

LEBANESE AMERICAN UNIVERSITY

**A Robust Deep Learning Approach for
Distribution System State Estimation with
Distributed Generation**

By

Ronald Kfour

A thesis
submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Engineering

School of Engineering
January 2023

© 2023

Ronald Kfour

All Rights Reserved

THESIS APPROVAL FORM

Student Name: Ronald Kfour I.D. #: 201004099

Thesis Title: A Robust Deep Learning Approach for Distribution System State Estimation with
Distributed Generation

Program: Computer Engineering

Department: Electrical and Computer Engineering

School: Engineering

The undersigned certify that they have examined the final electronic copy of this thesis and approved it in Partial Fulfillment of the requirements for the degree of:

Master of Science in the major of Computer Engineering

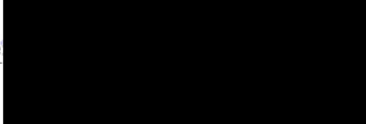
Thesis Advisor's Name: Dr. Harag Margossian

Signature:  Date: 9 / January / 2023
Day Month Year

Committee Member's Name: Dr. Wissam Fawaz

Signature:  Date: 9 / January / 2023
Day Month Year

Committee Member's Name: Dr. Raymond Ghajar


Signature:  Date: 9 / January / 2023
Day Month Year

THESIS COPYRIGHT RELEASE FORM

LEBANESE AMERICAN UNIVERSITY NON-EXCLUSIVE DISTRIBUTION LICENSE

By signing and submitting this license, you (the author(s) or copyright owner) grants the Lebanese American University (LAU) the non-exclusive right to reproduce, translate (as defined below), and/or distribute your submission (including the abstract) worldwide in print and electronic formats and in any medium, including but not limited to audio or video. You agree that LAU may, without changing the content, translate the submission to any medium or format for the purpose of preservation. You also agree that LAU may keep more than one copy of this submission for purposes of security, backup and preservation. You represent that the submission is your original work, and that you have the right to grant the rights contained in this license. You also represent that your submission does not, to the best of your knowledge, infringe upon anyone's copyright. If the submission contains material for which you do not hold copyright, you represent that you have obtained the unrestricted permission of the copyright owner to grant LAU the rights required by this license, and that such third-party owned material is clearly identified and acknowledged within the text or content of the submission. IF THE SUBMISSION IS BASED UPON WORK THAT HAS BEEN SPONSORED OR SUPPORTED BY AN AGENCY OR ORGANIZATION OTHER THAN LAU, YOU REPRESENT THAT YOU HAVE FULFILLED ANY RIGHT OF REVIEW OR OTHER OBLIGATIONS REQUIRED BY SUCH CONTRACT OR AGREEMENT. LAU will clearly identify your name(s) as the author(s) or owner(s) of the submission, and will not make any alteration, other than as allowed by this license, to your submission.

Name: Ronald Kfoury

Signature 

Date: 15 / 12 / 2022
Day Month Year

PLAGIARISM POLICY COMPLIANCE STATEMENT

I certify that:

1. I have read and understood LAU's Plagiarism Policy.
2. I understand that failure to comply with this Policy can lead to academic and disciplinary actions against me.
3. This work is substantially my own, and to the extent that any part of this work is not my own I have indicated that by acknowledging its sources.

Name: Ronald Kfourj

Signature:

Date: 15 / 12 / 2022

Day Month Year

ACKNOWLEDGMENT

Firstly, my immense gratitude goes to Dr. Wissam Fawaz and Dr. Raymond Ghajar, who generously provided their time to serve on the committee. Secondly, my appreciation goes to Dr. Harag Margossian, my thesis supervisor, for his patience, time, and guidance throughout this project. Finally, this thesis could not have been completed without the endless support of my family and friends.

A Robust Deep Learning Approach for Distribution System State Estimation with Distributed Generation

Ronald Kfoury

ABSTRACT

Distribution System State Estimation (DSSE) remains a challenging problem due to the nature of distribution grids. Conventional methods, which are used to solve state estimation on the transmission level, require the grid to be observable. This is not directly applicable to distribution grids. In addition, the high integration of renewable energy introduces uncertainty, which makes the DSSE problem more complex. This work proposes a deep neural network approach that solves the DSSE problem with and without distributed generation, without using highly inaccurate pseudo-measurements. Due to the lack of public frameworks, we create a dataset that emulates real-life scenarios to train and test the neural network. Also, to evaluate the robustness of the algorithms, we test the neural network, without retraining it, on multiple scenarios with noisier data and bad data. The algorithms are tested on three different networks. The proposed approach solves the DSSE problem with limited measurements as inputs, which cannot be solved

using conventional state estimation methods. Our approach also achieves highly accurate results, despite the additional noise introduced to the measurements.

Keywords: Bad Data, Deep Learning, Distributed Generation, Distribution System State Estimation, Renewable Energy Integration.

TABLE OF CONTENTS

ACKNOWLEDGMENT	v
ABSTRACT	vi
TABLE OF CONTENTS	viii
LIST OF TABLES	x
LIST OF FIGURES	xi
I - INTRODUCTION	1
II - BACKGROUND	4
2.1 State Estimation	4
2.1.1 Definition.....	4
2.1.2 Modeling the Measurements	5
2.1.3 Observability	8
2.1.4 Bad Data Detection and Identification	8
2.2 Neural Network	9
III - LITERATURE REVIEW	11
IV - CREATING THE DATASET	13
V - METHODOLOGY	17
5.1 Neural Network	18
5.2 Testing the Robustness of the Neural Network	23
5.2.1 Scenario 1 - Noisier Data	23

5.2.2 Scenario 2 - Bad Data.....	23
VI - RESULTS AND DISCUSSION	24
6.1 Software and Equipment	24
6.2 Cases Without DG	25
6.2.1 18-bus System Demo.....	25
6.2.2 The Remaining Cases	28
6.3 Cases with DG.....	28
6.3.1 18-bus System Demo.....	28
6.3.2 The Remaining Cases	30
6.4 Noisier Data.....	30
6.5 Bad Data.....	31
6.6 Discussion	33
VII - CONCLUSION AND FUTURE WORK	34
VIII - REFERENCES	35

LIST OF TABLES

Table 1 - Simulation Results for Voltage Magnitudes and Phase Angles Without DG for the 3 Test Cases.....	25
Table 2 - Comparing WLS to NN for the 18-bus System.....	26
Table 3 - Simulation Results for Voltage Magnitudes and Phase Angles with DG for the 3 Test Cases.....	29
Table 4 - Noisier Data Results for Cases Without DG.....	31
Table 5 - Noisier Data Results for Cases with DG.....	31
Table 6 - Bad Data Results for Cases Without DG.....	32
Table 7 - Bad Data Results for Cases with DG.....	32

LIST OF FIGURES

Figure 1 - Two-port π -model of a Network Branch	7
Figure 2 - Measurement Distribution for the 18-bus System.....	14
Figure 3 - Flowchart for Creating the Dataset with DG.....	16
Figure 4 - Methodology Summary.	17
Figure 5 - Sigmoid Activation Function.....	18
Figure 6 - Tanh Activation Function.....	19
Figure 7 - ReLU Activation Function.	19
Figure 8 - Neural Network Architecture.	22
Figure 9 - Simulation Results for Predicting Voltage Magnitudes for the 18-bus System Without DG.....	27
Figure 10 - Simulation Results for Predicting Phase Angles for the 18-bus System Without DG.....	27
Figure 11 - Simulation Results for Predicting Voltage Magnitudes for the 18-bus System with DG.....	29
Figure 12 - Simulation Results for Predicting Phase Angles for the 18-bus System with DG.....	30

CHAPTER ONE

INTRODUCTION

Distribution systems are different from transmission systems. They have a low X/R ratio, are highly unobservable, and have a radial or weakly-meshed topology. Also, high penetration of renewable energy resources and Distributed Generation (DG) render the generation less predictable. Hence, the Distribution System State Estimation (DSSE) problem has been heavily studied since the 1990s [1]-[3]. Researchers first adopted the State Estimation (SE) methods, which are used for transmission systems, to distribution systems: for example, the Weighted Least Squares (WLS) method based on node voltages [4]-[6] or branch currents [7]-[9], the Least Absolute Value (LAV) method [10], and the Generalized Maximum Likelihood (GM) method [11]. However, WLS is susceptible to bad data, LAV is computationally expensive and sensitive to measurement uncertainty, and GM is sensitive to parameter selection [12].

Moreover, to solve the unobservability problem, pseudo-measurements are employed [13]. These are generated from historic data, standard load data, or weather data and thus are highly inaccurate. Probabilistic or statistical-based methods are used to generate and handle the errors of these pseudo-measurements: for example, Gaussian Mixture Models (GMMs) [14], correlation analysis [15], [16], and multivariate Gaussian modeling [17].

In addition, with the ubiquitous renewable energy resources and DG integration, uncertainty is introduced to the grid, leading researchers to solve the DSSE in presence of DGs. For instance, forecasting aided DSSE is studied in [18] using GM and in [19] using L_1 regularization; optimal meter placement in active distribution grids is studied in [20], and pseudo-measurements generation with high DG penetration is studied in [21], [22].

Also, [23] proposes models of photovoltaic (PV) arrays to be integrated within the grid model, while [24] proposes an approach to generate pseudo-measurements for an active distribution grid.

Another problem that affects SE in general is bad data in measurements [25]-[27]. The term bad data refers to the measurements that suffer from anomalies due to communication noise or meter errors. In [28], a Lagrangian-based approach is used to identify bad data. In [29], L_1 -norm is used to identify corrupted measurements and estimate the states. But the L_1 penalty function is susceptible to outliers that have large magnitudes. As an improvement on [30], the authors in [31] propose a statistical approach based on iterative reweight least squares and an improved ADMM approach. Also, [32] formulates the Phasor Measurement Unit (PMU)-based SE as a quadratic programming problem and solves it in a decentralized way to maintain privacy between operators.

However, these methods get computationally expensive for high dimensions. Also, the DSSE remains a nonlinear problem, whose complexity is increasing with the increase of renewable energy integration, thus traditional methods cannot handle this problem and are most of the times prone to introducing relaxations or simplifications or guessing initial conditions. This leads the algorithm to either get stuck in local optima or not converge.

Thus, researchers introduced Artificial Intelligence (AI) algorithms to solve DSSE-related problems. These algorithms shift the computational burden to an offline stage. Examples include integrating AI algorithms with traditional algorithms, such as [33] that uses a Neural Network (NN) to initialize the Gauss-Newton method, [34] that proposes a NN to generate pseudo-measurements, and [35] that uses a Nonlinear Auto-Regressive Exogenous algorithm to predict the load and feed it back into the state estimator. Another research path is to solve the DSSE problem using only AI methods. For instance, [36] uses a Bayesian inference approach for unobservable distribution systems via deep learning.

Auto-encoders are studied in [37] to ensure two-way communication with unobservable distribution grids. Authors in [38] introduce a physics aware NN that follows the topology of the grid to solve the DSSE problem.

Although these methods have achieved impressive results, the reviewed papers suffer from at least one of the following problems:

- The methods use pseudo-measurements, with an error up to 50%, to compensate for the unobservability issue; this leads to inaccurate results following from the garbage in garbage out concept.
- The methods are not verified against outliers or practical anomalies.

This work proposes a deep learning approach that solves the DSSE problem without the need to generate pseudo-measurements, which cannot be performed using conventional SE methods. We present two scenarios: the first scenario calculates the system states without DG integration, while the second scenario calculates them with DG. The contributions are threefold:

1. We propose a deep neural network approach to solve the DSSE problem without using erroneous pseudo-measurements. The suggested algorithm requires a limited number of measurements, since distribution grids are unobservable, and can be used in real-time to cater to the need of frequently solving the DSSE problem.
2. Given the additional complexity introduced by DG, the approach shows a very good performance compared with the case of passive grids.
3. To test for robustness, we perform different test cases. We check the performance given noisier measurements and bad data.

CHAPTER TWO

BACKGROUND

2.1 State Estimation

2.1.1 Definition

Power systems are composed of generation, transmission, sub-transmission, and distribution systems, and loads. Generators inject power into the grid, while loads absorb it. Transmission systems embed substations that are interconnected via transmission lines, transformers, and other devices for control and protection. Distribution systems, which are connected to the transmission systems, usually operate in a radial or weakly-meshed configuration [39].

Operational constraints of the grid include the upper and lower limits on the voltage magnitudes and transmission lines capacity. A grid operates in a normal state if all the demand is supplied by the generators without violating these operational constraints. The operating conditions of a grid are determined given the network topology and the complex phasor voltages at every bus. The set of complex phasor voltages is referred to as the static state of the system [39], [40].

State Estimation (SE) is a data processing algorithm that converts meter measurements and other data into an estimate of the state vector [41]. The state vector consists of the voltage magnitudes and phase angles in static state estimation. Generally, measurements include active and reactive power flows, power injections, complex phasor voltages, current, generator outputs, loads, circuit breaker and switch status information, etc. These measurements are taken from sensors and smart meters, placed on the grid. The

state estimator filters the noise introduced to these measurements and estimates the state of the grid.

2.1.2 Modeling the Measurements

The Maximum Likelihood Estimation (MLE) can be used to estimate the most likely state of the grid. We assume that the measurements follow a previously determined probability distribution, e.g., Gaussian, with unknown parameters. The joint probability density function (pdf) is referred to as the likelihood function, which reaches its peak when the predicted parameters are closest to their actual values. Thus, an optimization problem is set to maximize the likelihood function [39].

Assuming independent measurements, the joint pdf can be expressed as follows:

$$f_m(\mathbf{z}) = f(z_1)f(z_2) \dots f(z_m) \quad (1)$$

where $f(z_i)$ is the individual pdf of every z_i measurement. $f_m(\mathbf{z})$ is referred to as the likelihood function for \mathbf{z} . MLE maximizes $f_m(\mathbf{z})$, or its log (to simplify the optimization process), thus, the problem reduces to the following:

$$\begin{aligned} & \text{maximize} \quad \log f_m(\mathbf{z}) \\ & \text{OR} \\ & \text{minimize} \quad \sum_{i=1}^m \left(\frac{z_i - \mu_i}{\sigma_i} \right)^2 \end{aligned} \quad (2)$$

where μ_i is the mean and σ_i is the standard deviation of z_i . Equation (2) can be re-written in terms of the residual r_i , defined as:

$$r_i = z_i - \mu_i \quad (3)$$

The mean can be expressed as $h_i(\mathbf{x})$, which is a nonlinear function that relates the state vector to the i^{th} measurement. The square of the residual, r_i^2 , is weighted by the matrix \mathbf{W} , having $W_{ii} = \sigma_i^{-2}$ entries, which represent the inverse error variance. Thus, the minimization problem reduces to the following:

$$\begin{aligned}
& \text{minimize} && \sum_{i=1}^m W_{ii} r_i^2 \\
& \text{subject to} && z_i = h_i(\mathbf{x}) + r_i, \quad i = 1, \dots, m
\end{aligned} \tag{4}$$

The solution of this optimization problem is referred to as the Weighted Least Squares (WLS) estimation of \mathbf{x} [39].

The measurement model thus becomes as follows:

$$\mathbf{z}_m = \begin{bmatrix} z_1 \\ \vdots \\ z_m \end{bmatrix} = \begin{bmatrix} h_1(x_1, \dots, x_n) \\ \vdots \\ h_m(x_1, \dots, x_n) \end{bmatrix} + \begin{bmatrix} e_1 \\ \vdots \\ e_m \end{bmatrix} = h_i(\mathbf{x}) + \mathbf{e} \tag{5}$$

where

\mathbf{z}_m is the measurement vector that includes m measurements,

$\mathbf{x} = (x_1, \dots, x_n)$ is a vector denoting n system states, composed of voltage magnitudes and phase angles,

$h_i(\mathbf{x})$ is a nonlinear function of \mathbf{x} , and

\mathbf{e} is a vector denoting the measurement noise that is usually assumed to follow a Gaussian distribution with zero-mean and a covariance matrix $\mathbf{R} = \text{diag}\{\sigma_1^2, \dots, \sigma_m^2\}$, σ_i^2 being the variance of \mathbf{z} .

The WLS is then equivalent to minimizing the following objective function:

$$J(\mathbf{x}) = \sum_{i=1}^m \frac{1}{R_{ii}} (z_i - h_i(\mathbf{x}))^2 \tag{6}$$

This optimization problem is usually solved using the iterative Gauss-Newton method.

However, this method is sensitive to initialization, computationally expensive, and is not guaranteed to converge [39], [40].

In this manuscript, we use three types of measurements, whose relations with the state variables are determined by the following equations:

- Voltage magnitudes at bus i : V_i
- Real and reactive power injections at bus i :

$$P_i = V_i \sum_{j=1}^N V_j (G_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij}) \quad (7)$$

$$Q_i = V_i \sum_{j=1}^N V_j (G_{ij} \sin \theta_{ij} - B_{ij} \cos \theta_{ij}) \quad (8)$$

- Real and reactive power flows from bus i to bus j :

$$P_{ij} = V_i^2 (g_{si} + g_{ij}) - V_i V_j (g_{ij} \cos \theta_{ij} + b_{ij} \sin \theta_{ij}) \quad (9)$$

$$Q_{ij} = -V_i^2 (b_{si} + b_{ij}) - V_i V_j (g_{ij} \sin \theta_{ij} - b_{ij} \cos \theta_{ij}) \quad (10)$$

where

V_i and θ_i are the voltage magnitude and phase angle at bus i ,

$$\theta_{ij} = \theta_i - \theta_j,$$

$G_{ij} + jB_{ij}$ is the ij^{th} element of the complex bus admittance matrix,

$g_{ij} + jb_{ij}$ is the admittance of the series branch connecting buses i and j ,

$g_{si} + jb_{si}$ is the admittance of the shunt branch connected at bus i , following the

two-port π -model of a network branch, as shown in Figure 1, and

N is the number of buses.

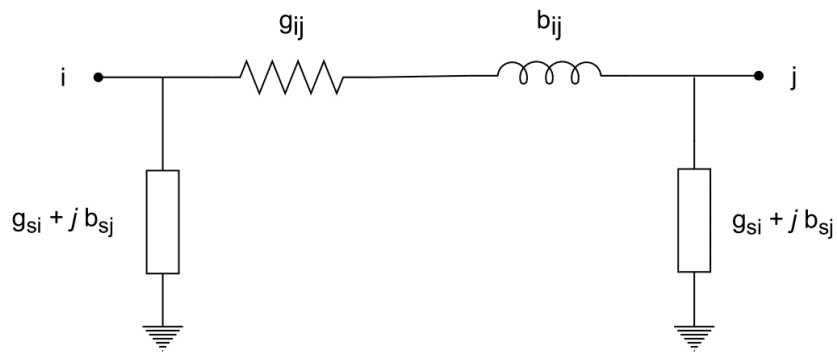


Figure 1 - Two-port π -model of a Network Branch.

2.1.3 Observability

The network observability determines whether the state can be uniquely estimated, given the type of available measurements and their corresponding locations. This analysis checks the sufficiency of the existing measurements. If the system is not observable, then additional sensors or meters need to be placed on the grid. Observability can be determined using numerical or topological methods, which also determine observable islands and critical measurements [39].

Distribution grids, however, are highly unobservable due to the scarcity of the sensors and smart meters placed on such grids [42]. Thus, the SE problem cannot be solved using conventional methods. To solve the unobservability problem, pseudo-measurements are employed [13]. These are generated from historic data, standard load data, or weather data. Thus, they are assigned high variances to compensate for their high inaccuracy.

2.1.4 Bad Data Detection and Identification

The term bad data refers to the measurements that contain errors due to communication noise or meter errors. Detection refers to determining whether the measurement set contains any bad data. Identification refers to determining which particular measurement(s) truly contains bad data [39].

Given that there are redundant measurements, state estimators are expected to detect such errors, identify them, and filter them out. Some types of bad data are detected a priori such as negative voltage magnitudes or very large power flows or injections. Other types of bad data need to be detected and eliminated a posteriori. This process depends on the method used to solve the SE problem. For example, upon using the WLS method, identifying and detecting bad data are executed after the estimation process [39].

2.2 Neural Network

Our model consists of a feedforward neural network having k layers with \mathbf{z} inputs and \mathbf{y} outputs. Each layer has $\hat{\mathbf{u}}_i$ inputs and \mathbf{u}_i outputs, with $\mathbf{u}_0 = \mathbf{z}$ and $\mathbf{u}_k = \mathbf{y}$, where:

$\hat{\mathbf{u}}_i$ represents the values *before* the activation function, and

\mathbf{u}_i represents the values *after* the activation function. $\hat{\mathbf{u}}_i$ and \mathbf{u}_i are defined, respectively,

as $\hat{\mathbf{u}}_i = \boldsymbol{\omega}_i \mathbf{u}_{i-1} + \mathbf{b}_i$ and $\mathbf{u}_i = \sigma_i(\boldsymbol{\omega}_i \mathbf{u}_{i-1} + \mathbf{b}_i)$, where:

$\boldsymbol{\omega}_i$ is the weight matrix,

\mathbf{b}_i is the bias vector, and

$\sigma_i(\cdot)$ is the activation function.

The model is provided first with a *Training* set, that is usually labeled. Having labeled data, i.e., data with both the inputs and outputs known, allows the neural network to be trained to learn the pattern of the data. Once the model is trained, it can calculate the values of new unseen data, which is commonly referred to as a *Testing* set. If the output comprises two or more discrete categories, then the problem is known as a *classification* problem. If the output comprises one or more continuous variables, then it is known as a *regression* problem, which is the case in this manuscript. Calculating and storing \mathbf{u}_i , $\hat{\mathbf{u}}_i$, and \mathbf{y} in order from the input layer to the output layer is known as *Forward Propagation* [43].

Backward Propagation refers to calculating and storing the gradients (or the partial derivatives) of the above variables in reverse order, i.e., from the output layer to the input layer. We define the error function $E(\boldsymbol{\omega})$ that compares the calculated output with the desired output. Our goal is to train the neural network by minimizing the error function. That is, find a weight matrix and bias vector, $\boldsymbol{\omega}$ and \mathbf{b} , that minimize the chosen error function. The smallest value would occur where the gradient of the error function is zero:

$$\nabla E(\omega) = 0 \quad (11)$$

However, this function is highly nonlinear, thus it can be stuck in local minima. Numerous methods have been proposed to solve the optimization problem; one efficient procedure is the error back propagation process, mainly using the chain rule of calculus [43].

CHAPTER THREE

LITERATURE REVIEW

In [34], Manitsas *et. al* propose an artificial neural network approach to generate pseudo-measurements. The authors provide real measurements and typical load profiles to a neural network that outputs active and reactive power injections to be used as pseudo-measurements in DSSE. The approach is demonstrated using the 95-bus section of the U.K. generic distribution system model. Results show that the estimated states are within the confidence bounds where voltage magnitudes estimation performed significantly better than phase angles estimation.

Hayes *et. al* in [35] propose a load forecasting and SE tool for distribution grids. The predictive database is designed using a Nonlinear Auto-Regressive Exogenous algorithm to predict the load. Load estimations are fed back into the state estimator to create a closed-loop information flow that is used to monitor and improve the DSSE. This method also provides system operators with warnings of potential issues in medium voltage grids. The algorithm is tested using a case study of a real 10kV distribution grid with a weakly-meshed topology. Results show a reduced load forecasting error and reduced estimation error.

In [33], Zamzam *et. al* propose using AI techniques to initialize the Gauss-Newton method. Specifically, they use historical or simulated data to train and test a neural network that maps the measurements to the true network voltages. Simulations are demonstrated using the IEEE-37 distribution feeder. The authors introduce renewable energy resources to be installed on the grid and use linear and quadratic measurements, as well as pseudo-measurements. This hybrid optimization approach is demonstrated to have

a better performance than optimization-only approach in terms of accuracy, runtime, and stability.

Authors in [36] propose a deep learning approach for unobservable distribution grids with applications in real-time. The algorithm consists of a Monte Carlo sampling technique to train a neural network for SE, followed by a bad data detection and filtering method using a Bayesian inference approach. The method is demonstrated using the 85-bus distribution grid. Results show that the computational complexity of the Bayesian approach is lower than that of the WLS techniques when used online and has a higher accuracy level.

Authors in [38] introduce a physics aware NN that follows the topology of the grid to solve the DSSE problem. Instead of fully connected layers, the authors suggest the neurons of the NN to be connected according to the grid topology. This reduces the complexity of the training stage and prevents overfitting. They also propose a greedy algorithm to minimize the complexity of the NN through optimally placing micro-PMUs. Simulations are done on the IEEE 37-bus system. Results show a better performance than Gauss-Newton and a robustness against corrupted measurements.

Sundaray *et. al* in [37] study auto-encoders to provide two-way communication with unobservable distribution grids. The approach incorporates constraints on the latent layer as per the voltage measurements and employ symbolic regression for explainability. Moreover, nonlinear power flow kernels are embedded into the decoder to adjust the forward and inverse mappings. The algorithm is tested on IEEE distribution systems up to 141 and 8500-bus systems. Simulations show accurate two-way information flow with improved computational complexity.

CHAPTER FOUR

CREATING THE DATASET

In this manuscript, two different datasets are created: a dataset for the passive grids without DG and a dataset for the active grids with DG. In each case, the dataset dictates the variations in the states of the distribution grid based on variations in generation and load. We use the 18, 85, and 141-bus cases from MATPOWER 7.1 [44].

For the case without DG, Step 1 is to vary the load at different buses and solve the AC power flow problem for each load variation. The load is varied between 50% and 200% of the original MATPOWER load values. To do that, we run nested *for* loops to change the load. A fragment of the code for the 18-bus system is presented below:

```
for PD2_Value = 0.2:0.1:0.4 % 3 values
    mpc.bus(2, PD) = PD2_Value;
    mpc.bus(2, QD) = 0.6*PD2_Value;

for PD4_Value = 0.5:0.5:1.5 % 3 values
    mpc.bus(4, PD) = PD4_Value;

for PD5_Value = 0:0.2:0.4 % 3 values
    mpc.bus(5, PD) = PD5_Value;
    mpc.bus(5, QD) = 0.625*PD5_Value;

for PD6_Value = 0.4:0.2:0.8 % 3 values
    mpc.bus(6, PD) = PD6_Value;
    mpc.bus(6, QD) = 0.625*PD6_Value;

for PD7_Value = 0.2:0.1:0.4 % 3 values
    mpc.bus(7, PD) = PD7_Value;
    mpc.bus(7, QD) = 0.6*PD7_Value;

for PD9_Value = 0.3:0.15:0.6 % 3 values
    mpc.bus(9, PD) = PD9_Value;

for PD11_Value = 0.3:0.15:0.6 % 3 values
    mpc.bus(11, PD) = PD11_Value;

for PD12_Value = 0.1:0.1:0.3 % 3 values
    mpc.bus(12, PD) = PD12_Value;

for PD14_Value = 0.2:0.15:0.5 % 3 values
    mpc.bus(14, PD) = PD14_Value;

for PD16_Value = 0.2:0.15:0.5 % 3 values
    mpc.bus(16, PD) = PD16_Value;
    mpc.bus(16, QD) = 0.6*PD16_Value;
```

```

for PD17_Value = 0.5:0.5:1.5 % 3 values
a = a + 1;
mpc.bus(17, PD) = PD17_Value;
results = runpf(mpc);

```

The default Newton Raphson algorithm is used to solve the AC power flow, by calling the runpf command. There are 11 *for* loops for the 18-bus system, with 3 values each, that is we have $3^{11} = 177,147$ cases.

In Step 2, we check for convergence at every data instance: if the algorithm converges, then the result is saved, else it is discarded. This would result in a number of data instances that is, for the 18-bus system, less than 177,147, depending on how many cases converged. We save these data instances in the following *csv* files:

- Bus voltage magnitudes
- Bus voltage angles
- Generator real power injections
- Generator reactive power injections
- Real power injected into “from” end of branch
- Reactive power injected into “from” end of branch

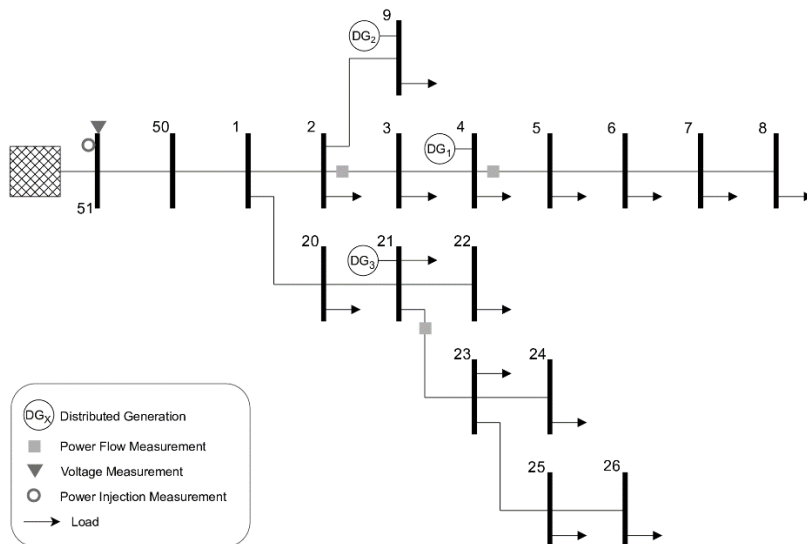


Figure 2 - Measurement Distribution for the 18-bus System.

For the case with DG, we modify approximately 15% of the buses of each grid, by making them generators, in addition to varying the loads. For example, for the 18-bus system, we add a generator with automatic voltage regulation on bus 21 and two constant active and reactive power generators, to emulate Photovoltaic (PV) generators, on buses 4 and 9 as shown in Figure 2. We then solve the AC power flow problem and repeat Step 2. The chosen measurements for the 18-bus system are the following:

- Voltage measurement at bus: 51
- Power injection measurement at bus: 51
- Power flow measurements (from-to): 2-3, 4-5, and 21-23, which represent 20% of the power flow measurements at one side.

This measurement distribution is shown in Figure 2. We note that this measurement distribution cannot be used with conventional SE approaches, because in such approaches, the grid has to be observable.

Step 3 is to convert all the values (except the angles) to the per-unit (pu) system. In Step 4, we emulate real measurements by adding Gaussian noise to the chosen measurements, with zero-mean and the following standard deviations [39]:

- 0.004 pu for voltage measurements
- 0.01 pu for power injection measurements
- 0.008 pu for power flow measurements

In Step 5, we shuffle the data and randomly select 100,000 data points to train and test the neural network. The algorithm is summarized as a flowchart in Figure 4.

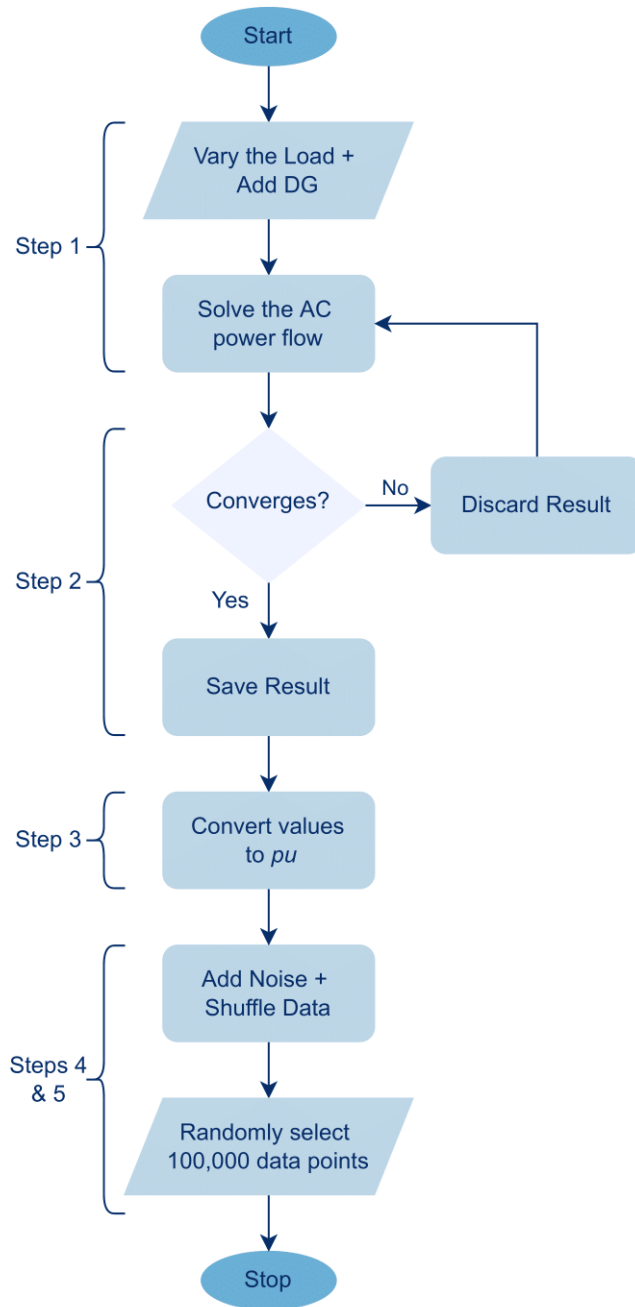


Figure 3 - Flowchart for Creating the Dataset with DG.

The same approach is repeated for the 85 and 141-bus systems, while maintaining 1 voltage measurement, 1 power injection measurement, and 20% of the power flow measurements. For the cases with DG, we also add varying generation at 15% of the buses.

CHAPTER FIVE

METHODOLOGY

In this chapter, we explain the approach that we use to solve the DSSE. We first present the neural network architecture and then the robustness tests. The whole process is summarized in the block diagram in Figure 4. It is performed for the 3 different bus systems.

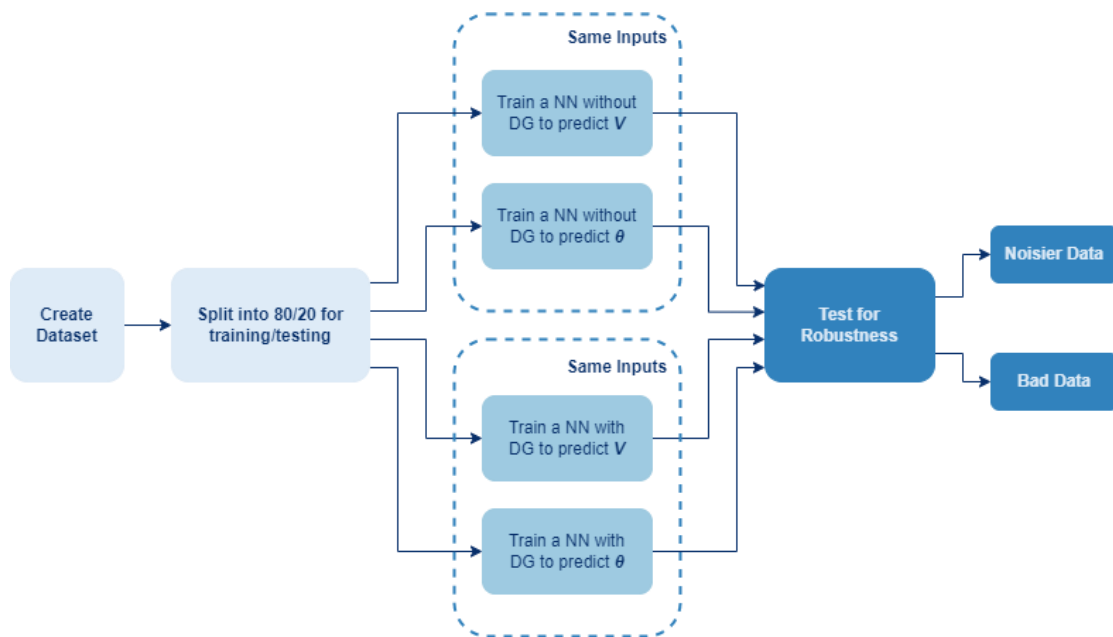


Figure 4 - Methodology Summary.

5.1 Neural Network

To build the neural network (NN), we need to select some parameters and hyperparameters such as:

1. The number of hidden layers and the number of neurons inside the hidden layers: usually, if the problem at hand is linear, one hidden layer is enough. If it is nonlinear then two and more layers would be used. We can try to increase the number of hidden layers or increase the number of neurons per layer to get more accurate results.
2. The activation function of each layer: several activation functions have been used in the literature such as *sigmoid*, *tanh*, and *ReLU*, among others. The most efficient one is the ReLU function.
 - a. Historically, the sigmoid function was one of the earliest activation functions used.

It is defined as $\sigma(u) = \frac{1}{1+e^{-u}}$. Its graph is shown in Figure 5.

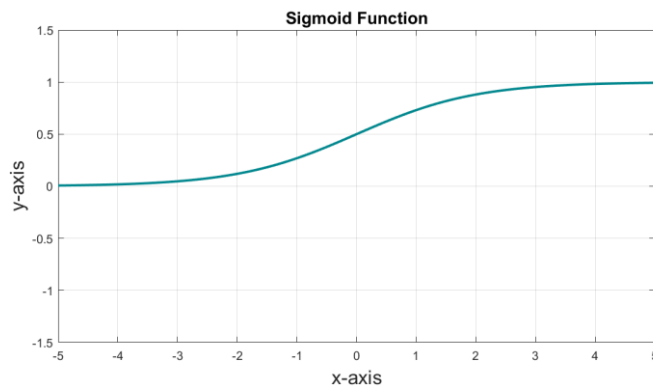


Figure 5 - Sigmoid Activation Function.

- b. Then, tanh became preferred over the sigmoid. Tanh is defined as

$\sigma(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}}$. Its graph is shown in Figure 6.

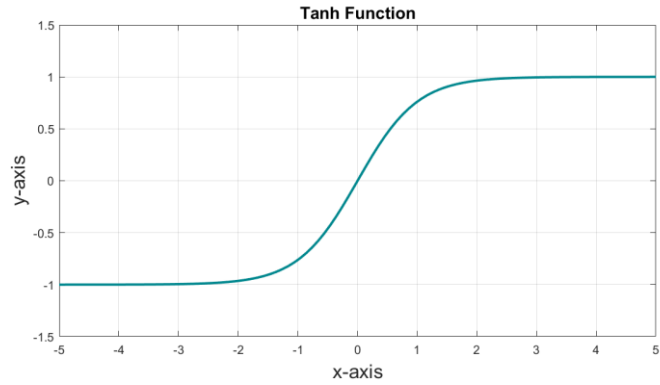


Figure 6 - Tanh Activation Function.

- c. Recently, the ReLU function is the most used activation function. It is defined as $\sigma(u) = \max(0, u)$. Its graph is shown in Figure 7. It is our selected activation function to use.

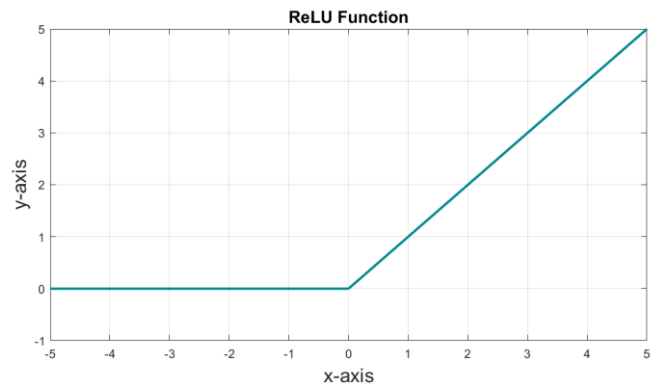


Figure 7 - ReLU Activation Function.

3. The optimizer: is a method or algorithm that is used to minimize the loss function. The optimizer usually changes the weights and biases of a neural network so the predicted output would reach the actual output as close as possible, i.e., to minimize the error. Some well-known optimizers are RMSProp, SGD, Adagrad, Adadelta, Adamax, Adam, Momentum, Nesterov Momentum, among others, each having its own advantages, disadvantages, and usages [43]. It has been shown that the Adam (or

Adaptive Moment Estimation) optimizer combines the advantages of both RMSProp and Momentum optimizers, hence it is used in this manuscript.

4. The learning rate: is a number that is usually between 0 and 1. It depicts how fast the gradient descent moves towards the global minimum. A very large learning rate leads the algorithm to oscillate around the global minimum, hence diverging; a very small learning rate might lead the algorithm to be trapped in a local minimum. After several modifications, the learning rate used for our algorithm is 0.001.
5. The number of epochs denotes the number of times the algorithm runs through the complete training dataset

The NN is now trained using the *training* dataset.

Then, we test the trained NN and check its performance. To test the algorithm, we use the training loss or error, to know the percentage of error that is found between the given testing data and the predicted data.

The above steps are iterated to find the algorithm with the best performance. That is, we check loss metrics: the training loss, the validation loss, and the testing loss. There are no strict rules that dictate how to select the above parameters and hyperparameters, but there are trial-and-error methods and best practices, found in the literature [43], to perform adjustments to the parameters. Some of these best practices are:

1. If the training loss is small, but the validation loss is large, then this is due to high variance, that is, the algorithm is overfitting or not generalizing well, then:
 - a. Check the validation set performance
 - b. Try using regularization techniques
2. If the training loss and the validation loss are both large, then the algorithm is underfitting due to high bias, then:
 - a. Get a bigger network (either collect more data, or augment them)

- b. Train for longer epochs
 - c. Try a more advanced optimization algorithm
3. If the training loss and the validation loss are both small and comparable, then the algorithm is pretty good; we have low bias and low variance.

On another note, we can follow simple rules such as:

1. If the training loss is not satisfactory:
 - a. Try a bigger network (more neurons per layer or more layers)
 - b. Try a different optimization algorithm
 - c. Train for longer epochs
2. If the validation loss is not satisfactory:
 - a. Try regularization
 - b. Use a bigger training set
 - c. Hyperparameter search
3. If testing loss is not satisfactory
 - a. Get a bigger validation set

The validation split that is selected in the code is 20%, i.e., 20% of the training dataset is used as a validation dataset.

In this paper, we use a deep feedforward NN that consists of one input layer, 4 hidden layers, and one output layer. For every grid topology, we build a separate NN:

1. NN to predict V for a grid without DG
2. NN to predict θ for a grid without DG
3. NN to predict V for a grid with DG
4. NN to predict θ for a grid with DG

In every case, the inputs are m measurements, which range from z_1 to z_m , depending on the size of the grid. The outputs are $n - 1$ state variables (voltage magnitudes and phase angles), n being the number of buses. We predict $n - 1$ variables because the slack bus is pre-determined, then its voltage magnitude and phase angle are already known thus, there is no need to include them in the NN to be predicted. The NN architecture is shown in Figure 8.

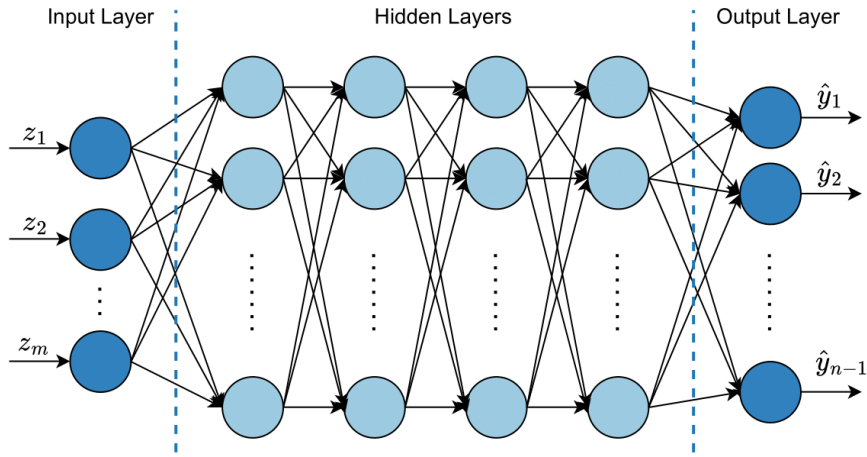


Figure 8 - Neural Network Architecture.

For every NN, we split the data into 80%, called *Training* dataset, and 20%, called *Testing* dataset. We note that, instead of predicting the angles, we predict the sine of the angles, i.e., $\sin \theta$. This step can be considered as a pre-processing step, to scale the output angles between -1 and 1 . The angles can be recovered in post-processing by performing an inverse sine, since the sign is preserved. The error (or loss function) used with most regression networks is Mean Squared Error (MSE) or Mean Absolute Error (MAE). In this manuscript, we use the MSE given by the following formula:

$$MSE = \frac{1}{n} \sum_{i=1}^n (\mathbf{y}_i - \hat{\mathbf{y}}_i)^2 \quad (12)$$

\mathbf{y}_i is the ground truth vector. It is \mathbf{V} for the first and third NNs and $\boldsymbol{\theta}$ for the second and fourth NNs;

$\hat{\mathbf{y}}_i$ is the predicted vector. It is $\hat{\mathbf{V}}$ for the first and third NNs and $\hat{\boldsymbol{\theta}}$ for the second and fourth NNs.

5.2 Testing the Robustness of the Neural Network

After the NN learns the mapping between the input measurements and the output states, we perform two test scenarios to verify its robustness.

5.2.1 Scenario 1 - Noisier Data

We assume that we have measurements with higher noise due to communication anomalies or extreme weather events. The measurement distribution is the same one provided in Chapter Four, but we add more noise in three steps: 2σ , 3σ , and 4σ . We add noise on all the data points of the *Testing* dataset, which corresponds to 20,000 data points, then, based on these noisier measurements, we evaluate the performance of the previously trained NN, without retraining it.

5.2.2 Scenario 2 - Bad Data

We assume that we have bad data due to meter errors or communication noise. The measurement distribution is the same one provided in Chapter Four, but we further corrupt the data by adding 10σ in three stages: we assume first that 10% of the measurements are bad data, followed by 20% and then 30%. This is done on the *Testing* dataset, which corresponds to 20,000 data points, to emulate bad measurements. Then, based on these measurements, we evaluate the performance of the previously trained NN, without retraining it.

CHAPTER SIX

RESULTS AND DISCUSSION

6.1 Software and Equipment

The following software, packages, and platforms were used to write the codes and perform simulations for this thesis.

MATPOWER [44] is a free package that is used with MATLAB for solving optimal power flow and power flow problems. It contains case files that are transmission or distribution grids with different number of buses. MATLAB R2021b was used with the academic license. MATPOWER was used to create the datasets.

Google Colaboratory, or *Colab* for short, is a virtual environment where codes can be written and simulated in the browser without any configurations. A *Colab* notebook is stored in a Google Drive account and easily shareable. GPUs and TPUs can be used to accelerate the simulation performance [45]. *Colab* was used to create and simulate the neural networks.

Tensorflow [46] is an end-to-end Python-based platform that is used to build and deploy machine learning and deep learning projects. The neural network simulations are performed using the TensorFlow [46] platform.

Pandapower [47] is a Python-based power system analysis tool for solving power flow, optimal power flow, and state estimation, among others. Pandapower was used to execute the conventional SE approach.

6.2 Cases Without DG

6.2.1 18-bus System Demo

For the 18-bus system without DG, we build a Neural Network (NN) having \mathbf{z}_{18bus} inputs, in accordance with the measurement distribution presented in Chapter Four. The \mathbf{z}_{18bus} vector comprises V_{51} , P_{51} , Q_{51} , P_{2-3} , Q_{2-3} , P_{4-5} , Q_{4-5} , P_{21-23} , and Q_{21-23} . The NN consists of 4 hidden layers, having 16 neurons each. The Rectified Linear Unit (ReLU) is used as an activation function. The output layer consists of $n - 1 = 17$ voltage magnitudes, with no activation function. We calculate 17 voltage magnitudes because the magnitude of the slack bus, in this case bus 51, is known beforehand. The NN is trained with a validation split of 20%, using the Adam optimizer, having a learning rate of $1e - 03$, over 200 epochs. After several simulations, the average testing error is $6.83e - 07$, as shown in Table 1. The average simulation time to calculate one data point is $64 ms$, also shown in Table 1.

Table 1 - Simulation Results for Voltage Magnitudes and Phase Angles Without DG for the 3 Test Cases.

Case	Variable	MSE	Time (in ms)
18-bus	\mathbf{V}	6.83e-07	64
	$\boldsymbol{\theta}$	1.63e-07	63
85-bus	\mathbf{V}	2.52e-07	64
	$\boldsymbol{\theta}$	1.11e-08	67
141-bus	\mathbf{V}	1.90e-07	68
	$\boldsymbol{\theta}$	4.23e-08	66

The same NN architecture is used to calculate the sine of the phase angles, except the NN now is trained for different outputs. The same measurements are used as inputs. The results are also summarized in Table 1.

For the 18-bus case, the NN is compared with conventional SE method. We use the SE module of Pandapower [47], which in turn uses the WLS method. To perform the WLS method, we provide the following measurements:

- Voltage measurement at bus: 51; this is similar to the one chosen for the NN.
- Power injection measurement at bus: 51; this is similar to the one chosen for the NN.
- 100% of the power flow measurements, as opposed to the case of the NN, where we choose 20% of the power flow measurements. With 20% only, the grid is unobservable thus, WLS method cannot be performed. Therefore, we increase the number of measurements to make the grid observable.

We refer to these measurements as $\mathbf{z}_{18busWLS}$. Similar to the NN, Gaussian noise is used with zero-mean, and the same variances provided in Chapter Four. We also do not use any pseudo-measurements in the case of WLS. As for the NN, we provide the algorithm with the measurements listed as \mathbf{z}_{18bus} , for the same data instance. We compare the MSE of the WLS method with the one of the NN method, for the same data instant. Results are summarized in Table 2.

Table 2 - Comparing WLS to NN for the 18-bus System

Case	Variable	WLS	Proposed Method
With DG	V	1.81e-05	8.27e-07
	θ	1.03e-05	2.68e-07
Without DG	V	1.96e-05	7.61e-07
	θ	1.01e-05	1.49e-07

Table 2 shows that our approach performs significantly better than the classic SE, based on WLS method, in all cases. We note that despite unobservability, our proposed approach

achieves better results than the conventional SE method, in which the grid is fully observable.

Figures 9 and 10 show the graphical representation of the errors of the estimated states with respect to the true system states, for one data instance of the 18-bus system. The graphs show that the states estimated by the NN closely follow the actual states.

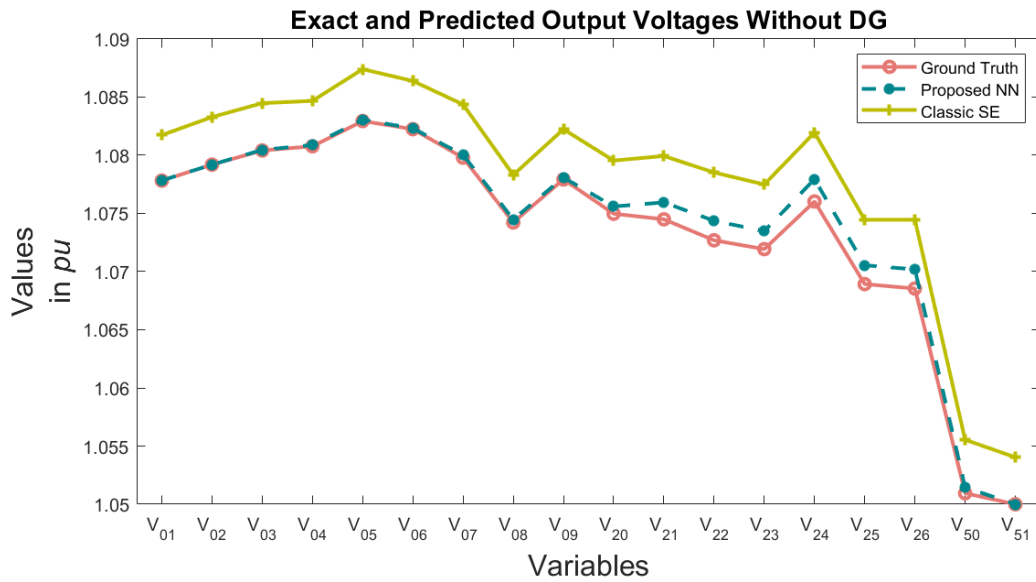


Figure 9 - Simulation Results for Predicting Voltage Magnitudes for the 18-bus System Without DG.

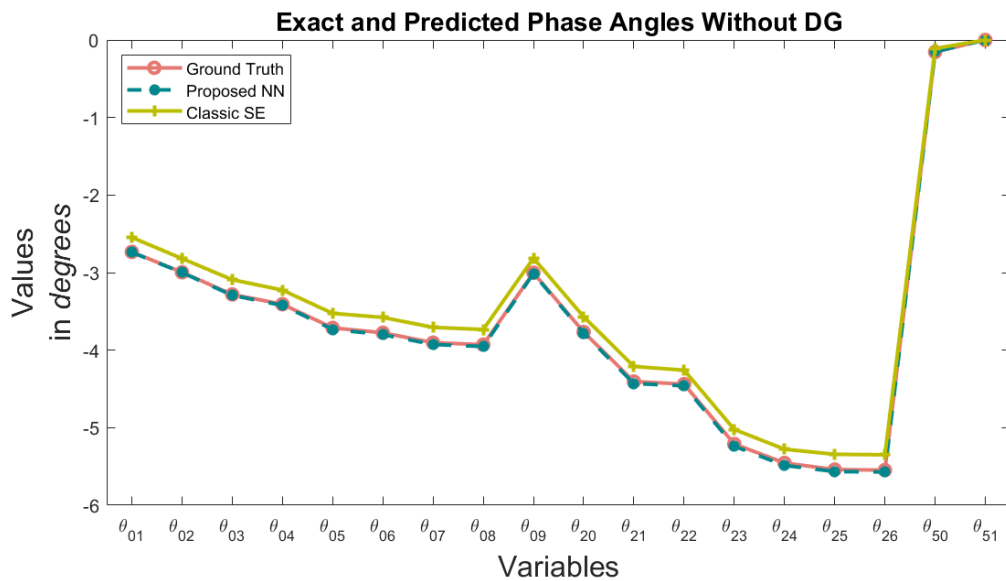


Figure 10 - Simulation Results for Predicting Phase Angles for the 18-bus System Without DG.

6.2.2 The Remaining Cases

For the 85-bus system, we also use a NN with 4 hidden layers, but this time, every layer has 64 neurons. The rest of the parameters are the same as the ones for the 18-bus system. Also, one NN is trained for predicting voltage magnitudes and another is trained for predicting phase angles. The results are summarized in Table 1.

For the 141-bus case, the hidden layers are also 4, but with 128 neurons each. This is done to compensate for the larger dimensionality of the bigger grids. The results are also summarized in Table 1.

Table 1 shows that our approach performs very well, under noisy measurements and unobservability. We note that conventional SE methods cannot be used with similar inputs. In addition, the NN shifts the computational burden to an offline stage. Thus, we evaluate the simulation time when this algorithm is used online to calculate a single output. In every case, the simulation time spent to calculate one data instance is in the milliseconds range. Moreover, even with larger grid topologies, the NN is achieving impressive results, in terms of accuracy and simulation time.

6.3 Cases with DG

6.3.1 18-bus System Demo

For the 18-bus system with DG, the same NN architecture is adopted, albeit with different input and output values. The inputs are represented by the \mathbf{z}_{18bus} vector and the outputs are the voltage magnitudes for the first NN, and the phase angles for the second NN, but the values are updated by adding DG to the grid as mentioned in Chapter Four. Results are shown in Table 3.

Table 3 - Simulation Results for Voltage Magnitudes and Phase Angles with DG for the 3 Test Cases.

Case	Variable	MSE	Time (in ms)
18-bus	V	5.17e-07	67
	θ	1.96e-07	62
85-bus	V	3.57e-07	64
	θ	3.97e-08	65
141-bus	V	7.39e-07	67
	θ	6.42e-08	68

Moreover, we compare our proposed approach with the conventional SE method.

We use the same measurement distributions presented before: one data instance of the \mathbf{z}_{18bus} vector for the NN method, and one data instance of $\mathbf{z}_{18busWLS}$ for the WLS method, to make the grid observable. Results are shown in Table 2.

Figures 11 and 12 show the graphical representation of the errors of the estimated states with respect to the true system states, for one data instance of the 18-bus case. The graphs show that the states estimated by the NN closely follow the actual states.

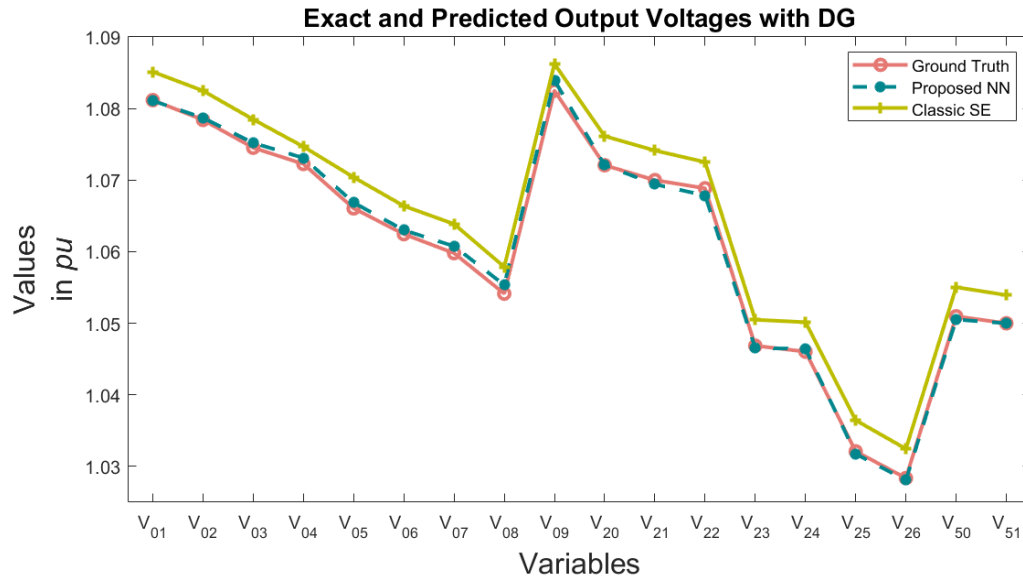


Figure 11 - Simulation Results for Predicting Voltage Magnitudes for the 18-bus System with DG.

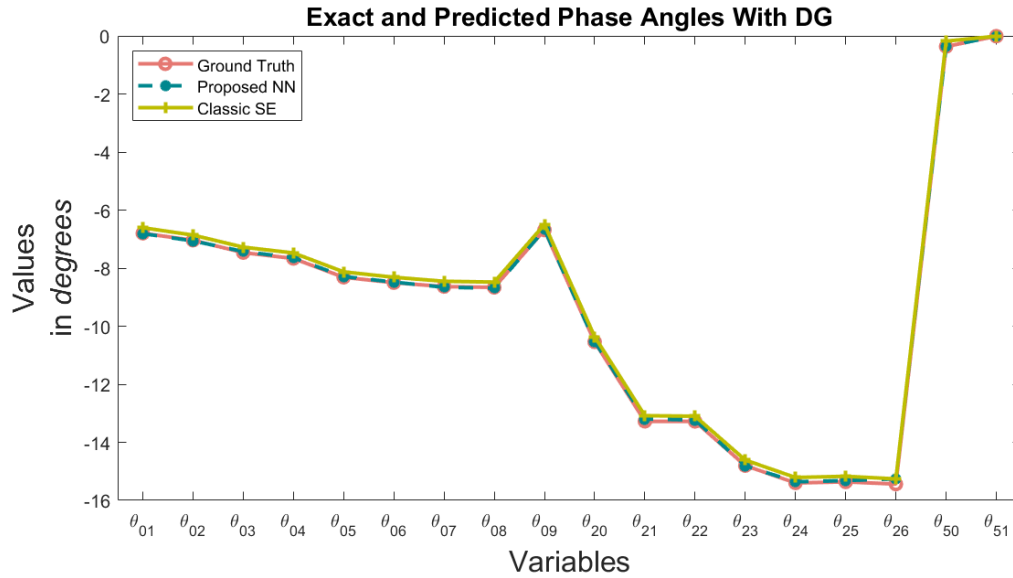


Figure 12 - Simulation Results for Predicting Phase Angles for the 18-bus System with DG.

6.3.2 The Remaining Cases

For the 85 and the 141-bus systems, we also use the same NN architectures with the input and output values changed, to dictate DG. Also, one NN is trained for predicting voltage magnitudes and another is trained for predicting phase angles. The results are summarized in Table 3.

Similar conclusions can be drawn for the cases with DG. Despite the added complexity, the proposed approach is performing very well in terms of accuracy and simulation time.

6.4 Noisier Data

For this scenario, given the trained NN, we test the algorithm using noisier data. That is, we add 2σ , 3σ , and 4σ to the measurements, available in the *Testing* dataset, and check the performance of the NN without retraining it. The results for the cases without DG and with DG are summarized in Tables 4 and 5, respectively. Results show that the

proposed approach can still be used with noisier measurements, despite being scarce. Also, the accuracy remains higher than conventional SE method which requires more data.

Table 4 - Noisier Data Results for Cases Without DG

Case	Variable	Original MSE	MSE with 2σ	MSE with 3σ	MSE with 4σ
18-bus	V	6.83e-07	7.81e-07	1.06e-06	1.53e-06
	θ	1.63e-07	1.96e-07	2.88e-07	4.27e-06
85-bus	V	2.52e-07	2.79e-07	3.61e-07	4.92e-07
	θ	1.11e-08	2.70e-08	3.43e-08	6.27e-08
141-bus	V	1.90e-07	1.96e-07	2.10e-07	2.35e-07
	θ	4.23e-08	5.11e-08	7.27e-08	1.53e-07

Table 5 - Noisier Data Results for Cases with DG

Case	Variable	Original MSE	MSE with 2σ	MSE with 3σ	MSE with 4σ
18-bus	V	5.17e-07	6.42e-07	1.01e-06	1.65e-06
	θ	1.96e-07	2.36e-07	3.43e-07	5.43e-07
85-bus	V	3.57e-07	3.60e-07	3.67e-07	3.91e-07
	θ	3.97e-08	4.40e-08	5.93e-08	1.04e-07
141-bus	V	7.39e-07	7.88e-07	8.24e-07	9.01e-07
	θ	6.42e-08	6.86e-08	8.54e-08	1.05e-07

6.5 Bad Data

For this scenario, we further corrupt the data by adding 10σ to the *Testing* dataset in three stages: we assume first that 10% of the measurements are bad data, followed by 20% and then 30%. We test the NN without retraining it. The results for the case without DG and with DG are summarized in Tables 6 and 7, respectively. Results show that the

performance remains higher than the conventional SE method, while 10% of the measurements are bad data, but as expected, starts to decline when the percentage of bad data increases to 20% and 30%.

Table 6 - Bad Data Results for Cases Without DG

Case	Variable	Original MSE	MSE with 10% Bad Data	MSE with 20% Bad Data	MSE with 30% Bad Data
18-bus	V	6.83e-07	1.78e-06	3.13e-05	3.20e-05
	θ	1.63e-07	2.12e-06	6.62e-05	1.84e-04
85-bus	V	2.52e-07	7.24e-06	3.97e-05	7.47e-05
	θ	1.11e-08	1.52e-06	1.02e-05	5.71e-04
141-bus	V	1.90e-07	4.98e-06	2.28e-05	7.54e-04
	θ	4.23e-08	2.14e-06	6.12e-05	4.37e-04

Table 7 - Bad Data Results for Cases with DG

Case	Variable	Original MSE	MSE with 10% Bad Data	MSE with 20% Bad Data	MSE with 30% Bad Data
18-bus	V	5.17e-07	4.73e-06	2.80e-05	3.51e-04
	θ	1.96e-07	2.78e-06	2.73e-05	2.38e-05
85-bus	V	3.57e-07	1.14e-06	5.79e-05	3.64e-04
	θ	3.97e-08	6.39e-06	3.21e-05	6.72e-05
141-bus	V	7.39e-07	2.31e-06	2.95e-05	6.11e-04
	θ	6.42e-08	3.16e-06	1.60e-05	3.52e-04

6.6 Discussion

- Our proposed approach solves the DSSE problem with high accuracy, despite the grid being highly unobservable, as shown in Tables 1 and 3. We recall that we only have 1 voltage injection measurement, 1 power injection measurement, and 20% of the power flow measurements. We do not use any pseudo-measurements. Moreover, the simulation time is in the order of milliseconds, even for the large 141-bus grid, which would enable the operators to frequently solve the DSSE problem. This approach is suitable for grids with and without DG.
- We compare our proposed approach with the WLS method, as shown in Table 2 and Figures 9, 10, 11, and 12. We show that the proposed method is more accurate despite having less inputs than WLS. Also, the WLS method cannot be performed using the same number of measurements that was provided to the NN.
- Tables 4 and 5 show that the NN is robust to additional noise. This is attributed to the fact that the NN was trained on a large dataset, making it more immune to noise.
- Tables 6 and 7 show that the NN still has good accuracy, with 10% of bad data. The performance, however, starts declining as more data are corrupted.

CHAPTER SEVEN

CONCLUSION AND FUTURE WORK

DSSE is still a challenging problem due to the nature of distribution grids.

Conventional SE methods, which are used for transmission SE, cannot be used for DSSE mainly because distribution grids are unobservable. Conventional SE methods are also computationally expensive, sensitive to initial conditions, may not always converge, might get stuck in local optima, and use highly inaccurate pseudo-measurements to compensate for the unobservability issue.

Neural networks, on the other hand, shift the computational complexity to an offline stage, are immune to noise, and can handle large networks if used online. Also, a NN does not require the grid to be observable and can easily work with the addition of renewable energy resources.

In this paper, we propose a deep neural network approach to solve the DSSE problem, with and without DG. Due to the lack of public frameworks, we create a dataset to train and test the neural network. We also check its robustness by testing several scenarios with noisier data and bad data. Our proposed approach has a better performance than conventional SE methods, even with less measurements, without using pseudo-measurements. The algorithms can also perform fast to cater to the need of frequently solving the DSSE problem.

Future work directions can be to design a similar approach to incorporate dynamic state estimation, on the distribution level, and to consider the unbalanced distribution grids.

REFERENCES

- [1] I. Roytelman and S. Shahidehpour, "State estimation for electric power distribution systems in quasi real-time conditions," *IEEE Transactions on Power Delivery*, vol. 8, no. 4, pp. 2009–2015, 1993.
- [2] M. E. Baran and A. W. Kelley, "State estimation for real-time monitoring of distribution systems," *IEEE Transactions on Power systems*, vol. 9, no. 3, pp. 1601–1609, 1994.
- [3] C. Lu, J. Teng, and W.-H. Liu, "Distribution system state estimation," *IEEE Transactions on Power systems*, vol. 10, no. 1, pp. 229–240, 1995.
- [4] K. Li, "State estimation for power distribution system and measurement impacts," *IEEE Transactions on Power Systems*, vol. 11, no. 2, pp. 911–916, 1996.
- [5] Y. Deng, Y. He, and B. Zhang, "A branch-estimation-based state estimation method for radial distribution systems," *IEEE Transactions on power delivery*, vol. 17, no. 4, pp. 1057–1062, 2002.
- [6] F. Therrien, I. Kocar, and J. Jatskevich, "A unified distribution system state estimator using the concept of augmented matrices," *IEEE Transactions on Power Systems*, vol. 28, no. 3, pp. 3390–3400, 2013.
- [7] M. E. Baran and A. W. Kelley, "A branch-current-based state estimation method for distribution systems," *IEEE transactions on power systems*, vol. 10, no. 1, pp. 483–491, 1995.
- [8] H. Wang and N. N. Schulz, "A revised branch current-based distribution system state estimation algorithm and meter placement impact," *IEEE Transactions on Power Systems*, vol. 19, no. 1, pp. 207–213, 2004.

- [9] M. Pau, P. A. Pegoraro, and S. Sulis, "Efficient branch-current-based distribution system state estimation including synchronized measurements," *IEEE Transactions on Instrumentation and Measurement*, vol. 62, no. 9, pp. 2419–2429, 2013.
- [10] M. Göl and A. Abur, "Lav based robust state estimation for systems measured by pmus," *IEEE Transactions on Smart Grid*, vol. 5, no. 4, pp. 1808–1814, 2014.
- [11] J. Zhao, G. Zhang, M. La Scala, and Z. Wang, "Enhanced robustness of state estimator to bad data processing through multi-innovation analysis," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 1610–1619, 2016.
- [12] K. Dehghanpour, Z. Wang, J. Wang, Y. Yuan, and F. Bu, "A survey on state estimation techniques and challenges in smart distribution systems," *IEEE Transactions on Smart Grid*, vol. 10, no. 2, pp. 2312–2322, 2018.
- [13] A. K. Ghosh, D. L. Lubkeman, and R. H. Jones, "Load modeling for distribution circuit state estimation," *IEEE Transactions on Power Delivery*, vol. 12, no. 2, pp. 999–1005, 1997.
- [14] R. Singh, B. C. Pal, and R. A. Jabr, "Statistical representation of distribution system loads using gaussian mixture model," *IEEE Transactions on Power Systems*, vol. 25, no. 1, pp. 29–37, 2009.
- [15] D. T. Nguyen, "Modeling load uncertainty in distribution network monitoring," *IEEE Transactions on Power Systems*, vol. 30, no. 5, pp. 2321–2328, 2014.
- [16] C. Muscas, M. Pau, P. A. Pegoraro, and S. Sulis, "Effects of measurements and pseudomeasurements correlation in distribution system state estimation," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 12, pp. 2813–2823, 2014.

- [17] A. Arefi, G. Ledwich, and B. Behi, "An efficient dse using conditional multivariate complex gaussian distribution," *IEEE Transactions on Smart Grid*, vol. 6, no. 4, pp. 2147–2156, 2015.
- [18] J. Zhao, G. Zhang, Z. Y. Dong, and M. La Scala, "Robust forecasting aided power system state estimation considering state correlations," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 2658–2666, 2016.
- [19] R. Dutta, S. J. Geetha, S. Chakrabarti, and A. Sharma, "An l-1 regularized forecasting-aided state estimator for active distribution networks," *IEEE Transactions on Smart Grid*, vol. 13, no. 1, pp. 191–201, 2021.
- [20] J. Liu, F. Ponci, A. Monti, C. Muscas, P. A. Pegoraro, and S. Sulis, "Optimal meter placement for robust measurement systems in active distribution grids," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 5, pp. 1096–1105, 2014.
- [21] A. Angioni, T. Schlösser, F. Ponci, and A. Monti, "Impact of pseudo-measurements from new power profiles on state estimation in low-voltage grids," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 1, pp. 70–77, 2015.
- [22] H. Z. Margossian, "Iterative state estimation with weight tuning and pseudo-measurement generation," *IEEE Systems Journal*, vol. 15, no. 3, pp. 3165–3172, 2021.
- [23] Z. Fang, Y. Lin, S. Song, C. Li, X. Lin, and Y. Chen, "State estimation for situational awareness of active distribution system with photovoltaic power plants," *IEEE Transactions on Smart Grid*, vol. 12, no. 1, pp. 239–250, 2020.
- [24] J. Liu, R. Singh, and B. C. Pal, "Distribution system state estimation with high penetration of demand response enabled loads," *IEEE Transactions on Power Systems*, vol. 36, no. 4, pp. 3093–3104, 2021.

- [25] H. M. Merrill and F. C. Schweppe, "Bad data suppression in power system static state estimation," *IEEE Transactions on Power Apparatus and Systems*, no. 6, pp. 2718–2725, 1971.
- [26] E. Handschin, F. C. Schweppe, J. Kohlas, and A. Fiechter, "Bad data analysis for power system state estimation," *IEEE Transactions on Power Apparatus and Systems*, vol. 94, no. 2, pp. 329–337, 1975.
- [27] A. Monticelli and A. Garcia, "Reliable bad data processing for real-time state estimation," *IEEE transactions on power apparatus and systems*, no. 5, pp. 1126–1139, 1983.
- [28] E. Caro, A. J. Conejo, and R. Minguez, "Decentralized state estimation and bad measurement identification: An efficient lagrangian relaxation approach," *IEEE Transactions on Power Systems*, vol. 26, no. 4, pp. 2500–2508, 2011.
- [29] V. Kekatos and G. B. Giannakis, "Distributed robust power system state estimation," *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 1617–1626, 2012.
- [30] J. Zhao, L. Mili, and R. C. Pires, "Statistical and numerical robust state estimator for heavily loaded power systems," *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 6904–6914, 2018.
- [31] C. H. Ho, H. Wu, S. Chan, and Y. Hou, "A robust statistical approach to distributed power system state estimation with bad data," *IEEE Transactions on Smart Grid*, vol. 11, no. 1, pp. 517–527, 2019.
- [32] C. Lin, W. Wu, and Y. Guo, "Decentralized robust state estimation of active distribution grids incorporating microgrids based on pmu measurements," *IEEE Transactions on Smart Grid*, vol. 11, no. 1, pp. 810–820, 2019.

- [33] A. S. Zamzam, X. Fu, and N. D. Sidiropoulos, "Data-driven learning-based optimization for distribution system state estimation," *IEEE Transactions on Power Systems*, vol. 34, no. 6, pp. 4796–4805, 2019.
- [34] E. Manitsas, R. Singh, B. C. Pal, and G. Strbac, "Distribution system state estimation using an artificial neural network approach for pseudo measurement modeling," *IEEE Transactions on power systems*, vol. 27, no. 4, pp. 1888–1896, 2012.
- [35] B. P. Hayes, J. K. Gruber, and M. Prodanovic, "A closed-loop state estimation tool for mv network monitoring and operation," *IEEE Transactions on Smart Grid*, vol. 6, no. 4, pp. 2116–2125, 2014.
- [36] K. R. Mestav, J. Luengo-Rozas, and L. Tong, "Bayesian state estimation for unobservable distribution systems via deep learning," *IEEE Transactions on Power Systems*, vol. 34, no. 6, pp. 4910–4920, 2019.
- [37] P. Sundaray and Y. Weng, "Alternative auto-encoder for state estimation in distribution systems with unobservability," *IEEE Transactions on Smart Grid*, 2022.
- [38] A. S. Zamzam and N. D. Sidiropoulos, "Physics-aware neural networks for distribution system state estimation," *IEEE Transactions on Power Systems*, vol. 35, no. 6, pp. 4347–4356, 2020.
- [39] A. Abur and A. G. Exposito, *Power system state estimation: theory and implementation*. CRC press, 2004.
- [40] A. Monticelli, *State estimation in electric power systems: a generalized approach*. Springer Science & Business Media, 2012.
- [41] F. C. Schweppe and J. Wildes, "Power system static-state estimation, part i: Exact model," *IEEE Transactions on Power Apparatus and systems*, no. 1, pp. 120–125, 1970.

- [42] A. M. Law, W. D. Kelton, and W. D. Kelton, *Simulation modeling and analysis*. Mcgraw-hill New York, 2007, vol. 3.
- [43] C. A. Charu, “Neural networks and deep learning: a textbook,” 2018.
- [44] R. Zimmerman and C. Murillo-Sanchez, “Matpower (version 7.1)(2020).”
- [45] “Colaboratory.” [Online]. Available: <https://research.google.com/colaboratory/faq.html>
- [46] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015, software available from tensorflow.org. [Online]. Available: <https://www.tensorflow.org/>
- [47] L. Thurner, A. Scheidler, F. Schäfer, J.-H. Menke, J. Dollichon, F. Meier, S. Meinecke, and M. Braun, “pandapower—an open-source python tool for convenient modeling, analysis, and optimization of electric power systems,” *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 6510–6521, 2018.