

Physics-guided Deep Learning for Power System State Estimation

Lei Wang, Qun Zhou, and Shuangshuang Jin

Abstract—In the past decade, dramatic progress has been made in the field of machine learning. This paper explores the possibility of applying deep learning in power system state estimation. Traditionally, physics-based models are used including weighted least square (WLS) or weighted least absolute value (WLAV). These models typically consider a single snapshot of the system without capturing temporal correlations of system states. In this paper, a physics-guided deep learning (PGDL) method is proposed. Specifically, inspired by autoencoders, deep neural networks (DNNs) are used to learn the temporal correlations. The estimated system states from DNNs are then checked against physics laws by running through a set of power flow equations. Hence, the proposed PGDL is both data-driven and physics-guided. The accuracy and robustness of the proposed PGDL method are compared with traditional methods in standard IEEE cases. Simulations show promising results and the applicability is further discussed.

Index Terms—State estimation, deep learning, deep neural network (DNN), temporal correlation, power system.

I. INTRODUCTION

MACHINE learning is booming everywhere around the world [1]. Deep learning has been applied in computer vision, speech recognition, natural language process, and many other areas [2]. As an notable example, AlphaGo developed by Google Deepmind employs deep reinforcement learning and has become the first computer program to defeat a professional human Go player [3]. With these achievements in computer science, the question arises on how to bridge machine learning with power systems.

In the power industry, machine learning has been used for load forecasting [4], [5]. In such application where the underlying physical models are unknown, machine learning presents a powerful tool to perform predictive analytics. However, most applications in power system analysis have concrete physical models, e. g., power flow equations, and machine learning may fall short compared to model-based approaches.

Power system state estimation is such an application

backed by detailed physical models. State estimation is critical in power grid operations. It receives raw measurements from a supervisory control and data acquisition (SCADA) system and generates critical inputs for other applications that require reliable estimates of current system states [6], [7].

Weighted least square (WLS) based methods are commonly used to estimate system states [7]. The measurement data usually consist of real and reactive power injections, real and reactive power flows, and bus voltage magnitudes, while the state vector contains the voltage magnitudes and phase angles at all buses. Due to instrument errors and communication noises, a stochastic error term needs to be incorporated. State estimation aims to obtain true states by maximizing the likelihood of estimated states, equivalent to minimizing the residuals between estimated and actual measurements.

Specifically, WLS-based state estimation can be represented by:

$$z = h(x) + \varepsilon \quad (1)$$

where z is the measurement vector; $h(\cdot)$ is the nonlinear power flow equation; x is the state vector consisting of voltage magnitudes and phase angles; and ε is the noise vector assumed to follow Gaussian distribution with expectation $E(\varepsilon) = 0$ and covariance $Cov(\varepsilon) = R$ [8], [9]. The WLS-based state estimation will determine the optimal states by minimizing the weighted residuals:

$$\min_{\hat{x}} (z - h(\hat{x}))^T R^{-1} (z - h(\hat{x})) \quad (2)$$

where \hat{x} is the estimated state vector; and R is the covariance matrix. Many research efforts have been dedicated to improve the performance of state estimation. In [10], a summary of synchrophasor technology is carried out to show its improvement in power system. A robust least absolute value (LAV) estimator is employed by taking advantage of phasor measurement units (PMUs) to enhance the computation performance of LAV estimator [11]. Considering the coexistence of SCADA data and PMU data in a power system, a hybrid state estimator has been formulated in [12] to dynamically trace the states based on weighted LAV. With PMU-based linear estimation, it is proposed to run multiple copies of a multi-area state estimator and the estimation efficiency is improved [13].

Most approaches in the literature are considered as single-snapshot estimation. At the estimation time t , only the measurement data at that exact moment is used to estimate the system states. One of the concerns with this static view of

Manuscript received: August 20, 2019; accepted: January 7, 2020. Date of CrossCheck: January 7, 2020. Date of online publication: June 25, 2020.

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

L. Wang and Q. Zhou (corresponding author) are with the Department of Electrical and Computer Engineering, University of Central Florida, Orlando, FL 32816, USA (e-mail: ray_wang@knights.ucf.edu; qun.zhou@ucf.edu).

S. Jin is with the School of Computing, Clemson University, North Charleston, SC 29405, USA (e-mail: jin6@clemson.edu).

DOI: 10.35833/MPCE.2019.000565



the power system is that the dynamics are not fully captured. In particular, power system conditions are constantly changing due to weather, customer behaviors, and other events. Therefore, system states are also correlated from time to time, i.e., temporal correlations exist between states at different times. Researchers have recognized this dynamics and incorporated into state estimation. For instance, a forecasting aided state estimation is proposed to enhance the estimation performance [14] and anomaly detection [15]. Similarly, time correlations in the PMU measurements are modeled using autoregressive methods [16]. Incorporating historical data, a robust data-driven state estimation is based on robust nearest neighbor search [17]. In [18], a new state estimation framework is proposed to account for measurement correlations and imperfect synchronization.

Nevertheless, the state dynamics and correlations are very difficult to model explicitly, because the underlying physics is unknown with random behaviors involved. This is where machine learning finds its role in state estimation.

In this paper, we propose a physics-guided deep learning (PGDL) state estimator that explores the temporal correlations between system states. In contrast to black-box modeling, the proposed PGDL is both data-driven and first-principle-based. Specifically, the true dynamics of power system are unknown and difficult to model due to exogenous factors such as weather and customer behaviors. Therefore, we adopt the latest development in machine learning and apply deep neural networks (DNNs) to learn the state dynamics and correlations. The estimated system states are checked against physics laws by running through a set of power flow equations. It is noted that the combination of data-driven and physical models are emerging in the field of machine learning. Examples include using physics-informed neural networks to solve nonlinear partial differential equations [19] and physics-guided neural networks for temperature modeling [20]. The proposed PGDL incorporates the physical AC power flow model into the deep learning to form a hybrid learning structure. The proposed data-driven state estimation method has been tested in IEEE 14-bus and IEEE 118-bus systems. The performance improvement is clearly shown in the numerical results.

The remainder of this paper is organized as follows. Section II presents the machine learning framework. Section III presents the structure of DNNs and the hybrid learning method for state estimation. In Section IV, the performance is evaluated in IEEE 14-bus and IEEE 118-bus systems. Section V discusses the findings from our study and Section VI provides concluding remarks.

II. MACHINE LEARNING FRAMEWORK

The motivation behind our study is that machine learning may be able to learn temporal correlations among different states. The proposed deep learning based framework is described in Fig. 1.

In a power system, the load is constantly changing due to underlying causes such as weather and customer behaviors. Assume that the load at time t (d_t) is correlated with the load at time $t-1$ (d_{t-1}), denoted by $d_t = l(d_{t-1})$. This load cor-

relation is translated to state correlation through the complex operations of the power system to balance the load and generation. The state correlation is represented by $x_t = f(x_{t-1})$, where $f(\cdot)$ denotes the nonlinear nature of power system operations. The correlations among state variables are then translated to measurement correlations through the nonlinear physical power flow model, denoted by $z_t = g(z_{t-1})$. The temporal correlations of measurement data should be considered in state estimation, whereas a single-snapshot state estimation lacks this consideration. In Fig. 1, \hat{x}_t and x_t are the estimated and real state values at time step t , respectively.

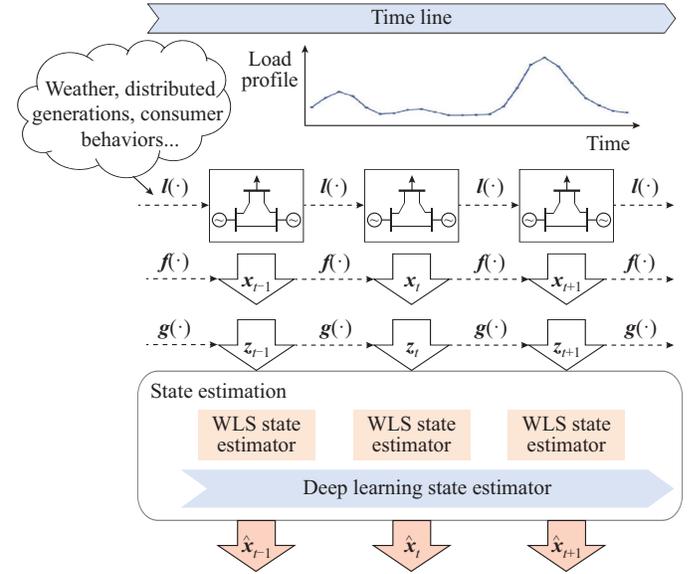


Fig. 1. Deep learning based framework to capture state correlations.

If we consider state estimation as an optimization problem, the traditional method is formulated as in (2), which minimizes the estimation errors for one time shot. On the contrary, the state estimation with temporal correlations is formulated as a look-ahead optimization problem:

$$\min_{\hat{x}_t} \sum_t (z_t - h(\hat{x}_t))^T R^{-1} (z_t - h(\hat{x}_t)) \quad (3)$$

s.t.

$$\hat{x}_t = f(\hat{x}_{t-1}) \quad (4)$$

Note that the true dynamics of states $f(\cdot)$ is unknown and difficult to model mathematically. Therefore, it is challenging to model the state correlations. This motivates a data-driven approach that learns the dynamics through historical data. A DNN with sufficient learning capacity is suitable for the large amount of data in a power system. The PGDL framework uses DNN to learn the temporal correlations of state variables while incorporating the physical model.

III. PGDL MODEL

The PGDL model that incorporates historical data and the physical power flow model is proposed in this section. This hybrid learning method employs DNNs to learn the state correlations while considering physical flows in a power system.

A. Physics-guided Learning Architecture

We propose a hybrid machine learning model inspired by the emerging autoencoder in the artificial intelligent (AI) field. As shown in Fig. 2, an autoencoder is a neural network that is trained to copy its input to its output with fully-connected internal layers [2]. Internally, it has a hidden layer (green box) that divides the neural network into two parts: encoder and decoder. The autoencoder is initially designed for dimension reduction so that the number of features can be reduced to represent a system [21].

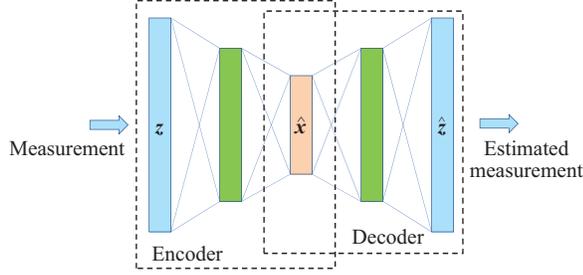


Fig. 2. Autoencoder for state estimation.

To apply autoencoders in state estimation, the measurement z is fed into the encoder, and the hidden layer output is the estimated system states \hat{x} , which can be considered as features to represent the power system. The estimated states \hat{x} then go through the decoder to output the reconstructed measurement data \hat{z} . In the context of power system state estimation, these measurements z come from the SCADA system consisting of real and reactive power injections at each bus (P_i and Q_i for bus i), and real and reactive power flows on each transmission line (P_{ij} and Q_{ij} for line ij). The estimated values of these measurements are denoted by \hat{z} . The system states x consist of voltage magnitude and angle at each bus (V_i and θ_i for bus i). The corresponding estimated system states are denoted by \hat{x} .

It would be a pure data-driven method to use autoencoder for state estimation without explanatory models. However, with the domain knowledge of power systems, we could improve the autoencoder by incorporating the first principles in the power system. The proposed physics-guided learning is both data-driven and first-principle-based.

Figure 3 depicts the architecture of physics-guided learning.

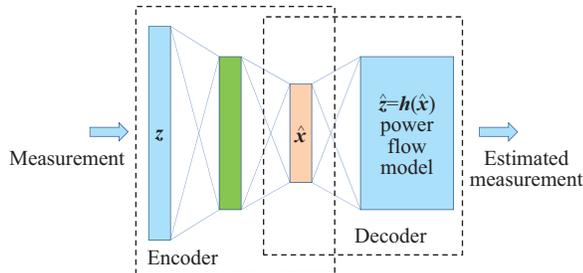


Fig. 3. Architecture of physics-guided learning.

At time k , the DNN input is the measurement set $z^k = [z_1^k, z_2^k, \dots, z_m^k]^T$ and the output is the corresponding estimated

state vector $\hat{x}^k = [\hat{x}_1^k, \hat{x}_2^k, \dots, \hat{x}_n^k]^T$, where m is the number of measurements and n is the number of states. The estimated states \hat{x} then go through the physical measurement model consisting of the power flow equations in (5).

$$\begin{cases} P_i = \sum_{j=1}^N V_i V_j (G_{ij} \cos(\theta_i - \theta_j) + B_{ij} \sin(\theta_i - \theta_j)) \\ Q_i = \sum_{j=1}^N V_i V_j (G_{ij} \sin(\theta_i - \theta_j) + B_{ij} \cos(\theta_i - \theta_j)) \\ P_{ij} = -V_i^2 G_{ij} + V_i V_j (G_{ij} \cos(\theta_i - \theta_j) + B_{ij} \sin(\theta_i - \theta_j)) \\ Q_{ij} = -V_i^2 B_{ij} + V_i V_j (G_{ij} \sin(\theta_i - \theta_j) - B_{ij} \cos(\theta_i - \theta_j)) \end{cases} \quad (5)$$

where G_{ij} and B_{ij} are the real and imaginary parts in the admittance matrix, respectively; and N is the number of buses in the power system.

As a result, the estimated measurement vector \hat{z} is generated. To train the DNN, the loss function is the cumulative error between actual and estimated measurement vectors. The error is then back-propagated through the physical model layer and the neural network layers, so that weights and biases of the neural network are adjusted accordingly.

B. Data and Training Flow

The implementation details are described in Fig. 4. The PGDL model consists of multiple stages: data acquisition, initialization, pre-training, validation, and online estimation, where \mathcal{L} is the loss function; w is the weight matrix of neural network; and w^0 is the parameter vector of the pre-trained DNNs.

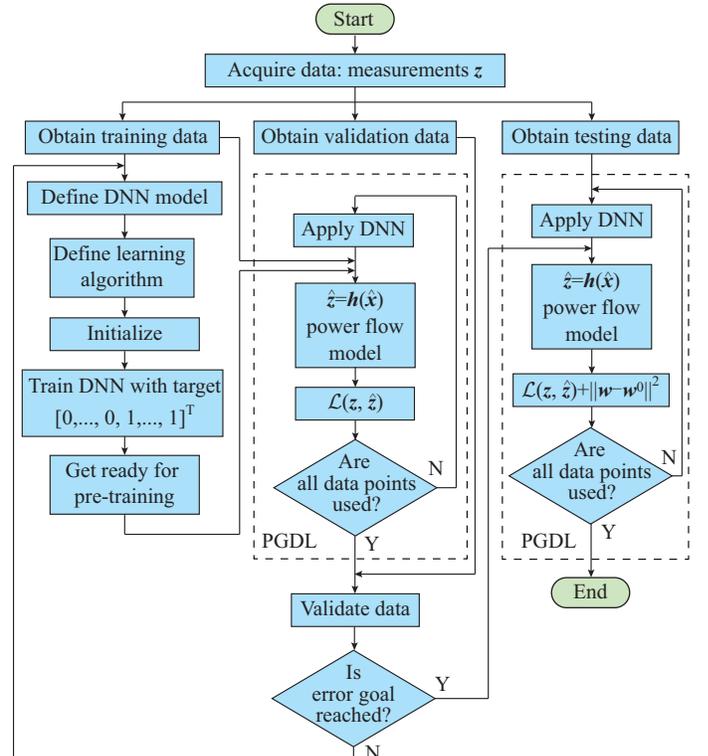


Fig. 4. Flowchart of PGDL.

First, the measurement data are acquired. In our case studies, the data come from simulations, but in reality, the data

are acquired from the SCADA system. The data are then divided into the training data, validation data, and testing data. The training data are used to train the DNNs (two DNNs are selected and introduced in Section III-C). For the pre-training stage, different learning rates and initialization methods are used. Note that the pre-training is an offline batch training, where the losses are calculated and back-propagated through the neural network for the update of the weights. After the DNNs are trained, validation data are used to assess the overall performance of the DNNs. If the performance is satisfied, go on to online estimation with new data. Otherwise, redefine the DNN network and repeat the process. More detailed training procedure can be found in Section III-D.

C. DNNs

DNNs are used as the encoder to map the nonlinear relationship of measurements z and states x . It is well-known that neural network has the universal approximation capability to approximate nonlinear functions [22]. In this paper, two typical DNNs are studied: feedforward neural networks (FFNNs) and long short-term memory (LSTM) neural networks.

1) FFNNs

A multi-layer FFNN is first utilized to build up the encoder in the PGDL model. As shown in Fig. 5, the input layer has m neurons, and the output layer has n neurons. Usually, $n=2N-1$. The width and depth of network will be determined by the complexity of the power system. For the hidden layers, the hyperbolic tangent (tanh) activation functions are exploited. It is proven that the hyperbolic tangent networks, unlike the sigmoid, do not suffer from the saturation behavior of the top hidden layers due to its symmetry around 0 [23].

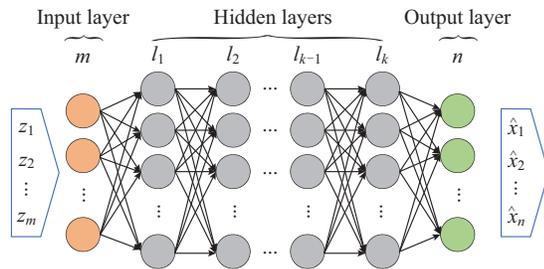


Fig. 5. FFNN-based state estimator.

Regarding the IEEE 14-bus and 118-bus systems, the details of FFNN structure are shown in Table I.

TABLE I
STRUCTURES OF FFNN IN IEEE TEST CASES

System	m	$l_i (i=1, 2, \dots, k-1)$	l_k	n
IEEE 14-bus	30	128	64	27
IEEE 118-bus	330	512	512	235

Note that FFNN only takes the measurement at time t as the input, and hence does not specifically learn the temporal correlations. FFNN is chosen as a comparing model to test

the hypothesis that adding temporal learning capability in the encoder DNN would enhance the overall performance of system estimation. This hypothesis is verified in the numerical results in Section IV.

2) LSTM Neural Network

LSTM neural network is recurrent neural network (RNN) that is suitable to capture the state dynamics [24]. Unlike the conventional RNN, LSTM can alleviate the exploding and vanishing gradient problems due to the memory cell and gating mechanism [25]. A common LSTM unit comprised of a memory cell and three gates is described in Fig. 6.

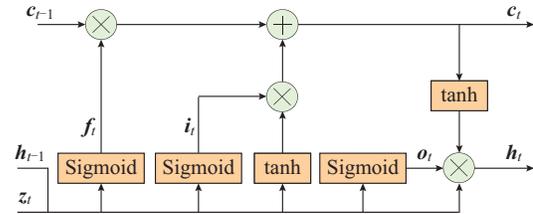


Fig. 6. LSTM unit.

For a single LSTM unit, the inputs are the measurement vector z_t at time t and hidden vector h_{t-1} from last time step, while the output is h_t . There are three gates: input gate i_t , forget gate f_t , and output gate o_t . The LSTM unit will learn the dependencies between the data in the input sequence. The input gate manages the values flowing into the memory cell from the inputs. The forget gate determines which part is passed to the next step. As for the output, it is a product of the result of output gate and the activation of the memory cell. The mathematical expressions of LSTM are given as follows: $f_t = \sigma(W_f z_t + U_f h_{t-1} + b_f)$, $i_t = \sigma(W_i z_t + U_i h_{t-1} + b_i)$, $c_t = f_t c_{t-1} + i_t (\tanh(W_c z_t + U_c h_{t-1} + b_c))$, $o_t = \sigma(W_o z_t + U_o h_{t-1} + b_o)$, and $h_t = o_t \tanh(c_t)$, where W_f , W_i , W_c , W_o , U_f , U_i , U_c and U_o are weights that connect different layers; b_f , b_i , b_c and b_o are the biases with each individual gate; and $\sigma(\cdot)$ is the sigmoid activation function.

To improve the learning capability, multiple LSTM units are stacked as shown in Fig. 7 and the last layer is a fully connected linear layer.

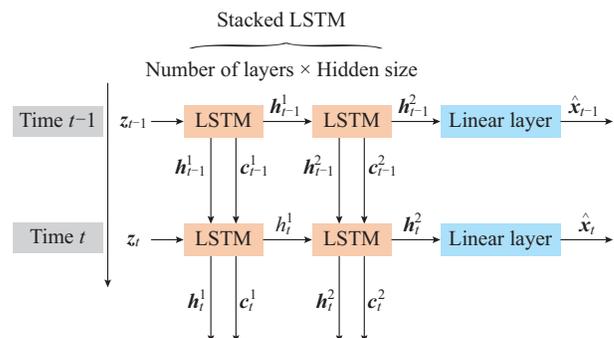


Fig. 7. Stacked LSTM unrolled in time.

Note that the input of LSTM is also the measurement at time t , but because of its recurrent nature, all the previous time steps are implicitly considered. Figure 7 shows an un-

rolled LSTM in time, where it is two dimensional in time and space.

For the IEEE cases in this paper, Table II provides the details of stacked LSTM network configuration.

TABLE II
STRUCTURES OF STACKED LSTM IN IEEE TEST CASES

System	Stacked LSTM			Size of linear layer
	Input size	No. of layers	Hidden size	
IEEE 14-bus	30	2	128	27
IEEE 118-bus	330	2	512	235

D. Training Procedure

The training procedure of two DNNs consists of multiple implementation steps including configuration and initialization of DNN, definition of loss functions and regularization, and selection of proper learning algorithms and learning rates.

1) Configuration and Initialization of DNN

The performance of DNNs can vary depending on the configuration and initialization. For each dataset, we run DNNs with 20 different configurations and random seeds, and select the DNNs with the best performance on the validation data, in order to achieve reliable online estimation for the test data. For the initialization, several methods are attempted including random initialization, uniform initialization, and Xavier initialization [23]. The neural networks are then trained using target values of voltage angles and magnitudes equal to $[0, 0, \dots, 1, 1, \dots, 1]^T$, where the numbers of zeros and ones equal to those of voltage angles and magnitudes in the state vector, respectively, which is similar to the flat start in WLS state estimation.

2) Definition of Loss Functions and Regularization

The loss function needs to be defined to train and update weights and biases of DNNs. The cumulative mean square error (MSE) loss function is used:

$$\mathcal{L} = \frac{1}{m} \sum_i (z_i - \hat{z}_i)^T (z_i - \hat{z}_i) \quad (6)$$

For online training and estimation, a regularization technique is implemented to maintain the stability of the trained DNNs. An L^2 -SP penalty [26] is added to encourage the similarity with the starting point, i.e., the parameters of pre-trained model. The loss function with L^2 -SP penalty can be formulated as (7).

$$\mathcal{L} = \frac{1}{m} (z - \hat{z})^T (z - \hat{z}) + \|w - w^0\| \quad (7)$$

3) Selection of Learning Algorithms and Learning Rates

The loss is then back-propagated through the physical model layer as well as the neural network layer. Hence, the chain rule is expressed by:

$$\frac{\partial \mathcal{L}}{\partial w} = \frac{\partial \mathcal{L}}{\partial \hat{z}} \frac{\partial \hat{z}}{\partial \hat{x}} \frac{\partial \hat{x}}{\partial w} \quad (8)$$

$$w^{new} = w^{old} - \eta \frac{\partial \mathcal{L}}{\partial w} \quad (9)$$

where η is the learning rate; and w^{new} and w^{old} are the new and old weight matrices of neural network, respectively. In

this work, a constant learning rate, a decayed learning rate, and a cyclical learning rate are used to improve the performance as in [27]. The update of weights and biases is based on the gradient-descent algorithm [28]. Specifically, mini-batch gradient descent is employed to pre-train the DNN models. For online training and estimation, stochastic gradient descent is used to continuously update model parameters as new data points come in.

IV. NUMERICAL RESULTS

In this section, the proposed PGDL is implemented in two standard IEEE power systems, and the code can be found on Github [29]. The performance is evaluated and compared with conventional WLS-based and WLAV-based state estimations in terms of accuracy and robustness. The performance measure is mean absolute estimated error (MAEE):

$$MAEE = \frac{1}{N} \sum_{k=1}^N |\hat{x}_k - x_k| \quad (10)$$

The simulation is carried out in a predefined time period, and the mean and standard deviation of the MAEEs are compared.

A. Simulation Setup

The simulation data are generated in the IEEE 14-bus and 118-bus systems. The IEEE 14-bus system has 2 generators and 20 loads, and the 118-bus system has 19 generators and 91 loads. The load profile is downloaded from NYISO and scaled down for the two systems [30]. A one-day normalized load profile with 5 min time interval is given in Fig. 8.

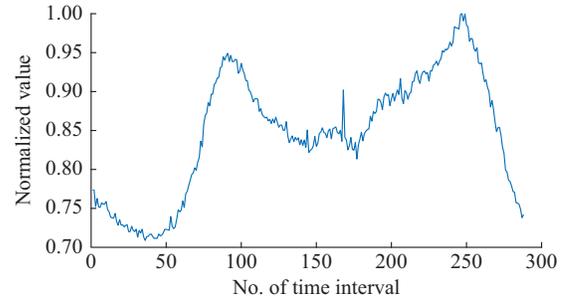


Fig. 8. Normalized load profile from NYISO.

The true states and measurement vectors are generated using MATPOWER 6.0 [31]. The states consist of voltage magnitude and angle of each bus, which are 27 and 235 for the IEEE 14-bus and 118-bus systems, respectively (the voltage angle of reference bus is 0).

After obtaining the measurement vectors \bar{z} including $[V_i, P_i, Q_i, P_{ij}, Q_{ij}]^T$ from MATPOWER, white Gaussian noises are added to form the noised measurements $z = \bar{z} + a\sigma$, where $a \sim N(0, 1)$ and σ is the standard deviation. The number of each measurement and standard deviation are shown in Table III. Note that the number of measurements is greater than that of states to ensure the observability [32].

The time horizon of the dataset is 4 months, i.e., 34560 data points with 5 min time interval. The first two months with 17280 data points are split into two parts: training (70%) and validation (30%). The training data are used in

both initialization and pre-training stages. The validation data are used to validate the PGDL performance. The remaining data of the two months are used for online training and state estimation, and the performance of the last day (288 data points) is reported for comparative study.

TABLE III
MEASUREMENT SIZES AND STANDARD DEVIATION OF IEEE TEST SYSTEMS

Measurement	Size		Standard deviation
	IEEE 14-bus system	IEEE 118-bus system	
V_i	1	1	0.004
P_i	7	118	0.010
Q_i	5	118	0.010
P_{ij}	7	44	0.008
Q_{ij}	10	49	0.008

B. Accuracy Analysis

The MAEEs for IEEE 14-bus system are presented in Figs. 9 and 10. It is observed that WLS has the largest variations of estimation errors. In contrast, FFNN and LSTM result in small error variations. In terms of accuracy, the two neural networks also have smaller average MAEEs. To compare the configurations of two neural networks, it is found that while the average MAEEs are almost the same, LSTM has lower standard deviation of errors than FFNN.

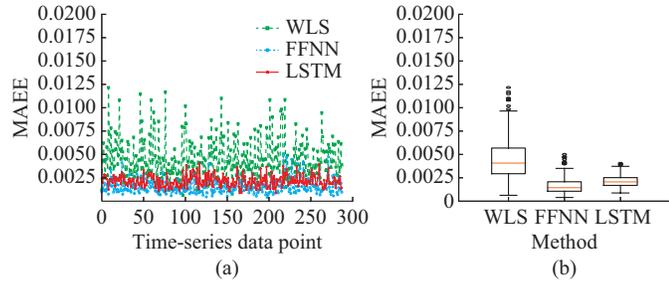


Fig. 9. MAEEs of voltage magnitudes in IEEE 14-bus system. (a) MAEE at time-series data points with 5 min time interval. (b) Boxplot of MAEE with different methods.

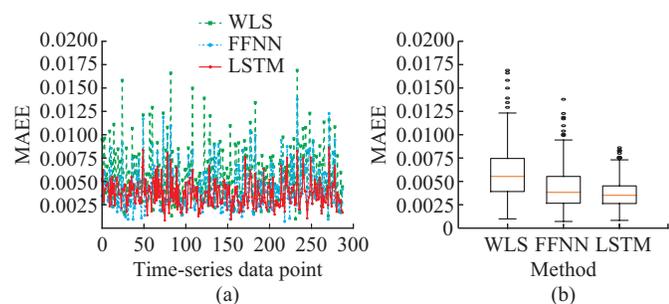


Fig. 10. MAEEs of voltage angles in IEEE 14-bus system. (a) MAEE at time-series data points with 5 min time interval. (b) Boxplot of MAEE with different methods.

For detailed analysis, the mean and standard deviation are described in Table IV. Generally, WLS has the highest average error for voltage magnitude V and angle θ , while two

deep learning based models are more accurate. FFNN and LSTM also have lower standard deviation of MAEE, which indicates more stable performance throughout the testing period. It is also noticed that the improvement of voltage magnitude estimation is more significant than voltage angle estimation.

TABLE IV
MEAN AND STANDARD DEVIATION OF MAEES IN IEEE 14-BUS SYSTEM

Method	Mean		Standard deviation	
	V	θ	V	θ
WLS	0.0046	0.0059	0.0022	0.0029
FFNN	0.0017	0.0044	0.0008	0.0023
LSTM	0.0021	0.0037	0.0006	0.0014

To generalize the proposed approach, the IEEE 118-bus system is used to test the performance in a larger system. Figures 11 and 12 provide a graphical illustration of the MAEEs between WLS and two DNN-based methods. It clearly shows that LSTM provides the best estimates for voltage magnitudes, while FFNN performs the best for voltage angles.

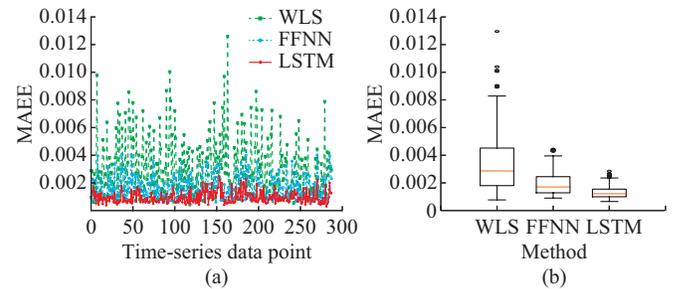


Fig. 11. MAEEs of voltage magnitudes in IEEE 118-bus system. (a) MAEE at time-series data points with 5 min time interval. (b) Boxplot of MAEE with different methods.

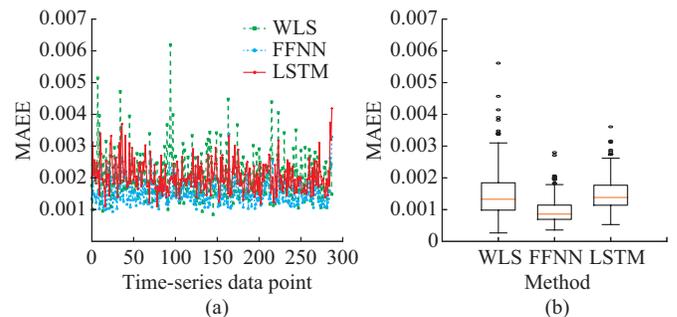


Fig. 12. MAEEs of voltage angles in IEEE 118-bus system. (a) MAEE at time-series data points with 5 min time interval. (b) Boxplot of MAEE with different methods.

Table V shows a statistical comparison of WLS and two DNN-based approaches. Overall, the DNN-based methods outperform WLS. LSTM performs significantly better on voltage magnitude, while its performance of voltage angle estimation are on par with WLS. On the other hand, the DNN-based method are much more stable than WLS estimation with lower error variance.

TABLE V
MEAN AND STANDARD DEVIATION OF MAEES IN IEEE 118-BUS SYSTEM

Method	Mean		Standard deviation	
	V	θ	V	θ
WLS	0.0032	0.0021	0.0021	0.0007
FFNN	0.0018	0.0015	0.0008	0.0004
LSTM	0.0011	0.0021	0.0004	0.0005

System-wise estimates versus true states are also plotted in Figs. 13 and 14. The results are consistent with the previous observations. Specifically, all three methods track the voltage angles very well while LSTM shows slightly better results.

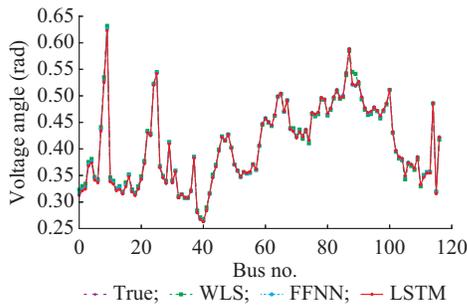


Fig. 13. Voltage angles for IEEE 118-bus system.

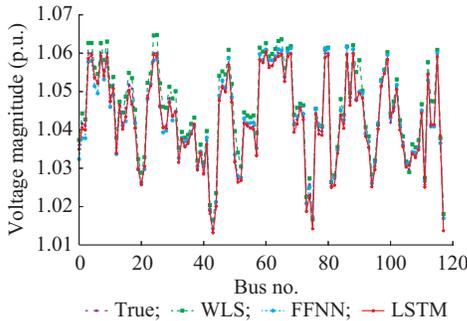


Fig. 14. Voltage magnitudes for IEEE 118-bus system.

For voltage magnitude, WLS has the worst performance, while LSTM is the best and outperforms the FFNN. The results further demonstrate that the temporal learning capability in the encoder DNN would enhance the overall estimation performance.

Regarding the runtime, both DNNs could achieve reasonably good execution time. Note that the network pre-training is conducted offline and does not affect the online execution time. For online estimation, only one-step forward calculation and one-step backward propagation are involved, and both are algebraic calculations leading to relatively fast runtime. Specifically, at each time step, DNNs are executed in 0.005 s for the IEEE 14-bus system and 0.02 s for the 118-bus system. FFNN is slightly faster than LSTM due to its simpler structure.

C. Robustness Analysis

A state estimator is considered statistically robust if the es-

timated states withstand deviations of measurements [7]. We compare the robustness of PGDL to WLS and the more robust WLAV. WLAV attempts to find the estimated states by minimizing the L_1 norm between real measurements and estimated measurements. We implement the WLAV state estimator using IPOPT [33].

To test the robustness, 2%-3% measurements are randomly selected and replaced with corrupted data. The results of the last day are used to test the robustness. Three corrupted data scenarios are considered as follows.

- 1) Scenario 1: the magnitudes of selected measurements are contaminated with 20 times of standard deviation.
- 2) Scenario 2: the selected measurements are set to be zero, representing missing data in reality.
- 3) Scenario 3: the signs of selected measurements are reversed, representing mis-communication in reality.

For each scenario, multiple simulation runs are conducted and each run has 24 hours (288 observations) of single system snapshots.

For the IEEE 14-bus system with 30 measurements, 50 simulation runs are conducted with 2 corrupted data points randomly selected for each run. Note that both WLS and WLAV have non-converging cases. For example, WLS has 1, 7, and 11 non-converging cases for three scenarios, respectively, while WLAV has 1 non-converging case for scenario 1. Note that PGDL methods will always produce results without convergence concerns. We remove non-convergence cases for WLS and WLAV, and the results are shown in Fig. 15. It is observed that LSTM-based model outperforms the others in all three scenarios. The LSTM-based model produces the lowest median in errors and smaller inter-quartile range compared to WLS and WLAV. Surprisingly, FFNN shows the worst performance in all four methods. This is because non-converging cases are not counted in WLS and WLAV.

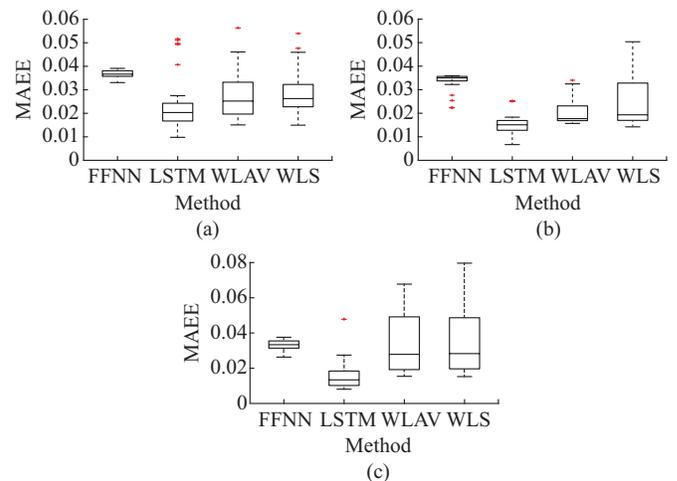


Fig. 15. MAEE distribution with corrupted data in IEEE 14-bus system. (a) Scenario 1. (b) Scenario 2. (c) Scenario 3.

For the IEEE 118-bus system with 330 measurements, 20 simulation runs are conducted with 5 corrupted data and 10 corrupted data that are randomly selected, respectively. The results are reported in Fig. 16. It can be observed that the

overall performance of LSTM is reasonably good compared to those of other methods. In all three scenarios, LSTM has the lowest average errors while WLS has the lowest error variance.

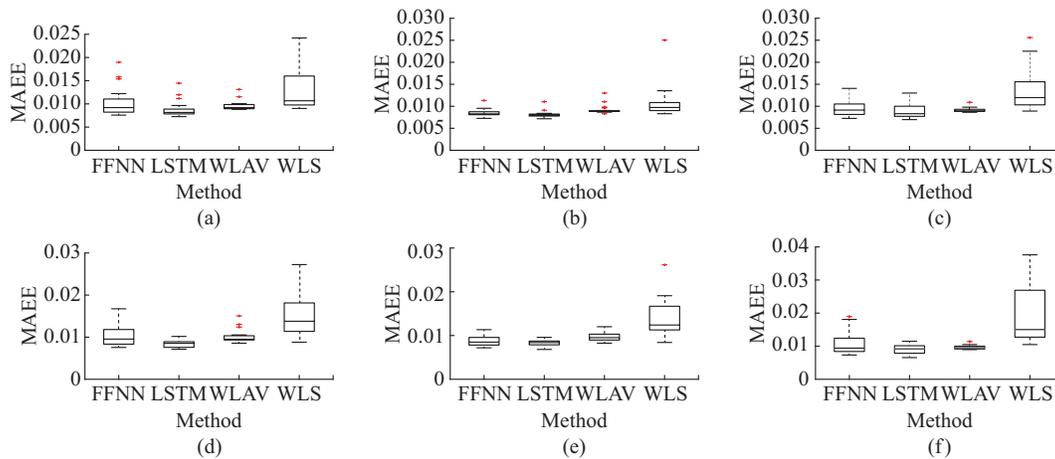


Fig. 16. MAEE distribution with corrupted data in IEEE 118-bus system. (a) Scenario 1 with 5 corrupted data. (b) Scenario 2 with 5 corrupted data. (c) Scenario 3 with 5 corrupted data. (d) Scenario 1 with 10 corrupted data. (e) Scenario 2 with 10 corrupted data. (f) Scenario 3 with 10 corrupted data.

Overall, comparing two different DNNs, the mean accuracy improvement of one over the other is subtle given that both methods achieve far better results than WLS. However, LSTM is definitely more stable than FFNN in terms of lower standard deviation of the error index. This can be further seen from the robustness test, where FFNN fails to maintain a low error index with corrupted data, while LSTM remains high accuracy and stable performance.

V. DISCUSSION

This paper investigates the possibility of combining machine learning with physical models for state estimation. The accuracy and robustness are examined in the IEEE 14-bus and 118-bus systems. Some aspects regarding the applicability are worthy to be discussed:

- 1) Machine learning can be used in state estimation. However, it is not intended to replace physics-based methods. Rather, it can be used in combination when WLS or WLAV could not yield converged solutions.
- 2) Incorporating physical models in the machine learning algorithms results in meaningful results for state estimation, and improves both accuracy and robustness.
- 3) The performance of state estimator can be improved by learning the existing temporal correlations. This is demonstrated by the better performance of LSTM compared to that of FFNN.

VI. CONCLUSION

A PGDL state estimator is proposed in the paper. The learning capacity of DNNs is fully exploited to model the temporal correlations among system states, in contrast to snapshot estimation by traditional methods. In addition, unlike the purely data-driven supervised learning, nonlinear power equations are attached and the deviation of estimated measurements from actual observed values is taken as the loss to train the neural networks. As a result, the proposed PGDL method shows higher accuracy compared to WLS,

and is more robust to corrupted data compared to WLS and WLAV. The machine-learning-based state estimation appears to be promising in real control centers, but it is not intended to replace but assist in physics-guided state estimation. As for the future work, system parameters errors need to be considered.

REFERENCES

- [1] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, May 2015.
- [2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge: MIT Press, 2016.
- [3] D. Silver, A. Huang, C. J. Maddison *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, p. 484, Jan. 2016.
- [4] S. Fan and L. Chen, "Short-term load forecasting based on an adaptive hybrid method," *IEEE Transactions on Power Systems*, vol. 21, no. 1, pp. 392-401, Jan. 2006.
- [5] 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.
- [6] F. C. Schweppe and J. Wildes, "Power system static state estimation, Part I: exact model," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-89, no. 1, pp. 120-125, Jan. 1970.
- [7] A. Abur and A. Exposito, *Power System State Estimation: Theory and Implementation*, 1st ed., Boca Raton: CRC Press, 2004.
- [8] T. A. S. Corporation and A. Gelb, *Applied Optimal Estimation*. Cambridge: MIT Press, 1974.
- [9] A. Wood, B. Wollenberg, and G. Sheble, *Power Generation, Operation, and Control*. 3rd ed., Hoboken: Wiley, Nov. 2013.
- [10] M. U. Usman and M. O. Faruque, "Applications of synchrophasor technologies in power systems," *Journal of Modern Power Systems and Clean Energy*, vol. 7, no. 2, pp. 211-226, Mar. 2019.
- [11] M. Gol and A. Abur, "LAV based robust state estimation for systems measured by PMUs," *IEEE Transactions on Smart Grid*, vol. 5, no. 4, pp. 1808-1814, Jul. 2014.
- [12] M. Gl and A. Abur, "A hybrid state estimator for systems with limited number of PMUs," *IEEE Transactions on Power Systems*, vol. 30, no. 3, pp. 1511-1517, May 2015.
- [13] C. Xu and A. Abur, "Robust linear state estimation using multi-level power system models with different partitions," in *Proceedings of 2017 IEEE Manchester PowerTech*, Manchester, UK, Jun. 2017.
- [14] 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.
- [15] J. Luo, T. Hong, and M. Yue, "Real-time anomaly detection for very

- short-term load forecasting,” *Journal of Modern Power Systems and Clean Energy*, vol. 6, no. 2, pp. 235-243, Mar. 2018.
- [16] Y. Chakhchoukh, V. Vittal, and G. T. Heydt, “PMU based state estimation by integrating correlation,” *IEEE Transactions on Power Systems*, vol. 29, no. 2, pp. 617-626, Mar. 2014.
- [17] Y. Weng, R. Negi, C. Faloutsos *et al.*, “Robust data-driven state estimation for smart grid,” *IEEE Transactions on Smart Grid*, vol. 8, no. 4, pp. 1956-1967, Jul. 2017.
- [18] 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.
- [19] M. Raissi, P. Perdikaris, and G. E. Karniadakis. (2017, Nov.). Physics informed deep learning. Part I: data-driven solutions of nonlinear partial differential equations. [Online]. Available: <http://arxiv.org/abs/1711.10561>
- [20] A. Karpatne, W. Watkins, J. S. Read *et al.* (2018, Feb.). Physics-guided neural networks (PGNN): an application in lake temperature modeling. [Online]. Available: <http://arxiv.org/abs/1710.11431>
- [21] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504-507, Jul. 2006.
- [22] K. Hornik, “Approximation capabilities of multilayer feedforward networks,” *Neural Networks*, vol. 4, no. 2, pp. 251-257, Mar. 1991.
- [23] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, Sardinia, Italy, May 2010, pp. 1-8.
- [24] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, Nov. 1997.
- [25] R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training recurrent neural networks,” in *Proceedings of the 30th International Conference on Machine Learning*, Atlanta, USA, Jun. 2013, pp. 1-9.
- [26] X. Li, Y. Grandvalet, and F. Davoine. (2018, Feb.). Explicit inductive bias for transfer learning with convolutional networks. [Online]. Available: <http://arxiv.org/abs/1802.01483>
- [27] L. N. Smith. (2017, Apr.). No more pesky learning rate guessing games. [Online]. Available: <http://www.oalib.com/paper/4083635>
- [28] S. Ruder. (2017, Jun.). An overview of gradient descent optimization algorithms. [Online]. Available: <https://arxiv.org/abs/1609.04747>
- [29] L. Wang. (2019, Jun.). Physics-guided deep learning for power system state estimation. [Online]. Available: <https://github.com/RayUCF/PSSECode>
- [30] NYISO. (2019, Mar.). NYISO load profile. [Online]. Available: <http://www.nyiso.com>
- [31] MATPOWER. (2019, Mar.). MARPOWER 6.0 Manual. [Online]. Available: <http://www.pserc.cornell.edu/matpower/docs>
- [32] E. E. Fetzter and P. M. Anderson, “Observability in the state estimation of power systems,” *IEEE Transactions on Power Apparatus and Systems*, vol. 94, no. 6, pp. 1981-1988, Nov. 1975.
- [33] J. Currie and D. I. Wilson, “OPTI: lowering the barrier between open source optimizers and the industrial MATLAB user,” in *Proceedings of Foundations of Computer Aided Process Operations Conference*, Savannah, USA, Jan. 2012, pp. 1-6.

Lei Wang received the M.S. degree in electrical engineering from University of Central Florida, Orlando, USA, in 2019. He is currently working toward the Ph.D. degree in the Department of Electrical Engineering, Centre for Advanced Power System, Florida State University, Tallahassee, USA. His research interests include cyber-security, robust state estimation and control of power system, smart grid and data analytics.

Qun Zhou is an Assistant Professor at University of Central Florida, Orlando, USA. She is the director of Smart Infrastructure Data Analytics Laboratory, University of Central Florida. Before joining University of Central Florida, she worked for Genscape and GE Grid Solutions as a power system engineer. She received her Ph.D. degree in electrical engineering from Iowa State University, Ames, USA. She is devoted to improving energy efficiency and customer engagement through data analytics, probabilistic modeling, and advanced pricing schemes. Her research interests include grid-edge resources, including smart buildings, rooftop PVs, and batteries, and their interactions with the grid.

Shuangshuang Jin received the Bachelor’s degree in computer science from Wuhan University, Wuhan, China, in 2001, and the Master’s and Ph.D. degrees in computer science from Washington State University, Pullman, USA, in 2003 and 2007, respectively. She joined PNNL as a research engineer in 2008 and became a senior research scientist in 2013. She is currently an Associate Professor at College of Engineering, Computing and Applied Sciences, Clemson University, North Charleston, USA. Her research interests include high-performance computing (HPC) and parallel programming on shared or distributed memory platforms, application of these technologies into power system modeling and simulation, renewable energy integration, and advanced grid analytics.