Robust PSSE Using Graph Neural Networks for Data-driven and Topology-aware Priors

Qiuling Yang, Alireza Sadeghi, Gang Wang, Georgios B. Giannakis, Fellow, IEEE, and Jian Sun

Abstract—Distributed renewable generation, elastic loads, and purposeful manipulation of meter readings challenge the monitoring and control of today’s power systems (PS). In this context, to maintain a comprehensive view of the system in real time, fast and robust state estimation (SE) methods are urgently needed. Conventional PSSE solvers typically entail minimizing a nonlinear and nonconvex least-squares by e.g., the workhorse Gauss-Newton method. Those iterative solvers however, are sensitive to initialization and may get stuck in local minima. To overcome these hurdles and inspired by recent image denoising techniques, this paper advocates a learnable regularization term for PSSE that uses a deep neural network (DNN) prior. For the resultant regularized PSSE problem, a “Gauss-Newton-like” alternating minimization solver is first developed. To accommodate real-time monitoring, a novel end-to-end DNN is constructed by unrolling the proposed alternating minimization solver. Interestingly, the power network topology can be easily incorporated into the DNN by designing a graph neural network (GNN) based prior. To further endow the physics-based DNN with robustness against bad data, an adversarial DNN training method is discussed. Numerical tests using real load data on the IEEE 118-bus benchmark system showcase the improved estimation and robustness performance of the proposed scheme compared with several state-of-the-art alternatives.

Index terms—State estimation, deep prior, graph neural network, robust optimization.

I. INTRODUCTION

In today’s smart grid, reliability and accuracy of state estimation are central for several system control and optimization tasks, including optimal power flow, unit commitment, economic dispatch, and contingency analysis [2]. However, frequent and sizable state variable fluctuations caused by fast variations of renewable generation, increasing deployment of electric vehicles, and human-in-the-loop demand response, are challenging these functions.

As state variables are difficult to measure directly, the supervisory control and data acquisition (SCADA) system offers abundant measurements, including voltage magnitudes, power flows, and power injections. Given SCADA measurements, the goal of PSSE is to retrieve the state variables, namely complex voltages at all buses [2]. PSSE is typically formulated as a (weighted) least-absolute-value (WLAV) or a least-squares (WLS) problem, both of which are ill-posed, and nonconvex in general [31].

To address these challenges, several efforts have been devoted. For example, the LAV estimation problem was converted into a constrained optimization, for which a sequential linear programming solver was devised in [13], and improved (stochastic) proximal-linear solvers were developed in [29]. On the other hand, focusing on the WLS problem, the Gauss-Newton solver is widely employed in practice [2]. Unfortunately, due to the nonconvexity and quartic loss function, there are two issues in implementing the Gauss-Newton solver: i) sensitivity to initialization; and ii) no convergence guarantee in general [38]. Semidefinite programming approaches can mitigate these issues to some extent, but they incur a heavy computational burden [38]. In a nutshell, the grand challenge of these methods, remains to develop fast and robust PSSE solvers attaining or approximating the global optimum.

To bypass the nonconvex optimization hurdle, recent works have focused on developing data- (and model-) driven neural network (NN) solutions [5], [20], [37], [36], [55], [8], [24], [9]. Such NN-based PSSE solvers aim at directly approximating the mapping from measurements to state variables based on a training set of measurement-state pairs generated using simulators or available from historical data [56]. Unfortunately, these NN architectures are not effective in exploring the power network topology. On the other hand, a common approach to tackling challenging ill-posed problems in image processing has been to regularize the loss function with appropriate priors [25]. Popular priors include sparsity, total variation, and low-rank penalty [10]. Recent efforts have also focused on data-driven priors that can be learned from exemplary data [18], [26], [3].

Building on [25], [5], this paper advocates a deep (D) NN-based trainable prior for standard ill-posed PSSE, to promote physically meaningful PSSE solutions. To tackle the resulting regularized PSSE problem, an alternating minimization-based solver is first developed, which entails Gauss-Newton iterations as a critical algorithmic component. As with Gauss-Newton iterations, our solver requires inverting a matrix per iteration, thus incurring a heavy computational burden that may discourage it from real-time monitoring of large networks.

To accommodate real-time operations and building on our previous works [37], [36], we unroll this alternating minimization solver to construct a new DNN architecture, called unrolled Gauss-Newton with deep priors (GN-DP). As the name suggests, our DNN model consists of a Gauss-Newton iteration as a basic building block, followed by a proximal step
to account for the regularization term. Interestingly, by means of employing a graph (G) NN-based prior, our model exploits the structure of the underlying power network. Different from \cite{36}, our GN-DP method offers a systematic and flexible framework to incorporate topology-aware prior information into standard PSSE tasks.

In practice, measurements collected by the SCADA system may be grossly corrupted due to e.g., parameter uncertainty, instrument mis-calibration, and unmonitored topology changes \cite{21, 29}. As a cyber-physical system, power networks are also vulnerable to adversarial attacks \cite{11, 32}, as asserted by the first hacker-caused Ukraine power blackout in 2015 \cite{7}. Furthermore, it has recently been demonstrated that adversarial attacks can severely deteriorate NNs’ performance \cite{16, 10}. Prompted by this, to endow our GN-DP scheme with robustness against bad data and even adversaries, we pursue a principled GN-DP training method through a distributionally robust optimization perspective. Numerical tests using the IEEE 118-bus benchmark system corroborate the estimation performance and robustness of the proposed scheme.

Paper outline. Regarding the remainder of the paper, Section \textbf{I} introduces the power system model and formally states the PSSE problem. Section \textbf{III} presents a general framework for incorporating data-driven and topology-aware priors into PSSE, and an alternating minimization solver for the resultant regularized PSSE, followed by the unrolled GN-GNN. Section \textbf{IV} develops a robust GN-GNN version. Numerical tests using the IEEE 118-bus test feeder are provided in Section \textbf{V} with concluding remarks drawn in Section \textbf{VI}.

Notation. Lower- (upper-) case boldface letters denote column vectors (matrices), with the exception of vectors \(V, P\) and \(Q\), and normal letters represent scalars. The \((i,j)\)-th entry, \(i\)-th row, and \(j\)-th column of matrix \(X\) are \([X]_{i,j}, [X]_{i},\) and \([X]_{j}\), respectively. Calligraphic letters are reserved for sets except operators \(\mathcal{E}\) and \(\mathcal{P}\). Symbol \(^\top\) stands for transposition; \(\mathbf{0}\) denotes all-zero vectors of suitable dimensions; and \(\|x\|\) is the \(l_{2}\)-norm of vector \(x\).

II. BACKGROUND AND PROBLEM FORMULATION

Consider an electric grid comprising \(N\) buses (nodes) with \(E\) lines (edges) that can be modeled as a graph \(G := (\mathcal{N}, \mathcal{E}, W)\), where the set \(\mathcal{N} := \{1, \ldots, N\}\) collects all buses, \(\mathcal{E} := \{(n, n')\} \subseteq \mathcal{N} \times \mathcal{N}\) all lines, and \(W \in \mathbb{R}^{N \times N}\) is a weight matrix with its \((n, n')\)-th entry \(W_{nn'} = w_{nn'}\) modeling the impedance between buses \(n\) and \(n'\). In particular, if \((n, n') \in \mathcal{E},\) then \([W]_{nn'} = w_{nn'}\); and \([W]_{nn'} = 0\) otherwise. For each bus \(n \in \mathcal{N},\) let \(V_n := v_n^r + jv_n^i\) be its complex voltage with magnitude denoted by \(|V_n|\), and \(P_n + jQ_n\) its complex power injection. For reference, collect the voltage magnitudes, active and reactive power injections across all buses into the \(N\)-dimensional column vectors \([V], P,\) and \(Q\), respectively.

System state variables \(v := [v_1^r \ v_1^i \ \ldots \ v_N^r \ v_N^i]^\top \in \mathbb{R}^{2N}\) can be represented by SCADA measurements, including voltage magnitudes, active and reactive power injections, as well as active and reactive power flows. Let \(\mathcal{S}_V, \mathcal{S}_P, \mathcal{S}_Q, \mathcal{E}_P,\) and \(\mathcal{E}_Q\) denote the sets of buses or lines where meters of corresponding type are installed. For a compact representation, let us collect the measurements from all meters into \(z := [(V_n)^2]_{n \in \mathcal{S}_V}, \{P_n\}_{n \in \mathcal{S}_P}, \{Q_n\}_{n \in \mathcal{S}_Q}, \{P_{nn'}\}_{(n, n') \in \mathcal{E}_P}, \{Q_{nn'}\}_{(n, n') \in \mathcal{E}_Q}\) \(\in \mathbb{R}^M\). Moreover, the \(\ell\)-th entry of \(z_m := \{z_m\}_{m = 1}^M\), can be described by the following model

\[
    z_m = h_m(v) + \epsilon_m, \quad \forall m = 1, \ldots, M
\]

where \(h_m(v) = v^\top H_m v\) for some symmetric measurement matrix \(H_m \in \mathbb{R}^{2N \times 2N}\), and \(\epsilon_m\) captures the modeling error as well as the measurement noise.

The goal of PSSE is to recover the state vector \(v\) from measurements \(z\). Specifically, adopting the least-squares criterion and vectorizing the terms in \(z\), PSSE can be formulated as the following nonlinear least-squares (NLS)

\[
    v^* := \arg \min_{v \in \mathbb{R}^{2N}} \|z - h(v)\|^2.
\]

A number of algorithms have been advocated for \cite{15}, including e.g., Gauss-Newton iterations \cite{2}, feasible-point pursuit-based, and semidefinite programming-based solvers in \cite{30} and \cite{38}, respectively. Starting from an initial \(v_0\), most of these schemes (the former two) iteratively implement a mapping from \(v_i\) to \(v_{i+1}\), in order to generate a sequence of iterates that hopefully converges to \(v^*\) or some point nearby. In the ensuing subsection, we will focus on the ‘workhorse’ Gauss-Newton PSSE solver.

A. Gauss-Newton Iterations

The Gauss-Newton method is the most commonly used one for minimizing NLS \cite{13} Sec. 1.5.1. It relies on Taylor’s expansion to linearize the function \(h(v)\). Specifically, at a given point \(v_i\), it linearly approximates

\[
    \tilde{h}(v, v_i) \approx h(v_i) + J_i(v - v_i)
\]

where \(J_i := \nabla h(v_i)\) is the \(M \times 2N\) Jacobian of \(h\) evaluated at \(v_i\), with \([J_i]_{m,n} := \partial h_m/\partial v_n\). Subsequently, the Gauss-Newton method approximates the nonlinear term \(h(v)\) in \(2\) via \(3\), and finds the next iterate as its minimizer; that is,

\[
    v_{i+1} = \arg \min_{v} \|z - h(v_i) - J_i(v - v_i)\|^2.
\]

Clearly, the per-iteration subproblem \(4\) is convex quadratic. If matrix \(J_i^\top J_i\) is invertible, the iterate \(v_i\) can be updated in closed-form as follows

\[
    v_{i+1} = v_i + (J_i^\top J_i)^{-1} J_i^\top (z - h(v_i))
\]

until some stopping criterion is satisfied. In practice however, due to the matrix inversion, the Gauss-Newton method becomes computationally expensive; it also suffers sensitivity to initialization, and it can diverge in certain cases. These limitations prevent its use for real-time monitoring of large-scale networks. To address these challenges, instead of solving every PSSE instance (corresponding to having a new set of measurements in \(z\)) with repeated iterations, an end-to-end approach based on DNNs is pursued next.
III. UNROLLED GAUSS-NEWTON WITH DEEP PRIORS

To mitigate the ill-posedness of PSSE, this section puts forth a flexible topology-aware prior that can be incorporated as a regularizer into standard PSSE tasks such as (2). To solve the resulting regularized PSSE, an alternating minimization-based solver is developed. Later, an end-to-end DNN architecture is constructed by unrolling the alternating minimization solver. Our obtained DNN comprises as building blocks several layers of unrolled Gauss-Newton iterations followed by proximal steps to account for the regularization term. Interestingly, upon utilizing a GNN-based prior, the power network topology can be exploited in PSSE.

A. Regularized PSSE with Deep Priors

In practice, recovering $v$ from $z$ is ill-posed, for instance when $J_i$ is a rectangular matrix. Building on the data-driven deep priors in imaging denoising [26], [3], we advocate regularizing any PSSE loss (here, the NLS in (2)) with some trainable prior information, as follows

$$\min_{v \in \mathbb{R}^{2N}} \left\| z - h(v) \right\|^2 + \lambda \left\| P(v) \right\|^2$$  \hspace{1cm} (6)

where $\lambda \geq 0$ is a tuning hyper-parameter, and regularizer the second term denotes the prior. Similarly to [18], [26], [3], define

$$D(v) := (I - P)(v) = v - P(v)$$  \hspace{1cm} (7)

where $I$ is the identity operator. Here, the intuition is that $D(\cdot)$ finds the residual error upon relying on the prior $P$. Hence, $P(v) = v - D(v)$ is the residual error in $v$ estimate.

Given this interpretation, we propose the following regularized PSSE problem

$$\min_{v \in \mathbb{R}^{2N}} \left\| z - h(v) \right\|^2 + \lambda \left\| v - D(v) \right\|^2.$$  \hspace{1cm} (8)

In general, when measurements $z$ are contaminated by noise or modeling inaccuracies, the second term in (8) will be high. Hence, the regularizer $\left\| P(v) \right\|^2$ encourages solutions that are minimally affected by noise. To encompass a large family of priors, we advocate a DNN-based prior $P_\theta(v) = v - D_\theta(v)$, where $\theta$ collects all the weights of a specified DNN which are to be learned from past data. The DNN $D_\theta(\cdot)$ can be viewed as a predictor, which, upon taking a Bayesian perspective, generates the posterior for a given input.

Although this regularizer promotes desirable properties into PSSE solutions, problem (8) remains nonconvex. In addition, the nested structure of the NN-based regularizer $D_\theta(\cdot)$ further challenges this problem. Similar to the Gauss-Newton method for NLS in (2), we first develop an alternating minimization algorithm to iteratively approximate the solution of (6). Starting with some initial guess $v_0$, per iteration $i$ uses a linearized data consistency term to compute the next iterate $v_{i+1}$: i.e.

$$v_{i+1} = \arg \min_{v} \left\| z - h(v_i) - J_i(v_i - v) \right\|^2 + \lambda \left\| v - D_\theta(v_i) \right\|^2$$

$$= A_i z + B_i u_i + b_i$$

where the coefficients are given by

$$A_i := (J_i^T J_i + \lambda I)^{-1} J_i^T$$

Specifically, with initialization $v_0 = 0$ and input $z$, the first iteration yields $v_1 = A_0 z + B_0 u_0 + b_0$. Upon passing $v_1$ through the DNN $D_\theta(\cdot)$, the output $u_1$ at the first iteration, which is also the input at the second iteration, is found by (10a). In principle, state estimates can be obtained by repeating these alternating minimizing iterations whenever a new measurement $z$ becomes available. However, at every iteration $i$, the Jacobian matrix $J_i$ must be evaluated, followed by matrix inversions to form $A_i$, $B_i$, and $b_i$. The associated computational burden could be prohibitive for real-time monitoring tasks of large-scale power systems.

For fast implementation, we pursue an end-to-end learning approach that trains a DNN constructed by unrolling iterations of this alternative minimizing solver to approximate directly the mapping from measurements $z$ to states $v$; see Fig. 1 for an illustration of the resulting GN-DP DNN architecture. Recalling that to derive the alternating minimization solver, the DNN prior $D_\theta(\cdot)$ in (10a) was assumed pre-trained, with weights $\theta$ fixed in advance. In our unrolled GN-DP however, we consider without loss of generality all the coefficients $\{A_i\}_{i=0}^I$, $\{B_i\}_{i=0}^I$, $\{b_i\}_{i=0}^I$ as well as the DNN weights $\{\theta_i\}_{i=0}^I$ to be learnable from data.

This end-to-end GN-DP can be trained using backpropagation using historical or simulated measurement-state training pairs $\{(z^t, v^t)\}_{t=1}^T$. Entailing only several matrix-vector multiplications, our GN-DP achieves competitive PSSE performance compared with other iterative solvers such as the Gauss-Newton method. Furthermore, relative to the existing data-driven NN approaches, our GN-DP enjoys avoiding gradients vanishing and exploding, through having direct connections from the input layer to intermediate and output layers (a.k.a skip-connections).

Interestingly, by carefully choosing the specific model for $D_\theta(\cdot)$, desirable properties such as scalability and high estimation accuracy can be achieved. For instance, if we use feed forward NNs as $D_\theta(\cdot)$, we can get a scalable solution for large power networks. However, without exploiting the topology of the power grid, feed forward NN is usually physics-agnostic. This prompts us to focus on GNNs, which will capture the topology and physics of the power network. The resultant unrolled Gauss-Newton with GNN priors (GNN-GN) is discussed in details next.

B. Graph Neural Network Deep Prior

To predict state vector $v^*$, we model $D_\theta(\cdot)$ through GNNs, which are known for their rich expressivity. GNNs have recently demonstrated remarkable performance in several tasks, including classification, recommendation, and robotics [15].
By operating directly over graphs, GNNs can explicitly leverage the power network topology. Hence, they are attractive options for parameterization in cases where there exists a graph structure [15].

Suppose there is a graph of $N$ nodes associated with a weight matrix $W$ capturing the connectivity. The GNN takes a signal $X \in \mathbb{R}^{N \times F}$ as input, whose $n$-th row $x_n := [X]_n$; represents a feature vector of size $F$ for node $n$. For the PSSE problem at hand, features are real and imaginary parts of the nodal voltage, i.e. $F = 2$. Upon pre-multiplying the input $X$ by $W$, features are propagated over the network, yielding a diffused version $Y \in \mathbb{R}^{N \times F}$ of the original signal as follows

$$ Y = WX. $$

(11)

**Remark 1.** To model feature propagation, a common option is to rely on the adjacency matrix or any other matrix that preserves the structure of the power network (i.e. $W_{nn'} = 0$ if $(n, n') \notin \mathcal{E}$). Examples include the Laplacian matrix, the random walk Laplacian, and their normalized versions.

Basically, the shift operation in (11) linearly combines the $f$-th features of all neighbors to obtain its propagated feature. In particular, for bus $n$, the shifted feature $[Y]_{nf}$ is found by

$$ [Y]_{nf} = \sum_{i \in \mathcal{N}_n} [W]_{ni} [X]_{if} = \sum_{i \in \mathcal{N}_n} W_{ni} x_i^f $$

(12)

where $\mathcal{N}_n = \{ i \in \mathcal{N} : (i, n) \in \mathcal{E} \}$ denotes the set of neighboring buses for bus $n$. Clearly, this interpretation generates a diffused copy or shift of the signal over the graph.

The convolution operation in GNNs exploits topology information to linearly combine features, namely

$$ [Y]_{nd} := \mathcal{H} \ast X; W_{nd} := \sum_{k=0}^{K-1} [W^k X]_{ni} [H_k]_d $$

(13)

where $\mathcal{H} := [H_0 \cdots H_{K-1}]$ with $H_k \in \mathbb{R}^{F \times D}$ concatenates all filter coefficients; $Y \in \mathbb{R}^{N \times D}$ is the intermediate (hidden) graph signal with $D$ features per bus; and, $W^k X$ linearly combines features of buses within the $k$-hop neighborhood by recursively applying the shift operator $W$.

To obtain a GNN with $L$ hidden layers, by abuse of notation, let us now denote by $X_{l-1}$ the output of the $l$-th layer, which is also the input of the $l$-th layer for $l = 1, \ldots, L$, and $X_0 = X$ is the input signal. The hidden signal $Y_l \in \mathbb{R}^{N \times D_l}$ with $D_l$ features is obtained by applying the graph convolution operation [13] at layer $l$, that is

$$ [Y_l]_{nd} := \sum_{k=0}^{K_l-1} [W^k X_{l-1}]_{ni} [H_{lk}]_d $$

(14)

where $H_{lk} \in \mathbb{R}^{F_{l-1} \times F_l}$ are the graph convolution coefficients for $k = 0, \ldots, K_l - 1$. The output $X_l$ at layer $l$ is computed by applying a graph convolution followed by a point-wise nonlinear operation $\sigma_\ell()$, such as the rectified linear unit $\sigma_\ell(t) := \max\{0, t\}$ for $t \in \mathbb{R}$; see Fig. 2 for a depiction. Rewriting (14) in a compact form, we arrive at

$$ X_l := \sigma_\ell(Y_l) = \sigma_\ell \left( \sum_{k=0}^{K_l-1} W^k X_{l-1} H_{lk} \right). $$

(15)

The GNN-based PSSE provides a nonlinear functional mapping $X_L = \Phi(X_0; \Theta, W)$ that maps the GNN input $X_0$ to voltage estimates by taking into account the graph structure through $W$, through

$$ \Phi(X_0; \Theta, W) := \sigma_L \left( \sum_{k=0}^{K_L-1} W^k \left( \cdots \left( \sigma_1 \left( \sum_{k=0}^{K_1-1} W^k X_0 H_{1k} \right) \cdots \right) H_{Lk} \right) \right) $$

(16)

where the parameter set $\Theta$ contains all the filter weights; i.e, $\Theta := \{H_{lk}, \forall l, k\}$, and also recall that $X_0 = X$.

**Remark 2.** With $L$ hidden layers, $F_l$ features and $K_l$ filters per layer, the total number of parameters to be learned is $|\Theta| = \sum_{l=1}^L K_l \times F_l \times F_{l-1}$. 

![Fig. 1: The structure of the proposed GN-DP.](image-url)
To accommodate the GNN implementation over the proposed unrolled architecture, at the \( t \)-th iteration, we reshape the states \( v_t \in \mathbb{R}^{2N} \) to an \( N \times 2 \) matrix, which is the input signal \( X_0^t \in \mathbb{R}^{N \times 2} \) to the GNN. Afterwards, we vectorize the GNN output \( X_T \in \mathbb{R}^{N \times 2} \) to get vector \( u_T \in \mathbb{R}^{2N} \) (c.f. \( 10a \)). For convenience, concatenate all trainable parameters of the GNN in vector \( \omega := \{ \{ \Theta \}_j \}_{j=0}^t, \{ A_i \}_{i=0}^t, \{ B_i \}_{i=0}^t, \{ b_i \}_{i=0}^t \} \), and denote the end-to-end GNN parametric model by \( \pi(z; \omega) \), which for given measurements \( z \) predicts the voltages across all buses, i.e., \( \hat{v} = \pi(z; \omega) \). The GNN-GNN weights \( \omega \) can be updated using backpropagation, upon specifying some loss \( \ell(v^*, v_{t+1}) \) measuring how well the estimated voltages \( v_{t+1} \) by the GN-GNN matches the ground-truth voltages \( v^* \). The proposed method is summarized in Algorithm 1.

IV. ROBUST PSSE SOLVER

In real-time inference, our proposed GN-GNN that has been trained using past data outputs an estimate of the system state vector \( v^t \) per time slot \( t \) based on the observed measurements \( z^t \). However, due to impulsive communication noise and/or cyberattacks, our proposed GN-GNN in Section III may yield grossly biased estimation results. To obtain estimators robust to bad data, classical formulations including H"uber estimation, Huber M-estimation, and Schweppe-H"uber generalized M-estimation, have considered the \( \ell_1 \)-contaminated models for measurements; see e.g., [22], [31]. In this paper, we assume instead that measurements-ground truth voltages are drawn from some nominal yet unknown distribution \( P_0 \), that is \( (z, v^*) \sim P_0 \). Tying the PSSE problem with distributional assumption, we train the parameters \( \omega \) upon solving

\[
\min_{\omega} \mathbb{E}_{P_0}[\ell(\pi(z; \omega), v^*)] \tag{23}
\]

Note that, in practice only i.i.d. data samples \( \{(z^t, v^t+1)\}_{t=1}^T \sim \hat{P}_0^{(T)} \) (a.k.a training data) are given and \( P_0 \) is unknown. Therefore, a meaningful PSSE solver entails minimizing the following empirical loss

\[
\min_{\omega} \frac{1}{T} \sum_{t=1}^T \ell(\pi(z^t; \omega), v^t) \tag{17}
\]

To cope with uncertainties and even adversaries, robustness can be infused into the solution of (17), by considering optimization over a set \( \mathcal{P} \) of probability distributions centering at \( \hat{P}_0^{(T)} \), and minimizing the worst-case expected loss with respect to the choice of any distribution satisfying \( P \in \mathcal{P} \). Concretely, this can be formulated as the following distributionally robust optimization

\[
\min \sup_{P \in \mathcal{P}} \mathbb{E}_P[\ell(\pi(z; \omega), v^*)]. \tag{18}
\]

Compared with (17), the worst-case formulation in (18) ensures a reasonable performance across a continuum of distributions in \( \mathcal{P} \). A broad range of ambiguity sets \( \mathcal{P} \) could be considered here. Featuring a strong duality enabled by the optimal transport theory [28], such distributionally robust optimization approaches have become a popular solution to make machine learning models robust [4]. Indeed, this tractability is the key impetus for this section.

To formalize, consider probability density functions \( P \) and \( Q \) defined over the support \( Z \), and let \( \Pi(P, Q) \) be the set of all joint probability distributions defined over \( Z^2 \), with marginals \( P \) and \( Q \). Also let \( c : Z \times Z \rightarrow [0, \infty) \) be some cost function representing the cost of transporting a unit of mass from \( z \) in \( P \) to another element \( z' \) in \( Q \). The so-called optimal transport between two distributions \( P \) and \( Q \) is given by [28] Page 111

\[
W_c(P, Q) := \inf_{\pi \in \Pi} \mathbb{E}_{\pi}[c(z, z')]. \tag{19}
\]

Intuitively, \( W_c(P, Q) \) denotes the minimum cost associated with transporting all the mass from distribution \( P \) to \( Q \). Under mild conditions over the cost function and distributions, \( W_c \) gives the well-known Wasserstein distance between \( P \) and \( Q \); see e.g., [27].

Having introduced the distance \( W_c \), let us define the uncertainty set for the given empirical distribution \( \hat{P}_0^{(T)} \), as \( \mathcal{P} := \{ P \mid W_c(P, \hat{P}_0^{(T)}) \leq \rho \} \) that includes all probability distributions having at most \( \rho \)-distance from \( \hat{P}_0^{(T)} \). Incorporating this ambiguity set \( \mathcal{P} \) into problem (18) yields the following optimization for distributionally robust GN-GNN estimation

\[
\min_{\rho} \sup_{P \in \mathcal{P}} \mathbb{E}_P[\ell(\pi(z; \omega), v^*)] \tag{20a}
\]

s.t. \( W_c(P, \hat{P}_0^{(T)}) \leq \rho \). \tag{20b}

Observe that the inner functional optimization in (20a) runs over all probability distributions \( P \) characterized by (20b).

Evidently, optimizing directly over the infinite-dimensional distribution functions is intractable. Fortunately, for continuous loss as well as transportation cost functions, the inner maximization satisfies strong duality condition; that is, the optimal objective value of the inner maximization is equal to its dual optimal objective value. In addition, the dual problem involves optimization over only a one-dimension variable, that can be carried out efficiently. These two observations prompt us to solve (20) in the dual domain. To formally obtain this tractable surrogate, we call for a result from [6].

Proposition 1. Let \( \ell : \omega \times \mathcal{Z} \rightarrow [0, \infty) \), and \( c : \mathcal{Z} \times \mathcal{Z} \rightarrow [0, \infty) \) satisfy be continuous functions. Then, for any given \( \hat{P}_0^{(T)} \), and \( \rho > 0 \), it holds that

\[
\sup_{P \in \mathcal{P}} \mathbb{E}_P[\ell(\pi(z; \omega), v^*)] = \inf_{\gamma \geq 0} \left\{ \frac{1}{\gamma} \mathbb{E}_{\gamma \pi(\zeta; \omega), \zeta \sim \hat{P}_0^{(T)}} \left[ \sup_{\zeta \in \mathcal{Z}} \ell(\pi(\zeta; \omega), v^*) + \gamma(\rho - c(z, \zeta)) \right] \right\} \tag{21}
\]

where \( \mathcal{P} := \{ P \mid W_c(P, \hat{P}_0^{(T)}) \leq \rho \} \).

Remark 3. Thanks to the strong duality, the right-hand side in (21) simply is a univariate dual reformulation of the primal problem given on the left-hand side. In contrast with the
primal formulation, the expectation in the dual domain is taken only over the empirical distribution \( \hat{P}_0^{(\pi)} \) rather than over any \( P \in \mathcal{P} \). Furthermore, since this reformulation circumvents the need for finding the optimal coupling \( \pi \in \Pi \) to define \( \mathcal{P} \), and characterizing the primal objective for all \( P \in \mathcal{P} \), it is practically appealing and convenient.

Capitalizing on Proposition 1, we can replace the inner maximization with its dual reformulation to arrive at the following distributionally robust PSSE optimization

\[
\min_{\omega} \inf_{\gamma \geq 0} \mathbb{E}_{\[z, v^*\] \sim \hat{P}_0^{(\pi)}} \left[ \sup_{\zeta \in \mathcal{Z}} \ell(\pi(\zeta; \omega), v^*) + \gamma(\rho - c(z, \zeta)) \right].
\]  

**Remark 4.** Although the robust surrogate in (22) looks similar to minimax (saddle-point) optimization problems, it requires the supremum to be solved separately per observed measurements \( z \), that cannot readily be handled by existing minimax optimization solvers.

Finding the optimal solution \((\omega^*, \gamma^*)\) of (22) is in general challenging. A common approach to bypassing this hurdle is to approximate the optimal \( \omega^* \) by solving (22) with a preselected and fixed \( \gamma > 0 \) (24). Indeed, it has been shown in (27) that for any strongly convex transportation cost function, such as \( c(z, z') := \|z - z'\|_p^2 \) for any \( p \geq 1 \), a sufficiently large \( \gamma > 0 \) ensures that the inner maximization is strongly convex, hence efficiently solvable. Note that having a fixed \( \gamma \) is tantamount to tuning \( \rho \), which in turn controls the level of infused robustness. Fixing some large enough \( \gamma > 0 \) in (22), our robustified GN-GNN model can thus be obtained by solving

\[
\min_{\omega} \mathbb{E}_{\[z, v^*\] \sim \hat{P}_0^{(\pi)}} \left[ \sup_{\zeta \in \mathcal{Z}} \psi(\omega, \zeta, z, v^*) \right]
\]  

where

\[
\psi(\omega, \zeta, z, v^*) := \ell(\pi(\zeta; \omega), v^*) + \gamma(\rho - c(z, \zeta)).
\]  

Intuitively, (23) can be understood as first ‘adversarially’ perturbing the measurements \( z \) into \( \zeta^* \) by maximizing \( \psi(\cdot) \), and subsequently seeking a model that minimizes the empirical loss with respect to even such perturbed inputs. In this manner, robustness of the sought model is achieved to future data that may be contaminated by adversaries.

Initialized with some \( \omega^0 \), and given a datum \((z^t, v^*)\), we form \( \psi(\cdot) \) (c.f. (24)), and implement a single gradient ascent step for the inner maximization as follows

\[
\zeta^t = z^t + \eta^t \nabla \psi(\omega^t, \zeta; z^t, v^*) |_{\zeta = z^t},
\]  

where \( \eta^t > 0 \) is the stepsize. Upon evaluating (25), the perturbed data \( \zeta^t \) will be taken as input (replacing the ‘healthy’ data \( z^t \)) fed into Algorithm 1. Having now the loss \( \ell(\pi(\zeta^t; \omega^t), v^*) \) as solely a function of the GN-GNN weights \( \omega^t \), the current iterate \( \omega^t \) can be updated again by means of backpropogation.

V. NUMERICAL TESTS

This section the estimation performance as well as robustness of our proposed methods on the IEEE 118-bus benchmark system.

A. Simulation Setup

The simulations were carried out on an NVIDIA Titan X GPU with a 12 GB RAM. For numerical tests, we used real load consumption data from the 2012 Global Energy Forecasting Competition (GEFC) [1]. Using this data, training and testing data were prepared by solving the AC power flow equations using the MATPOWER toolbox [49]. To match the
scale of power demands, we normalized the load data, and fed it into MATPOWER to generate 1,000 pairs of measurements and ground-truth voltages, 80% of which were used for training while the remaining for testing. Measurements include all sending-end active power flows as well as voltage magnitudes, corrupted by additive white Gaussian noise. Standard deviations of the noise added to power flows and voltage magnitudes were set to 0.02 and 0.01 [29], respectively.

A reasonable question to ask is whether explicitly incorporating the power network topology using a trainable regularizer offers improved performance over competing alternatives or not. In addition, it is of interest to study how an adversarial training method enhances the PSSE performance in the presence of bad data and even adversaries. To this aim, three baseline PSSE methods were numerically tested, including: i) the prox-linear network [36]; ii) 6-layer plain-vanilla feed-forward NN (6-layer FNN); and iii) 8-layer feed-forward NN (8-layer FNN). The weights of these NNs were trained using the ‘Adam’ optimizer to minimize the Huber loss. The learning rate was fixed to $10^{-3}$ throughout 500 epochs, and the batch size was set to 32.

B. GN-GNN for PSSE

In the first experiment, we implemented a GN-GNN network by unrolling $I = 6$ iterations of the proposed alternating minimizing solver. A GNN with $K = 2$ hops and $D = 8$ hidden units with ReLU activations per unrolled GN-CNN iteration was used for the deep prior. The GN-GNN architecture was designed so that its total number of weight parameters is roughly the same as that of the prox-linear network.

The first set of results depicted in Figs. 3 and 4 show the estimated voltage profiles obtained at buses 80 and 100 from test slots 50 to 100, respectively. The ground-truth and estimated voltages for the first 50 buses on the test slot 100 are presented in Fig. 5. These plots corroborate the competitive or improved performance of the developed GN-GNN relative to the simulated PSSE solvers.

C. Robust PSSE

Despite their remarkable performance in standard PSSE, DNNs may fail to yield meaningful estimates in practice due to bad data. Obviously, this challenges their application in safety-critical power networks. Here, we examine the robustness of our GN-GNN trained with the described adversarial learning method. To this aim, we implemented distributionally robust learning framework to manipulate the input of GN-GNN, prox-linear net, 6-layer FNN, and 8-layer FNN models. Specifically, in distributional attacks, we postulated an ambiguity set $\mathcal{P}$ comprising distributions centered at the nominal data-generating $P_0$. Although the training samples were generated according to $P_0$, testing samples were obtained using a distribution $P \in \mathcal{P}$ that yields the worst empirical loss. Fig. 6 and 7 demonstrate estimated voltage profiles under distributional attack for a fixed $\gamma = 0.13$ (c.f., [23] and [24]). The plots showcase the robustness of the proposed method against distributional uncertainties, where it outperforms the competing alternatives.

VI. CONCLUSIONS

This paper advocated topology-aware DNN-based regularizers to address the ill-posedness as well as the nonconvexity of standard PSSE. An alternating minimization solver was developed to approach the solution of the regularized PSSE, which is further unrolled to construct a DNN model. For real-time monitoring of large-scale networks, the resulting DNN was trained using historical or simulated measurement and ground-truth voltages. A basic building block of our GN-GNN consists of a Gauss-Newton iteration followed by a proximal...
step to deal with the regularization term. Numerical tests showed the competitive performance of our proposed GN-GNN relative to several existing ones. Further, an adversarial training method was presented to endow the GN-GNN with robustness against bad data. Future directions include exploring such data-driven and topology-aware regularizers for optimal power flow and unit commitment problems.

Fig. 7: The estimated voltage magnitudes and angles by the four schemes under distributional attacks for the first 50 buses on slot 50.

REFERENCES

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

[2] A. Abur and A. G. Exposito. Power System State Estimation: Theory and Implementation. New York, USA: CRC Press, 2004.

[3] H. K. Aggarwal, M. P. Mani, and M. Jacob, “ModL: Model-based deep learning architecture for inverse problems,” IEEE Trans. Med. Imag., vol. 38, no. 2, pp. 394–405, Aug. 2018.

[4] C. Bandi and D. Bertsimas, “Robust option pricing,” Eur. J. Oper. Res., vol. 239, no. 3, pp. 842–853, Dec. 2014.

[5] P. P. Barbeiro, J. Krstulovic, H. Teixeira, J. Pereira, F. J. Soares, and J. P. Iria, “State estimation in distribution smart grids using autoencoders,” in IEEE Int'l. Power Engineering and Optimization Conf., 2014, pp. 358–363.

[6] J. Blanchet and K. Murthy, “Quantifying distributional model risk via optimal transport,” Math. Oper. Res., vol. 44, no. 2, pp. 565–600, May 2019.

[7] D. U. Case, “Analysis of the cyber attack on the Ukrainian power grid,” E-ISAC, vol. 388, Mar. 2016.

[8] Y. Chen, Y. Shi, and B. Zhang, “Input convex neural networks for optimal voltage regulation,” arXiv:2002.08684, 2020.

[9] R. Dobbe, W. Van Westering, S. X. Liu, D. Arnold, D. S. Callaway, and C. Tomlin, “Linear single- and three-phase voltage forecasting and bayesian state estimation with limited sensing,” IEEE Trans. Power Syst., vol. 35, no. 3, pp. 1–10, Aug. 2020.

[10] H. W. Engl, M. Hanke, and A. Neubauer, Regularization of Inverse Problems. Berlin, HR: SSBM, 1996, vol. 375.

[11] P. Fairley, “Cybersecurity at U.S. utilities due for an upgrade: Tech to detect intrusions into industrial control systems will be mandatory,” IEEE Spectr., vol. 53, no. 5, pp. 11–13, May 2016.

[12] X. Hu, H. Hu, S. Verma, and Z.-L. Zhang, “Physics-guided deep neural networks for powerflow analysis,” arXiv:2002.00097, 2020.

[13] R. Jabr and B. Pal, “Iteratively re-weighted least absolute value method for state estimation,” IET Trans. Gener., Transmiss., Distrib., vol. 150, no. 4, pp. 385–391, Jul. 2003.

[14] M. Jin, J. Lavaei, and K. H. Johansson, “Power grid AC-based state estimation: Vulnerability analysis against cyber attacks,” IEEE Trans. Autom. Sci. Eng., vol. 15, no. 4, pp. 1784–1799, July 2018.

[15] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” arXiv:1609.02907, 2016.

[16] A. Kurakin, I. Goodfellow, and S. Bengio, “Adversarial examples in the physical world,” Int’l. Conf. Learn. Rep., Vancouver, BC, Canada, Apr.

[17] Y. Lan, H. Zhu, and X. Guan, “Fast nonconvex SPD solvers for large-scale power system state estimation,” IEEE Trans. Power Syst., 2020.

[18] J. Schlemper, J. Caballero, J. V. Hajnal, A. N. Price, and D. Rueckert, “A deep cascade of convolutional neural networks for dynamic MRI image reconstruction,” IEEE Trans. Med. Imag., vol. 37, no. 2, pp. 491–503, Oct. 2017.

[19] A. Sinha, H. Namkoong, and J. Duchi, “Certifying some distributional robustness with principled adversarial training,” Int’l. Conf. Learn. Rep., Vancouver, BC, Canada, May 2018.

[20] C. Villani, Optimal Transport: Old and New. Berlin, HR: SSBM, 2008, vol. 338.

[21] G. Wang, G. B. Giannakis, and J. Chen, “Robust and scalable power system state estimation via composite optimization,” IEEE Trans. Smart Grid, vol. 10, no. 6, pp. 6137–6147, Nov. 2019.

[22] G. Wang, A. S. Zamzam, G. B. Giannakis, and N. D. Sidiropoulos, “Power system state estimation via feasible point pursuit: Algorithms and cramer-Rao bound,” IEEE Trans. Signal Process., vol. 66, no. 6, pp. 1649–1658, Mar. 2018.

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

[24] G. Wu, J. Sun, and L. Xiong, “Optimal switching attacks and countermeasures in cyber-physical systems,” IEEE Trans. Syst., Man, Cybern.: Syst., vol. 50, no. 5, pp. 1–10, Jun. 2020.

[25] Q. Yang, G. Wang, A. Sadeghi, G. B. Giannakis, and J. Sun, “Two-timescale voltage control in distribution grids using deep reinforcement learning,” IEEE Trans. Smart Grid, pp. 1–11, 2019.

[26] A. Zamzam and K. Baker, “Learning optimal solutions for extremely fast ac optimal power flow,” arXiv preprint arXiv:1910.01213, 2019.

[27] A. S. Zamzam and N. D. Sidiropoulos, “Physics-aware neural networks for powerflow analysis,” IEEE Trans. Power Syst., vol. 60, no. 5, pp. 1784–1799, July 2018.

[28] L. Zhang, G. Wang, and G. B. Giannakis, “Real-time power system state estimation and forecasting via deep unraveled neural networks,” IEEE Trans. Signal Process., vol. 67, no. 15, pp. 4069–4077, Aug. 2019.

[29] L. Zhang, G. Wang, and G. B. Giannakis, “Real-time power system state estimation via deep unraveled neural networks,” in IEEE Global Conf. on Signal and Info. Process., 2018, pp. 907–911.

[30] H. Zhu and G. B. Giannakis, “Power system nonlinear state estimation using distributed semidefinite programming,” IEEE J. Sel. Topics Signal Process., vol. 8, no. 6, pp. 1039–1050, Jun. 2014.

[31] J. M. Zimmerman, C. H. Tseng, 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, Feb. 2011.