red line), and slightly better than the replaced-by-mean case (blue line). Lastly, note that these studies are conducted on the trained DNN created in Section IV-A, i. e., only the inputs to the DNN in the testing phase are changed while its architecture is left unaltered.

Considering the high speed at which DeNSE is expected to operate during its online implementation (30 samples per second), it must be ensured that the Wald test and data pre-processing are performed within that time frame. The most time-consuming portion in this regard is the proposed bad data correction module, which must compare the current testing sample with all the samples in the training database to find the optimal replacement(s). It is observed that with 10000 training samples and 41 phasor measurements as inputs, the bad data replacement for the IEEE 118-bus system could be carried out in (7.74 ± 0.35) ms. As this is much less than the speed at which a PMU produces an output (≈ 33 ms), the proposed algorithm meets the high speed and high accuracy expectations of purely PMU-based state estimation.

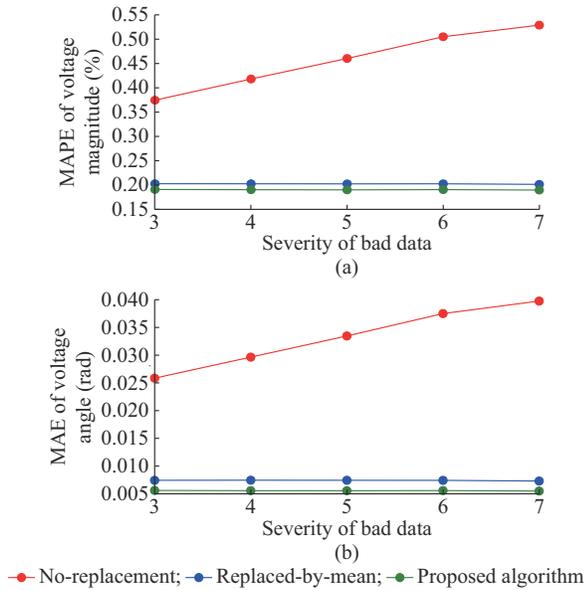


Fig. 7. Bad data replacement with increasing severity of bad data. (a) Average MAPE of voltage magnitude. (b) Average MAE of voltage angle.

F. Tackling of Extreme Scenarios

In Section IV-E, the superiority of the BDDC based on the Wald test and NOC is demonstrated. In this sub-section, the need and impact of the extreme scenario filter are discussed. 1000 extreme scenarios are created for the IEEE 118-bus system by significantly increasing the loading of buses 8 and 10. Due to the physics of the power system, PMUs located at buses 8 and 10 as well as the ones located in the vicinity of the two buses are impacted in these scenarios. Consequently, one or more measurements coming from the impacted PMUs (i.e., input features of the DeNSE) are flagged as bad data by the Wald test. At the same time, bad data are also added to the PMUs placed at buses 68 and 81, which are far away from the stressed region of the system. The extreme scenario filter identifies the set of features for which the BDDC should be suppressed, using the logic described in Section III-B. Three different outcomes are analyzed, as shown in Table VII. Note that to obtain the results shown in this table, Gaussian noise is added to all the measurements.

TABLE VII
DENSE PERFORMANCE WHEN BAD DATA AND EXTREME SCENARIO
MANIFEST SIMULTANEOUSLY IN IEEE 118-BUS SYSTEM

Method	Average MAPE of voltage magnitudes (%)		Average MAE of voltage angles (rad)	
	Mean	Standard deviation	Mean	Standard deviation
DeNSE without BDDC	0.3337	0.0254	0.0267	0.0023
DeNSE with BDDC but without extreme scenario filter	0.1853	0.0035	0.0059	0.0002
DeNSE with BDDC and extreme scenario filter	0.1812	0.0037	0.0053	0.0002

The first row of Table VII depicts the outcome obtained when bad data are not corrected. Comparing this row with the first row of Table IV, it can be seen that the results are

significantly worse. This considerable deterioration of the results is due to the presence of bad data in the measurements coming from PMUs placed at buses 68 and 81. A large amount of variability is also observed across the 1000 scenarios as captured by the high standard deviation values. The second row of Table VII depicts the outcome that obtained when BDDC takes place but without the extreme scenario filter. The relatively high errors in this case are due to the presence of extreme scenarios around buses 8 and 10, whose corresponding PMU measurements are unnecessarily replaced. The best outcome is obtained when the proposed BDDC is applied to the PMU measurements coming from buses 68 and 81, but is suppressed by the extreme scenario filter for the PMU measurements coming from the region around buses 8 and 10, as depicted in the third row of Table VII. Thus, this analysis demonstrates the robust performance of the proposed DeNSE under diverse OCs.

G. Impact of Different Database Sizes

In the proposed DeNSE, it is necessary to solve a variety of power flows under different operating conditions to create a comprehensive database for DNN training. The determination of the requisite number of samples is contingent upon the accuracy of the DNN relative to the number of samples utilized. In general, augmenting the training samples can further diminish DNN error until a point of performance saturation is reached. This is realized for the IEEE 118-bus system by progressively training the DNN with an increasing number of samples. It is observed that beyond the threshold of 10000 samples, no discernible improvement occurs, as shown in Fig. 8. Hence, we conclude that 10000 samples (i.e., sum of number of training and validation samples in Table III) are sufficient for robust performance of the DeNSE for this system.

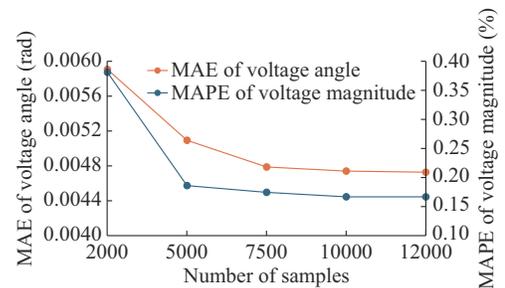


Fig. 8. Impact of database sizes on DNN performance.

H. State Estimation Results for 2000-bus Synthetic Texas System

To demonstrate the applicability of the DeNSE to large transmission systems, we use the publicly available 2000-bus synthetic Texas system [41], [42]. The number of highest-voltage buses in this system is 120, and it is assumed that PMUs are already placed on these buses such that the voltage phasors of these buses as well as the current phasors of the lines coming out of these buses are measured by PMUs. By employing the time-series data available online for this system, the training and testing data are generated and a DNN is trained using the DeNSE framework ex-

plained in Section III-C.

The error estimates obtained with PMUs placed at 120 buses and under different noise types are shown in Table VIII and Fig. 9, respectively. The outcomes presented in Table VIII correspond to a TVE of 1%, which is equivalent to a signal-to-noise ratio lying between 52 dB to 49 dB for Gaussian noise, 85 dB to 47 dB for GMM noise, and 90 dB to 85 dB for Laplacian noise, respectively. Note that LSE for this system requires the placement of PMUs at 512 optimally selected buses. It can be observed from the table that with PMUs placed at less than one-quarter of the buses ($120/512=0.234$), the DeNSE has similar performance as LSE even in presence of non-Gaussian noise in PMU measurements. From Fig. 9, it can be realized that the deterioration in the estimation performance is small even for buses that are 8 to 10 hops away. The hyperparameters and dataset size of the DeNSE for this system are summarized in Table IX. Note that the trained DNN takes only 2.6 ms on average to produce the state estimates. This validates the ability of the DeNSE to estimate the states of large systems at high speeds.

TABLE VIII
PERFORMANCE OF DENSE AND LSE UNDER DIFFERENT NOISE TYPES FOR 2000-BUS SYNTHETIC TEXAS SYSTEM

Method (noise type)	Average MAPE of voltage magnitudes (%)	Average MAE of voltage angles (rad)	Number of buses with PMUs
LSE (Gaussian)	0.2809	0.0026	512*
DeNSE (Gaussian)	0.2800	0.0024	120
DeNSE (GMM)	0.2714	0.0024	120
DeNSE (Laplacian)	0.2890	0.0027	120

Note: * means that PMUs are optimally placed to ensure complete system observability.

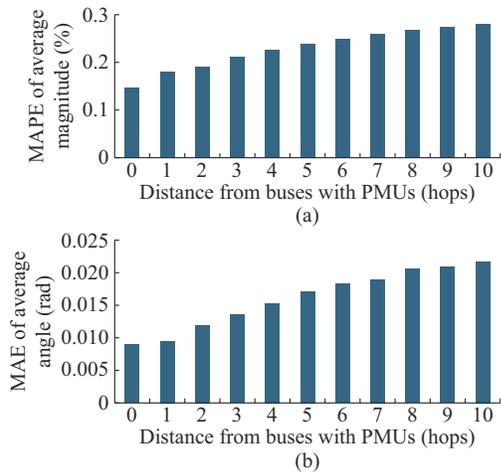


Fig. 9. Performance evaluation of DeNSE for 2000-bus synthetic Texas system as a function of distance from buses where PMUs are placed. (a) Average MAPE of voltage magnitudes. (b) Average MAE of voltage angles.

Remark 1: note that for the test systems analyzed in this paper, the DeNSE performs state estimation using (1) in real-time based on a limited set of PMU measurements and without requiring knowledge of the system model and parameters. However, in the offline training phase, essential infor-

mation is derived from power flow computations, which requires knowledge of the system model and parameters. In other words, the proposed DeNSE remains model-agnostic during online operation but depends on system model and parameters during offline training. One way to avoid this dependency for an actual power system implementation is by directly utilizing historical SCADA state estimator results for creating the requisite training database of the DNN.

TABLE IX
HYPERPARAMETERS AND DATASET SIZE OF DENSE FOR 2000-BUS SYNTHETIC TEXAS SYSTEM

Type	Name	Value
Hyperparameter	Number of hidden layers	4
	Number of neurons per hidden layer	500
	Activation functions	ReLU (hidden layers), linear (output layer)
	Loss function	MSE
	Optimizer	ADAM
	Batch size	256
	Learning rate	0.001
	Number of epochs	3000
	Early stopping	Patience=10
	Dropout	30%
Dataset size	Training	7500
	Validation	2500
	Testing	4000
	Total	14000

Remark 2: when making additions to any existing system, a variety of factors must be considered. Therefore, it is not surprising that the final locations where PMUs would be placed are often decided based on negotiations with the grid operators, rather than through a purely mathematical optimization procedure (such as solving an OPP problem) [47], [48]. However, this decision (of where to place the PMUs) does not affect the DeNSE because the locations of the PMUs are input to the DeNSE, and not determined by the DeNSE. This means that the DeNSE is not limited to PMUs being placed only at the highest voltage buses of the system. In other words, even for a power system that has PMUs placed at low-voltage buses, the DeNSE will simply take all available PMU measurements into consideration during training to give the state estimates of all the buses of that system during testing.

V. CONCLUSION

In this paper, a Bayesian framework for high-speed time-synchronized TSSE is proposed, which does not require complete observability of the system by PMUs for its successful execution. The proposed state estimator, i. e., the DeNSE, overcame unobservability by indirectly combining inferences drawn from slow-timescale SCADA data with fast-timescale PMU measurements. The robustness of the DeNSE is demonstrated by its ability to successfully tackle practical challenges such as topology changes, non-Gaussian measurement noise, and different types of bad data under diverse operat-

ing conditions.

The IEEE 118-bus system and the 2000-bus synthetic Texas system are used as the test systems for the analysis conducted here. In comparison to conventional methods, the proposed DeNSE is able to bring the estimation errors of all the buses to reasonable levels, which requires less than half the number of PMUs required for full observability for the IEEE 118-bus system and less than one-quarter for the 2000-bus Synthetic Texas system. The future scope of this study will involve developing strategies to further improve accuracy of the DeNSE by determining locations for adding new PMUs, extending the proposed framework to handle events such as faults and load/generation losses, and providing provable performance guarantees [49].

APPENDIX A

A. Logical Explanation of DeNSE Functioning

The DeNSE is an MMSE estimator, in which the DNN approximates the conditional expectation $\mathbb{E}(x_i|z)$. For the i^{th} state x_i , the conditional expectation $\mathbb{E}(x_i|z)$ can be written in terms of the probability distributions as shown below:

$$\mathbb{E}(x_i|z) = \int_{-\infty}^{+\infty} x_i p(x_i|z) dx_i = \int_{-\infty}^{+\infty} x_i \frac{p(x_i, z)}{p(z)} dx_i \quad (\text{A1})$$

where $p(x_i|z)$ and $p(x_i, z)$ are the conditional probability and the joint probability between x_i and z , respectively; and $p(z)$ is the probability distribution of z . Now, it can be inferred from (1) and (A1) that $\hat{x}_i(z)$ can be obtained for any value of m (where $z \in \mathbb{R}^m$), as long as $p(x_i|z)$ is known. Moreover, increasing m can improve the estimation quality only if the new measurements are not correlated with the existing measurements, or are constant.

To better understand these inferences in the context of TSSE, consider the 3-bus system shown in Fig. A1. The reference bus (bus 1) has an angle of 0° , but its magnitude is an unknown variable. Bus 2 has both load and generation, while bus 3 has only load. The system has three sensors (depicted by blue boxes) that are measuring the magnitude of the current flowing in lines 1-2 and 2-1, and the magnitude of the current injection at bus 3.

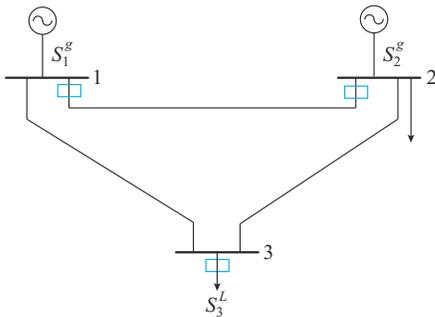


Fig. A1. 3-bus system.

Let the goal be to estimate the voltage magnitude of bus 3, i.e., $x_i = |V_3|$. The system is unobservable because $|V_3|$ cannot be estimated from the given measurements in the conven-

tional least squares sense. Note that this example simply illustrates how the Bayesian framework of DeNSE can be used to estimate states that cannot be estimated using conventional methods due to limited observability. In an actual system, the DeNSE will estimate all bus voltage magnitudes and angles without differentiating among unobserved buses, directly observed buses, and indirectly observed buses as it only relies on the joint PDF $p(x_i, z)$ between the PMU measurements and the states.

To generate $p(x_i, z)$ and $p(z)$ for this system, $F=10000$ power flows are solved. The simulation parameters used for solving the power flows of the 3-bus system are provided in Table AI.

TABLE AI
SIMULATION PARAMETERS USED FOR SOLVING POWER FLOWS OF 3-BUS SYSTEM

Parameter	Value (p.u.)	Parameter	Value (p.u.)
Series Imp_1-2	0.05 + j0.1	P_2^g	$2+N(0, 0.04)$
Series Imp_2-3	0 + j0.05	P_2^l	$0.5+N(0, 0.04)$
Series Imp_3-1	0.02 + j0.05	Q_2^l	$0.1+N(0, 0.04)$
Shunt Imp_1	-j100	P_3^l	$2+N(0, 0.04)$
Shunt Imp_2	Inf	Q_3^l	$0.5+N(0, 0.04)$
Shunt Imp_3	-j40	$ V_1 $	$1+N(0, 0.0001)$

Due to the reasons mentioned in Section II-A, it is usually not possible to analytically compute $\mathbb{E}(x_i|z)$ for all x_i and z , which is why its approximation by a DNN is needed in the first place. However, for this 3-bus system, it is observed that the probability distributions of the relevant random variables $|V_3|$, $|I_{12}|$, $|I_{21}|$, and $|I_3|$ could be well-approximated by multivariate normal distributions. In such a scenario, the conditional probability of x_i given $z = [z_1, z_2, \dots, z_m]$ is written as [50]:

$$p(x_i|z = [z_1, z_2, \dots, z_m]) = \frac{1/\sqrt{(2\pi)^{m+1} |\Sigma_{y_p}|} \exp\left[-\frac{1}{2} (y_p - \mu_{y_p})^\top \Sigma_{y_p}^{-1} (y_p - \mu_{y_p})\right]}{1/\sqrt{(2\pi)^m |\Sigma_{y_q}|} \exp\left[-\frac{1}{2} (y_q - \mu_{y_q})^\top \Sigma_{y_q}^{-1} (y_q - \mu_{y_q})\right]} \quad (\text{A2})$$

where y_p and y_q are obtained from power flow solutions, with y_p comprising all variables in x_i and z , and y_q comprising variables in z only; μ and Σ are the mean and covariance, respectively, and $|\Sigma|$ is determinant of the covariance. Now, using (A1) and (A2), we compare $\mathbb{E}(x_i|z)$ with the actual value of x_i for five MMSE estimator cases: ① Case 1: $z = \{\angle V_1\}$; ② Case 2: $z = \{|I_{12}|\}$; ③ Case 3: $z = \{|I_3|\}$; ④ Case 4: $z = \{|I_{12}|, |I_{21}|\}$; ⑤ Case 5: $z = \{|I_{12}|, |I_3|\}$. Note that $|V_3|$, $|I_{12}|$, $|I_{21}|$, and $|I_3|$ are dependent variables as they correspond to converged power flow solutions, while $\angle V_1$ is a constant. The estimation results are shown in Table AII. In Case 1, z is a constant, and so $\mathbb{E}(x_i|z) = \mathbb{E}(x_i)$, which is the mean value of $|V_3|$ across all F power flows. As this case is

not able to track the variations in OCs across different power flows, its estimate is the worst. Cases 2 and 3 give similar results as they separately track the variations in $|I_{12}|$ and $|I_3|$ to estimate $|V_3|$. Despite having two measurements, the results of Case 4 are worse than those in Cases 2 and 3 because $|I_{12}|$ and $|I_{21}|$ are highly correlated. The expected values of Case 5 are closest to the ground-truth values as this estimator is able to use both $|I_{12}|$ and $|I_3|$ to estimate $|V_3|$. This analysis confirms that the knowledge of $p(\mathbf{x}|\mathbf{z})$ and a non-large value of m are the basis for the DeNSE to overcome unobservability. It is also worth mentioning that the estimation quality of the DeNSE improves if F is increased, because with more training samples, the DNN will be able to better approximate the probability distributions, and in turn, $\mathbb{E}(\mathbf{x}|\mathbf{z})$.

TABLE AII
STATE ESTIMATION RESULTS OF CASE STUDIES DONE ON 3-BUS SYSTEM

Case	\mathbf{z}	$MAE = \frac{1}{F} \sum_{k=1}^F x_i^k - \mathbb{E}(x_i^k \mathbf{z}^k) $
1	$\{\angle V_1\}$	0.00100
2	$\{ I_{12} \}$	0.00014
3	$\{ I_3 \}$	0.00021
4	$\{ I_{12} , I_{21} \}$	0.00094
5	$\{ I_{12} , I_3 \}$	0.00005

B. Python Resources for DeNSE Implementation

All the Python codes required for implementing the DeNSE method developed in this paper can be accessed through the following GitHub repository: <https://github.com/Anamitra-Pal-Lab/DeNSE>. The Read Me file provided in this repository contains all the information that is needed to run the files and obtain the results.

REFERENCES

- [1] G. Wang, G. B. Giannakis, and J. Chen, "Robust and scalable power system state estimation via composite optimization," *IEEE Transactions on Smart Grid*, vol. 10, no. 6, pp. 6137-6147, Nov. 2019.
- [2] S. Chatzivasileiadis, P. Aristidou, I. Sassiou *et al.*, "Micro-flexibility: challenges for power system modeling and control," *Electric Power Systems Research*, vol. 216, p. 109002, Mar. 2023.
- [3] G. Wang, H. Zhu, G. B. Giannakis *et al.*, "Robust power system state estimation from rank-one measurements," *IEEE Transactions on Control Network Systems*, vol. 6, no. 4, pp. 1391-1403, Dec. 2019.
- [4] A. S. Dobakhshari, M. Abdolmaleki, V. Terzija *et al.*, "Robust hybrid linear state estimator utilizing SCADA and PMU measurements," *IEEE Transactions on Power Systems*, vol. 36, no. 2, pp. 1264-1273, Mar. 2021.
- [5] M. Kabiri and N. Amjadi, "A new hybrid state estimation considering different accuracy levels of PMU and SCADA measurements," *IEEE Transactions on Instrumentation & Measurements*, vol. 68, no. 9, pp. 3078-3089, Sept. 2019.
- [6] K. Sun, M. Huang, Z. Wei *et al.*, "High-refresh-rate robust state estimation based on recursive correction for large-scale power systems," *IEEE Transactions on Instrumentation & Measurements*, vol. 72, p. 9002413, May 2023.
- [7] J. Zhao, S. Wang, L. Mili *et al.*, "A robust state estimation framework considering measurement correlations and imperfect synchronization," *IEEE Transactions on Power Systems*, vol. 33, no. 4, pp. 4604-4613, Jul. 2018.
- [8] P. Yang, Z. Tan, A. Wiesel *et al.*, "Power system state estimation using PMUs with imperfect synchronization," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 4162-4172, Nov. 2013.
- [9] J. Zhao, G. Zhang, K. Das *et al.*, "Power system real-time monitoring by using PMU-based robust state estimation method," *IEEE Transactions on Smart Grid*, vol. 7, no. 1, pp. 300-309, Jan. 2016.
- [10] N. M. Manousakis and G. N. Korres, "A hybrid power system state estimator using synchronized and unsynchronized sensors," *International Transactions on Electrical Energy Systems*, vol. 38, no. 8, p. e2580, Aug. 2018.
- [11] Z. Jin, P. Wall, Y. Chen *et al.*, "Analysis of hybrid state estimators: accuracy and convergence of estimator formulations," *IEEE Transactions on Power Systems*, vol. 34, no. 4, pp. 2565-2576, Jul. 2019.
- [12] T. Chen, H. Ren, Y. Sun *et al.*, "Optimal placement of phasor measurement unit in smart grids considering multiple constraints," *Journal of Modern Power Systems and Clean Energy*, vol. 11, no. 2, pp. 479-488, Mar. 2023.
- [13] A. Pal, G. A. Sanchez-Ayala, V. A. Centeno *et al.*, "A PMU placement scheme ensuring real-time monitoring of critical buses of the network," *IEEE Transactions on Power Delivery*, vol. 29, no. 2, pp. 510-517, Apr. 2014.
- [14] N. P. Theodorakatos, M. Lytras, and R. Babu, "Towards smart energy grids: a box-constrained nonlinear underdetermined model for power system observability using recursive quadratic programming," *Energies*, vol. 13, no. 7, pp. 1-17, Apr. 2020.
- [15] M. A. R. S. Cruz, H. R. O. Rocha, M. H. M. Paiva *et al.*, "PMU placement with multi-objective optimization considering resilient communication infrastructure," *International Journal of Electrical Power & Energy Systems*, vol. 141, p. 108167, Apr. 2022.
- [16] N. P. Theodorakatos, R. Babu, and A. P. Moschoudis, "The branch-and-bound algorithm in optimizing mathematical programming models to achieve power grid observability," *Axioms*, vol. 12, pp. 1-46, Nov. 2023.
- [17] R. S. Biswas, B. Azimian, and A. Pal, "A micro-PMU placement scheme for distribution systems considering practical constraints," in *Proceedings of IEEE PES General Meeting*, Montreal, Canada, pp. 1-5, Aug. 2020.
- [18] K. Amare, V. A. Centeno, and A. Pal, "Unified PMU placement algorithm for power systems," in *Proceedings of IEEE North American Power Symposium (NAPS)*, Charlotte, USA, pp. 1-6, Oct. 2015.
- [19] M. Ghamsari-Yazdel, M. Esmaili, F. Aminifar *et al.*, "Incorporation of controlled islanding scenarios and complex substations in optimal WAMS design," *IEEE Transactions on Power Systems*, vol. 34, no. 5, pp. 3408-3416, Sept. 2019.
- [20] U.S. Department of Energy, Office of Electricity Delivery and Energy Reliability. (2014, Oct.). Factors affecting PMU installation costs. [Online]. Available: <https://www.smartgrid.gov/files/documents/PMU-cost-study-final-10162014.pdf>
- [21] A. Pal, C. Mishra, A. K. S. Vullikanti *et al.*, "General optimal substation coverage algorithm for phasor measurement unit placement in practical systems," *IET Generation, Transmission & Distribution*, vol. 11, no. 2, pp. 347-353, Jan. 2017.
- [22] A. Pal, A. K. S. Vullikanti, and S. S. Ravi, "A PMU placement scheme considering realistic costs and modern trends in relaying," *IEEE Transactions on Power Systems*, vol. 32, no. 1, pp. 552-561, Jan. 2017.
- [23] K. R. Mestav, J. Luengo-Rozas, and L. Tong, "Bayesian state estimation for unobservable distribution systems via deep learning," *IEEE Transactions on Power Systems*, vol. 34, no. 6, pp. 4910-4920, Nov. 2019.
- [24] B. Azimian, R. S. Biswas, S. Moshagh *et al.*, "State and topology estimation for unobservable distribution systems using deep neural networks," *IEEE Transactions on Instrumentation & Measurements*, vol. 71, pp. 1-14, Apr. 2022.
- [25] K. R. Mestav and L. Tong, "Learning the unobservable: high-resolution state estimation via deep learning," in *Proceedings of 57th Annual Allerton Conference on Communication, Control, and Computing*, Monticello, USA, Sept. 2019, pp. 171-176.
- [26] G. Tian, Y. Gu, D. Shi *et al.*, "Neural-network-based power system state estimation with extended observability," *Journal of Modern Power Systems and Clean Energy*, vol. 9, no. 5, pp. 1043-1053, Sept. 2021.
- [27] V. Chakati, M. Pore, A. Pal *et al.*, "Challenges and trade-offs of a cloud hosted phasor measurement unit-based linear state estimator," in *Proceedings of IEEE PES Conference on Innovative Smart Grid Technologies*, Washington DC, USA, Apr. 2017, pp. 1-5.
- [28] R. Raz, "On the complexity of matrix product," in *Proceedings of 34th Annual ACM Symposium on Theory Computing*, New York, USA,

- May 2002, pp. 144-151.
- [29] E. Klarreich, "Multiplication hits the speed limit," *Communications of the ACM*, vol. 63, no. 1, pp. 11-13, Jan. 2020.
- [30] T. Ahmad and N. Senroy, "Statistical characterization of PMU error for robust WAMS based analytics," *IEEE Transactions on Power Systems*, vol. 35, no. 2, pp. 920-928, Mar. 2020.
- [31] D. Salls, J. Ramirez, A. Varghese *et al.*, "Statistical characterization of random errors present in synchrophasor measurements," in *Proceedings of IEEE PES General Meeting*, Washington DC, USA, Jul. 2021, pp. 1-5.
- [32] A. C. Varghese, A. Pal, and G. Dasarathy, "Transmission line parameter estimation under non-Gaussian measurement noise," *IEEE Transactions on Power Systems*, vol. 38, no. 4, pp. 3147-3162, Jul. 2023.
- [33] J. Zhao and L. Mili, "A framework for robust hybrid state estimation with unknown measurement noise statistics," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 5, pp. 1866-1875, May 2018.
- [34] Y. Gu, Z. Yu, R. Diao *et al.*, "Doubly-fed deep learning method for bad data identification in linear state estimation," *Journal of Modern Power Systems and Clean Energy*, vol. 8, no. 6, pp. 1140-1150, Nov. 2020.
- [35] W. Liu, J. Liu, H. Li *et al.*, "Multichannel signal detection based on Wald test in subspace interference and Gaussian noise," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 3, pp. 1370-1381, Jun. 2019.
- [36] Y. Yang, Z. Yang, J. Yu *et al.*, "Fast economic dispatch in smart grids using deep learning: an active constraint screening approach," *IEEE Internet of Things Journal*, vol. 7, no. 11, pp. 11030-11040, Nov. 2020.
- [37] J. A. D. Massignan, J. B. A. London, and V. Miranda, "Tracking power system state evolution with maximum-correntropy-based extended Kalman filter," *Journal of Modern Power Systems and Clean Energy*, vol. 8, no. 4, pp. 616-626, Jul. 2020.
- [38] S. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345-1359, Oct. 2010.
- [39] Y. Zhang, Y. Zhang, and Q. Yang, "Parameter transfer unit for deep neural networks," in *Advances in Knowledge Discovery and Data Mining*, Cham: Springer International Publishing, pp. 82-95, Mar. 2019.
- [40] K. D. Jones, A. Pal, and J. S. Thorp, "Methodology for performing synchrophasor data conditioning and validation," *IEEE Transactions on Power Systems*, vol. 30, no. 3, pp. 1121-1130, May 2015.
- [41] H. Li, J. Yeo, A. L. Bornsheuer *et al.*, "The creation and validation of load time series for synthetic electric power systems," *IEEE Transactions on Power Systems*, vol. 36, no. 2, pp. 961-969, Mar. 2021.
- [42] A. B. Birchfield, T. Xu, and T. J. Overbye, "Power flow convergence and reactive power planning in the creation of large synthetic grids," *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 6667-6674, Nov. 2018.
- [43] F. Chollet. (2015, Dec.). Keras. [Online]. Available: <https://keras.io>
- [44] WANDB. (2023, Nov.). Weights & biases. [Online]. Available: <https://wandb.ai/site>
- [45] *IEEE/IEC International Standard – Measuring Relays and Protection Equipment – Part 118-1: Synchrophasor for Power Systems – Measurements*, IEC/IEEE 60255-118-1:2018, Dec. 2018.
- [46] Q. Yang, A. Sadeghi, and G. Wang, "Data-driven priors for robust PSSE via Gauss-Newton unrolled neural networks," *IEEE Journal of Emerging and Selected Topics on Circuits Systems*, vol. 12, no. 1, pp. 172-181, Mar. 2022.
- [47] D. Borkowski, A. Wetula, J. Kowalski *et al.*, "Experimental setup for harmonic impedance measurement in a real HV power grid," *Electric Power Components and Systems*, vol. 47, no. 8, pp. 733-742, Aug. 2019.
- [48] N. Theodorakatos, N. Manousakis, and G. Korres, "Optimal placement of phasor measurement units with linear and non-linear models," *Electric Power Components and Systems*, vol. 43, no. 4, pp. 357-373, Feb. 2015.
- [49] B. Azimian, S. Moshtagh, A. Pal *et al.*, "Analytical verification of performance of deep neural network based time-synchronized distribution system state estimation," *Journal of Modern Power Systems and Clean Energy*, vol. 12, no. 4, pp. 1126-1134, Jul. 2024.
- [50] H. Pishro-Nik. (2014, Dec.). Introduction to probability, statistics, and random processes. [Online]. Available: <https://www.probabilitycourse.com>

Antos Cheeramban Varghese received the B.Tech. degree in electrical engineering from the Indian Institute of Technology Jodhpur, Rajasthan, India, in 2016, and the M.Tech. degree in electrical engineering from the Indian Institute of Technology Bombay, Mumbai, India, in 2018. Currently, he is pursuing the Ph.D. degree in electrical engineering at Arizona State University, Tempe, USA. His research interests include leveraging machine learning, statistical analysis, and optimization techniques to address various challenges within the field of power systems.

Hritik Shah received the B.E. degree in electronics engineering from the DJ Sanghvi College of Engineering, Mumbai, India, in 2021, and the M.S. degree in industrial engineering from Arizona State University, Tempe, USA, in 2023. He is a Data Science Analyst with Reliability and Resiliency Group at Eversource Energy, Berlin, Germany. His research interests include power grid reliability, outage management, and advanced load forecasting.

Behrouz Azimian received the B.Sc. degree in electrical engineering from Iran University of Science and Technology, Tehran, Iran, in 2016, and the M.Sc. degree in electrical engineering from Alfred University, Alfred, USA, in 2019. He is currently pursuing the Ph.D. degree at Arizona State University, Tempe, USA. His research interests include machine learning, deep learning, application of artificial intelligence and optimization methods in power system problems such as state estimation, electricity market, and electric vehicle charging scheduling.

Anamitra Pal received the B.E. degree (summa cum laude) in electrical and electronics engineering from the Birla Institute of Technology at Mesra, Ranchi, India, in 2008, and the M.S. and Ph.D. degrees in electrical engineering from Virginia Tech, Blacksburg, USA, in 2012 and 2014, respectively. He is currently an Associate Professor in the School of Electrical, Computer, and Energy Engineering at Arizona State University (ASU), Tempe, USA. His research interests include data analytics with a special emphasis on time-synchronized measurements, applications of artificial intelligence in power systems, renewable generation integration studies, and critical infrastructure resilience.

Evangelos Farantatos received the Diploma in electrical and computer engineering from the National Technical University of Athens, Athens, Greece, in 2006, and the M.S. and Ph.D. degrees from the Georgia Institute of Technology, Atlanta, USA, in 2009 and 2012, respectively. He is a Sr. Principal Team Lead with the Transmission Operations and Planning R&D Group at Electric Power Research Institute (EPRI), Palo Alto, USA. His research interests include synchrophasor technology, power system monitoring and control, power system stability and dynamics, renewable energy resource modeling, grid operation protection and control with high levels of inverter-based resources.