POWER SYSTEM STATE FORECASTING VIA DEEP RECURRENT NEURAL NETWORKS

Liang Zhang, Gang Wang, and Georgios B. Giannakis

Dept. of ECE and Digital Tech. Center, Univ. of Minnesota, Mpls, MN 55455, USA
E-mails: {zhan3523, gangwang, georgios}@umn.edu

ABSTRACT

State forecasting plays a critical role in power system monitoring, by offering system awareness even ahead of the time horizon, enhancing system observability, and providing efficient identification of the grid topology and link parameter changes. However, available approaches relying on linear estimators or single-hidden-layer feed-forward neural networks (FNNs), cannot capture long-term nonlinear dependencies in the voltage time series, and lead to suboptimal performance. To bypass these hurdles, this paper advocates deep recurrent neural networks (RNNs) for power system state forecasting. Deep RNNs capture long-term dependencies, and are easy to implement. By also leveraging the physics behind power systems, a novel architecture based on prox-linear nets (RPLN) is further developed for state forecasting based on past measurements. Simulated tests show improved performance of the proposed RNN and RPLN predictors when compared to FNN and vector autoregression based alternatives.

***Index Terms*—** Power system state forecasting, recurrent neural network, recurrent prox-linear net, data validation.

1. INTRODUCTION

Power grids are currently facing major challenges related to rapid voltage fluctuations due to massive integration of electric vehicles, and intermittent generation of renewable energy. As a result, monitoring and tracking system states becomes increasingly critical, not only for system protection [15], but also for energy management [17]. To that end, static power system state estimation (PSSE) that aims at recovering all nodal voltages from a subset of related measurements, must be revisited to account for variations in the state when PSSE results become available to the system operator [15]. In addition, PSSE works well only if the grid topology and link parameters are perfectly known, and there are enough measurements ensuring system observability. To cope with unknown dynamics and create extra voltage measurements, power system state forecasting is well motivated.

Power system state forecasting has been pursued using Kalman filtering with an identity state transition matrix [2]. To improve prediction performance, (block) diagonal transition matrices that are updated as new measurements become available, have been also considered [8, 11]; see also [9] for a recent approach to state prediction based on first-order vector auto-regressive (VAR) modeling. All aforementioned predictors however, assume a *linear* state transition function; yet the dependence of system state on previous states is often nonlinear (quadratic with power measurements).

To render *nonlinear* estimators tractable, state forecasting based on feed-forward neural networks (FNNs) has been suggested with the state transition map modeled by a single hidden layer [3,4]. Once trained off-line using past data, the FNN enables real-time state forecasting at affordable complexity in the operational phase. Unfortunately, the number of FNN parameters grows linearly with the length of input sequences, which either discourages FNNs from capturing long-term dependencies in time series of voltages, or, lowers forecasting performance when short-term memory is adopted.

In this context, we advocate deep recurrent neural networks (RNNs) that are well poised to forecast power system states from time-series measurements. Deep RNNs can capture complex nonlinear dependencies present in time series data, they involve a fixed number of parameters even with input sequences of variable length, and are easy to implement using publicly available platforms such as 'TensorFlow.' Per time slot t, the deep RNN leverages an estimate of the current state $\mathbf{v}_t$ to predict $\mathbf{v}_{t+1}$. However, estimating $\mathbf{v}_t$ using optimization methods can be time consuming, which may hamper scalable and efficient state forecasting. To bypass this limitation, we further introduce what we term recurrent prox-linear nets (RPLNs), which nicely wed prox-linear nets with RNNs, to predict system states based on past measurements. Without our novel combination with RNNs here for state forecasting, prox-linear nets were recently developed in [19] for static PSSE by unrolling the corresponding solver [14]. Numerical tests on real data will be also presented to corroborate the merits of the developed deep RNN and RPLN approaches relative to existing alternatives.

Regarding notation, lower- (upper-) case boldface letters represent column vectors (matrices). Symbol $^\top$ ($^\mathcal{H}$) stands for matrix (conjugate) transposition. Finally, sets are represented by calligraphic fonts.

This work was supported in part by NSF grants 1508993, 1509040, and 1711471.

2. METHODOLOGY

Consider a power network comprising N buses, modeled as a graph $\mathcal{G} := \{\mathcal{N}, \mathcal{L}\}$, where $\mathcal{N} := \{1, \ldots, N\}$ and $\mathcal{L} := \{(i,j)\} \in \mathcal{N} \times \mathcal{N}$ collect all buses and edges, respectively. Per bus $n \in \mathcal{N}$, let $V_n := |V_n|e^{j\theta_n}$ be its voltage phasor, and $S_n := P_n + jQ_n$ its complex power injection, with P_n (Q_n) denoting the active (reactive) power injection. Per line $(i,j) \in \mathcal{L}$, let $S_{ij}^f := P_{ij}^f + jQ_{ij}^f$ denote the complex power flow at the 'forwarding' end with P_{ij}^f (Q_{ij}^f) representing the active (reactive) power flow. Likewise, let S_{ij}^e, P_{ij}^e, and Q_{ij}^e denote the complex, active, and reactive power flows at the 'terminal' end of edge (i,j). To perform system state forecasting, suppose that M_t system variables are measured at time t. For brevity, let $\mathbf{z}_t := \big[\{|V_{n,t}|^2\}_{n\in\mathcal{N}_t^o}, \{P_{n,t}\}_{n\in\mathcal{N}_t^o}, \{Q_{n,t}\}_{n\in\mathcal{N}_t^o}, \{P_{ij,t}^f\}_{(i,j)\in\mathcal{E}_t^o}, \{Q_{ij,t}^f\}_{(i,j)\in\mathcal{E}_t^o}, \{P_{ij,t}^e\}_{(i,j)\in\mathcal{E}_t^o}, \{Q_{ij,t}^e\}_{(i,j)\in\mathcal{E}_t^o}\big]^\top$ concatenate all observed quantities at time t, where the sets $\mathcal{N}_t^o$ and $\mathcal{E}_t^o$ signify the locations where the corresponding nodal and line quantities are measured.

Per time t, given measurements $\mathbf{z}_t$ along with corresponding measurement matrices, the system state $\mathbf{v}_t \in \mathbb{C}^N$ can be estimated using standard static PSSE modules [15,19]. The obtained time-series $\{\mathbf{v}_\tau\}_{\tau=0}^t$ can be subsequently used to predict the system state at the next time step, namely, $\mathbf{v}_{t+1}$ [3]. Mathematically, the estimation and prediction steps are summarized using the following equations

$$\mathbf{v}_{t+1} = \boldsymbol{\phi}(\mathbf{v}_t, \mathbf{v}_{t-1}, \mathbf{v}_{t-2}, \ldots, \mathbf{v}_{t-r+1}) + \boldsymbol{\eta}_t \quad (1)$$

$$\mathbf{z}_{t+1} = \mathbf{h}_{t+1}(\mathbf{v}_{t+1}) + \boldsymbol{\epsilon}_{t+1} \quad (2)$$

where $r \geq 1$ denotes the number of lagged states used to predict $\mathbf{v}_{t+1}$; $\{\boldsymbol{\eta}_t, \boldsymbol{\epsilon}_{t+1}\}$ account for modeling inaccuracies and measurement noise; $\boldsymbol{\phi}$ is an unknown function describing the state transition map, and $\mathbf{h}_{t+1}(\cdot)$ is the measurement function at time $t+1$. To perform state forecasting, we need to estimate $\boldsymbol{\phi}$ that we will model using RNNs, as we present next.

2.1. Deep RNNs for Forecasting and PSSE

RNNs are NNs designed to learn from correlated time series data. They are not only scalable to long sequence inputs, meaning sequences with large r, but also capable of dealing with input sequences of variable length [7]. Given input sequence $\{\mathbf{v}_\tau\}_{\tau=t-r+1}^t$, and initial state $\mathbf{s}_{t-r}$, an RNN finds the hidden state vector sequence $\{\mathbf{s}_\tau\}_{\tau=t-r+1}^t$ using

$$\mathbf{s}_\tau = f(\mathbf{W}^0\mathbf{v}_t + \mathbf{W}^{ss}\mathbf{s}_{\tau-1} + \mathbf{b}^0) \quad (3)$$

where $f(\cdot)$ is a pre-selected nonlinear activation function (such as a sigmoid or a rectified linear unit), that is understood applied to a vector entry-wise, while matrices $\mathbf{W}^0$, $\mathbf{W}^{ss}$, and vector $\mathbf{b}^0$ contain time-invariant weight coefficients.

Deep RNNs are RNNs of multiple (≥ 3) processing layers, which are responsible for learning representations of time series with hierarchical nonlinear transformations. Deep RNN-based approaches have dramatically improved upon the state-of-the-art in diverse sequence processing applications, such as machine translation and music prediction [7]. One way to construct deep RNNs is by stacking up multiple recurrent hidden layers one on top of another, as follows [12]

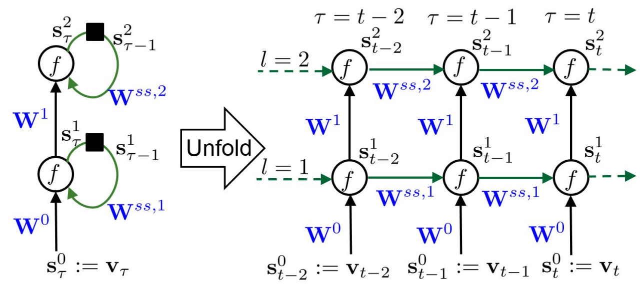

Fig. 1: An unfolded deep RNN with no outputs.

$$\mathbf{s}_\tau^l = f\big(\mathbf{W}^{l-1}\mathbf{s}_\tau^{l-1} + \mathbf{W}^{ss,l}\mathbf{s}_{\tau-1}^l + \mathbf{b}^{l-1}\big), \quad l \geq 1 \quad (4)$$

where l is the layer index, $\mathbf{s}_\tau^l$ denotes the so-called hidden state of the l-th layer at time τ with $\mathbf{s}_\tau^0 := \mathbf{v}_\tau$, and $\{\mathbf{W}^l, \mathbf{W}^{ss,l}, \mathbf{b}^l\}$ comprise unknown weights. Fig. 1 (left) shows the computational graph representing (4) for $l = 2$, with the bias vectors $\mathbf{b}^l = \mathbf{0}\ \forall l$ for simplicity in depiction, and the black squares indicating a one-step delay unit. The unfolded version of this graph is in Fig. 1 (right) with rows representing layers, and columns corresponding to time slots.

The RNN output can come in various forms, including one output per time step, or, one output after a sequence of steps. The latter matches the rth-order nonlinear regression in (1). Concretely, the output of our deep RNN is given by

$$\check{\mathbf{v}}_{t+1} = \mathbf{W}^{out}\mathbf{s}_t^l + \mathbf{b}^{out} \quad (5)$$

where $\check{\mathbf{v}}_{t+1}$ denotes the forecast of $\mathbf{v}_{t+1}$, and $\mathbf{W}^{out}$ and $\mathbf{b}^{out}$ contain weights of the output layer. Given historical nodal voltage time series, $\{\mathbf{W}^{out}, \mathbf{b}^{out}, \mathbf{W}^l, \mathbf{W}^{ss,l}, \mathbf{b}^l\}$ can be learned end-to-end using a back-propagation solver [7].

Although the focus here is on one-step forecasting, it is worth stressing that with minor modifications, our proposed approaches can be adapted to predict the power system states multiple steps ahead.

So far, we have seen how RNNs enable nonlinear predictors for power state forecasting, that is how past voltages and (4) can yield $\check{\mathbf{v}}_{t+1}$ in (5). In addition, $\check{\mathbf{v}}_{t+1}$ can be employed as a prior to aid PSSE when a new measurement $\mathbf{z}_{t+1}$ becomes available, by solving a regularized problem as

$$\hat{\mathbf{v}}_{t+1} := \arg\min_{\mathbf{v}_{t+1}} \ell_1(\mathbf{z}_{t+1} - \mathbf{h}_{t+1}(\mathbf{v}_{t+1})) + \lambda\ell_2(\mathbf{v}_{t+1} - \check{\mathbf{v}}_{t+1}) \quad (6)$$

where $\ell_1(\cdot)$ and $\ell_2(\cdot)$ are pre-selected fitting loss functions, e.g., least-squares or least-absolute-value errors, and λ is a

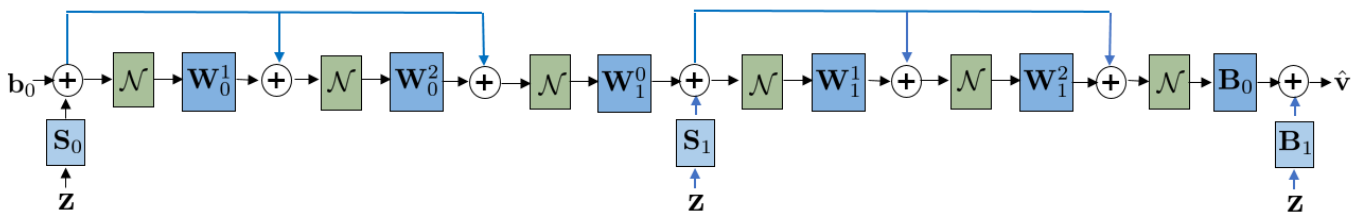

Fig. 2: Prox-linear net with 6 hidden layers.

regularization parameter. When $\lambda > 0$, the prediction $\check{\mathbf{v}}_{t+1}$ serves as an additional set of virtual (a.k.a. pseudo) measurements, which are known to improve system observability. The minimization in (6) is carried using off-the-self PSSE solvers; see e.g., [15, 19]. One remark is now in order.

Remark 1. A forecasting-aided PSSE approach based on FNNs with a single hidden layer, was introduced by [3]. It can be viewed as a special case of (6), which broadens the class of nonlinear predictors with memory by invoking RNNs. As will be seen in our numerical tests, forecasting performance improves considerably leveraging the deep RNNs. Our regularized PSSE in (6) is also reminiscent of the predictor-corrector estimator form emerging with dynamic state estimation problems using Kalman filters.

2.2. Deep RPLNs for Computational Efficiency

The RNN-based approach of the previous section is attractive provided that (6) can be solved efficiently. But with power measurements being quadratic functions of voltages, the fitting loss function in (6) is nonconvex. As a result, convergence and performance of iterative solvers relies critically on the initialization [16]. To cope with these challenges, we designed deep FNNs for PSSE in [19], by approximately linearizing the nonconvex loss in (6) per iteration.

Here we will design related prox-linear nets (PLNs), but for deep RNNs. To this end, consider the 6-layer PLN in Fig. 2, where $\mathbf{z}$ is the input measurement vector, $\mathcal{N}$ in all green blocks represents the pre-selected entry-wise nonlinear activation functions, such as rectified linear units (ReLU), 'tanh', or soft-thresholding operator; while $\mathbf{b}_0$, $\{\mathbf{S}_k\}_{k=0}^{l}$, $\{\mathbf{W}_k^l\}_{0\le k\le 1}^{1\le l\le 3}$, and $\{\mathbf{B}_k\}_{k=0}^{1}$ contain weights learned in the training via backpropagation. Relative to conventional FNNs, the developed PLN features 'skip-connections' (the bluish lines on top of Fig. 2) that directly connect $\mathbf{z}$ to intermediate/output layers. Such connections have been empirically shown to improve training efficiency of FNNs [10, 19].

After training our PLN off-line using historical and/or simulated data, we will employ it to approximately solve (6) in real time. As with the FNNs in [19], this will markedly improve computational efficiency of our RNN-based forecasting and PSSE. When inferring $\check{\mathbf{v}}_{t+1}$ at time t, our PLN-based approach proceeds in two stages: the first stage yields the PLN-based estimate $\hat{\mathbf{v}}_t$; and the second stage uses $\{\hat{\mathbf{v}}_\tau\}_{\tau=t-r+1}^{t}$ as input to the trained RNN described in Sec. 2 to obtain $\check{\mathbf{v}}_{t+1}$. Note that PLN and RNN parameters here are learned separately. Recent studies however, suggest that end-to-end learning, which refers to the deep learning approaches where all parameters are learned jointly, leads to improved performance [7, 13]. Prompted by this, we will connect the PLN output directly with the RNN input, and train the two networks jointly. Fig. 3 shows our novel network that we naturally term recurrent prox-linear net (RPLN). Note that PLN parameters in Fig. 3 are also time-invariant, and they can be initialized using the learned PLN parameters in [19].

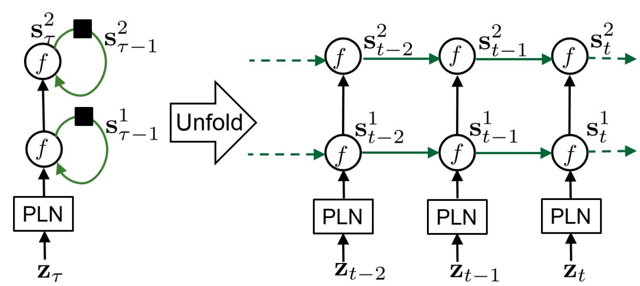

Fig. 3: An unfolded recurrent prox-linear net with no outputs.

Remark 2. Besides PSSE, state forecasting can aid system observability by providing extra (pseudo) measurements [5, 18]. It can also facilitate grid topology and link parameter identification (see e.g., [6]). Indeed, if a topology change between t and $t+1$ is not reported, the operator has only outdated information about the state-to-measurement map $\mathbf{h}_{t+1}(\cdot)$ in (2), which adversely affects $\hat{\mathbf{v}}_{t+1}$. As a result, entries of the measurement residual vector $\boldsymbol{\rho}_{t+1} := \mathbf{z}_{t+1} - \mathbf{h}_{t+1}(\hat{\mathbf{v}}_{t+1})$ cannot reveal this change. On the other hand, topology changes have no effect on the forecast $\check{\mathbf{v}}_{t+1}$, but certainly influence the so-termed measurement innovation vector [3]

$$\boldsymbol{\nu}_{t+1} := \mathbf{z}_{t+1} - \mathbf{h}_{t+1}(\check{\mathbf{v}}_{t+1}). \tag{7}$$

Statistical analysis of $\boldsymbol{\nu}_{t+1}$ and $\boldsymbol{\rho}_{t+1}$ can unveil and remove erroneous data; see [3] for an overview. In a nutshell, the benefits of our RNN and RPLN based forecasting permeate to critical power system tasks, including PSSE, observability, as well as identification of topology and link parameter changes.

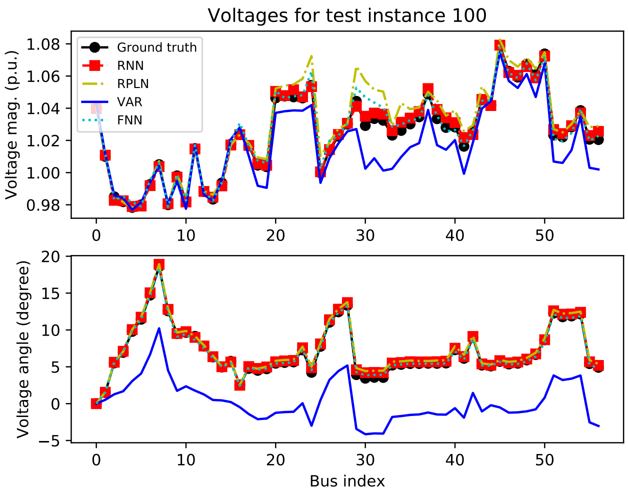

Fig. 4: Forecasting errors in voltage magnitudes and angles for all 57 buses at test instance 100.

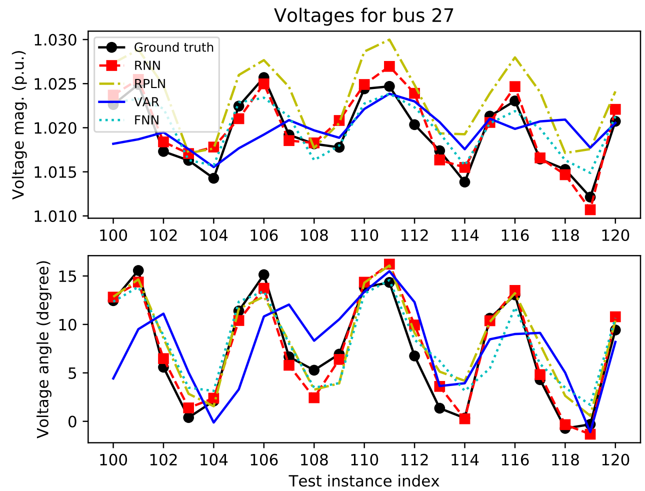

Fig. 5: Forecasting errors in voltage magnitudes and angles for bus 27 from test instances 100 to 120.

3. NUMERICAL TESTS

Performance of our deep RNN and RPLN methods was evaluated using the IEEE 57-bus test system. To obtain the training and testing time series, real load data provided by the 2012 Global Energy Forecasting Competition [1] were used. The series of loads was subsampled, and subsequently normalized to match the scale of power demands in the system. The MATPOWER toolbox [20] was employed to solve the AC power flow equations from the normalized load series, to obtain the actual voltage time series $\{\mathbf{v}_\tau\}$, and generate the measurement time series $\{\mathbf{z}_\tau\}$ that consists of all active (reactive) power flows at the forwarding end of each line and voltage magnitude. Concretely, time series $\{(\mathbf{z}_\tau, \mathbf{v}_\tau)\}_{\tau=0}^{7674}$ were obtained, where the first 80% $\left(\{(\mathbf{z}_\tau, \mathbf{v}_\tau)\}_{\tau=0}^{6174}\right)$ time instances were used for training, while the remaining ones were kept for testing. The forecasting performance of deep RNN and RPLN based approaches was assessed in terms of the normalized root mean-square error (RMSE) defined as $\|\check{\mathbf{v}} - \mathbf{v}\|_2/N$, in which $\mathbf{v}$ is the actual voltage profile, and $\check{\mathbf{v}}$ the estimate found by the deep RNN or RPLN.

Specifically, deep RNNs with $l = 3$, $r = 10$, and 'tanh' activation functions were simulated, whereas the RPLN was formed by combining a 6-hidden-layer PLN with the aforementioned RNN. The number of hidden units per layer in both RNN and RPLN were kept the same as the input dimension, namely $57 \times 2 = 114$. For comparison, the single-hidden-layer FNN [4], and a VAR(1) model [9] based state forecasting approaches were adopted as benchmarks. Note that the proposed deep RNN approach has $52,440$ parameters, while the single-hidden-layer FNN has $143,184$ parameters. All NNs were trained using 'TensorFlow' [1] on a NVIDIA Titan X GPU of 12GB RAM. The weight matrices of all NNs were learned by running the 'Adam' optimizer for 100 epochs with a start learning rate of 10^{-3}. The forecasted state using the VAR(1) model can be expressed in closed form [9]. With regards to estimation performance, the average RMSEs over $1,500$ testing instances for the deep RNN, RPLN, FNN and VAR(1) are 0.2172, 0.2694, 0.2958, and 0.6772, respectively. Evidently, these numbers confirm the markedly improved performance of our deep RNN approach. Although our RPLN approach predicts the voltage directly from past measurements $\{\mathbf{z}_\tau\}$, it achieves competitive performance relative to FNN and VAR(1). The true voltages and the forecasted ones provided by the deep RNN, RPLN, FNN, and VAR(1) for all buses on test instance 100 as well as for bus 27 from test instances 100 to 120, are reported in Figs. 4 and 5, respectively. Clearly, the deep RNN yields the best performance in both Figs. 4 and 5.

[1] https://www.kaggle.com/c/global-energy-forecasting-competition-2012-load-forecasting/data.

4. CONCLUSIONS

This paper dealt with power system state forecasting using nonlinear prediction based approaches. Specifically, deep RNNs and RPLNs were introduced for state forecasting based on past voltages and past measurements, respectively. The proposed approaches not only account for the long-term nonlinear dependencies present in time-series inputs, but also they are computationally inexpensive, easy-to-implement, and fast. Preliminary tests on the IEEE 57-bus benchmark system using real load data showcase the merits of our developed approaches relative to existing alternatives.

Our current and future research agenda includes 'on-the-fly' RNN-based algorithms to account for dynamically changing environments, and corresponding time dependencies.

5. REFERENCES

[1] M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[2] A. S. Debs and R. E. Larson, "A dynamic estimator for tracking the state of a power system," *IEEE Trans. Power App. Syst.*, vol. 89, no. 7, pp. 1670–1678, Sept. 1970.

[3] M. B. Do Coutto Filho and J. C. Stacchini de Souza, "Forecasting-aided state estimation–Part I: Panorama," *IEEE Trans. Power Syst.*, vol. 24, no. 4, pp. 1667–1677, Nov. 2009.

[4] M. B. Do Coutto Filho, J. C. Stacchini de Souza, and R. S. Freund, "Forecasting-aided state estimation–Part II: Implementation," *IEEE Tran. Power Syst.*, vol. 24, no. 4, pp. 1678–1685, Nov. 2009.

[5] G. B. Giannakis, V. Kekatos, N. Gatsis, S.-J. Kim, H. Zhu, and B. Wollenberg, "Monitoring and optimization for power grids: A signal processing perspective," *IEEE Signal Process. Mag.*, vol. 30, no. 5, pp. 107–128, Sep. 2013.

[6] G. B. Giannakis, Y. Shen, and G. V. Karanikolas, "Topology identification and learning over graphs: Accounting for nonlinearities and dynamics," *Proc. IEEE*, vol. 106, no. 5, pp. 787–807, May 2018.

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

[8] M. Hassanzadeh and C. Y. Evrenosoğlu, "Power system state forecasting using regression analysis," in *Proc. IEEE Power & Energy Society General Meeting*, San Diego, CA, USA, July 2012, pp. 1–6.

[9] M. Hassanzadeh, C. Y. Evrenosoğlu, and L. Mili, "A short-term nodal voltage phasor forecasting method using temporal and spatial correlation," *IEEE Trans. Power Syst.*, vol. 31, no. 5, pp. 3881–3890, 2016.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. Conf. Comput. Vision and Pattern Recognit.*, Las Vegas, NV, 2016, pp. 770–778.

[11] A. M. Leite da Silva, M. B. Do Coutto Filho, and J. F. De Queiroz, "State forecasting in electric power systems," *IEE Gen. Trans. Dist.*, vol. 130, no. 5, pp. 237–244, Sept. 1983.

[12] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, "How to construct deep recurrent neural networks," in *Proc. Intl. Conf. on Learning Representations*, Banff, Canada, Apr. 2014.

[13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, Canada, Dec. 2015.

[14] G. Wang, G. B. Giannakis, and J. Chen, "Robust and scalable power system state estimation via composite optimization," *IEEE Trans. Smart Grid*, 2019, DOI: 10.1109/TSG.2019.2897100.

[15] G. Wang, G. B. Giannakis, J. Chen, and J. Sun, "Distribution system state estimation: An overview of recent developments," *Front. Inf. Technol. Electron. Eng.*, vol. 20, no. 1, pp. 4–17, Jan. 2019.

[16] G. Wang, G. B. Giannakis, Y. Saad, and J. Chen, "Phase retrieval via reweighted amplitude flow," *IEEE Trans. Signal Process.*, vol. 66, no. 11, pp. 2818–2833, Jun. 2018.

[17] L. Zhang, V. Kekatos, and G. B. Giannakis, "Scalable electric vehicle charging protocols," *IEEE Trans. Power Syst.*, vol. 32, no. 2, pp. 1451–1462, Mar. 2017.

[18] L. Zhang, G. Wang, and G. B. Giannakis, "Real-time power system state estimation and forecasting via deep neural networks," *arXiv:1811.06146*, Nov. 2018.

[19] ——, "Real-time power system state estimation via deep unrolled neural networks," in *Proc. Global Conf. on Signal and Info. Process.*, Anaheim, CA, USA, Nov. 2018.

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