Deep Statistical Solver for Distribution System State Estimation

Benjamin Habib [iD], Elvin Isufi [iD], *Member, IEEE*, Ward van Breda, Arjen Jongepier [iD], and Jochen L. Cremer [iD], *Member, IEEE*

Abstract—Implementing accurate Distribution System State Estimation (DSSE) faces several challenges, among which the lack of observability and the high density of the distribution system. While data-driven alternatives based on Machine Learning models could be a choice, they suffer in DSSE because of the lack of labeled data. In fact, measurements in the distribution system are often noisy, corrupted, and unavailable. To address these issues, we propose the Deep Statistical Solver for Distribution System State Estimation (DSS²), a deep learning model based on graph neural networks (GNNs) that accounts for the network structure of the distribution system and the governing power flow equations of the problem. DSS² is based on GNN and leverages hypergraphs to model the network as a graph into the deep-learning algorithm and to represent the heterogeneous components of the distribution systems. A weakly supervised learning approach is put forth to train the DSS²: by enforcing the GNN output into the power flow equations, we force the DSS² to respect the physics of the distribution system. This strategy enables learning from noisy measurements and alleviates the need for ideal labeled data. Extensive experiments with case studies on the IEEE 14-bus, 70-bus, and 179-bus networks showed the DSS² outperforms the conventional Weighted Least Squares algorithm in accuracy, convergence, and computational time while being more robust to noisy, erroneous, and missing measurements. The DSS² achieves a competing, yet lower, performance compared with the supervised models that rely on the unrealistic assumption of having all the true labels.

Index Terms—State estimation, distribution system, deep learning, graph neural network, physic-informed neural network, weakly supervised learning.

I. INTRODUCTION

DISTRIBUTION systems are taking a more active role in the energy transition. These active distribution systems require more extensive monitoring and control, which is possible by developing Distribution System State Estimation (DSSE) [1]. Currently, state estimation (SE) is mostly only possible in the transmission systems, and several challenges exist to extending SE to distribution systems successfully. First, conventional SE algorithms for transmission systems are challenging to adopt to distribution systems as the assumptions differ. Additionally, the distribution grid lacks real-time measurements. Conventional algorithms assume full observability of the grid with redundant measurements, which is impractical [2]. To address the observability issue of distribution systems, forecasted values based on historical data called *pseudomeasurements*[1] are used to compensate for the lack of measurements, but they are often inaccurate and can impact the SE accuracy [3]. Also, the Weighted Least Squares (WLS) method used for SE is time-consuming and sensitive to data noise for large distribution systems [4]. Multi-area SE has been widely investigated to speed up the estimation process [5], [6], and has been extended to distribution systems [7]. However, the convergence and sensitivity issues remain, and division into multiple areas brings in communication and time-synchronization challenges.

Different algorithms have been proposed to improve the robustness and convergence of SE, notably the branch-current WLS, the Least Absolute Value, and the Generalized Maximum Likelihood [4], [8]. Branch-current algorithms are more robust to parameter selection and uncertainty and are more suited for the weakly meshed and radial topologies in the distribution system [9]. Although, these algorithms suffer from the lack of qualitative measurements in wide distribution systems and the uncertainty of distributed loads and generators. Kalman Filters aim to improve speed and estimation performance under low observability. They are linked to the Forecast-Aided SE concept, where model-based approaches use the previous states as extra information to enhance accuracy and speed [4], [10], [11], [12], [13]. However, Kalman Filter SE is limited by the assumptions of system linearization and the Gaussian distribution of the measurements, which reduce its accuracy and robustness. Indeed, power systems are highly nonlinear, and measurements can show a non-Gaussian distribution [14].

Data-driven techniques showed promising results in performing fast DSSE without the above-mentioned assumptions. Deep learning models showed remarkable results to fit data for the SE task [15], [16], [17], [18]. This *supervised learning* approach

Benjamin Habib is with the Department of Electrical Sustainable Energy, TU Delft, 2628 CD Delft, Netherlands (e-mail: benjaminhabib@lilo.org).

Elvin Isufi is with the Department of Intelligent Systems, TU Delft, 2628 CD Delft, Netherlands (e-mail: e.isufi-1@tudelft.nl).

Ward van Breda and Arjen Jongepier are with the Stedin Group, 3011 TA Rotterdam, Netherlands (e-mail: Ward.vanBreda@stedin.net; arjen.jongepier@stedin.net).

Jochen L. Cremer is with the Department of Electrical Sustainable Energy, TU Delft, 2628 CD Delft, Netherlands, and also with the Department of Electrical and Electronic Engineering, Imperial College London, SW7 2AZ London, U.K. (e-mail: j.l.cremer@tudelft.nl).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TPWRS.2023.3290358.

Digital Object Identifier 10.1109/TPWRS.2023.3290358

[1]It is common in DSSE to define pseudomeasurements as values estimated from historical data, as opposed to defining them as unmeasured variables calculated from the measurement of neighbor variables.

trains neural networks to fit *labels*, which are the grid's state variables. These labels are usually provided from simulations, as getting them from the grid is often impossible. As such, these approaches suffer from the scarcity of real labelled data and supervised learning is only possible using simulation data. Therefore, the models can only fit simulators exclusively and not real systems. Even though some approaches try to improve this technique by introducing some inductive bias [19] and physic-awareness [20], they all require extensive supervised learning using large labelled databases [4].

Combining model-based and data-driven approaches is a promising research direction to overcome the limitations of the model-based techniques with data-driven tools [11]. This approach is considered in [21] to combine the efficiency of Kalman Filters with the robustness of supervised Deep Learning architectures. It showed interesting results in low-dimensional problems; however, it suffers from high-dimensional problems due to the need for labelled data and unstable training. In the field of 'hybrid' approaches, [22] combines data with physics and develops a model-specific deep neural network (DNN) by unrolling a SE solver to enhance estimation performance and alleviate computation expenses. However, the model is trained with fully labeled data, and the physic-awareness of the approach is limited and does not include the structure of the system.

To address the lack of labeled data for training data-driven models, the concept of *weakly supervised learning* has been proposed [23]. This is similar to supervised learning but it is used for tasks where data is only partially or inaccurately labeled, relying on physical information, mathematical tools, or model-based modules to enhance the training process. While unsupervised learning also trains models without labels, weakly supervised learning still relies on imperfect values defined as labels. Although weakly supervised learning is highly practical for tasks such as DSSE, it has not yet been investigated for this particular problem.

Topology and parameter estimation in distribution systems from limited measurements have also been explored in the literature, with promising results. In [24], a model-based algorithm using linear regression is proposed for topology and parameter estimation from limited measurements. In [25], smart meter data is utilized to estimate the topology through an Ordinary Least Squares method. Meanwhile, [26] proposes a data-driven Lasso algorithm for topology estimation.

Graph Neural Networks (GNNs) are a particular family of deep learning models that use the underlying network structure as an inductive bias [27], [28] to tackle the curse of dimensionality and reduce the data demand. GNNs have also shown robustness to perturbations in the network topology [29], [30], [31], which makes them appealing data-driven alternatives for the DSSE task. GNNs are investigated for power system applications, where the electrical lines correspond to the graph's edges and the buses correspond to the graph's nodes [32], and the data varies for the specific application.

GNNs have been investigated for their potential use in SE in power systems, as shown in [19], [33], [34], [35]. The models in [19], [33] demonstrate that GNNs can accurately perform

SE while being robust against noise and missing data. Moreover, [34] shows that GNNs can provide fast and robust SE, and any inaccuracies in the data would only impact local estimation. Additionally, [35] highlights that GNN models can handle fast sampling measurements, thereby improving SE in power systems. However, the heterogeneity of components in power systems cannot be accurately modeled using simple graphs, and thus, GNNs have limited expressivity in the graph model for power system applications.

Despite the increasing development of GNNs applications in power systems and growing research on deep learning for DSSE, the literature on GNN for DSSE is limited to parallel works [36], [37] In [36], an electrical-model-guided GNN is used to perform DSSE and compared to conventional methods and other machine learning techniques. This approach demonstrated higher accuracy and robustness, indicating the potential of GNN-based approaches. In [37], a GNN model is combined with matrix completion techniques to perform DSSE without the need for a detailed system model, highlighting the robustness of GNN approaches to model inaccuracies. While these approaches show promising results, they rely on labeled data for training, which is impractical due to the limited observability of the system's state.

In this article, we propose the *Deep Statistical Solver for Distribution System State Estimation (DSS^2)*, a GNN model based on the Deep Statistical Solver architecture [38] specialized for optimization tasks on power systems. The model is trained in *weakly supervision* manner to tackle the issues of data scarcity and inaccurate labeling. The success of such weak supervision is conveyed by considering physical information of the network and the physical laws of the power flow equations in the training loss function, rendering labels obsolete. Specific contributions include:

1) DSS^2, the Deep Statistical Solver model for accurate data-driven DSSE using a weakly supervised approach.
2) adding physical constraints as penalization to the loss function, enhancing the model's performance.
3) the innovative use of *weakly supervision* in the context of data-driven DSSE, which leverages the power flow equations to restrict the model's search for the mapping function, hence reducing the data demand and improving robustness to inaccurate measurements.

Our goal with this proposition is to enhance the accuracy of grid estimation in situations where there are limited measurements and low-quality input values, such as pseudomeasurements. To achieve this, we propose using a deep learning model that learns solely from the grid's measurements and the physical information of the system.

We validate the model using various case studies on the IEEE 14-bus, 70-bus, and 179-bus systems and compare it to the WLS algorithm baseline and other Deep Learning architectures. The proposed DSS^2 is up to 15 times faster, 4 times more accurate, and more robust than the standard WLS algorithm. Our model also outperforms supervised learning approaches, being 10 times more accurate in line-loading estimation while alleviating the need for labelled data. Interestingly, our approach is better for larger networks as the GNN learns in the neighborhood of buses,

and the larger the power network, the more data to learn from. The source code of this work is available on GitHub [39].

This article presents the methodology in Sections II and III, introducing the Deep Statistical Solver model and extending its usage to DSS, respectively. Section IV are the case studies and compares the performances to the baseline algorithm and other data-driven models. Section V concludes this work.

II. PROPOSED APPROACH FOR STATE ESTIMATION

A. Conventional Problem Formulation

The state estimation problem aims at finding the state vector $\boldsymbol{x}$ based on a noisy measurement vector $\boldsymbol{z}$. Conventionally, we consider the voltage amplitude and angle at every grid bus as state variables, and $\boldsymbol{z}$ can include any measurement type:

$$\boldsymbol{x} = [V_0, V_1, \ldots, V_{n-1}, \varphi_0 = 0., \varphi_1, \ldots, \varphi_{n-1}] \in \mathbb{R}^{2n \times 1} \tag{1a}$$

$$\boldsymbol{z} = [z_0, z_1, \ldots, z_{m-1}] \in \mathbb{R}^{m \times 1}, \tag{1b}$$

where we consider n buses and m measurements. V_i is the voltage amplitude at bus i, and φ_i the voltage phase angle. We have $2n - 1$ state variables, as φ_0 is set to 0 by the slack bus convention. Linking the measurement vector $\boldsymbol{z}$ to the state vector $\boldsymbol{x}$, we define a measurement function $\boldsymbol{h}(\boldsymbol{x})$:

$$\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{\varepsilon}, \tag{2}$$

where $\boldsymbol{\varepsilon} \in \mathbb{R}^{m \times 1}$ is the measurement noise vector, and $\boldsymbol{h}(\boldsymbol{x})$ are the power flow equations shown in (3) [40].

$$\boldsymbol{h}(\boldsymbol{x}) = \begin{cases} V_i = V_i \\ \varphi_i = \varphi_i \\ P_{ij_\rightarrow} = -V_i V_j \left[\mathbb{R}(Y_{ij})\cos(\Delta\varphi_{ij}) + \mathbb{I}(Y_{ij})\sin(\Delta\varphi_{ij})\right] \\ \qquad + V_i^2 \left[\mathbb{R}(Y_{ij}) + \dfrac{\mathbb{R}(Y_{s_{ij}})}{2}\right] \\ P_{ij_\leftarrow} = V_i V_j \left[-\mathbb{R}(Y_{ij})\cos(\Delta\varphi_{ij}) + \mathbb{I}(Y_{ij})\sin(\Delta\varphi_{ij})\right] \\ \qquad + V_j^2 \left[\mathbb{R}(Y_{ij}) + \dfrac{\mathbb{R}(Y_{s_{ij}})}{2}\right] \\ Q_{ij_\rightarrow} = V_i V_j \left[-\mathbb{R}(Y_{ij})\sin(\Delta\varphi_{ij}) + \mathbb{I}(Y_{ij})\cos(\Delta\varphi_{ij})\right] \\ \qquad - V_i^2 \left[\mathbb{I}(Y_{ij}) + \dfrac{\mathbb{I}(Y_{s_{ij}})}{2}\right] \\ Q_{ij_\leftarrow} = V_i V_j \left[\mathbb{R}(Y_{ij})\sin(\Delta\varphi_{ij}) + \mathbb{I}(Y_{ij})\cos(\Delta\varphi_{ij})\right] \\ \qquad - V_j^2 \left[\mathbb{I}(Y_{ij}) + \dfrac{\mathbb{I}(Y_{s_{ij}})}{2}\right] \\ I_{ij_\rightarrow} = \left|\dfrac{P_{ij_\rightarrow} - jQ_{ij_\rightarrow}}{\sqrt{3}V_i e^{-j\varphi_i}}\right| = \dfrac{|P_{ij_\rightarrow} - jQ_{ij_\rightarrow}|}{\sqrt{3}V_i} \\ I_{ij_\leftarrow} = \left|\dfrac{P_{ij_\leftarrow} - jQ_{ij_\leftarrow}}{\sqrt{3}V_j e^{-j\varphi_j}}\right| = \dfrac{|P_{ij_\leftarrow} - jQ_{ij_\leftarrow}|}{\sqrt{3}V_j} \\ P_i = -\sum_{j \in \mathcal{N}(i)} P_{ij_\leftarrow} + P_{ij_\rightarrow} \\ Q_i = -\sum_{j \in \mathcal{N}(i)} Q_{ij_\leftarrow} + Q_{ij_\rightarrow} \end{cases} \tag{3}$$

In this measurement function, (3), $\Delta\varphi_{ij} = \varphi_i - \varphi_j + \phi_{ij}$ is the voltage angle difference across the line that connects bus i to
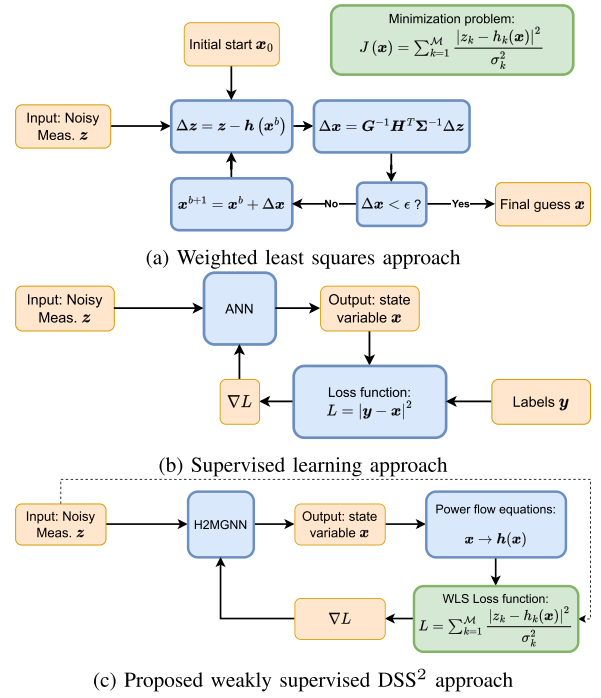

(a) Weighted least squares approach

(b) Supervised learning approach

(c) Proposed weakly supervised DSS² approach

Fig. 1. State estimation with (a) WLS with Newton-Raphson solver uses an initial guess $\boldsymbol{x}_0$ of the state vector $\boldsymbol{x}$ that iteratively updates $\boldsymbol{x}^b$ until the tolerance $\Delta\boldsymbol{x} > \epsilon$ or a maximum of iterations is reached, (b) supervised learning uses a label vector $\boldsymbol{y}$ to train an ANN to fit the output $\boldsymbol{x}$ to the input $\boldsymbol{z}$ and (c) the proposed weakly supervised approach considers the power flow equations $\boldsymbol{h}(\boldsymbol{x})$ ((3)) to get the *estimated* measurements and fits the output $\boldsymbol{x}$ of the GNN model to the input $\boldsymbol{z}$ without labels. The target optimization is similar to WLS.

bus j, ϕ_{ij} is the shift angle of the transformer if any, and Y_{ij} and $Y_{s_{ij}}$ are respectively the line and shunt admittance of the line between bus i and bus j. Measuring flows at bus i, we have $P_{ij_\rightarrow}$ and $Q_{ij_\rightarrow}$ as the active and reactive power flow from bus i to bus j, and $P_{ij_\leftarrow}$ and $Q_{ij_\leftarrow}$ as the power flow from bus j to bus i. Current flow I_{ij} follows the same convention. Finally, we derive the active and reactive power injections at bus i, P_i, and Q_i from the power flows. All these outputs are possible elements of $\boldsymbol{z}$, depending on the measurement infrastructure.

The measurement function $\boldsymbol{h}(\boldsymbol{x})$ contains equations that connect the state variables V_i and φ_i to all types of measurements in the network. The first two lines correspond to identity functions that link the state variables to their direct measurements. Lines 3-6 consist of AC power flow equations used to derive power flows from the state variables. Lines 7-8 derive current flows from the previously derived power flows. Finally, lines 9-10 derive the nodes' power injection by ensuring exact power balance in each network node.

The measurement function $\boldsymbol{h}(\boldsymbol{x})$ is nonlinear, and (2) includes the probabilistic noise vector $\boldsymbol{\varepsilon}$. In SE, we are interested in finding the inverse relation $\boldsymbol{h}^{-1}(\boldsymbol{z})$ to estimate the state vector $\boldsymbol{x}$ while compensating the error $\boldsymbol{\varepsilon}$. The conventional SE approach, shown in Fig. 1(a), uses the iterative Newton-Raphson algorithm to minimize a WLS objective function [40]. This technique uses the redundancy of measurements to provide an accurate estima-tion. However, the approach requires at least the same number of

measurements as state variables, meaning $m \geq 2n - 1$, and the system needs to be fully observable. Moreover, matching this requirement but failing to provide enough redundancy highly impacts the estimation accuracy. In practice, $m \approx 4n$ achieves satisfying results, which is impractical for the distribution system [41]. The iterative process may even diverge in case of poor observability or high noise level in the measurements [4].

Another approach to approximate $\boldsymbol{h}^{-1}(\boldsymbol{z})$ is to train an Artificial Neural Network (ANN) to map this function. ANNs approximate functions using a series of nonlinear operations parameterized by their *trainable weights* $\boldsymbol{\theta}$ [42]. These weights are estimated during the training phase to approximate the given relation. For the SE task, the model is trained to approximate the inverse relation $\boldsymbol{h}^{-1}(\boldsymbol{z})$, considering the measurement vector $\boldsymbol{z}$ as the input of the ANN and the state vector $\boldsymbol{x}$ as its output. We use the state estimation's convention of $\boldsymbol{x}$ as output and $\boldsymbol{z}$ as input of the model.:

$$f_{\boldsymbol{\theta}}(\boldsymbol{z}) := \boldsymbol{h}^{-1}(\boldsymbol{z}) \to \boldsymbol{x} \tag{4}$$

In a common supervised learning approach, the approach assigns a label vector $\boldsymbol{y}$ as the *true value* of $\boldsymbol{x}$ for each measurement sample (one measurement vector $\boldsymbol{z}$). In the training process, shown in Fig. 1(b), the model fits the data using available labels as reference. This approach, although quite efficient, is impractical for DSSE due to the lack of labelled data. Instead, our contribution combines Deep Learning and WLS optimization to propose a weakly supervised learning approach, alleviating the need for labels.

B. Weakly Supervised Learning

To develop a weakly supervised learning approach for the DSSE task, we incorporate the power flow equations from (3) within the training phase, as shown in Fig. 1(c). The H2MGNN algorithm takes the measurement vector $\boldsymbol{z}$ as an input, and provide an estimated state vector $\boldsymbol{x}$ as an output, which is then used to retrieve the estimated values $\boldsymbol{h}(\boldsymbol{x})$ of the network using the power flow equations ((3)). The loss function of the model is then set to be the minimization objective of the WLS approach:

$$L(\boldsymbol{z}, \boldsymbol{x}) = \sum_{k \in \mathcal{M}} \frac{|z_k - h_k(\boldsymbol{x})|^2}{\sigma_k^2} \tag{5}$$

with σ_k the standard deviation of the measurement k's uncertainty assumed as Gaussian distribution, and $\mathcal{M}$ the measurements set. While SE aims at considering the actual noise φ of the measurement vector, a 'first guess' of this value is assumed. σ_k. $|\mathcal{M}| = m$ is the number of measurements where $|\cdot|$ is the cardinality of a set. We assume uncorrelated measurements.

The detailed training procedure shown in Fig. 1(c) can be divided into four steps:

1) An estimated state vector **x** is given by the H2MGNN algorithm from an input measurement vector **z**.
2) The power flow equations integrated into the measurement function **h(x)** are used to retrieve network's values from the estimated state **x**.
3) The *estimated* values **h(x)** are compared to the actual measurement vector **z** in the Weighted Least Squared

loss function. An estimation error is retrieved from each input measurement and weighted by the inverse of the measurement's variance.
4) The sum of all estimation errors consists in the loss function, and we apply gradient descent to find the partial derivatives and tune the H2MGNN accordingly.

In this training process, the model is trained by minimizing the error between the measurements $\boldsymbol{z}$ and the estimated values $\boldsymbol{h}(\boldsymbol{x})$, and the uncertainty σ^2 of the measurements is used as weights to emphasize learning from the most accurate measurements. With this loss function, we implement a weakly supervised learning approach where we use the input measurements $\boldsymbol{z}$ as *noisy, imperfect labels* that the H2MGNN needs to fit through the power flow equations, and no ground-truth labels are used for training. Function (3) is differentiable w.r.t the output state variables, and the gradient can be expressed using the measurement Jacobian matrix $\boldsymbol{H}(\boldsymbol{x}) = \nabla \boldsymbol{h}(\boldsymbol{x})$.

With this method, the target optimization of the training phase is exactly the conventional WLS minimization problem, allowing the model to learn an input-output mapping that represents this function. Our goal is to achieve a similar level of performance as the WLS approach while improving the model's numerical stability, computation time, robustness, and observability requirements.

C. Physical Penalization Terms in the Loss Function

We propose aiding the WLS learning loss (5) with different penalization terms to reduce the number of local minima and 'guide' the outputs towards physically-feasible solutions. We assume our model does not need to estimate unstable states as protection schemes are faster and more reliable, so we guide the learning process to only estimate stable states in the output. Considering stable networks, we add three terms to the loss:

- Voltage level stability criteria: power systems ensure a voltage level between $V^{LB} = 95\%$ and $V^{UB} = 105\%$ per unit to remain stable. Therefore, a two-sided penalization $[V - V^{UB}]_+ + [V^{LB} - V]_+$ is added to the loss function to enforce this criterion.[2]
- Phase angle stability criteria: large variations in phase angles are improbable in stable systems. For example, a phase angle difference of more than $\Delta\varphi^{UB} = 0.25$ rads between two neighbouring buses would characterize an unstable network. Therefore, we add a second two-sided penalization $[\Delta\varphi - \Delta\varphi^{UB}]_+ + [-\Delta\varphi^{UB} - \Delta\varphi]_+$ to the loss function to constrain this phase angle difference to $\Delta\varphi^{UB} = 0.25$.
- Line loading stability criteria: power systems regulators ensure the network's security by applying safety margins to line loading. To keep the model output within a physical range, we apply a third penalization $[l - l^{UB}]_+$ on the line loading when the prediction gives a loading higher than $l^{UB} = 100\%$.

---

[2]$[x]_+ = \max(0, x)$

(a) 4-bus network (b) Standard graph model

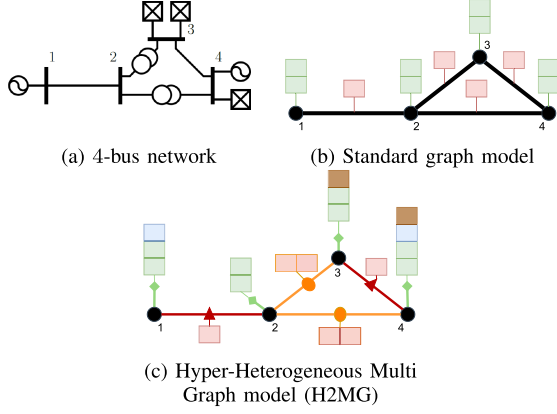(c) Hyper-Heterogeneous Multi Graph model (H2MG)

Fig. 2. Modelling the network (a) with two generators, three loads, two lines, and two transformers as (b) a standard graph and (c) an H2MG. The standard graph (b) has vertices (●) and edges (▬) with their features represented as boxes. The H2MG (c) models the components, bus (◆▬), line (▲▬), and transformer (●▬) as hyperedges connected to any number of connection ports with their features.

Adding these terms to the loss function, the equation used in the training process becomes:

$$L(\boldsymbol{z}, \boldsymbol{x}) = \sum_{k \in \mathcal{M}} \frac{|z_k - h_k(\boldsymbol{x})|^2}{\sigma_k^2} + \lambda_0 [\lambda_1 [V - V^{UB}]_+$$

$$+ \lambda_1 [V^{LB} - V]_+ + \lambda_2 [\Delta\varphi - \Delta\varphi^{UB}]_+$$

$$+ \lambda_2 [-\Delta\varphi^{UB} - \Delta\varphi]_+ + \lambda_3 [l - l^{UB}]_+], \quad (6)$$

where $\lambda_0, \lambda_1, \lambda_2, \lambda_3$ are hyperparameters set to balance the effect of each mathematical term during training. These terms penalize the model output towards physically plausible boundaries and avoid diverging toward local minima that are well beyond the physical margins of the system.

III. THE DEEP STATISTICAL SOLVER MODEL

This section proposes the H2MG structure, the modelling of the heterogeneous components of the distribution grid, the Hyper-Heterogeneous Multi Graph Neural Network (H2MGNN) and how to learn the H2MGNN in weak supervision for DSS by applying Section II.

A. Hyper-Heterogeneous Multi Graph (H2MG)

The H2MG uses hypergraphs to model power grids. Power grids are complex networks where different heterogeneous components are connected as shown in Fig. 2(a). Modelling power networks solely with vertices and edges, as done with standard graph models, Fig. 2(b), leads to information losses when merging grid components together into graph objects. More versatile modelling of such networks is possible using hypergraphs, Fig. 2(c), where each component can be modelled as a specific hyperedge which can mitigate the loss of information.

The H2MG formalism is defined by:
- Objects as hyperedges: every object in the network is modelled as a hyperedge that can connect to any number

of vertices. This is shown in Fig. 2 where each component is modelled separately as a hyperedge: we represent lines and transformers as hyperedges connected to two vertices, whereas buses are modelled as hyperedges connected to one vertex.
- Vertices as ports: vertices represent the interfaces between objects. In a hypergraph, vertices are *connection points* between the components (the hyperedges). These connection points between components in a power system are the network's buses. Therefore, we model buses as both hyperedges as network components and vertices as network interfaces.
- Hyper-Heterogeneous Multi Graph: the collection of hyperedges connected through vertices forms a hypergraph, and we call this hypergraph *heterogeneous* if it contains multiple classes of objects.

Hyperedges carry features and outputs, while vertices, as connection ports, do not carry input-output information.

B. H2MG Neural Network (H2MGNN)

The H2MGNN is a GNN architecture that works with H2MG models. It uses a recursive process to learn information from the hypergraph and related features. It is a recurrent and residual GNN architecture, with trainable mappings implemented as standard ANNs and trained through standard back-propagation. As presented in Algorithm 1, we consider four types of variables:
- Vertex latent variables, considering a vertex set $\mathcal{V}$ corresponding to the *interface* role of buses: $\boldsymbol{h}_i^v, \ \forall i \in \mathcal{V}$;
- hyperedge latent variables, considering $c \in \mathcal{C}$ as the objects' class, and $e \in \mathcal{E}^c$ as the objects' hyperedge: $\boldsymbol{h}_e^c$;
- hyperedge inputs $\boldsymbol{z}_e^c$;
- hyperedge outputs $\hat{\boldsymbol{x}}_e^c$.

In our setting, the hyperedge index e refers to the object's connections: considering a vertex i and its neighbouring vertex j, $e = i$ for a bus, and $e = ij$ for a line.

In the initialization of Algorithm 1, the hyperparameter d sets the dimension of the latent variables. We initialize these latent variables with a flat start (zero values) and set predicted output variables to the initial values $\hat{\boldsymbol{x}}_{e,0}^c$ dependent on the task. For DSSE, a common initialization is $V_i = 1$ p.u. and $\varphi_i = 0$ rad. Then, the H2MGNN algorithm recursively updates these variables in the system with trainable mappings ϕ_θ. An iteration variable t is defined to weigh each iteration in the update process and T is the maximum number of iterations. At each iteration, latent variables are updated by an increment defined through the message-passing step similar to conventional GNN algorithms:

$$\Delta\boldsymbol{h}_i^v = \sum_{(c,e,o) \in \mathcal{N}(i)} \phi_\theta^{c,o} \left(\frac{t}{T}, \boldsymbol{h}_i^v, \boldsymbol{h}_e^c, \hat{\boldsymbol{x}}_e^c, \boldsymbol{z}_e^c \right), \quad \forall i \in \mathcal{V} \quad (7a)$$

$$\Delta\boldsymbol{h}_e^c = \phi_\theta^{c,h} \left(\frac{t}{T}, \boldsymbol{h}_i^v, \boldsymbol{h}_e^c, \hat{\boldsymbol{x}}_e^c, \boldsymbol{z}_e^c \right) \quad (7b)$$

$$\Delta\hat{\boldsymbol{x}}_e^c = \phi_\theta^{c,y} \left(\frac{t}{T}, \boldsymbol{h}_i^v, \boldsymbol{h}_e^c, \hat{\boldsymbol{x}}_e^c, \boldsymbol{z}_e^c \right) \quad (7c)$$

Algorithm 1: H2MGNN Algorithm.

1: **procedure** $f_\theta z_e^c$
   ▷ Initialization
2:   **for** $i \in \mathcal{V}$ **do**
3:     $h_i^v \leftarrow \mathbf{0}^d$
4:   **for** $c \in \mathcal{C}$ **do**
5:     **for** $e \in \mathcal{E}^c$ **do**
6:       $h_e^c \leftarrow \mathbf{0}^d$
7:       $\hat{x}_e^c \leftarrow \hat{x}_{e,0}^c$
   ▷ Latent interaction
8:   $t \leftarrow 0$
9:   **while** $t < T$ **do**
10:    **for** $i \in \mathcal{V}$ **do**
11:      $h_i^v \leftarrow h_i^v + \frac{1}{T} \times \sum \phi_\theta^{c,o}(\frac{t}{T}, h_i^v, h_e^c, \hat{x}_e^c, z_e^c)$
12:    **for** $c \in \mathcal{C}$ **do**
13:      **for** $e \in \mathcal{E}^c$ **do**
14:        $h_e^c \leftarrow h_e^c + \frac{1}{T} \times \phi_\theta^{c,h}(\frac{t}{T}, h_i^v, h_e^c, \hat{x}_e^c, z_e^c)$
15:        $\hat{x}_e^c \leftarrow \hat{x}_e^c + \frac{1}{T} \times \phi_\theta^{c,y}(\frac{t}{T}, h_i^v, h_e^c, \hat{x}_e^c, z_e^c)$
16:    $t \leftarrow t + 1$
17:   **return** $\hat{x}_e^c$

with $\mathcal{N}(i)$ the set of hyperedges connected to vertex i, and o the connection port of a hyperedge (if connected to multiple ports). The final output of the model is stored in the hyperedge outputs $\hat{x}_e^c$.

C. Proposed DSS² Implementation

As presented in Fig. 1(c), we use the H2MGNN model to estimate the state variables x from the measurements z through Algorithm 1 and train it through the weakly supervised approach with the WLS as target optimization. For the DSSE described in Section II, the DSS² model approximates the inverse of the measurement function (3) as (4). To simplify the model, we consider balanced systems and only model the positive sequence of the networks. We model buses, lines and transformers and integrate generators and loads as nodes' power injection, as commonly done for the DSSE task.

The input features follow the WLS algorithm where, for each measurement, we consider the two, the measured value and its uncertainty. Voltage angles are considered as possible inputs of the model to allow the use of synchronized phasors, but are not required as most distribution systems do not carry such measurements. We also add all other parameters as features needed to compute the measurement function (3) as *topology parameters*. The features and parameters assigned to each class of components are listed in Table I. The model's output is every bus's voltage amplitude and angle, as typical in SE. Finally, we add booleans to detail components: $\mathbb{1}_z$ defines buses with zero-injection (no consumption or generation), $\mathbb{1}_s$ defines slack buses, and $\mathbb{1}_{cl}$ defines closed lines. These booleans simplify the model and provide more information about the network to the DSS² model. In other words, this simplification considers 'virtual measurements' to enforce zero power flow at buses without injection ($\mathbb{1}_z$), no power flow at the connected buses to an open line ($1 - \mathbb{1}_{cl}$) and $V_s = 1$ p.u. and $\phi_s = 0$ rad at the

TABLE I
FEATURES AND TOPOLOGY PARAMETERS OF THE H2MGNN (MODELLED AS AN H2MG)

	Buses	Lines
Topology param.	Bus port i Zero-inj. bool. $\mathbb{1}_z$ Slack bool. $\mathbb{1}_s$	Line ports (i, j) Closed line bool. $\mathbb{1}_{cl}$ Phase-shift ϕ_{ij}
Input features	Voltage magn.: V_i - σ_{V_i} Voltage angle: φ_i - σ_{φ_i} Active power inj.: P_i - σ_{P_i} Reactive power inj.: Q_i - σ_{Q_i}	Active PF: P_{ij} - $\sigma_{P_{ij}}$ Reactive PF: Q_{ij} - $\sigma_{Q_{ij}}$ Current magn.: I_{ij} - $\sigma_{I_{ij}}$ Line admittance: Y_{ij} Shunt admittance: $Y_{s_{ij}}$
Output features	Voltage magn.: V_i Voltage angle: φ_i	

slack bus s where s is the index of the vector $\mathbb{1}_s$ that equals 1. Since distribution grids typically have a limited number of measurements, we assume a low amount of measurements and incorporate pseudomeasurements to complete the observability of the system. These pseudomeasurements are based on historical demand data and are added as active and reactive power injections P_i and Q_i for buses where observability is lacking.

IV. CASE STUDIES

Case studies have been undertaken to provide insights into the proposed approach and evidence of its efficacy. After stating the case studies settings and showing the efficiency of the proposed weakly-supervised learning approach, we analyse the performance of the DSS² exploring the trade-off of providing labels and accuracy, subsequently, investigating the accuracy, convergence and computational speed for larger networks. Finally, we investigate the performance of the proposed approaches for different measurement noise, when the measurements are disturbed, and when we have higher and lower load levels and renewable powers.

A. Test Systems and Setup

We considered the 14-bus CIGRE MV distribution grid with PV and Wind distributed energy resources (DER) activated [43], the 70-bus Oberrhein MV/LV sub-grid, and the whole 179-bus Oberrhein grid from [41]. The networks are presented in Fig. 3. The measurement locations for each network are shown in Fig. 3. These measurements $\mathcal{M}$ either measure the power flow over lines or the voltages at buses and were assumed with different Gaussian noise, as further discussed.

For each network, 8640 load samples were collected, equivalent to one year of hourly data. Each load scenario considers load levels of 24 consecutive samples discretized hourly for all loads in the network. These load scenarios resulted from a Monte Carlo sampling on standard load profiles taken from [44], considering a 15% uncertainty. For each sample, in each scenario, assuming balanced systems, the AC power flow computed the full *true* state using PandaPower 2.9 [41] and Python 3.8. Subsequently, one sample's full *true* state considered all loads and generators'
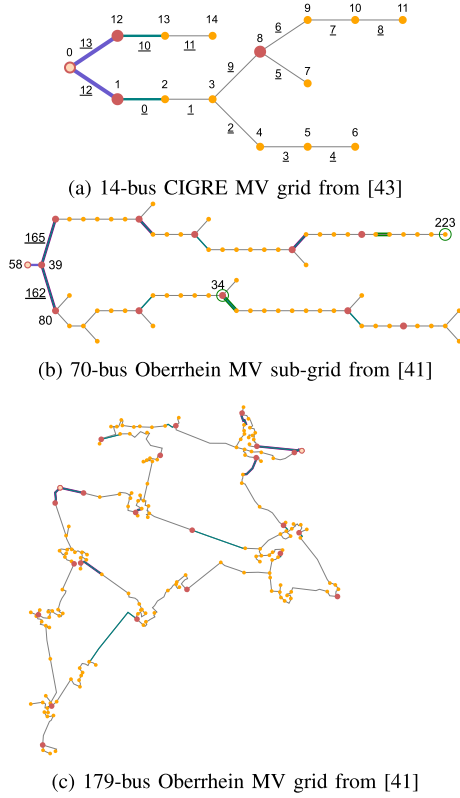
(a) 14-bus CIGRE MV grid from [43]

(b) 70-bus Oberrhein MV sub-grid from [41]

(c) 179-bus Oberrhein MV grid from [41]

Fig. 3. Three test networks consisting out of trafos (▬), lines (▬), MV/LV buses (●) and HV buses (●). The state can be estimated using the set of power flow measurements (▬) and voltage measurements (●). Relevant indices are indicated, and lines' indices are underlined. Case studies on the 70-bus grid focus on buses indicated with ○.

TABLE II
HYPERPARAMETER VALUES OF DSS^2 TRAINED FOR THREE POWER NETWORKS

Parameter	14-bus	70-bus	179-bus
Epochs	630	540	1020
λ	0.8	0.8	0.8
α	0.006	0.006	0.006
batch size	64	64	64
d	40	40	40
layers	3	3	3
T	7	20	25
r	0.4	0.4	0.4
ℓ_2	0.002	0.002	0.002
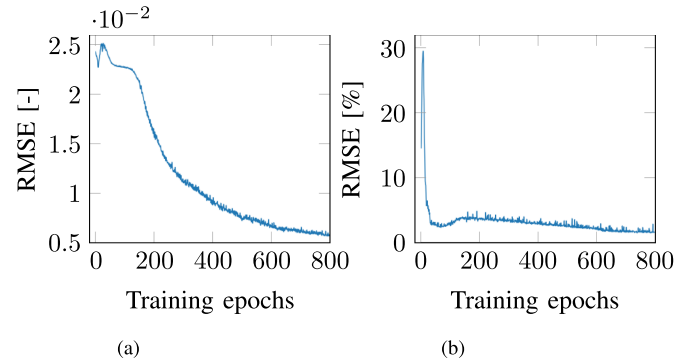
Fig. 4. Validation RMSEs of voltages (a) and line loading (b) during training.

active and reactive power levels, the bus voltage levels, phases, and line loadings. System operators do not have access to this full *true* state; however, some key variables are provided by the measurements specified earlier. These observed variables were assumed corrupted with zero-mean Gaussian white noise at the measurement locations. Between 0.5% and 2% standard deviations were assumed for the voltage and current measurement noises, and between 1% and 5% for the active and reactive power measurement errors. Pseudomeasurements of power levels were considered at every (unobserved) bus using generic load and generation profiles taken from [44].

The dataset was split into train, validation, and test sets, following an 80/10/10 split. In supervised learning, the measurement vector z at the measurement locations mentioned above represents the input to the model, and the full state represents the label y.

Several baseline models were assumed as follows. The standard SE WLS algorithm [45], a standard ANN model trained with supervised learning, and the DSS^2 model but trained with supervised learning (referenced with sup. DSS^2). The WLS algorithm from PandaPower 2.9 was used, and the deep learning models were implemented in Tensorflow 2.8 [46]. The ANN was designed with 5 layers of 32 hidden values, $tanh$ activation functions and a Glorot normal initializer. The code to reproduce the case studies of this article can be accessed in GitHub [39].

B. Efficiency of the Weakly-Supervised Learning

This section investigates the efficiency of the weakly supervised learning DSS^2 approach and hyperparameters that can impact the state estimation accuracy. The hyperparameters penalization factor $\lambda = \lambda_0 = \lambda_1 = \lambda_2$, batch size, dropout rate r, ℓ_2-regularizer, and the number of iteration T were fixed. A grid search tuned the hyperparameters learning rate within the ranges $\alpha \in \{0.001, 0.002, \ldots, 0.009, 0.01\}$, layer dimension $d \in \{8, 16, 24, 32, 40, 48\}$, and layers number $\in \{2, 3, 4, 5\}$. The selected hyperparameter values are in Table II for each network.

The efficiency of the learning approach is shown in Fig. 4(a) and (b) when training on the 14-bus network. The voltage and line loading estimation RMSE slowly decreased at each epoch, showing a learning curve through the power flow equations and using only the noisy measurements and pseudomeasurements. When training in weakly supervision, DSS^2 learned to minimize the different objectives using *noisy* measurements as 'reference values'. However, the computational time to train DSS^2 in weakly supervision was lower than to train in supervision.

C. Trade-Off Between Accuracy and Available Labels

This case study investigates the performance of the proposed weakly supervised DSS^2 model on the 14-bus system compared

TABLE III
MEAN (STANDARD DEVIATION IN PARENTHESES) VALUES OF PERFORMANCE METRICS IN *DEFAULT* CONDITIONS

Performance metric \ Model	14-bus CIGRE				70-bus Oberrhein			179-bus Oberrhein	
	WLS	ANN	sup. DSS²	DSS²	WLS	WLS*	DSS²	WLS**	DSS²
Voltage RMSE [10^{-3}]	9.9 (0.45)	2.7 (0.17)	**2.5 (0.11)**	3.4 (0.18)	31 (0.96)	5.9 (0.18)	**1.5 (0.01)**	9.9 (0.3)	**2.3 (0.01)**
Line loading RMSE [%]	**3.4 (0.05)**	42 (0.83)	12.7 (0.17)	3.8 (0.05)	17 (0.9)	15 (0.8)	**2.3 (0.01)**	5.9 (0.4)	**3.4 (0.01)**
Line & trafos loading RMSE [%]	**4.6 (0.06)**	39 (0.5)	14 (0.15)	8 (0.06)	39 (2.7)	24 (1.67)	**2.5 (0.01)**	4.2 (0.2)	**3.5 (0.01)**
Convergence [%]	100	100	100	100	25	100	100	53	**100**
Computational time [ms]	86 (41)	**3.5 (0.73)**	4.7 (1.4)	5.5 (4.5)	123 (31)	161 (40)	**26 (10.4)**	1212 (424)	**58 (26.7)**

WLS * is the WLS algorithm without flow measurements, and WLS ** with increased tolerance. The bold font shows the best model or algorithm for each metric.
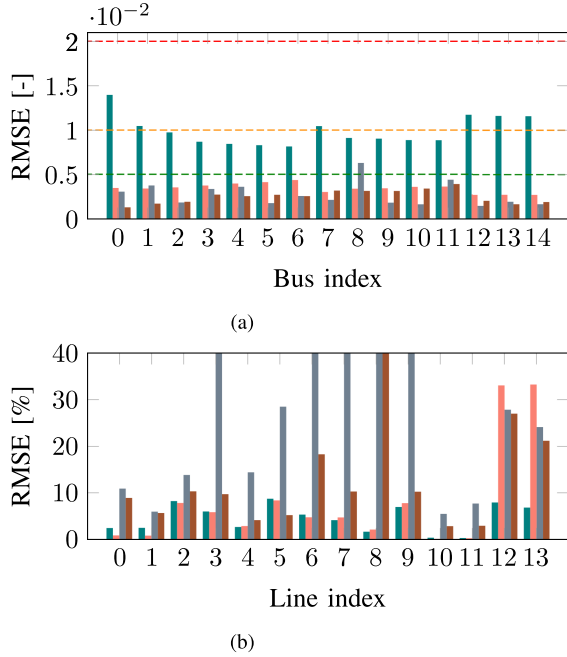

Fig. 5. Comparing RMSE of estimating (a) voltage levels and (b) line loadings in the 14-bus network between WLS (■), DSS² (■), FFNN (■) and sup. DSS² (■). In (a), the dashed lines show acceptable state estimators based on the standard deviation. Below the dashed green is acceptable, and above red is unacceptable. In (b), indexes 12 and 13 are transformers.

to three baselines. The second column in Table III summarizes the results.[3]

The RMSE for voltages of the proposed weakly supervised DSS² was three times lower than the WLS, 2.5% versus 9.9%. In more detail, Fig. 5(a) shows the voltage estimation RMSE per bus. The RMSE was lower than the 0.5% threshold for all buses, showing successful learning from voltage measurement data while handling measurements' noise. The difference in RMSE between the observed (buses 1,8, and 12) and the unobserved buses are small, showing the capability of our DSS² model to extrapolate to all buses. The supervised models (ANN, sup. DSS²) estimated the voltage more accurately, as expected, as they learned from the ideal *true* voltage data having an unfair, impractical advantage.

The RMSE of line loading of the weakly supervised DSS² reaches performances equivalent to the WLS, outperforming the

[3]Observability in the large networks is reached artificially by using pseudomeasurements in unobservable parts of the networks.

supervised models by a wide margin according to Table III. This observation offered insights. Supervised models poorly estimated *indirect* values such as the line loading that were calculated using the power flow equations. The models only outputted the state variables and supervised models poorly considered the *coupling* of the state variables in the estimations of line loading. However, the weakly supervised model learned directly through the power flow equations about the coupling with the effect of estimating line loading more accurately. In more detail, Fig. 5(b) shows the loading estimation error per line. The weakly supervised DSS² model had a very high accuracy on measured lines (lines 0 and 10) and their extension (lines 1 and 11). However, there was a clear drop in performance for the estimation of transformers' loading, shown at indexes 12 and 13. The simple modelling of transformers or slack may have led to this reduced accuracy as the transformers and lines were considered in the same class of models. As a result of this simple modelling, the H2MGNN considered the same mapping for these components, which may have reduced the accuracy of transformer estimations.

D. Convergence, Accuracy and Computation Speed in Large Networks

This case study investigates the performance of the proposed DSS² compared to the WLS in larger networks, the 70-bus and 179-bus networks, along three performance criteria: the convergence rate, the accuracy, and the computational time. The 2nd and 3rd columns in Table III summarize the results.

When analysing the convergence rate, the DSS² always converges, and the WLS never converges in the 179-bus network. The WLS was unstable in this large and noisy network, leading to these poor convergence rates. WLS' convergence issues with noisy measurements in large systems is already well-known [4], [10]. Many noisy measurements constrain the Newton-Raphson solver and can lead to divergence. More specifically, the WLS had issues in handling flow measurements. In response to these issues and to compare the accuracy and computational times of DSS² with WLS, only voltage measurements and pseudomeasurements were used in WLS to increase the convergence rate (WLS* in the table). This increased the convergence rate in the 70-bus system but did not increase the convergence of the WLS in the 179-bus. Therefore, in the 179-bus system, the tolerance of the Newton-Raphson iterative process and the number of iterations were increased (WLS** in the table). Increasing these
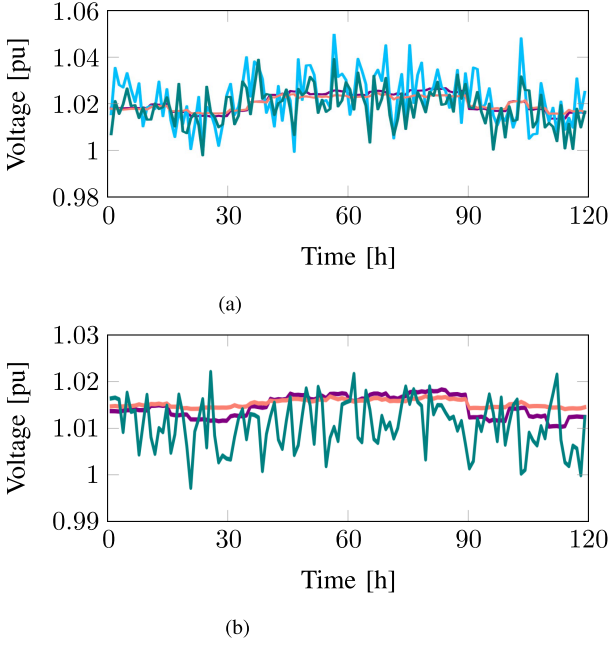
(a)

(b)

Fig. 6. Estimation of the voltage level at (a) bus 34 and (b) bus 223 of the 70-bus network under normal conditions and across the sampling period, using WLS (■), and DSS^2 (■). True voltage (■) and measurement (■) as references.
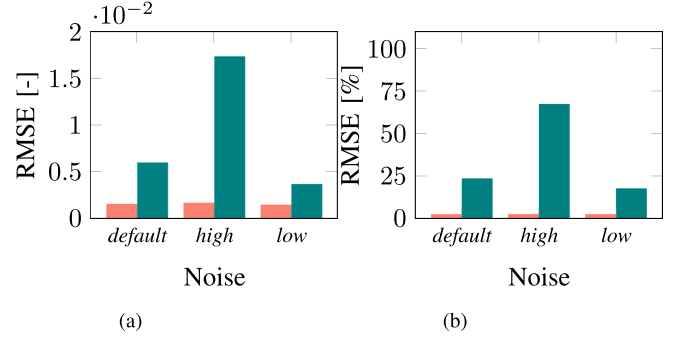

(a)   (b)

Fig. 7. Comparison of RMSE for the estimation of (a) voltage level and (b) line loading between the WLS (■) and DSS^2 (■) in the 70-bus network; considering different levels of measurement noise.
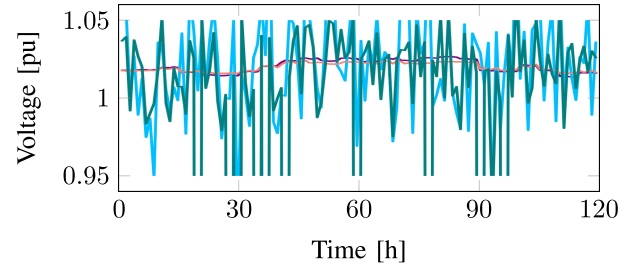

Fig. 8. Estimation of the voltage level at (a) bus 34 of the 70-bus network under *high* noise level and across the sampling period, using WLS (■), and DSS^2 (■). True voltage (■) and measurement (■) as reference.

parameters increased the convergence rate at the cost of lower accuracy and slower processing.

When analysing the accuracy, a key advantage of the DSS^2 becomes visible. DSS^2 outperformed the WLS in every metric in the two larger networks. The models based on GNN, such as DSS^2, learn from local operations (in the neighbourhood of buses) and extrapolate to other locations (to other neighbourhoods of buses). Therefore, the more buses and lines in the network, the more local operations to learn from that can further enhance the model's accuracy. Also, these networks have more static loads and less DER than the 14-bus network, so the variation of voltage and line loading was smaller, and the estimated values from the DSS^2 become more accurate. Fig. 6(a) and (b) compare the estimated voltage levels through a sampling period in the 70-bus system for the measured bus 34 and the unmeasured remote bus 223, respectively. The accuracy of the DSS^2 model estimating the voltage in measured nodes through noisy measurements was high. However, the model lacked generalizability when estimating voltage in remote, unmeasured nodes.

When analysing the computation times, in the last row of Table III, the DSS^2 increasingly outperformed WLS for larger networks. The computational time of the WLS and the DSS^2 increased from the 70-bus network to the 179-bus network by factors of 10 and 2, respectively. The DSS^2 scaled to a larger network 5-fold better than the WLS algorithm. The WLS needed more iterations for this larger system until the Newton-Raphson converged, although the tolerance was increased, which typically decreased the computational times. The DSS^2 also showed a lower variance in the computational times as it is not based on an iterative algorithm.

E. Measurement Noise

This case study compares the robustness to measurement noise of the DSS^2 to the WLS in the 70-bus network. The level of measurement noise refers to the standard deviation σ_i of the Gaussian noise added to the measurements. Three different levels of noise were considered. The *default* level had 1% noise on ideal measurements of voltage and current, and 2% noise on the ideal measurements of active and reactive power; the *low* level had 0.5% and 1% noise, and the *high* level 3% and 5%, respectively.

At *high* noise, Fig. 7 shows the RMSE of the DSS^2 was more than 10 times better than that of the WLS showing significantly higher robustness of DSS^2. DSS^2 had a similarly high accuracy at *low* and *high* noise as in the *default* noise level. DSS^2 learned to process many noisy signals with different standard deviations within the high noise level ranges and GNN structures. The dropout step during training improved the capability of the DSS^2 model to handle stochasticity, including noise. Fig. 8 compares the voltage level estimation at *high* measurement noise for the bus 34. The DSS^2 successfully cancelled the increased noise, whereas the WLS algorithm struggled to stay accurate.

F. Missing and Erroneous Measurements

This case study investigates the impact of missing and erroneous measurements on the DSS^2 and the WLS algorithm at the 70-bus network. Case (i) assumed a missing voltage measurement on bus 39 that was naively replaced with their
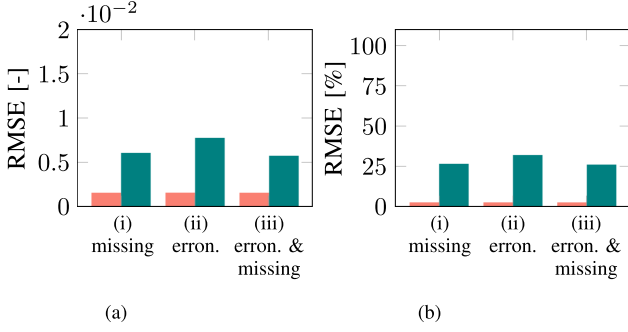
Fig. 9.   Comparison of RMSE for the estimation of (a) voltage level and (b) line loading between WLS (■) and DSS^2 (■) in the 70-bus network considering (i) missing voltage measurement, (ii) erroneous voltage measurement and erroneous active power flow measurements, and (iii) missing voltage measurement and erroneous voltage measurements.
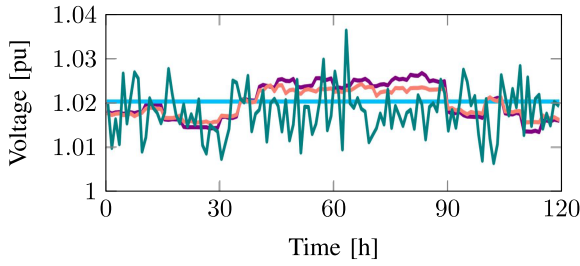

Fig. 10.   Estimation of voltage level at bus 34 of the 70-bus network under missing measurement conditions and across the sampling period, using WLS (■), and DSS^2 (■). True voltage (■) and measurement (■) as reference.

historical mean value. Case (ii) assumed erroneous voltage measurements on buses 39, 58 and 80, and erroneous active power flow measurements in lines 162 and 165 with a higher deviation from the *true* state values than the expected (standard) deviation. Case (iii) assumed missing voltage measurements on buses 34, 39 and 80 and erroneous voltage measurements on bus 58.

Fig. 9 shows the results. The DSS^2 had high robustness to missing and erroneous measurements in all three cases, with a similar RMSE as the *default* case (no missing or erroneous measurements). However, the erroneous measurement case (ii) impacted the WLS, showing an increase of around 20% on relative voltage RMSE. Fig. 10 focuses on one bus with erroneous measurements, the bus 34 in case (iii). The measurement in bus 34 was missing for the whole sequence and was naively replaced by the empirical mean value (light blue).

A key insight of this analysis is that the DSS^2 was not impacted by this missed value and successfully provided an accurate estimation. Interestingly, the DSS^2 model was not trained to handle such events. However, using known patterns from neighbouring information, the DSS^2 remained accurate. Indeed, the GNN architecture increased the interpolation capabilities by incorporating the data symmetries w.r.t. the underlying graph.

G. Changes in Power Levels of Load and Renewables

This case study investigates the generalization capability of the DSS^2 (and the WLS) to changes in levels of power in
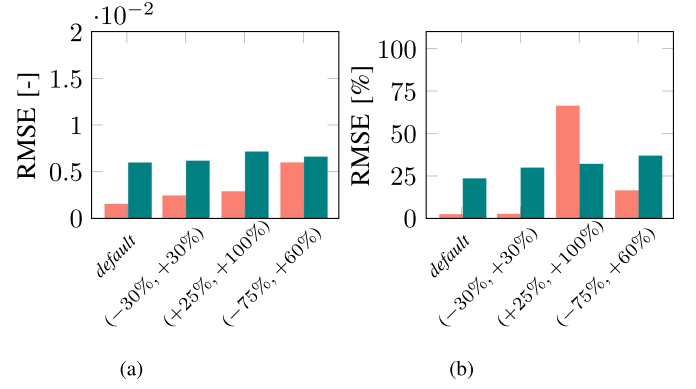

Fig. 11.   Comparing RMSE for (a) voltage level and (b) line loading between the WLS (■) and DSS^2 (■) in the 70-bus network considering from the *default* three changes in generation and load: 30% decrease in generation and 30% increase in load, 25% increase in generation and 100% increase in load and 75% decrease in generation and 60% increase in load.

the loads and distributed generation compared to the training dataset. Three cases altered the power levels for the testing dataset on the 70-bus network by:

$(-30\%, +30\%)$ 30% decrease in generation, 30% increase in load,

$(+25\%, +100\%)$ 25% increase in generation, 100% increase in load to simulate a system near overload.

$(-75\%, +60\%)$ 75% decrease in generation, 60% increase in load to simulate more voltage deviations

Note that the DSS^2 model was never trained on such cases; only *default* power levels were used for training.

Fig. 11 shows the results. In the case of a 'small' load change $(-30\%, +30\%)$, the DSS^2 showed good estimation performances with only a small increase in RMSE. However, in the cases $(+25\%, +100\%)$ and $(-75\%, +60\%)$ the RMSE significantly increased. The lines were highly loaded in the case $(+25\%, +100\%)$. Hence, the loading estimation was highly impacted. In the case $(-75\%, +60\%)$, the deviation in voltage was more harmful to the voltage estimation. These results explored the limitations of the changes in loading levels that the DSS^2 model could handle. Good results were perceived for changes in loads of around 30% showing good generalization capability of the DSS^2 model to handle state estimation tasks under limited uncertain changes. However, the model became sensitive when the network was extremely loaded or under strong voltage deviations, and then, the model does not generalize well anymore to extreme conditions.

V. DISCUSSION AND CONCLUSION

This article introduces the Deep Statistical Solver for Distribution System State Estimation. This Deep Learning architecture incorporates the power flow equations in the loss function for physics awareness. Our proposed DSS^2 approach uses the same objective function as the WLS, allowing to train of the model with a noisy and poorly labelled dataset. This approach is called weak supervision learning, and we combine it with a GNN

architecture to enhance the learning from local patterns and the robustness of the model. A remarkable advantage of the DSS^2 is that the larger the power network, the better the performance. The DSS^2 is based on a GNN architecture that learns from local patterns (in the neighbourhood of buses). Hence, the larger the network, the more local patterns the GNN-based architecture can learn from. We consider this remarkable as conventional power system analysis, for example, for estimating the state, often scales poorly with network size, whereas DSS^2 showed the reverse effect. Another outstanding advantage is that through learning in the neighbourhood of buses, the DSS^2 model becomes robust and invariant to changes in individual values, such as missing, erroneous measurements. This is an important practical advantage over other conventional methods (and the studied supervised models) that depend on the accuracy of individual measurements. Our different case studies show that the DSS^2 is faster, more robust, and more scalable than WLS as DSS^2 does not involve iterative algorithms and learns from local patterns and noisy measurements. Compared to supervised models, the weakly supervised DSS^2 shows equivalent speed and voltage accuracy while outperforming the supervised models in estimating indirect values such as line loading. We conclude that learning from the power flow equations and the neighbourhood are the strengths of DSS^2; these incorporate a coupling between voltage magnitudes and voltage angles to fit the measurements. Finally, the DSS^2 model does not require labels as the approach is weakly supervised learning from the power flow equations. This type of learning makes the DSS^2 model more practical than other ML-based approaches as labels are scarce.

Our implementation of the DSS^2 has limitations. First of all, the penalization method used in training impacts the quality of estimation but does not ensure any guarantee of convergence during testing. Feasible solutions cannot be guaranteed with a data-driven method that focuses on individual accuracy. Secondly, in our implementation of the H2MG architecture, the assumption to modelling transformers as lines may have particularly limited the accuracy of transformers' loading estimation. There, the model was 'forced' to learn a similar input-output mapping for lines and transformers that may reduce the expressivity of the model. Then, the DSS^2's estimation is impacted when the load power level in the network varies significantly. The generalization ability of the DSS^2 showed a limit of around 30% load changes. The changes in measurements are encouraging but should be improved.

Future work could investigate the types of measurements and meter placement decisions that would maximize the DSS^2 performances. Adding an algorithm that detects changes in the data could benefit quantifying the confidence of state estimations by DSS^2. Combining the DSS^2 for state estimation to a state-of-the-art anomaly detector could improve generalization. Also, an extension to unbalanced systems is deemed possible by extending to unbalanced systems modeling and power flow equations, and it should be investigated in the future. Then, the network's model in the deep learning architecture could be improved. The proposed model is simple; however, the H2MGNN architecture allows for advanced modelling of components that can further increase the expressivity and performance of the DSS^2. Finally,

future work should explore robustness to model inaccuracies and implementation for distribution grids that undergo topology changes. This can be achieved by leveraging the robustness of GNN to graph variation. Such an implementation will further improve the practicality, robustness, and accuracy of the model.

REFERENCES

[1] E. M. Lourenço, Ed., *Power Distribution System State Estimation* (Energy Engineering Series). Stevenage, U.K.: Inst. Eng. Technol., 2022. [Online]. Available: https://digital-library.theiet.org/content/books/po/pbpo183e

[2] A. Primadianto and C. N. Lu, "A review on distribution system state estimation," *IEEE Trans. Power Syst.*, vol. 32, no. 5, pp. 3875–3883, Sep. 2017.

[3] D. Della Giustina, M. Pau, P. A. Pegoraro, F. Ponci, and S. Sulis, "Electrical distribution system state estimation: Measurement issues and challenges," *IEEE Instrum. Meas. Mag.*, vol. 17, no. 6, pp. 36–42, Dec. 2014.

[4] M. Fotopoulou, S. Petridis, I. Karachalios, and D. Rakopoulos, "A review on distribution system state estimation algorithms," *Appl. Sci.*, vol. 12, no. 21, 2022, Art. no. 11073. [Online]. Available: https://www.mdpi.com/2076-3417/12/21/11073

[5] V. Kekatos and G. B. Giannakis, "Distributed robust power system state estimation," *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 1617–1626, May 2013.

[6] G. N. Korres, "A distributed multiarea state estimation," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 73–84, Feb. 2011.

[7] M. Pau, F. Ponci, A. Monti, C. Muscas, and P. A. Pegoraro, "Distributed state estimation for multi-feeder distribution grids," *IEEE Open J. Instrum. Meas.*, vol. 1, pp. 1–12, 2022.

[8] K. Dehghanpour, Z. Wang, J. Wang, Y. Yuan, and F. Bu, "A survey on state estimation techniques and challenges in smart distribution systems," *IEEE Trans. Smart Grid*, vol. 10, no. 2, pp. 2312–2322, Mar. 2019.

[9] M. Pau, P. A. Pegoraro, and S. Sulis, "WLS distribution system state estimator based on voltages or branch-currents: Accuracy and performance comparison," in *Proc. IEEE Int. Instrum. Meas. Technol. Conf.*, 2013, pp. 493–498.

[10] F. Ahmad, A. Rasool, E. Ozsoy, R. Sekar, A. Sabanovic, and M. Elitaş, "Distribution system state estimation-a step towards smart grid," *Renewable Sustain. Energy Rev.*, vol. 81, pp. 2659–2671, 2018.

[11] X. B. Jin, R. J. R. Jeremiah, T. L. Su, Y. T. Bai, and J. L. Kong, "The new trend of state estimation: From model-driven to hybrid-driven methods," *Sensors*, vol. 21, pp. 1–25, 2021.

[12] H. Liu, F. Hu, J. Su, X. Wei, and R. Qin, "Comparisons on Kalman-filter-based dynamic state estimation algorithms of power systems," *IEEE Access*, vol. 8, pp. 51035–51043, 2020.

[13] N. Kumari, R. Kulkarni, M. R. Ahmed, and N. Kumar, "Use of Kalman filter and its variants in state estimation: A review," in *Artificial Intelligence for a Sustainable Industry 4.0*. Berlin, Germany: Springer, 2021, pp. 213–230, doi: 10.1007/978-3-030-77070-9_13.

[14] P. A. Pegoraro et al., "Bayesian approach for distribution system state estimation with non-Gaussian uncertainty models," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 11, pp. 2957–2966, Nov. 2017.

[15] K. R. Mestav, J. Luengo-Rozas, and L. Tong, "Bayesian state estimation for unobservable distribution systems via deep learning," *IEEE Trans. Power Syst.*, vol. 34, no. 6, pp. 4910–4920, Nov. 2019.

[16] M. Q. Tran, A. S. Zamzam, P. H. Nguyen, and G. Pemen, "Multi-area distribution system state estimation using decentralized physics-aware neural networks," *Energies*, vol. 14, 2021, Art. no. 3025.

[17] B. Zargar, A. Angioni, F. Ponci, and A. Monti, "Multiarea parallel data-driven three-phase distribution system state estimation using synchrophasor measurements," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 9, pp. 6186–6202, Sep. 2020.

[18] S. de Jongh, F. Müller, H. Li, B. Georgieva, M. Suriyah, and T. Leibfried, "Machine-learning-based Bayesian state estimation in electrical energy systems," in *Proc. CIRED 2020 Berlin Workshop*, 2020, pp. 341–344.

[19] O. Kundacina, M. Cosovic, and D. Vukobratovic, "State estimation in electric power systems leveraging graph neural networks," in *Proc. IEEE 17th Int. Conf. Probabilistic Methods Appl. Power Syst.*, 2022, pp. 1–6. [Online]. Available: https://doi.org/10.1109%2Fpmaps53380.2022.9810559

[20] A. S. Zamzam and N. D. Sidiropoulos, "Physics-aware neural networks for distribution system state estimation," *IEEE Trans. Power Syst.*, vol. 35, no. 6, pp. 4347–4356, Nov. 2020.

[21] G. Revach, N. Shlezinger, R. J. G. van Sloun, and Y. C. Eldar, "KalmanNet: Data-driven Kalman filtering," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2021, pp. 3905–3909.

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

[23] Z.-H. Zhou, "A brief introduction to weakly supervised learning," *Nat. Sci. Rev.*, vol. 5, no. 1, pp. 44–53, 2017, doi: 10.1093/nsr/nwx106.

[24] M. Lave, M. J. Reno, and J. Peppanen, "Distribution system parameter and topology estimation applied to resolve low-voltage circuits on three real distribution feeders," *IEEE Trans. Sustain. Energy*, vol. 10, no. 3, pp. 1585–1592, Jul. 2019.

[25] H. Li, Y. Weng, Y. Liao, B. Keel, and K. E. Brown, "Distribution grid impedance & topology estimation with limited or no micro-PMUs," *Int. J. Elect. Power Energy Syst.*, vol. 129, 2021, Art. no. 106794. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S014206152100034X

[26] Y. Liao, Y. Weng, G. Liu, and R. Rajagopal, "Urban MV and LV distribution grid topology estimation via group lasso," *IEEE Trans. Power Syst.*, vol. 34, no. 1, pp. 12–27, Jan. 2019.

[27] F. Gama, E. Isufi, G. Leus, and A. Ribeiro, "Graphs, convolutions, and neural networks: From graph filters to graph neural networks," *IEEE Signal Process. Mag.*, vol. 37, no. 6, pp. 128–138, Nov. 2020.

[28] P. W. Battaglia et al., "Relational inductive biases, deep learning, and graph networks," 2018. [Online]. Available: https://arxiv.org/abs/1806.01261

[29] F. Gama, J. Bruna, and A. Ribeiro, "Stability properties of graph neural networks," *IEEE Trans. Signal Process.*, vol. 68, pp. 5680–5695, 2020.

[30] Z. Gao, E. Isufi, and A. Ribeiro, "Stability of graph convolutional neural networks to stochastic perturbations," *Signal Process.*, vol. 188, 2021, Art. no. 108216.

[31] Y. Ma, X. Liu, T. Zhao, Y. Liu, J. Tang, and N. Shah, "A unified view on graph neural networks as graph signal denoising," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.*, 2021, pp. 1202–1211, doi: 10.1145/3459637.3482225.

[32] W. Liao, B. Bak-Jensen, J. R. Pillai, Y. Wang, and Y. Wang, "A review of graph neural networks and their applications in power systems," *J. Modern Power Syst. Clean Energy*, vol. 10, no. 2, pp. 345–360, Mar. 2022.

[33] M. Ringsquandl et al., "Power to the relational inductive bias: Graph neural networks in electrical power grids," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.*, 2021, pp. 1538–1547.

[34] O. Kundacina, M. Cosovic, D. Miskovic, and D. Vukobratovic, "Distributed nonlinear state estimation in electric power systems using graph neural networks," in *Proc. IEEE Int. Conf. Commun., Control, Comput. Technol. Smart Grids*, 2022, pp. 8–13.

[35] Z. Wu, Q. Wang, and X. Liu, "State estimation for power system based on graph neural network," in *Proc. IEEE 5th Int. Elect. Energy Conf.*, 2022, pp. 1431–1436.

[36] H. Lin and Y. Sun, "ElegNN: Electrical-model-guided graph neural networks for power distribution system state estimation," in *Proc. IEEE Glob. Commun. Conf.*, 2022, pp. 5292–5298.

[37] R. Madbhavi, B. Natarajan, and B. Srinivasan, "Graph neural network-based distribution system state estimators," *IEEE Trans. Ind. Informat.*, early access, Feb. 23, 2023, doi: 10.1109/TII.2023.3248082.

[38] B. Donon, "Deep statistical solvers & power systems applications," Doctoral dissertation, Orsay, France, Université Paris-Saclay, 2022. [Online]. Available: https://tel.archives-ouvertes.fr/tel-03624628

[39] B. Habib, E. Isufi, W. van Breda, A. Jongepier, and J. Cremer, "Implementation of deep statistical solver for distribution system state estimation," 2023. [Online]. Available: https://github.com/TU-Delft-AI-Energy-Lab/Deep-Statistical-Solver-for-Distribution-System-State-Estimation

[40] A. Abur and A. G. Exposito, *Power System State Estimation : Theory and Implementation*. New York, NY, USA: Marcel Dekker, 2004.

[41] L. Thurner et al., "Pandapower—An open source python tool for convenient modeling, analysis and optimization of electric power systems," *IEEE Trans. Power Syst.*, vol. 33, no. 6, pp. 6510–6521, Nov. 2018, doi: 10.1109/TPWRS.2018.2829021.

[42] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, 1989. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0893608089900208

[43] K. Strunz, E. Abbasi, R. Fletcher, N. Hatziargyriou, R. Iravani, and G. Joos, *TF C6.04.02: TB 575—Benchmark Systems for Network Integration of Renewable and Distributed Energy Resources*, 2014.

[44] K. Rudion, A. Orths, Z. A. Styczynski, and K. Strunz, "Design of Benchmark of Medium Voltage Distribution Network for Investigation of DG Integration," in *Proc. IEEE Power Eng. Soc. Gen. Meeting*, 2006.

[45] F. C. Schweppe and J. Wildes, "Power system static-state estimation, Part I: Exact model," *IEEE Trans. Power App. Syst.*, vol. PAS- 89, no. 1, pp. 120–125, Jan. 1970.

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