Laboratoire des Signaux et Systèmes – GeePs –
CNRS - Centrale Supélec - Université Paris Saclay

TECHNICAL REPORT

Low-Voltage Distribution System State Estimation via Machine Learning Techniques

Supervised by: Dr. ALESSIO IOVINE, L2S
Dr. TRUNG DUNG LE, GeePs
Mr. ELIO EL SEMAAN, GeePs & L2S

Implemented by: Mr. DAT TIEN NGUYEN

Gif-sur-Yvette, 28 April 2022

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1 INTRODUCTION

1.1 CONTEXT AND MOTIVATION

In power systems, State Estimation (SE) consists in applying techniques to estimate system's state to serve for the real-time monitoring operations of power grid [1][2]. It plays an important role in multiple purposes of automation and control including Volt/VAr optimization, feeder reconfiguration and restoration and also in network 'forensics' tasks like bad measurements detection gross modeling error identification [1]. To achieve these objectives, physics laws combined with measurements such as nodal voltages, injections, line flows are utilized for the estimation of state variables including normally bus voltage magnitude, voltage angle, active power, reactive power, current magnitude and power factor [1].

However, considering the availability and observability of measurement devices which are feasible in Transmission Network (TN) where all can be installed in the buses, those of Distribution Network (DN) are often scarcer because of the exponential increment of the number of buses from the substations to the consuming sides, putting pressure on the practical and economic considerations [1][3]. Additionally, DS operates under Low-Voltage (LV) level, causing limitations for measurement units' operation [3]. Also, the values received from them have to cope with poor readability issues rooted from the different measured results under the same transformer [3]. Moreover, the increasing penetration of electrical vehicles (EVs) and distributed energy resources (DERs) like renewable energy also makes the state of buses in the DS more dynamic, unstable and uncertain [2].

In this context, the term Distribution System State Estimation (DSSE) appears to be highly crucial, which can be considered as a mapping process using a limited number of measured values to infer the state variables of the system for a large quantity of non-measured buses [4]. The scarcity of the real-time measurements in monitoring are often compensated by pseudo-measurements which are important to enable DSSE [1][3]. The high variance and low accuracy of pseudo-measurements can influence detrimentally on the reliability of DSSE [3].

Regarding metering instruments, Phasor Measurement Units (PMU), Supervisory Control and Data Acquisition systems (SCADA) and Smart Meters (SM) are usual measuring technologies utilized to capture input data to cope with the lack of information, especially in Low-Voltage level

[5]. SM considers the measurement of load profile normally presented in graph which demonstrates the power consumed on each time step interval (30minutes, 1 hour, …) in a day of a customer [3]. Particularly, the micro-PMU is considered as the highest accuracy method despite having the high cost of installation [6]. Technically, the limitation of bandwidth, infrequent delays and synchronization problems in communication prevent the large number of buses with high unobservability in Distribution System from grasping all the data at the same time; therefore, the installation of these measuring devices in every bus is considered to be practically unfeasible [3].

In the past 30 years, the adoption of Machine Learning techniques in the power systems is under a slow progress despite the large number of academic researches. This is because the power domain acquires high requirements regarding safety criteria in energy, health care or automobiles, which are obliged to be considered under an insightful understanding about the problem and the underlying formulas. Machine Learning, however, is trained to optimize and capture the equations demonstrating the relationships of features, which is hidden under a black box and highly risks this understanding [8]. As mentioned in the literatures, ML appears to be highly potential to cope with the significantly increasing complexity of power systems and the high uncertainty with integration of renewable energy and electrical devices [2]. This is because ML is capable of outperforming the conventional techniques by fastening the decision-making process and capturing complicated relationships through the continuous learning and adaptation to the environment [8].

In this research, the available approaches for DSSE including the conventional method Weighted Least Square (WLS) and the Machine Learning-based ones: Feed Forward Neural Network (FFNN), Linear Regression (LR), Support Vector Machine (SVM), Auto-Encoder (AE), Deep Neural Network (DNN), Swallow Neural Network (SNN) and Physics-Aware Neural Network (PAWNN) are analyzed in terms of literature review to grasp their general ideas in order to evaluate and consider which ones should be tested and applied for the studied case. In machine learning techniques, Artificial Neural Network-based (ANN) methods (FFNN, DNN, SNN) are inspired from learning weights, bias and activation parameters on each constructed node in the neuron to capture the non-linear pattern connecting the input and desired output features [1][3]. Besides, there are several works on linear relation including LR and SVM by learning only the weights and biases [3]. Lastly, auto-encoder (AE) consists in reconstructing and fitting the missing values based on extracting the context of the original pattern on an encoding and decoding symmetry structure [5][7].

Regarding the tested phases in the considered articles, only SNN and PAWNN are indicated to be tested under 3-phase consideration [1][6], the remaining methods consider only a single phase assuming the similarity and balance in the state among all the phases.

1.2 OBJECTIVES OF THE REPORT

The objectives of this report can be divided into 2 main sections. The first one gives the literature review about the context of distribution system from which raises the need and importance of State Estimation compared to that of the transmission system. Subsequently, several details of the issue regarding measuring instruments and required input parameters for SE are also introduced and indicated before demonstrating specifically the theoretical idea and performance of the current approaches in comparison with the conventional method. These evaluations facilitate the selection of the potential approaches used for the implementation section afterwards. Also, the data measurement, simulation and generation process are considered and discussed as the beginning to further grasp the idea of possible input features for training section.

The second one concerns the implementation of the selected method by specifying the principles of model including structure, training method and estimation approach. After that, the application of hyperparameters is introduced aiming to automate the selection of training parameters before mentioning the metrics and visualization methods used to evaluate the model's performance which are tested with different categories of bus system, topology's complexity and amount of missing data. Lastly, the results and the proposed improvements of the testing strategies are demonstrated in a simple grid and, especially in a realistic case to verify the model's adaptability on situations in reality.

CHAPTER 2. CURRENT METHODS FOR DISTRIBUTION SYSTEM STATE ESTIMATION

2.1 WEIGHTED LEAST SQUARES

Concerning the parameterized notations in state measurement, the true state vector $\boldsymbol{x}$ (size $n \times 1$) is mapped into measurement vector $\boldsymbol{z}$ (size $m \times 1$), captured and recorded by measuring instruments, by a measurement function $\boldsymbol{h}$ and measurement error $\boldsymbol{e}$ [4]. In particular, the measurement vector is the value which is captured and recorded by installed measuring instruments [4]. The measurement function $\boldsymbol{h}$ depends on the type of the device introducing whether linear or quadratic relation [1]. Lastly, error respects the Normal Distribution with zero mean [1][4]:

$$\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{e} \tag{1}$$

Under the assumption of the normal distribution for measurements, Weighted Least Squares (WLS) is mainly used as a conventional approach for SE. The algorithm aims at finding an estimated argument as a state vector in a maximum likelihood problem [4]:

$$\hat{\boldsymbol{x}} = arg_{\boldsymbol{x}} \min \left(\boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x})\right)^{T} \boldsymbol{W}(\boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x})) \tag{2}$$

Accordingly, the error vectors inffered from the substraction in the equation (2) under transposed and untransposed form are represented with the weight matrix $\boldsymbol{W}$ identifying the confidence of user in the measured values [4]. The selection of the weight matrix varies with respect to different approaches of the conventional method. The usual form of this considers the variance (squared of standard deviation) of the vector $\boldsymbol{z}$ along its m measurements as diagonal parameters assuming that the statistical dependance of measurement errors exists in different measurements [4]:

$$\boldsymbol{W} = diag\{\sigma_1{}^{-2}, \dots, \sigma_m{}^{-2}\} \tag{3}$$

To enhance the accuracy of WLS, the weight matrix is modified and added by modeling the correlation between 2 different non-diagonal terms among variables measured by SM or PMUs method, namely active power (P), reactive power (Q), voltage magnitude (V), current magnitude (I), power factor (cos (ϕ)) [4]:

$$\sigma_{P,V} = \sigma_V{}^2 I \cos(\phi), \quad \sigma_{V.Q} = \sigma_V{}^2 I \sin(\phi)$$

$$\sigma_{P,Q} = \sigma_V{}^2 I^2 \sin(2\phi) - \sigma_\phi{}^2 I^2\, Q^2 \sin(2\phi) + \sigma_I{}^2\, V^2 \sin(2\phi) \tag{4}$$

In another approach, to cope with the issue of the conventional Weighted Least Square (WLS) which is the high sensitivity to bad data measurements, different thresholds are set using the residual level corresponding to each of m recorded samples [4]. Therefore, to reduce the unreliable impact of bad data, a quantity evaluating this issue D'_i is considered to make comparisons with those levels so that the diagonal values of weight function are modified to be reduced by a factor $\boldsymbol{\zeta_i}$or diminished to be 0 [4]:

$$\overline{w_\iota} = \begin{cases} \sigma_i^{-2}, & D'_i \le k_0 \\ \sigma_i^{-2}\zeta_i, & k_0 < D'_i \le k_1 \\ 0, & D'_i \ge k_1 \end{cases} \tag{5}$$

In order to deal with an optimization problem stated in equation (1), Gauss-Newton can be used as a method to solve the gradient of its objective function J by iteratively updating the estimated state variable which takes into account the Jacobian H (derivative of J with respect to the state variables) and the system gain matrix G [4]. After k iterations, the process is represented by:

$$\begin{gathered} \boldsymbol{H}\big(\boldsymbol{x}(k)\big) = \frac{\partial J}{\partial \boldsymbol{x}(\boldsymbol{k})} \\ \boldsymbol{G}(k) = \boldsymbol{H}\big(\boldsymbol{x}(k)\big)^T \boldsymbol{W}\boldsymbol{H}\big(\boldsymbol{x}(k)\big) \\ \Delta\boldsymbol{x}(k) = \boldsymbol{G}(k)^{-1}\boldsymbol{H}\big(\boldsymbol{x}(k)\big)^T \boldsymbol{W}\big(\boldsymbol{z} - \boldsymbol{h}\big(\boldsymbol{x}(k)\big)\big) \\ \boldsymbol{x}(k+1) = \boldsymbol{x}(k) + \Delta\boldsymbol{x}(k) \end{gathered} \tag{6}$$

The Weighted Least Square is known as a widely used method for SE which is also fast and simple to approach. However, its drawback is the high unreliablity against bad data measurements [4]. To solve this issue, several alterative DSSE structures were also introduced with different considerations of objective function to be more robust against outliers or leverage points such as Least Median Square (LMS), Least Trimmed Squares (LTS) or to remove automatically bad data as in the Least Absolute Value (LAV) method [4]. However, the trade-offs in terms of the algorithm performance regarding high computational cost or memory requirement are also needed to be considered when considering a real-time DSSE [4].

Also, its solution using Gaussian-Newton method introduces a non-convex function which possibly gives multiple local minima, especially when dealing with a limited number of measurements of the usual DSSE context [1]. Besides, Gaussian-Newton appears to be highly dependent on initialization; therefore, the issues regarding multiple iterations needed or convergence failure can be the consequence [1].

2.2 SUPERVISED LEARNING FOR CONSECUTIVE BUSES

In [3], three unsupervised learning approaches (Feed Forward Neural Network (FFNN), Linear Regression (LR) and Support Vector Machine (SVM)) are presented to estimate the state including voltage magnitude and voltage angle of a large number of consecutive buses with missing data given a limited number of available measured buses. By doing this, data measurement of all buses is unnecessary for these methods, which is one of the big advantages of this work.

2.2.1 Data Generation

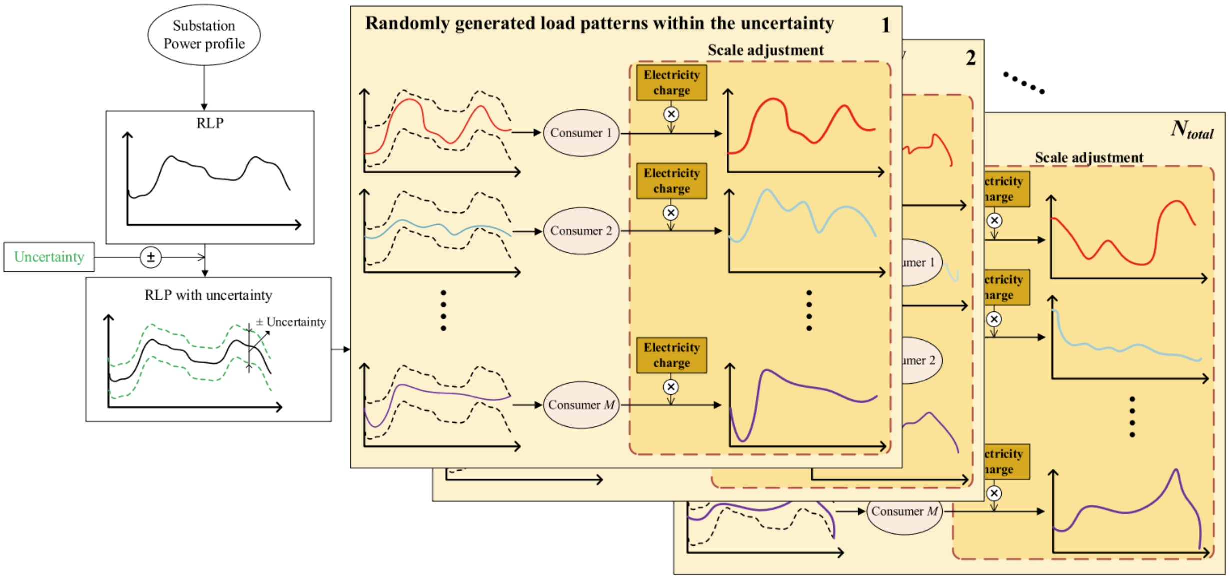


Figure 1: *Generation of N datasets for daily load (t) of M customers with the addition of uncertainty intervals.*

In this paper, a n-th training set in N total datasets for time t in a day for the m-th in M customers is generated from Representative Load Profile (RLP) calculated from the data regarding the power curve of the substation. RLP demonstrates the demand pattern of a single or a group of customers over a period of time and it is normally utilized to capture pseudo-measurements without metering instruments [3].

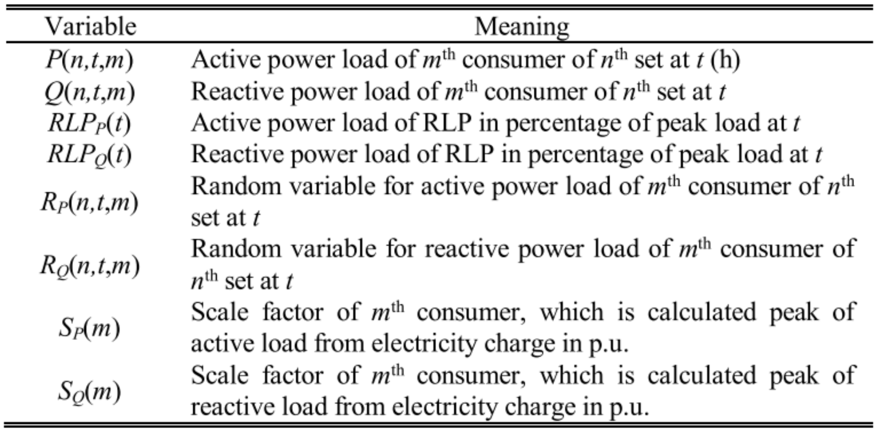
To cope with high variance of load pattern in the demanding side, the RLP is added with an uncertainty value depending on the uniformly distributed random percentage (ranging from

10% to 50% and spaced by 10%) of the measured power profile. This allows the evaluation of recommended models on all possible case scenarios of error and low accuracy obtained in reality [3]. By doing this, the data profile prepared for the training section includes Active/Reactive power, voltage magnitude and voltage angle:

$$P(n,t,m) = RLP_P(t) \times \left(1 + R_P(n,t,m) \times S_P(m)\right) \tag{7}$$

$$Q(n,t,m) = RLP_Q(t) \times \left(1 + R_Q(n,t,m) \times S_Q(m)\right)$$

Table 1: *Parameters for load profile generation.*

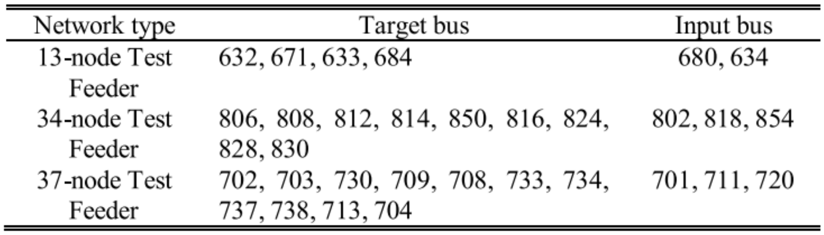


Variable	Meaning
$P(n,t,m)$	Active power load of m^{th} consumer of n^{th} set at t (h)
$Q(n,t,m)$	Reactive power load of m^{th} consumer of n^{th} set at t
$RLP_P(t)$	Active power load of RLP in percentage of peak load at t
$RLP_Q(t)$	Reactive power load of RLP in percentage of peak load at t
$R_P(n,t,m)$	Random variable for active power load of m^{th} consumer of n^{th} set at t
$R_Q(n,t,m)$	Random variable for reactive power load of m^{th} consumer of n^{th} set at t
$S_P(m)$	Scale factor of m^{th} consumer, which is calculated peak of active load from electricity charge in p.u.
$S_Q(m)$	Scale factor of m^{th} consumer, which is calculated peak of reactive load from electricity charge in p.u.

2.2.2 Study Cases and Machine Learning Models

In this case, FFNN is used to learn the non-linear relationship between the input and output through activation functions after transmitting forward the information through layers of nodes using the dot product with the learning weights. The model is learnt gradually through updating the weights, biases and activation parameters in optimization problem. Compared with the non-linear relation of FFNN, that of the input and target features in LR method is formed and trained by a linear straight line. Finally, the more complex input-output connection of ambiguous data relations is expected to be handled by SVM by finding a hyper plane in a large dimensional space to categorize the data based on margin maximization.

Table 2: *3 study cases with increasing complexity and target bus number.*



Network type	Target bus	Input bus
13-node Test Feeder	632, 671, 633, 684	680, 634
34-node Test Feeder	806, 808, 812, 814, 850, 816, 824, 828, 830	802, 818, 854
37-node Test Feeder	702, 703, 730, 709, 708, 733, 734, 737, 738, 713, 704	701, 711, 720

.

Concerning the study case, the author used 3 network structures based on IEEE standard with the increasing number of target bus with missing data for SE. As can be witnessed, the target buses are determined based on the consecution with the input ones.

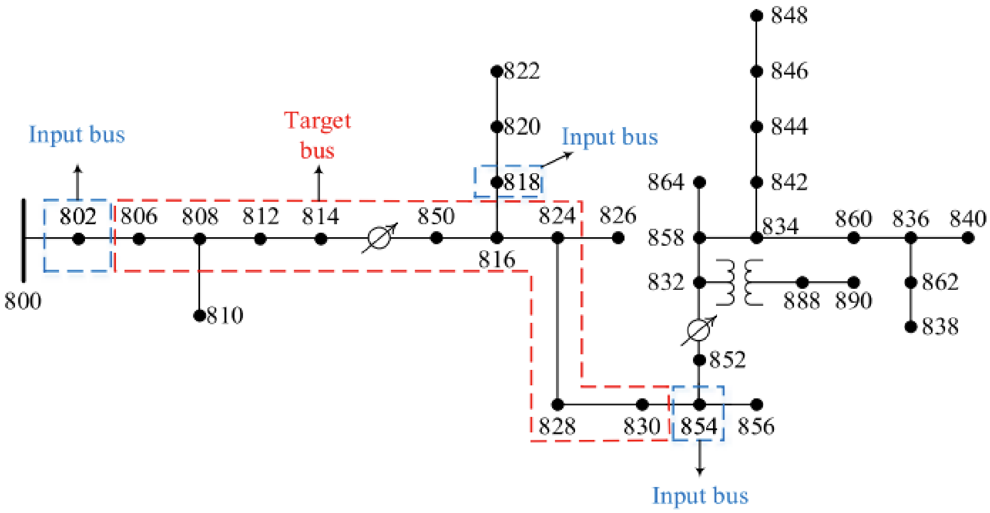


Figure 2: *Topology simulation of IEEE 34-node Test Feeder.*

Accordingly, the training concept of the approaches takes Active/ Reactive power ($\boldsymbol{P_S}, \boldsymbol{Q_S}$) of the slack bus, time measured in hour ($\boldsymbol{t}$) (aiming at evaluating the impact of the time in the day on the state of target bus) and finally the state of the input buses including voltage magnitude $\boldsymbol{V_{in}}$ and voltage angle $\boldsymbol{\theta_{in}}$. The SE variables ($\boldsymbol{V_{tar}}, \boldsymbol{\theta_{tar}}$) are the output of the model which are then used to compared with $\boldsymbol{V_{true}}$ and $\boldsymbol{\theta_{true}}$ for the model evaluation section.

Concerning the relations and impact of the parameters, the input slack bus, defined as the Reference Bus in which voltage magnitude and angular are set to 0 and 1p.u respectively as reference for all other buses in the system to balance its active and reactive power, is considered in this paper probably because it facilitates the learning task of the ML model on the connection between the considered input buses, target buses and the reference one. By doing this, the relation between them can be inferred and helps estimating the state on other real-time situations.

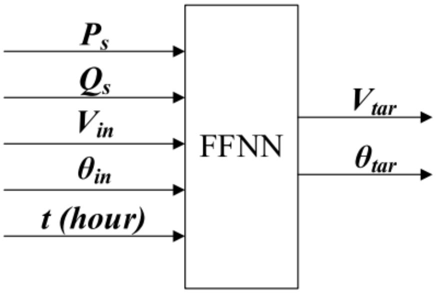


Figure 3: *Data Flow of FFNN's training concept which is the same in LR and SVM.*

2.2.3 Model Performance

For the m-th customer at n-th dataset at time t, the state estimations of Voltage magnitude and Voltage angle are used to calculate the absolute error with the true ones. The maximum values of all calculations in the set are selected for the performance comparison and evaluation of each approach for each tested case:

$$\begin{gathered}\varepsilon_{V,p.u}(m) = max\begin{pmatrix} a_{11} & \cdots & a_{1t} \\ \vdots & \ddots & \vdots \\ a_{nt} & \cdots & a_{nt} \end{pmatrix} \\ m = 1, 2, \dots, M_{tar} \\ \\ \varepsilon_{V,deg}(m) = max\begin{pmatrix} b_{11} & \cdots & b_{1t} \\ \vdots & \ddots & \vdots \\ b_{nt} & \cdots & b_{nt} \end{pmatrix} \\ m = 1, 2, \dots, M_{tar} \\ \\ a_{nt} = |V_{true}(n,t,m) - V_{est}(n,t,m)| \\ b_{nt} = |\theta_{true}(n,t,m) - \theta_{est}(n,t,m)| \end{gathered} \tag{8}$$

Table 3: *Maximum Error of 3 Machine Learning methods for Voltage Magnitude.*

Algorithm type	Network type	Maximum values of $\varepsilon_{V_{p.u.}}(m)$ in 10^{-2} p.u. for all m according to the uncertainty, m=1,2, $\cdots$, M_{tar}				
		10% uncertainty	20% uncertainty	30% uncertainty	40% uncertainty	50% uncertainty
FFNN	13-node	0.42	0.42	0.77	1.00	0.86
	34-node	0.03	0.06	0.07	0.09	0.11
	37-node	0.04	0.06	0.22	0.52	0.48
LR	13-node	0.04	0.08	0.15	0.22	0.34
	34-node	0.02	0.05	0.07	0.08	0.08
	37-node	0.02	0.05	0.07	0.07	0.14
SVM	13-node	0.12	0.12	0.13	0.17	0.29
	34-node	0.33	0.37	0.46	0.70	0.67
	37-node	0.19	0.12	0.15	0.24	0.22

The comparison table for maximum error of 3 algorithm types for 3 study cases on 5 levels of added uncertainty demonstrates that Linear Regression method obtains lowest error and only produces a slight increase with higher uncertainties due to the linear relation between the Voltage and Power.

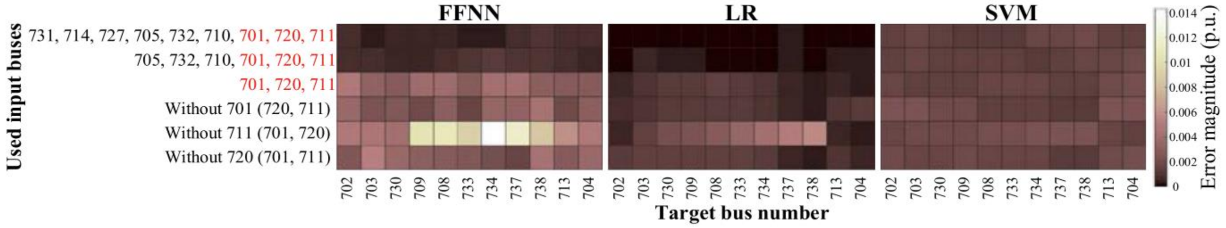


Figure 4: *Error Distribution in the case with high number of buses (37-nodes).*

One of the most qualified sections of this research is the consideration for the model performance under the impact of input buses. Accordingly, various tests are examined by modifying the data availability of different input buses to see how the models perform on estimating the state of a certain chosen set of remaining buses. By doing this, the robustness of the models on the more randomized situations of available input buses compared to that of the output ones can be evaluated. The result of the study case network with largest number of buses (37 nodes), which is more likely to exist more frequent in reality, is chosen to be analyzed in this report because it can provide multiple changing cases on the input buses. As can be seen from the Figure 4, SVM even give stable maximum error distribution for the case with high number of buses through various variations of used input created compared with the 2 other methods. This exhibits the advantage of this approach on learning the data features whose relations are not well-understood [3]. Therefore, due to the higher performance on capturing the relationship in more intricate scenarios in the availability of input data in reality, Support Vector Machine can be potential in the context of the tested case of the internship.

However, the case application of this paper is still limited with consecutive buses state estimation which can prevent the SE task from giving the guess of the voltage magnitude or voltage angle for all the missing values in a certain network. This situation can be handled by installing the measuring devices on the buses positioned at the end of each branch, which allows creating sets of consecutive buses covering all the buses in the concerning network.

2.3 CONTEXT LEARNING BY AUTO-ENCODER (AE)

2.3.1 Fundamental Ideas Behind the Application of Auto-Encoder

In the image processing, auto-encoder is normally used to identify and extract the features of the image (edges, shape, contours or color values) presented in the encoded vector so that these patterns can be transmitted and retained in the reconstructed image. This fundamental basement plays an important role in image restoration, deblurring, denoising and image generation tasks.

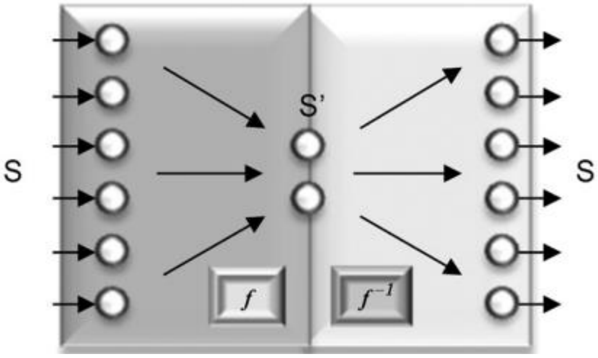


Figure 5: *Data Flow of a simple Autoencoder Neural Network [7].*

The idea of auto-encoder consists in learning a mapping function $\boldsymbol{f}$ from the original dimension of the input data $\boldsymbol{S}$ to a compressed encoding $\boldsymbol{S'}$ with lower dimension then attempting to reconstruct the initial information using the reversed function form $\boldsymbol{f^{-1}}$ as a decoder [7]. Accordingly, AE is trained so that the output is similar to the input in both value and dimension by learning and transmitting the information through the encoded layer which can be considered as a characteristic hidden vector. The weights of the model are learnt in a symmetrical architecture to be stored with data manifold [7]. For these reasons, when being applied for the DSSE, the problem can be represented under the form of missing data restoration task. This is because, similarly to the image processing, the model is capable of capturing the general patterns of the given values in the connection in relation of the context. Therefore, it is also expected to learn the electrical relationship among the nodes of the network given the available input nodes, which is then used to serve for estimating the states as the missing values in real-time application. In the more complex form of the AE applying for the case with multiple input values, the number of hidden layers can be extended to be more instead of being limited with only 1 as in the simple situation so that the characteristics of the network patterns can be learnt carefully. Besides, the dimension of the hidden layers can also be determined depending on different applications according to a reduction ratio between the lowest middle layer and the outermost one [7].

As introduced in [7], there are 3 categories of auto-encoder listed as the considerations for the task (figure 6). In particular, the first one relates to the Projection Onto Convex Sets (POCS) which converges the initially randomized missing values to the desired ones by feeding iteratively the corresponding result to the input of the missing input. Secondly, the minimization algorithm is used in unconstrained search for the difference between missing signals and their output ground-truth separately from the available ones. Lastly, all data is considered in the optimization iteration in constrained search instead of the separation in the previous method. Among these approaches, POCS is examined to be simple to be applied; however, the highest efficiency is still be witnessed in the constrained method. [7] This might be because the optimization weight is all concentrated on the missing positions instead of being equally separated for others. By doing this, the available measurements can play a role in orienting the missing ones to the objective values that give the lowest error on the reconstruction performance.

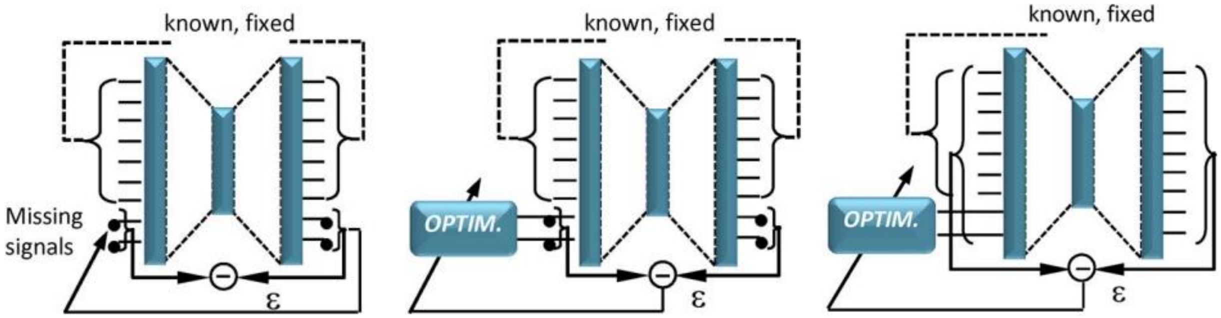


Figure 6: *3 types of Auto-Encoder for Missing Signals Reconstruction:*
a) POCS, b) constrained search, c) unconstrained search [7].

In another work, AE is slightly modified to become Extreme Learning Machine Auto-Encoder (ELM-AE) which does not run back-propagation as the conventional AE and is used for enhancing both accuracy and learning speed performance. In particular, the input values are initially standardized so that they are fit within the normalized interval [-1,1]. By doing this, it can be matched within the range of the activation function [5]. Regarding the algorithm, ELM approach only takes into account the computation of weights vector between the hidden and the output layer which is learnt through a non-iterative process to minimize both training error and the norm of output weights. The input weights and the biases of the hidden layer is randomly initialized based on Normal Distribution and are made orthogonal before being fed into the model.

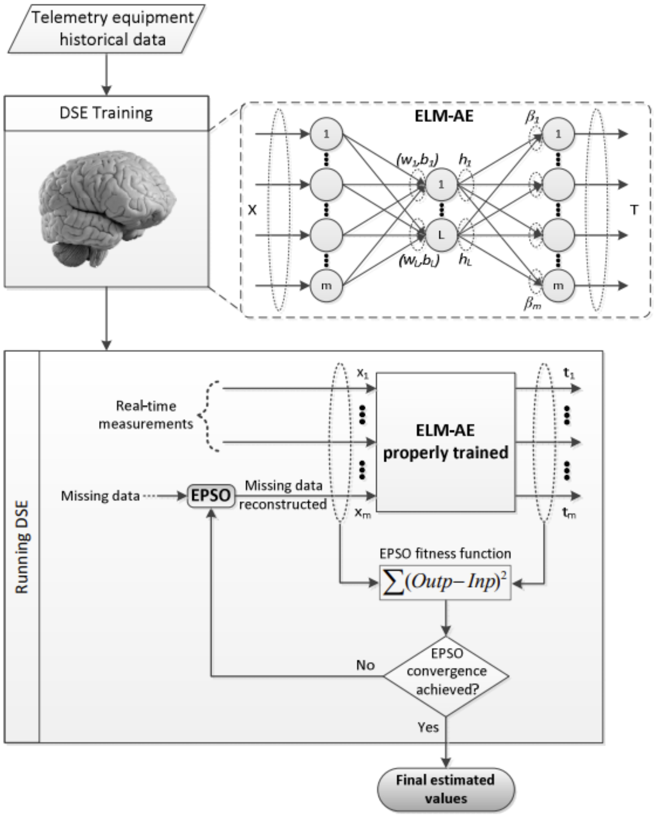


Figure 7: *2 Stages of Distribution State Estimator (DSE) [5].*

In terms of operation section, it consists of offline AE training and online DSE parts. The first one makes full use of the historical data (voltage magnitude, voltage angle, active and reactive power) of the buses to train the reconstruction ability of the model. The resulting AE representing under the trained weights and biases of layers are applied in real-time for State Estimation based on Evolutionary Particle Swarm Optimization (EPSO) technique based on a limited number of quasi real-time measurements.

2.3.2 Model Performance

The simple AE method shows a reasonable performance on estimating missing Voltage values whose absolute error mainly distributes around 0 in probability density distribution and obtains 1.95×10^{-3} as the average value [7]. This result can be rooted from the ability of the auto-encoder in capturing the underlying topology of the examined network through the general context learnt from the historical data that has been demonstrated, which can be considered as one of the important and interesting topics showing a parameter probably determining significantly the performance of DSSE and can be further discussed in the upcoming sections.

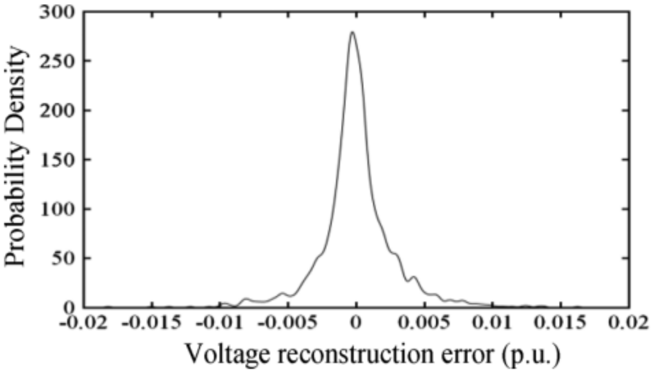
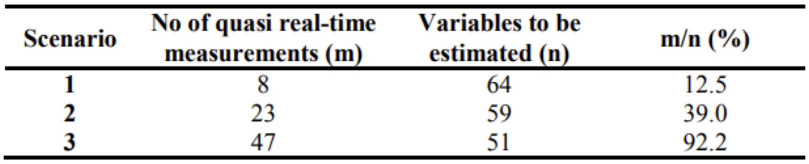


Figure 8: *Absolute Error Probability Desnity Distribution for simple AE [7].*

To further examine the performance of AE on the number of missing values, 3 scenarios are generated considering different available ratios presented in percentages (12.5, 39.0 and 92.2) considering missing measurements over the total number of buses in the tested network.

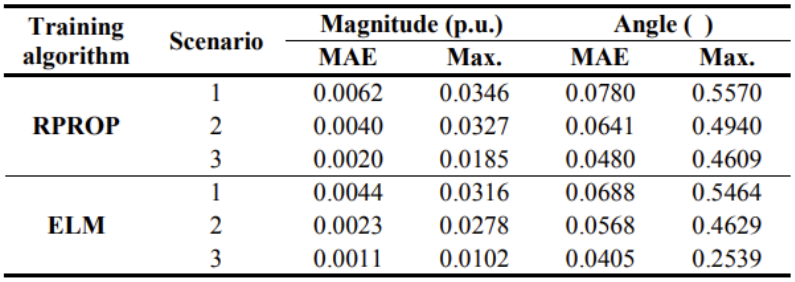
Table 4: *Quasi real-time measurement for 3 types of tested scenario for ELM-AE [7].*



Scenario	No of quasi real-time measurements (m)	Variables to be estimated (n)	m/n (%)
1	8	64	12.5
2	23	59	39.0
3	47	51	92.2

As can be seen in the *Table 5* showing the operation error on these tested cases, the average MAE of this approach is not so different compared to that of the simple AE (in the range of 1.1×10^{-3} to $4.4 \times 10^{-3}\ p.u$). However, the maximum range of this value is approximately 3 times greater than the previous method ($2.78 \times 10^{-2}\ p.u$ compared to $1.00 \times 10^{-2}\ p.u$). Additionally, the AE is still able to reconstruct and estimate missing value under the lowest measurement ratio (12.5%) with relatively acceptable average error ($4.4 \times 10^{-3}\ p.u$) with not large difference ($1.1 \times 10^{-3}\ p.u$) with respect to the low ratio situation. This shows an significant robustness performance of AE in DSSE task; however, when being applied in real cases in which large missing ratio is frequently observed, the high increase of error in guessing the state of this model should be considered.

Table 5: *State Estimation Error on ELM-AE on 3 tested scenarios [7].*



Training algorithm	Scenario	Magnitude (p.u.)		Angle ( )	
		MAE	Max.	MAE	Max.
RPROP	1	0.0062	0.0346	0.0780	0.5570
	2	0.0040	0.0327	0.0641	0.4940
	3	0.0020	0.0185	0.0480	0.4609
ELM	1	0.0044	0.0316	0.0688	0.5464
	2	0.0023	0.0278	0.0568	0.4629
	3	0.0011	0.0102	0.0405	0.2539

2.4 SHALLOW NEURAL NETWORK (SNN) FOR GAUSSIAN-NEWTON INITIALIZATION

2.4.1 Fundamental Ideas

First of all, the idea of Neural Network consists in the mapping process $\boldsymbol{F}(\cdot)$ from the initial inputs $\boldsymbol{z}$ to the desired target $\boldsymbol{v}$. Being utilized in a DSSE problem which requires the high similarity between the real states and the noise-contained measurements, the problem becomes the mapping approximation [1]:

$$\boldsymbol{F}(\boldsymbol{z}) \approx \boldsymbol{v} \tag{9}$$

Principally in a universal function approximator as NN, a simple (input, hidden, output) layer structure is able to provide an approximation for any continuous multivariate function with a prescribed accuracy [1]. The recent qualified success of this component in machine learning raises a promising potential on its application to learn offline the mapping $\boldsymbol{F}(\cdot)$ from historical measurements and then to solve the DSSE problem online [1]. In practical, a one hidden layer NN can contain multiple neurons connected and transmitting the data in t-th neuron by the parameters containing linear relation weights $\boldsymbol{w}_t$, the corresponding biases beta, the activation layer σ and its values $\boldsymbol{\alpha}_t$ to bridge the neurons' result to the NN's output $\boldsymbol{g}$ given the vector input $\boldsymbol{z}$.

$$\boldsymbol{g}_T(\boldsymbol{z}) = \sum_{t=1}^{T} \boldsymbol{\alpha}_t \sigma(\boldsymbol{w}_t^T \boldsymbol{z} + \boldsymbol{\beta}_t) \tag{10}$$

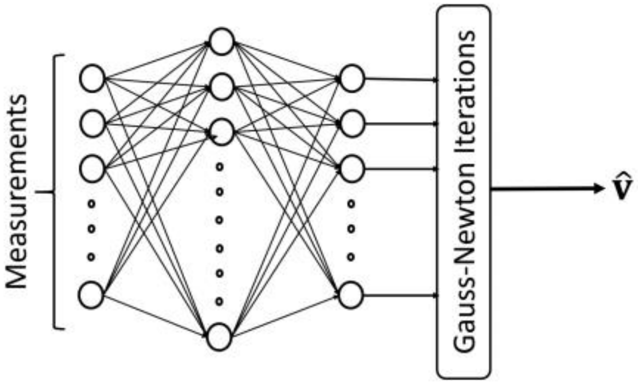


Figure 9: *SNN for Initialization of Gauss-Newton Iterations.[1]*

These parameters can be trained to be optimally fit the training pair (z^j, v^j) of input measurements and the target state through a minimization process of the cost function:

$$\min_{\{\boldsymbol{\alpha}_t, \boldsymbol{w}_t, \boldsymbol{\beta}_t\}_{t=1}^{T}} \sum_j \left\| \boldsymbol{v}^j - \boldsymbol{g}_T(\boldsymbol{z}^j) \right\|_2^2 \tag{11}$$

For a DSSE problem, it is considered to be highly ambitious, sample complicated or high complexity with large T or a deep multi-layer NN to learn an exact end-to-end mapping in this absolute mean cost function. To deal with this situation, a Shallow Neural Network is proposed to tune and only map the input measurements to a point in the neighboring area of the real state whose surrounding ball of radius is determined by ε such that $\left\| \boldsymbol{v}^j - \boldsymbol{g}_T(\boldsymbol{z}^j) \right\|_2^2 \leq \varepsilon^2$:

$$\min_{\{\boldsymbol{\alpha}_t, \boldsymbol{w}_t, \boldsymbol{\beta}_t\}_{t=1}^{T}} \sum_j max \left\{ \left\| \boldsymbol{v}^j - \boldsymbol{g}_T(\boldsymbol{z}^j) \right\|_2^2 - \varepsilon^2, 0 \right\} \tag{12}$$

2.4.2 Result and Model Performance

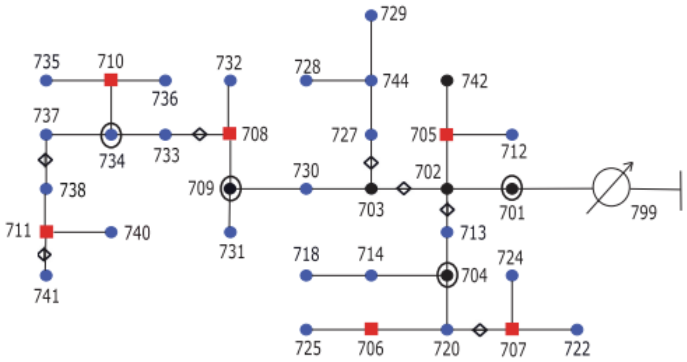


Figure 10: *IEEE-37 Distribution Feeder with Notations: Blue Circle: pseduo-measurements, Red Square: DER installation, Black Circled: PMUs, rhombus links: Current measurement.[1]*

The case study is selected to be implemented in IEEE-37 Distribution Feeder with 4 types of measurement devices installed at certain pre-defined buses and connecting lines. Particularly, PMU is marked at black Circled buses for voltage phasors measurement, pseduo-measurement in normal blue buses for active and reactive aggregate load demand estimation using historical and situational data and finally DER pseduo-measurement with integration of renewable energy sources. [1]

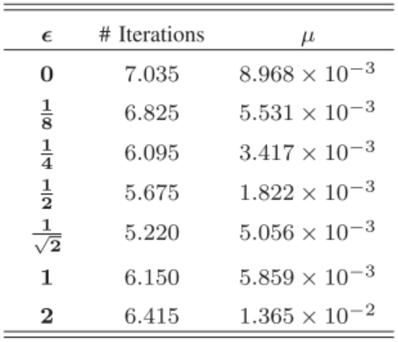
There are 2 methods for model performance evaluation. The first approach uses Root Mean Square Error (RMSE) for the training section with generated historical data, which is used to identify the lowest-training-error neighboring radius. The second one considers the Mean Square Error (MSE) between the estimated state and the real one which is more crucial to evaluate the performance of the trained model for the DSSE in other testing situations. [1]

$$\nu = \|\hat{\boldsymbol{v}} - \boldsymbol{v}^{true}\|_2^2$$

$$\mu = \sum_{l=1}^{L} (\boldsymbol{z}_l - \boldsymbol{h}_l(\hat{\boldsymbol{v}}))^2 \tag{13}$$

As can be seen in *Table 5*, the neighboring radius $1/2$ obtains the lowest error of $1.822 \times 10^{-3}\ p.u$ while still gives a reasonably low number of training iterations. Therefore, this number should be eventually selected for the implementation of SNN for the later results.

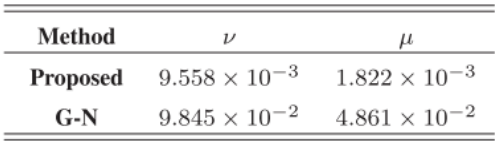
Table 6: *Performance with Different Radius of Ball ε [1].*



ϵ	# Iterations	μ
$\mathbf{0}$	7.035	8.968×10^{-3}
$\mathbf{\frac{1}{8}}$	6.825	5.531×10^{-3}
$\mathbf{\frac{1}{4}}$	6.095	3.417×10^{-3}
$\mathbf{\frac{1}{2}}$	5.675	1.822×10^{-3}
$\mathbf{\frac{1}{\sqrt{2}}}$	5.220	5.056×10^{-3}
$\mathbf{1}$	6.150	5.859×10^{-3}
$\mathbf{2}$	6.415	1.365×10^{-2}

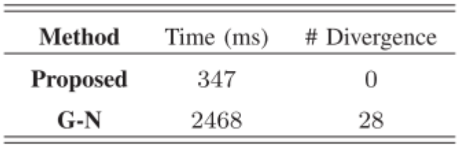
Accordingly, in comparison with the conventional Gaussian-Newton measure, SNN-GN obtains approximately 10 times lower estimation error ($9.558 \times 10^{-3}\ p.u$) and even exhibits 6 times faster in training time without any case of divergence, facilitating the application and calculation in quasi-realtime DSSE. However, when compared to Auto-Encoder method with MAE in the previous section, this approach gives poorer performance with about 10 times greater.

Table 7: *Performance Comparison of SNN-GN with Gaussian-Newton [1].*



Method	ν	μ
Proposed	9.558×10^{-3}	1.822×10^{-3}
G-N	9.845×10^{-2}	4.861×10^{-2}

Table 8: *Timing and Divergence Comparison of SNN-GN with Gaussian-Newton [1].*



Method	Time (ms)	# Divergence
Proposed	347	0
G-N	2468	28

2.5 DEEP NEURAL NETWORK (DNN)

2.5.1 Fundamental Ideas

Similar to SNN, DNN structure is constructed from multiple neurons to map the input features to the target ones. However, DNN is stacked by multiple layers instead of only 1 or 2 as presented in the shallow version so that these added layers permit a considerably fewer number of parameters, which can operate an exponential increase in SNN under a growing number of input features, to be applied more robustly in large distribution system [2]. By doing this, DNN is proposed to learn the non-linear relationship between measurements and states regardless the size of the system as the initialization for faster convergence in the forward/backward sweep SE [2]. Regarding the input features, the examined work mentioned SCADA, PMUs and SM measurements including customers' load profile, sub-station level voltage, power injection, real and reactive power flow. The implementation scheme of this paper also consists of offline training section and the online application one, which is the similar to that of AE and SNN.

Since being stacked with multiple hidden layers, the data is transmitted forwardly layer after layer and each of them is normally constructed by a predefined fully-connected layer, whose parameters are all connected by weights with those of the previous one, for linear relation and then followed by an activation layer with a similar number of neurons to capture non-linear relationship [2]. One of the renown activators is ReLU which filters only positive values and attaches $0s$ to the remaining ones:

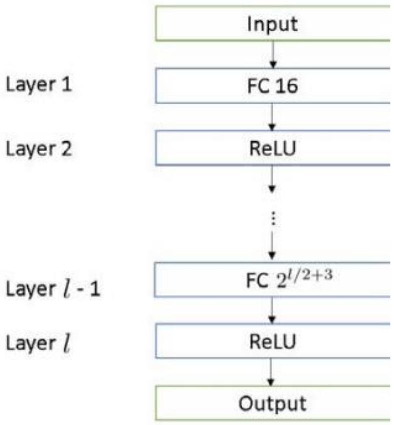


Figure 11: *Data Flow in a Structured DNN with multiple FC-ReLU layers. [2]*

2.5.2 Model Performance

The comparison between SNN and DNN is empathized in the evaluation of model performance for 123-bus and 8500-node structure as small and large distribution system respectively with the corresponding constructed DNN parameters:

Table 9: *DNN Structure for Different Test Feeders.*

System	123-bus	8500-node
Input size	190	2276
hidden layers l	10	14
Output size	494	15108

Instead of MSE evaluating the distance of the estimations (voltage magnitude in $p.u$ and phase angle in $degree$) from the ground-truth ones as in the previous approaches, RMSE is used for taking into account the influence of outliers as significantly high error results. As can be seen in the *Table 9*, DNN outperforms SNN-GN for both considered state variables, especially when being applied for the complex distribution system with 8500 nodes, DNN even shows a remarkable result with only 1.00×10^{-3} $p.u$. This can be explained by the DNN's capability to learn carefully through stacked deep hidden layers the relationship of the input highly dimensional features to predict significantly closed to the desired state values.

Table 10: *State Estimation Errors on Voltage Magnitude (upper) & phase angle (lower).*

Method	123-bus	8500-node
DNN	7.73×10^{-4}	1.00×10^{-3}
NN	2.05×10^{-3}	1.63×10^{-2}

Method	123-bus	8500-node
DNN	5.29×10^{-2}	8.66×10^{-2}
NN	1.25×10^{-1}	6.86×10^{-1}

Figure 12 demonstrates more clearly this performance by the specific estimation comparison on 19 Tested Instances. As mentioned in the previous part, SNN is used only for the convergence in the neighboring region of the target point, this proves how significantly its estimations are skewed from the true values even in the small distribution system (123-bus tested case). On the contrary, DNN acquires a quasi-closed distance with the ground-truth state especially in the complicated tested system 8500-bus.

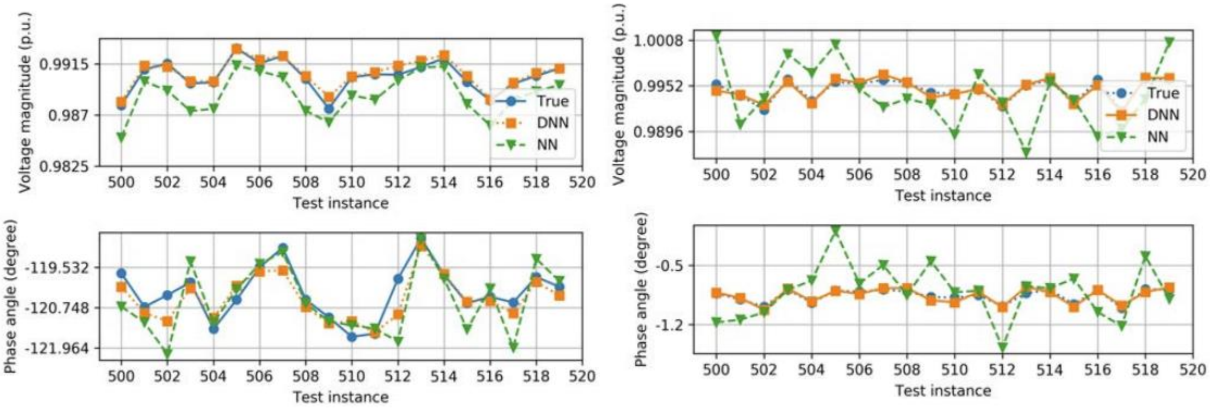


Figure 12: *SE Error in 123-bus (left) and 8500-node (right) for 19 Tested Instances [2].*

However, to express the trade-off this method to obtain these outperformed results when compared to the others, the deep learning technique require multiple layers which is equivalent to a large number of training parameters. Therefore, when considering this approach to be used in reality, the issue regarding the model's cost should also be considered. This will be further discussed in the later sections.

2.6 PHYSICS AWARE NEURAL NETWORK (PAWNN)

2.6.1 Fundamental Ideas

Considering the mentioned Machine Learning-based methods, the underlying physical connections (topology) between the node are overlooked as the concerning parameters impacting on the changes of the output result [6]. As a result, the mapping from the input measurement and the desired state is over-parameterized. This means the ML training process only takes into account the relationship of the input parameters and gradually forms the relations of the unconnected nodes, increasing the amount of needed training parameters.

Being inspired by this awareness, the structure of physics-aware neural network (PAWNN) method is modified by pruning the links of constructed multiple neuron layers which is positionally equivalent to the underlying physical structure of the considering distribution network [6]. Generally considering a NN consisting of multiple layers, the output formula of the t-th layer can be expressed under the relations of linear weight W_t, bias β_t and the non-linear activation σ_t:

$$h_{t+1} = \sigma_t(W_t h_t + \beta_t) \tag{14}$$

Accordingly, taking an example of a 2-layer NN as shown in *Figure 13 -a*, the model output y can be calculated from the input x consisting of 6 parameters as 6 nodes in the graph:

$$y = \sigma_2(W_2 \sigma_1(W_1 x + \beta_1) + \beta_2) \tag{15}$$

Particularly, W_t is a $N \times N$ sized weight matrix corresponding to the relationship among N input buses. Therefore, the tuple indexing location (i, j) of the weight W_t matrix is pruned if there is no direct connection between x_i and x_j. As a result, this can produce a sparse weight matrix, reducing the number of training parameters for the Machine Learning model, as exhibited in the *Figure 13 -b* as an example. Concerning the neuron connections in the NN structure, each multiplication by weight, addition by bias and non-linear activation generates a link between the neurons of hidden layers. Therefore, a normal NN can produce highly intricate connections from each neuron of a layer to every neuron of the next one. By pruning the irrelevant nodes and sparifying the weight matrices of hidden layers, the connections only exist in the relevant ones directly interconnected in the distribution system (*Figure 13 -c*).

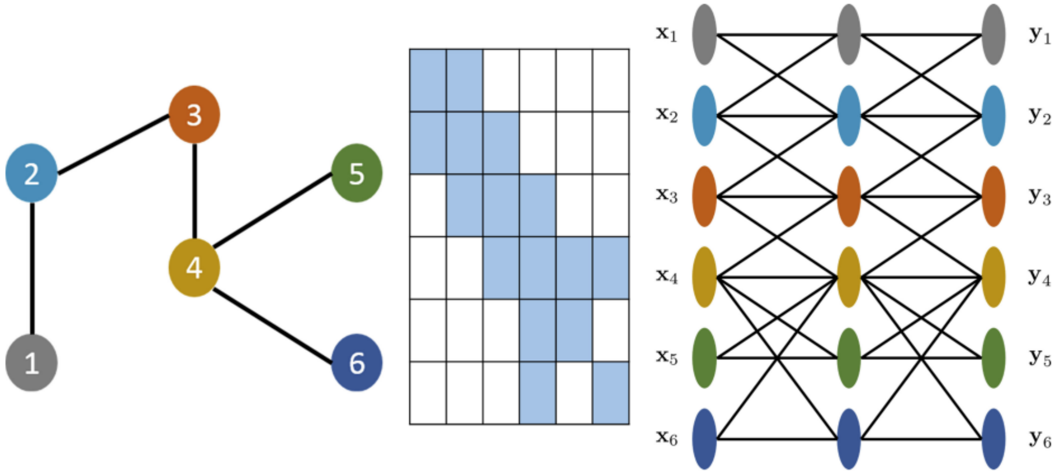
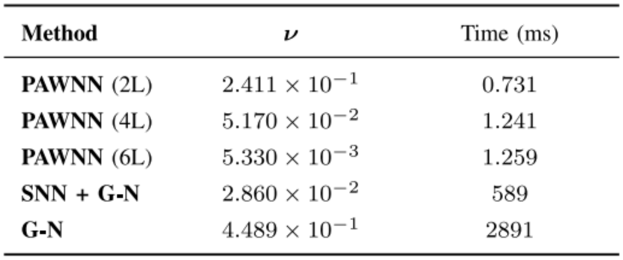


Figure 13: *a) An example 6-node graph. b) The sparsity of the training weights. c) 2 layers graph pruned NN. [6]*

2.6.2 Model Performance

The PAWNN is tested with different numbers of layers in the same network scenario as the previous SNN-GN method shown in *Figure 10*. As can be seen in *Table 11*, the operation time of PAWNN is significantly lower than the SNN-GN and the conventional G-N method because of the substantial reduction in number of training weights. Moreover, 6-layer PAWNN gives highest performance with only $5.330 \times 10^{-3}\ p.u$, which is about 10 times lower compared to the 4-layer case ($5.170 \times 10^{-2}\ p.u$), with negligible increase in operating duration.

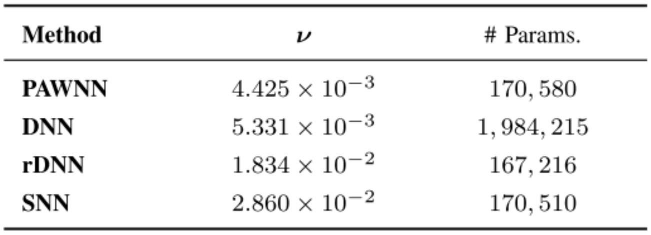
Table 11: *Performance Comparison with Different Numbers of Layers of PAWNN. [6]*



Method	ν	Time (ms)
PAWNN (2L)	2.411×10^{-1}	0.731
PAWNN (4L)	5.170×10^{-2}	1.241
PAWNN (6L)	5.330×10^{-3}	1.259
SNN + G-N	2.860×10^{-2}	589
G-N	4.489×10^{-1}	2891

In comparison with other different Neural Networks that have recently been introduced in a considerable number of works, PAWNN stills demonstrates its high potential in resolving DSSE problem with low error ($4.425 \times 10^{-3}\ p.u$) and a reasonable number of training parameters. In particular, DNN is still able to give a remarkable performance thanks to the multiple stacked hidden layers allowing the ML model to specifically learn the relationships between the input features and the target ones. Also because of this characteristic, this approach faces a trade-off regarding the model cost with significantly high number of training parameters compared to the other methods. This puts the method under a lot of economic and training time pressure when being considered in real case situations, especially with more complicated distribution network. Using the nearly similar number of parameters as the PAWNN, reduced Deep Neural Network rDNN and the SNN-GN model obtain much higher MAE error ($1.834 \times 10^{-2}\ p.u$ and $2.860 \times 10^{-2}\ p.u$ respectively).

Table 12: *Performance Comparison of PAWNN (4L) and Other State Estimators. [6]*



Method	ν	# Params.
PAWNN	4.425×10^{-3}	$170,580$
DNN	5.331×10^{-3}	$1,984,215$
rDNN	1.834×10^{-2}	$167,216$
SNN	2.860×10^{-2}	$170,510$

Therefore, by utilizing the underlying connections among the nodes in the distribution system to prune and equivalently reflect in the training weights, PAWNN method exhibits a highly potential approach for DSSE problem to both resolve the problem of high number of training parameters requirement in DSS and reduce significantly the error compared to SNN.

2.7 SUMMARY OF STUDIED METHODS & PLANIFICATION

Table 13: *Summary of Studied ML Methods*

Methods	Advantages	Disadvantages
WLS	Fast and simple.	Non-convex optimization function with multiple local minima. Poor performance.
FFNN	Reducing the complexity of SE task to matrix multiplications by shifting to an offline training stage utilizing historical or simulated data.	Challenging to avoid exploding or vanishing gradients. Less accurate than any other optimization-based approach.
SVM	Capable of capturing the relationship between the nodes. Robust to random missing bus data and the target buses. Able to learn the relations from the input features that are not well understood. Relatively low error.	Only tested under a local area in the system. Not examined performance for large scale distribution system.
AE	Able to capture the underlying topology of the network. Simple model structure. Applicable in quasi real-time DSSE with the offline pre-trained phase.	Decreasing performance under high missing ratio. Requiring a high ratio of availability measurements to be able to learn the underlying structure.
SNN-GN	Simplified structure by mapping the measurements into a point in the neighborhood of the true state. Lowering runtime and model complexity.	Not remarkable accuracy.
DNN	Significantly high performance benefited from deeply learnt parameters of multiple hidden layers.	High ambition with exact end-to-end mapping. Requiring large measurements or deep Neural Network. High training sample complexity. High cost in real applications. Considerably high training time for large system.
PAWNN	Able to consider the underlying physical connections in topology. Reduced complexity compared to DNN due to the pruned training weights. High performance.	Requiring another method to save and convert the data of nodal connections to prune the weight matrix. The training weight matrix can be exponentially large and highly sparse

	Low operating time for real-time DSSE.	when being applied in large and complex distribution systems.

To conclude, this chapter has provided the theoretical ideas and current situations, problems of Distribution System and also indicates the significant necessity of State Estimation for it. In that context, several renowned methods are selected to be considered in terms of both fundamental ideas and model performance. They are compared eventually so that potential candidates can be chosen for the next stage of implementation.

The comparison of the models is performed by taking into account their benefits and drawbacks rooted and inferred from their performance. Regarding high accuracy requirement, DNN can be considered as a potential method to test. However, SVM and PAWNN are evaluated to be more appropriate regarding the high requirement of low model complexity but still potentially result in an acceptable performance. Nevertheless, due to its ability to both capture the underlying physical connections of the context and give a potential low error, AE appears to be the most suitable method to be tested. . The solution can be initially tested with one phase under the balanced system assumption before considering the application of more computationally complicated 3-phase case. The complexity of AE (number of encoding/decoding hidden layers) can be modified corresponding to the increase of the considered bus number.

Concerning data generation process as the initial step of implementation, it is impractical to place the measurement devices onto all the tested buses in the distribution system. For NN-based techniques, different types of instruments for active/reactive power, current and phase can be reasonably planned to be installed at several suitable bus positions, as shown in [1] and [6]. After that, the training pairs containing measurements and the target ground-truth of the remaining buses can subsequently inferred and computed from these measurements based on physics formula. Moreover, the training data including active/reactive power, voltage and phase can also be calculated from load profile from Smart Meter [3].

For the first considered case, active/reactive power, voltage magnitude and voltage angle are indicated as the needed input features for the AE [5][7]. Due to the limitation in data availability, these features can be generated for the training process by simulating a network close to the real system and injecting the real measurements of load profile, which can be downloaded from online sources. Due to the high number of measurements for each bus, the complexity of the tested bus number can be examined from simple cases (3 and 5-bus network) to more complicated network (about 20 to 30 buses). To evaluate the generality of the trained model, the generated dataset can be divided into 2 sections for training and testing purposes. The training process requires the majority of the data to enlarge the possible sample case for the model to learn while the remaining minority is used to test and evaluate how the trained model performs in the cases that it has not faced before [8]. The lower the test error, the higher the model performance. Missing data is randomly created in positions of every parameter of the evaluated part. In the range of this project, voltage is considered as the main state variable to estimate for the initial section of the implementation.

CHAPTER 3. METHODOLOGY OF AE-PSO STATE ESTIMATOR

In this chapter, the specific principles of all techniques that have been used throughout all the procedures are introduced and demonstrated following the selected Auto-Encoder-based approach. They are also considered as preparation tools and materials for the implementation represented in the latter chapter.

3.1 AUTO-ENCODER STRUCTURE

3.1.1 General Picture of AE State Estimator

As mentioned in the section **2.3** of the previous chapter, Auto-Encoder is known as an unsupervised machine learning model using a symmetrical neural network (NN) aiming to reconstruct the initial input values following an encoder-decoder process [7, 9]. In particular, the encoder is utilized to compress the input vector $\boldsymbol{z}$ into a T latent-dim representation $\boldsymbol{y}$ by a fully-connected mapping function $\boldsymbol{f}$ using weights and bias as linear parameters $\boldsymbol{\theta} = \{\boldsymbol{w}, \boldsymbol{\beta}\}$ and an activation layer $\sigma()$ [9]:

$$\boldsymbol{y} = \boldsymbol{f}_{\theta}(\boldsymbol{z}) = \sum_{t=1}^{T} \sigma(\boldsymbol{w}_t^T \boldsymbol{z} + \boldsymbol{\beta}_t) \tag{16}$$

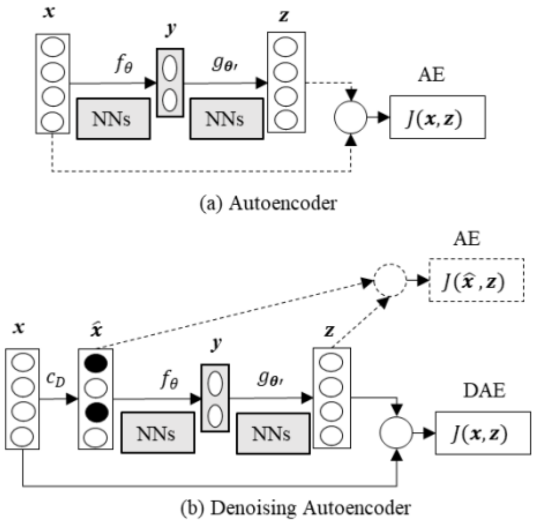
On the contrary, the decoder uses a reversed structure $\boldsymbol{f^{-1}}$ containing learning parameters $\boldsymbol{\theta'} = \{\boldsymbol{w'}, \boldsymbol{\beta'}\}$ to map back to a vector $\boldsymbol{z'}$ which is expected to be as similar as possible to the input $\boldsymbol{z}$ [9]:

$$\boldsymbol{z'} = \boldsymbol{f}_{\theta'}(\boldsymbol{y}) = \sum_{t=1}^{N} \sigma(\boldsymbol{w'}_t^T \boldsymbol{y} + \boldsymbol{\beta'}_t) \tag{17}$$

To accomplish this task, the optimal parameters $\boldsymbol{\theta}_{opt}$ and $\boldsymbol{\theta'}_{opt}$ are learnt by iteratively updating through N input samples $\{\boldsymbol{z}_i\}_{i=1}^{N}$ using a loss function J to minimize the error with each of its corresponding output $\boldsymbol{z'}_i$ [9]:

$$\{\boldsymbol{\theta}_{opt}, \boldsymbol{\theta'}_{opt}\} = \arg \min_{\boldsymbol{\theta}, \boldsymbol{\theta'}} \frac{1}{N} \sum_{i=1}^{N} J(\boldsymbol{z}_i, \boldsymbol{z'}_i) \tag{18}$$

Denoising Auto-Encoder (DAE) is a variant of AE normally used to deal with missing data reconstruction tasks is also applied to be tested in this case. Technically, the input data is randomly corrupted into zeros in serval parameters based on a specific scheme before being put into the model to follow the same forward and backward sections as in AE.



(a) Autoencoder

(b) Denoising Autoencoder

Figure 14: *Data flow of AE and DAE [9].*

The complexity of the model can be extended by simultaneously increasing the number of hidden layers in each half of the AE so that the deep structure can be formed to adapt with more intricate problems. On top of that, the changing in size after each layer is implemented with the same rate and is divided into 2 cases including overcomplete and undercomplete for increasing and decreasing trend respectively in the number of encoder layers. The latter is more frequently used because it only requires the model to capture the most relevant features from data [10]. Therefore, within the range of this project, one single simple latent representation is projected to be applied on all the cases whereas the capability of multi-layer structure can be considered for the realistic one whose network contains multiple nodes in a complex topology. Also, the power of overcomplete or undercomplete application is examined in the implementation process.

Particularly concerning the nonlinear mapping, activation function is considered as a crucial and differential component of neural network and it acts like a mathematical gate in between the input feeding the current neuron and its output going to the next layer. This dynamic property allows the model to extract complex features from data and to represent non-linear convoluted random functional mappings between input and output. Without this mapping, the model would be a simple linear regression transformation with limited complexity to extract complicated information. More importantly, the requirement of its differentiability also facilitates the back-propagation to update and fine-tune the learning parameters for optimization task. [12]

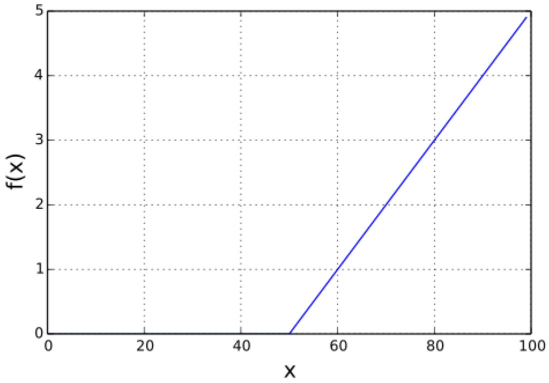


Figure 15: *ReLU activation function [13].*

Accordingly, ReLU (Rectified Linear Unit) is selected to be used in the hidden layer to filter only positive values because of its computational efficiency thanks to the non-simultaneous activation property. While $tanh()$ and $arctan()$ are projected to be tested separately for the output layer of the decoder to verify its performance. This is because these functions are able to map the input values into the range of $[-1, 1]$ and $[-\pi/2 , \pi/2]$ respectively to deal with the negative numbers (P and Q) of this context. [7, 12]

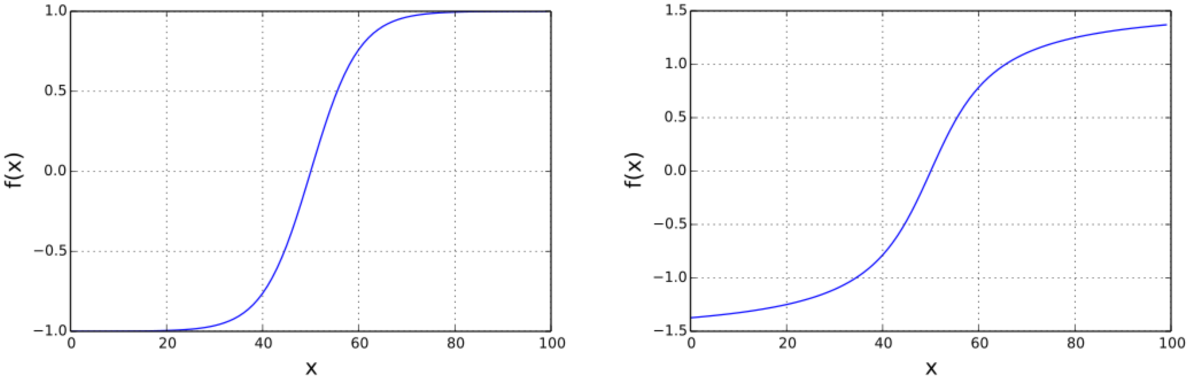


Figure 16: *Tanh and Arctan activation function [13].*

3.1.2 Training

Similar to the training section of other ANN types, that of AE concerns 5 components. Firstly, Mean Squared Error is selected as the criterion J to evaluate the error between the input data $\boldsymbol{z}$ and its estimations $\boldsymbol{z}'$. Secondly, the number of epochs to train over the dataset is tested so that the overfit situation can be avoided. Thirdly, the training data size and the testing ratio also play an important role in determining the performance of the model; therefore, different values are also examined to be suitable for each data and model complexity. Additionally, the batch size, which indicates the number of samples to be trained simultaneously, is preferred to be 1 so that the model can be trained to adapt with each of different situations in the context; however, the case of greater exponential of 2 is also considered for batch size in the case that the estimated state is time-series dependent. Lastly, in order to make the training parameters updated and improved iteratively based on the evaluation of the loss function, Adam is used as an improved gradient descent optimizer because of its high efficiency and less memory requirement against problem containing a large number of data or parameters. In particular, momentum property with exponentially weighted average of the gradients is considered in Adam optimization to boost the gradient descent performance to converge in a faster pace towards the minima. In parallel, an adaptive learning strategy, namely Root Mean Square Propagation, is also applied to adapt the oscillation of the weights suitably corresponding local or global minima. [10, 14]

More importantly, to adapt with this specific problem of missing value reconstruction, the weighted loss function is utilized as a modified version of MSE to adjust and put more weights on certain specified parameters [11]. This method not only takes advantages of giving missing positions more prioritized weights to be optimized in training Denoising Auto-Encoder with random missing value (more details in section **3.4**) but also facilitates different categories of search in Particle Swarm Optimization for state estimation in section **3.2**.

$$J(\boldsymbol{z}_i, \boldsymbol{z'}_i) = \alpha \|\boldsymbol{m} \circ (\boldsymbol{z}_i - \boldsymbol{z'}_i)\|_2^2 + (1 - \alpha) \, \|\bar{\boldsymbol{m}} \circ (\boldsymbol{z}_i - \boldsymbol{z'}_i)\|_2^2 \tag{19}$$

Where $\circ$ denotes parameter-wise product, $\boldsymbol{m}$ and $\bar{\boldsymbol{m}}$ are binary mask vectors for missing values and available ones orderly and α is the adjustable parameter in the range [0,1] to control the weighted loss. [11]

3.2 PARTICLE SWARM OPTIMIZATION

3.2.1 Principle of PSO

Particle Swarm Optimization is inspired by the idea of a group of animals or particles chasing an enemy or searching for a food source in a realistic wild space or environment. Being transitioned into mathematical term, these targets can be converted into the optimum as an optimization problem for those learners, the particles. The generation of new individuals as new positions of particles after each recording of time unit follows movement rule in which the search is oriented towards the global minimum of them [15].

Accordingly, to track the optimization process, position and velocity of the particles are recorded and updated iteratively based on those of the previous movement assuming that the iterative time step is equal to 1.

$$\boldsymbol{X}_i^{(k+1)} = \boldsymbol{X}_i^{(k)} + V_i^{(k+1)}$$

$$\boldsymbol{V}_i^{(k+1)} = \boldsymbol{A}\boldsymbol{V}_i^{(k)} + \boldsymbol{B}\left(b_i - \boldsymbol{X}_i^{(k)}\right) + \boldsymbol{C}\left(b_G - \boldsymbol{X}_i^{(k)}\right)$$

Particularly for the velocity, $\boldsymbol{A}$ denotes inertia demonstrating the continuous habits of the particles directing towards the previous movement. Secondly, $\boldsymbol{B}$ represents the attraction of the particles to the best points b_i they have swarmed in their past memory. Finally, $\boldsymbol{C}$ is also a representation of memory b_G but it is recorded and found by information exchange throughout all primitive particles' trajectory; therefore, it is called cooperation. [15, 16]

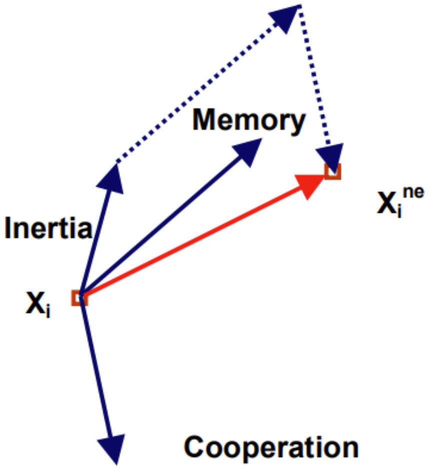


Figure 17: *Illustration of 3 parameters influencing the movement of individuals in PSO [16].*

3.2.2 AE as Objective Function & Different Types of Estimation Search

As mentioned in the section **2.3,** there are 3 categories of search applied for estimation process, namely POCS, constrained search and unconstrained search. However, only two last types work with optimization function and acquire the sufficient complexity and efficiency. Therefore, they are selected to be the tested methods for objective function of PSO algorithm.

Particularly, instead of Extreme Learning method shown in [5], the Auto-Encoder is projected to be pre-trained following back-propagation-based algorithm as specified in section **3.1** before being used as an objective function to be optimized by PSO for DSSE. Particularly, the particles are set to swarm over the space to search for values at missing marked indices so that the MSE they produce is minimized. Also, the parameters error criterion to search for those states is based on either constrained or unconstrained way to test for the performances. To facilitate these tests of search type, the weighted loss demonstrated in section **3.1** is utilized by adjusting the value of alpha $\alpha = 0$ and $\alpha = 1$ respectively. The space dimension of the search is equivalent to the number of missing values in the considered sample in estimation process.

Moreover, the performance of the estimation stage is highly dependent on how the number of particles and iterations are selected and the optimal values of them are problem-dependent. In particular, the former illustrates diversity of the swarm since the more particles swarming, the larger parts of the search space to be covered per iteration and therefore, the higher the chance it obtains the area of the optimal position. However, this also causes more complexity for the computational process as all the positions and velocities of the particles are required to be updated for each step. On the other hand, the latter parameter determines the number of updating run to reach the optimal position for solution. Accordingly, selecting a low value of iteration could cause a premature termination and result in a poor performance of the estimation. On the contrary, an extremely high number could also complexify the computation of the search and increase the amount of swarming time for DSSE. The trade-off is also witnessed between these parameters since a high number of parameters requires a lower number of iterations and vice versa. [17]

3.3 HYPERPARAMETERS OPTIMIZATION

As can be seen in both methods mentioned in the previous sections, they contain plenty of adjustable parameters which can influence significantly on their results as overfit or underfit and hence; on the output performance of the model. Therefore, to acquire a sufficient performance for a certain situation and a certain dataset, these parameters related to neural architectures or training procedures are required to be tested multiple times so that sufficient parameters (or hyperparameters) are selected for the run based on experience. [18]

This task is covered under the range of automated machine learning (AutoML) whose manner is operated by data-driven, objective and automation for hyperparameter optimization

(HPO). Given a certain dataset and a searching space, HPO can automatically recommend the approach that performs in a rival and quasi-optimal way while still consumes substantially reduced amounts of resources compared to the exhausted manual trial. Moreover, the features of the training section as reproducibility and fair comparisons are also facilitated. Because of these considerable advantages, the development of AutoML has been invested dramatically in recent years to democratize machine learning. [18]

However, the practical application of HPO still possesses several obstacles. First of all, large and deep model structures burden the cost of the operation and also complexify the pipeline of the system. Also because of this, the searching space can suffer from a multi-dimensional problem, a wide range of values and a diversity of parameter types including continuous, categorial and conditional ones. Moreover, the generalization performance of the model is often detached due to the limitation in interfering parameters like gradient, convexity, smoothness or a complete training dataset. [18]

Because of the first issue, Random Search is selected to perform HPO with multiple numbers and types of data in this case instead of Grid Search which can cause the exponential explosion in dimensionality of configuration space due to the finite set of values predefined for this method. More importantly, to avoid the burden of computation in searching process, two HPO operations are implemented separately to find sufficient parameters minimizing the objective function for AE-reconstruction (batch size, ReLU, Tanh, data size, learning rate, latent dimension, number of epochs) and percentage of outlier PSO estimation (number of particles and number of iterations).

To implement Random Search on these tasks, optuna library is projected to be utilized with model construction in PyTorch because it allows custom suggestions corresponding to each type of parameters and operates the number of searches based on a certain budget. In particular, suggest_categorial is used for the trial on the boolean availability of ReLu, Tanh and different values of batch size in exponentiation of 2. Whereas suggest_int can adapt with a range of integer value in the number of latent dimensions, epochs, data size, particles and iterations. Finally, suggest_loguniform can be used for negative exponential range of learning rate. [19] In order to give a sufficient range of values to test for each of these categories, several manual training sections can be used initially to see how the model reacts with different ranges of parameter. Moreover, to ease the burden of this exhausting procedure on a large dataset and a multivariable search, prune method is also applied by attaching report on the performance of each epoch as the consideration for early-stopping when a poor suggestion is early recognized.

3.4 DATA PRE-PROCESSING

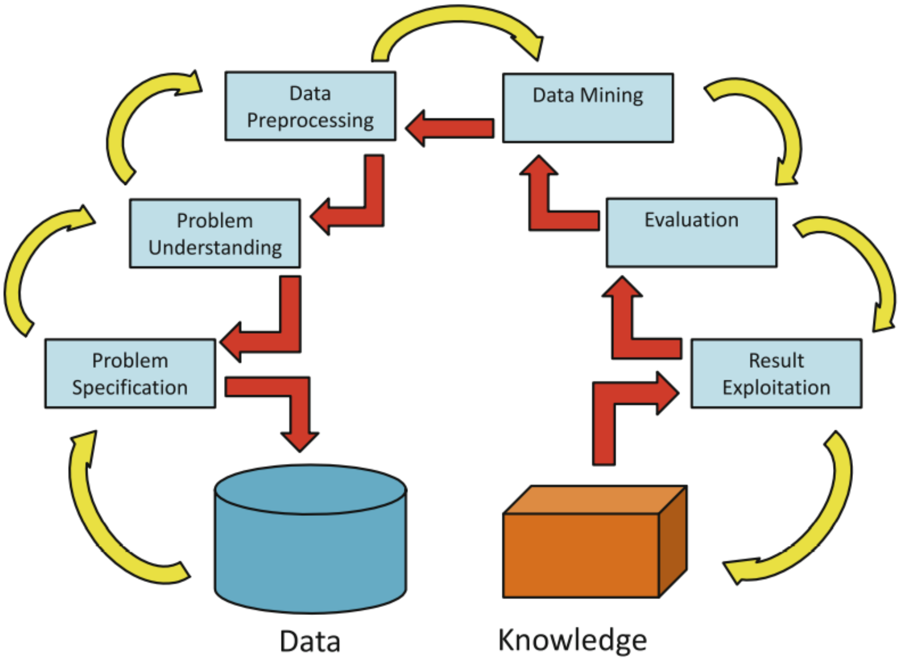


Figure 18: *Knowledge Discovery in Database process (KDD) [20].*

Data Preprocessing (DP) is considered as one of the most important steps in Knowledge Discovery Dataset which is used to extract and grasp the understanding and characteristics for any data-related task. Therefore, DP is also considered as a tool to facilitate the planification and decision-making on the up-coming stages, especially on Data Mining, Modelization and Evaluation. These important impacts also are main reasons why it is composed of multiple sub-processes including data cleaning (related to noise removal and data consistency), data integration (deal with multi-source dataset), data transformation (transform and consolidate datatype and data range for specific problem) and data reduction (select and extract both samples and features in the database). [20] However, DP is highly context-dependent; therefore, each of these sub-processes is selected and operated based on the requirements of the task to cope with its specific dataset.

Regarding the dataset of the DSSE problem to serve for a research and testing project, it is mainly generated by time-series simulation through ___ application which can cover all the characteristics, variables and noise factors of realistic cases. Accordingly, the data cleaning and data reduction problem regarding feature dimensionality are handled due to the completion of data simulation for training section and the selection of relevant features for each bus (active power P, reactive power Q and voltage V) in a DSSE task. Concerning the second aspect of data reduction, data size is projected to be added in HPO to scale a sufficient amount of data for both training and testing. A training subset of instances is selected randomly from the simulated dataset Random Sampling [21] so that the model can obtain the ability to adapt with different cases of sample separately. While that of the validation subset is retained sequentially to facilitate visualization and model evaluation with a test ratio of 0.3.

3.4.1 Data Rescaling

The remaining issue related to data transformation is the main concern data preprocessing task in this project. This is because general range of the values in the dataset is highly influenced by different features and outliers in each feature like voltage drop. Particularly as can be seen in ***Figure 19*** showing the boxplot distribution of all features created by data simulation, there is a large difference in the value range of Voltages compared to that of Power attributes. Moreover, a high skewness due to outliers is also witnessed in the distribution of each variable [22]. Therefore, data rescaling is considerably crucial to be applied for this case so that the scale of each attribute is transformed and brought closed to each other, which is also considered to facilitate the gradient descent in the training process afterwards.

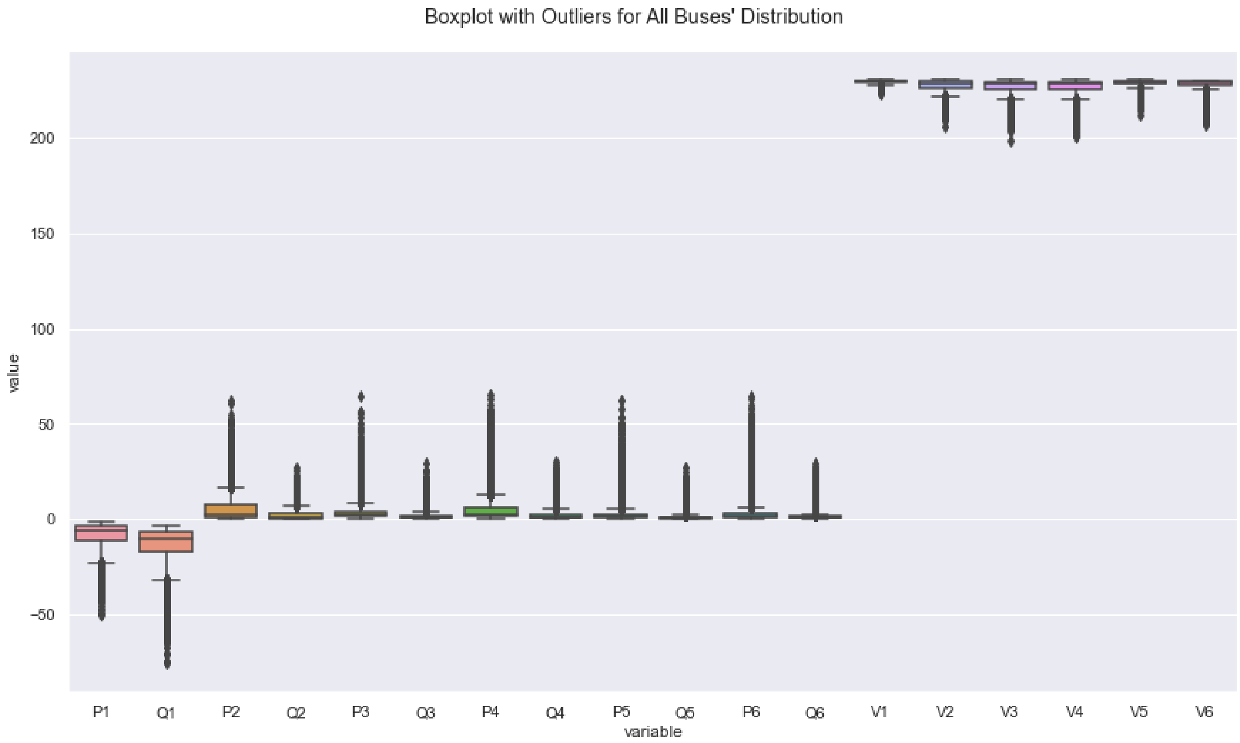


Figure 19: *Boxplot with Outliers for All Buses' Distribution.*

Instead of using the scalers like Min-Max Scaler (transform the feature into given range with [0, 1] as default) or Standard Scaler (standardize features by removing the mean and scaling to unit variance) which handle the transformation on the dimension of each feature, the normalization tool is planned to be applied so that the scale on each sample is considered separately. This is because the correlation of variables in a Neural Network problem is highly important so that the model is capable of capturing their relations, recording them in weights, biases and externalizing them in the reconstruction and DSSE tasks. By doing this, all the values in each instance are divided by their calculated norm. However, different norm calculations for each sample are witnessed to be an obstacle for data denormalization after finishing the estimation, which is highly important to restore the original range of the value to retain their physical properties. As a result, all the values in the dataset are projected to have the division by the base value which is equal to the voltage level of surge protection $(400/\sqrt{3}V)$.

3.4.2 Missing Mask & Random Missing Value

To create the missing data for the training process, two tools are customized for different specific test plans (more details in section **3.6**). Particularly in the case of Denoising Auto-Encoder, random positions in each instance are replaced by 0s following the binomial rule so that corrupted data is generated for the training section [11]. In parallel, the corresponding complete version of each is also retained to pass the ground-truth to the evaluation section. Secondly, to serve for the testing strategies using predefined missing positions, -1 is used to insert to them as a missing mark for PSO estimation. While 0 in the first case can impact on the performance of the model since it is injected directly into the model, the -1 is selected to only locate the missing indices without using it directly for initialization. This is because DAE method is tested for pure direct auto-encoder estimation whereas that of the other cases is performed separately by PSO based on AEs which are pre-trained on complete datasets.

3.5 EVALUATION METHODS

In general, to verify the adaptability of a ML model in both comprehension and hidden patterns identification against a certain problem or phenomenon, its ability to acquire proper performance on new dataset needs evaluating. Accordingly, Performance Fitness and Error Metrics (PFEMs) are utilized to validate the error or the closeness of real observations (ground-truth) to the estimations given by the model. Therefore, the lower the PFEMs, the better the model adapt with the situation. [23]

Normally, these PFEMs are the methods inserting the evaluation by giving one output on the accumulation of all dataset. For this reason, several types of figures are also added to test the model's performance on each individual and their impacts on the whole distribution of the dataset.

3.5.1 Performance Fitness & Error Metrics

One of the most frequently used PFEMs is Mean Square Error (MSE) which measures the average of the squares of the errors between actual $\boldsymbol{z}_i$ and the estimated state $\boldsymbol{z'}_i$ over T testing samples. Since being utilized as a criterion for optimization process of both AE and PSO, MSE can be used to track the change of loss in the training and validating section.

$$MSE = \frac{1}{N}\sum_{i=1}^{T}(\boldsymbol{z}_i - \boldsymbol{z'}_i)^2$$

Besides, one of the variants of MSE is Root Mean Squared Error which is calculated by taking the square root of MSE:

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{T}(\boldsymbol{z}_i - \boldsymbol{z'}_i)^2}$$

In comparison with MSE concerning the square of the value, RMSE is considered as the more straightforward metric to interpret because its unit is equivalent to that of the target value ($\boldsymbol{z}_i$ and $\boldsymbol{z'}_i$) which is originally represented under Voltage (V) before being processed and converted to power unit $(\mathrm{p.u})$. Reasonably, RMSE is often preferred for the interpretation of the average deviation between the estimation and the ground-truth, especially applicable for the numeric attributes in this case. However, because of the squared expression, RMSE is more sensitive to noise and outliers, which is projected to be highly adequate to verify the adaptability of the AE-PSO state-estimator on the dataset containing multiple voltage drops, uncertainties and unstable conditions as indicated in section **1.1**. [23, 26]

To evaluate the relative deviation of the estimated value without concerning its direction with respect to the real value, Mean Absolute Error (MAE) is also considered for this task. MAE is less prone to outliers in comparison with RMSE. [23, 26] Therefore, both of them can be measured to verify the influence of outliers on the main error distribution. This comparison allows assessing the sensitivity of the model to critical voltage situations.

$$MAE = \frac{1}{N}\sum_{i=1}^{T}|\boldsymbol{z}_i - \boldsymbol{z'}_i|$$

To grab a more particular view of individual' error instead of the accumulated summation approaches (MSE, RMSE, MAE), relative percentage error (RPE) is applied to interpret the percentage that the estimated value is deviated from the actual records on each instance. The term "relative" is used instead of "absolute" so that the level of overestimation and underestimation can also be evaluated, especially for this power-related problem. In parallel, the qualified range $[-1{,}1]\%$ is also introduced as a RPE threshold for a qualified estimation. Accordingly, the denser the error is distributed in this range, the more qualification the estimation is validated.

$$RPE_{\boldsymbol{z}_i} = \left(1 - \frac{\boldsymbol{z'}_i}{\boldsymbol{z}_i}\right) \times 100\%$$

3.5.2 Figure Visualization

Calculating individually the error of each testing instance as in RPE also means a group of error values is encountered for evaluation. In that case Boxplot and Kernel Density Estimation (KDE) are represented to display the distribution of RPE with respect to the qualified range. In parallel, Real-Time-Series Comparison plot is also considered to compare and observe the difference in real-time and in time-series measurement of these state values.

Particularly in Boxplot, the data is distributed based on the records of five parameters (“minimum”, first quartile (Q1), median, third quartile (Q3) and “maximum”). Their different arrangements on the plot can express the plentiful precipitations about the symmetricity, intensity of outliers, tightness or skewness. [24]

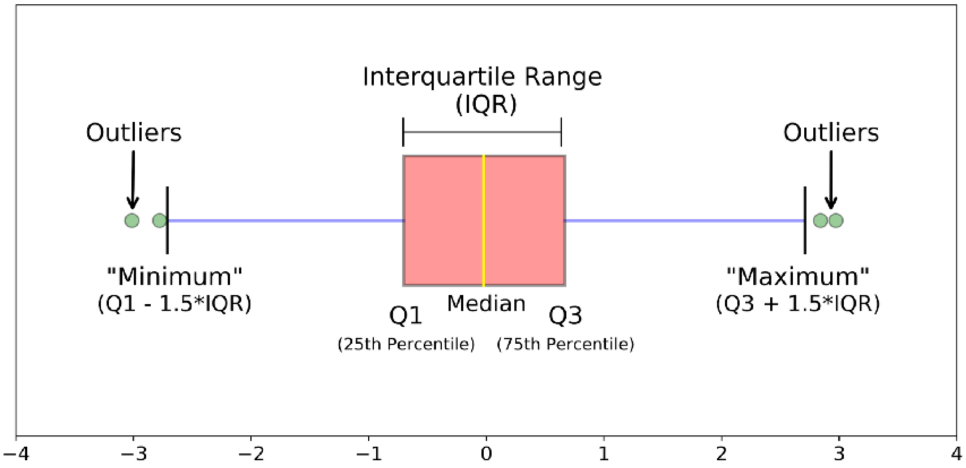


Figure 20: *Different components of Boxplot [24].*

Where median (second quartile Q2 or 50th percentile) is defined as the middle value of the whole dataset. The 25th Percentile Q1 are also denoted as the middle value between the smallest (not the “minimum”) and the median Q2 while the 75th percentile Q3 is the middle value between the median Q2 and the highest value (not the “maximum”). Accordingly, the range containing the majority of the data’s distribution is called Interquartile Range (IQR) ranging from Q1 to Q3.

On the other hand, Kernel Density Function (KDE) is a renowned method to identify the underlying continuous probability density function for a set of observations. This provides the information regarding the distribution of these recordings on a less cluttered and more interpretable manner. Not being restricted by any assumption, the probability density $\hat{P}_T(x)$ is estimated nonparametrically for each $RPE_{\boldsymbol{z}_i}$ error denoted by e_i on T testing samples. This facilitates the flexibility of the method on dealing with multiple data contexts [26]:

$$\hat{P}_T(x) = \frac{1}{Th}\sum_{i=1}^{T} K\left(\frac{x - e_i}{h}\right)$$

Intuitively, the density function is estimated based on the shape of a smooth bump determined by $K(x)$ for each instance. The summation of these bumps allows the demonstration of a large value on positions containing dense observations. Conversely, a small one is calculated for the regions distributed by a lower number of samples.

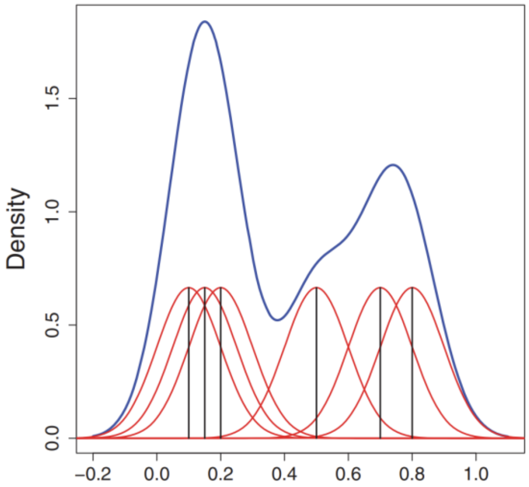


Figure 21: *Density curve (blue) based on acccumuation of bumps (red) on 6 observations (black) [24].*

Similar to a histogram, the performance of a KDF depends significantly on qualified selection of smoothing parameters, especially including kernel function $K(x)$ and smoothing bandwidth h. $K(x)$ can be selected within Gaussian Kernel or Spherical Kernel while the selection of h or standard deviation plays an important role in removing the distortions on the estimated distribution. An over-smoothed curve can remove important characteristics from the distribution's representation, while an under-smoothed one can generate distorted features out of random variability. The proper distribution is characterized with a smooth, unimodal and roughly bell-shaped curve.

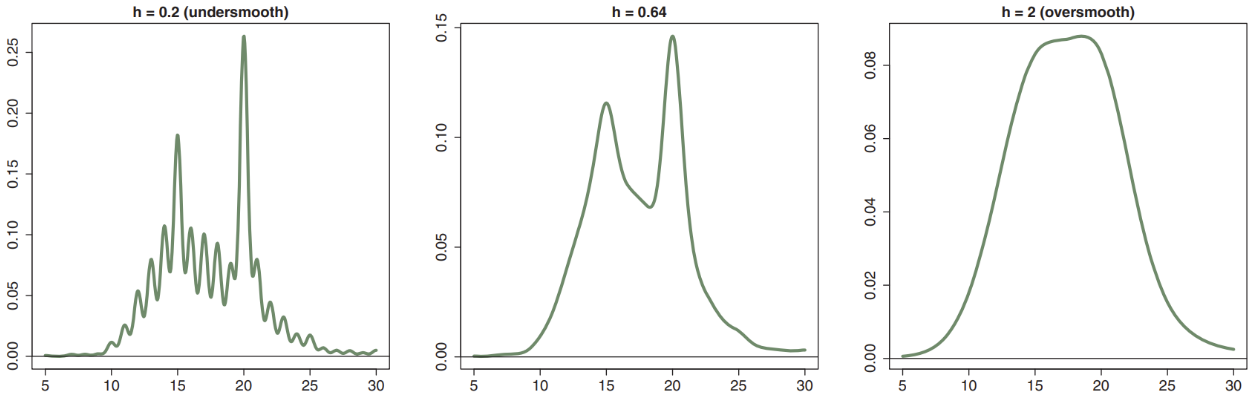


__Figure 22:__ An example of proper setting of banwidth in the middle compared to oversmooth and undersmooth [26].

3.6 TESTING STRATEGIES

3.6.1 Testing Bus-System

To test the adaptability of the model on the context, various versions of bus-system are used with increasing complexity in terms of both the bus number and the topology. In general, there are 3 parameters recorded for each bus including Voltage (V), Active Power (P) and Reactive Power (Q). Phase angle is also another common state attribute; however, only voltage is considered in the strategies with initial simple cases. In parallel, only 1 phase for all the cases is tested at the beginning before being replaced by 3-phase system when the model's performance is adequate. By doing this, there would exist 9 parameters for each bus by counting 3 base parameters $(P, Q$ and $V)$ for each phase.

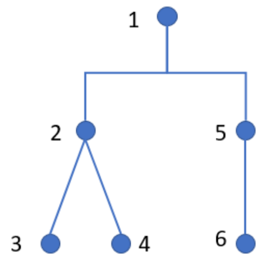


__Figure 23:__ Single-Phase 6-Bus System.

Originally, simple grid systems with 3 and 5 buses are simulated for initial trials, especially with the test for higher performance of Constrained than Unconstrained Search (*Appendix*). However, a 6-bus grid consist of 2 feeding branches (on 2 and 5) is predominantly used for the main testing sections including Denoising Auto-Encoder and Hyperparameters for AE-PSO estimator. As can be seen in ***Figure 24***, the states of these feeders are highly influenced by outliers with voltage drop ranging approximately below 220V. The Voltage distribution of the first branch contain a wider range of voltage outliers below 210V while that of the source one remains a more stable pattern at around 230V.

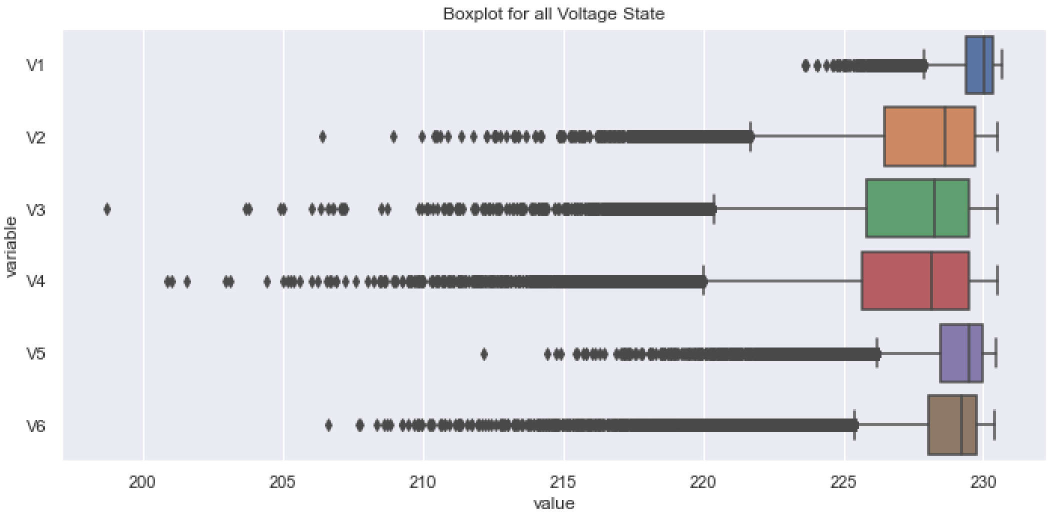


Figure 24: *Boxplot for Voltage Distribution of 6-Bus System.*

To serve for the AE-PSO estimation section, the model's performance is examined for each bus whose full base components (P, Q and V) are missing. This is attached to the missing meter situation in realistic bus-system utilizing one measurement device to record all of these parameters.

Table 14: *6 Cases of Missing Parameters (in Red) for Each Nodes.*

Missing Cases	**Node 1**			**Node 2**			**Node 3**			**Node 4**			**Node 5**			**Node 6**		
	V1	P1	Q1	V2	P2	Q2	V3	P3	Q3	V4	P4	Q4	V5	P5	Q5	V6	P6	Q6
1	🟥	🟥	🟥	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
2	x	x	x	🟥	🟥	🟥	x	x	x	x	x	x	x	x	x	x	x	x
3	x	x	x	x	x	x	🟥	🟥	🟥	x	x	x	x	x	x	x	x	x
4	x	x	x	x	x	x	x	x	x	🟥	🟥	🟥	x	x	x	x	x	x
5	x	x	x	x	x	x	x	x	x	x	x	x	🟥	🟥	🟥	x	x	x
6	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	🟥	🟥	🟥

3.6.2 Application on A Realistic Case

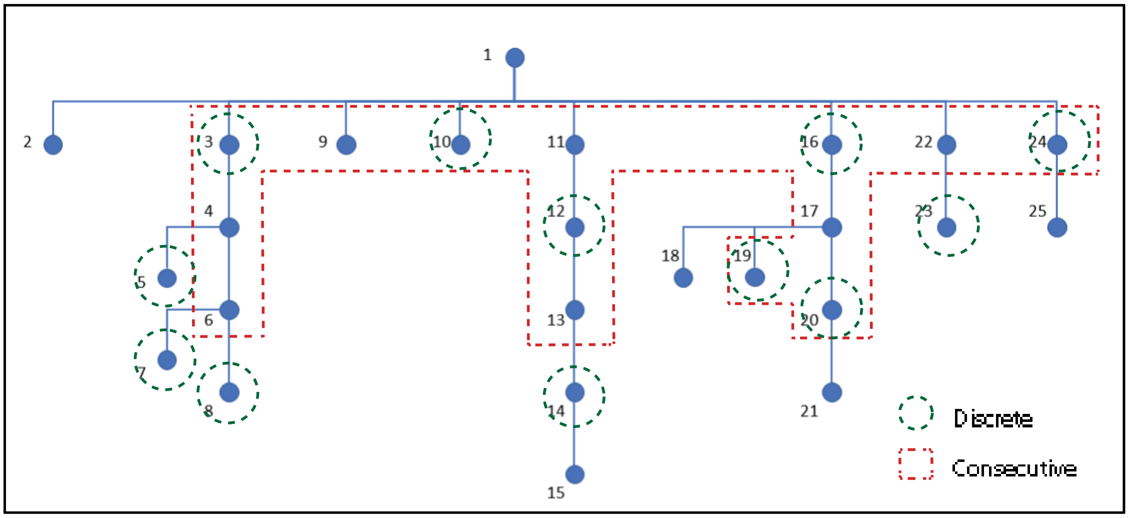


Figure 25: *Consecutive and Discrete Missing Range for 25-bus system.*

To be convinced about the adaptability of the AE-PSO estimator on a realistic case, a 25-bus system is simulated with 8 main branches feeding for different number of buses. As can be shown in ***Figure 26***, the 3 last buses of the fifth feeder (bus 13, 14 and 15) are distributed with more critical voltage drop to test the model's capability of dealing with sudden changes of bus state in reality. Moreover, the accumulation of multiple household's consumption reduces significantly the outliers in the boxplot of every voltage, which is witnessed reversely in the simple grid insisting of lower amount of power usage. Initially, the grid is assumed to have a balanced consumption on each phase for one-phase test before being applied with unbalanced data from 3 phases, which is equivalent to a realistic situation.

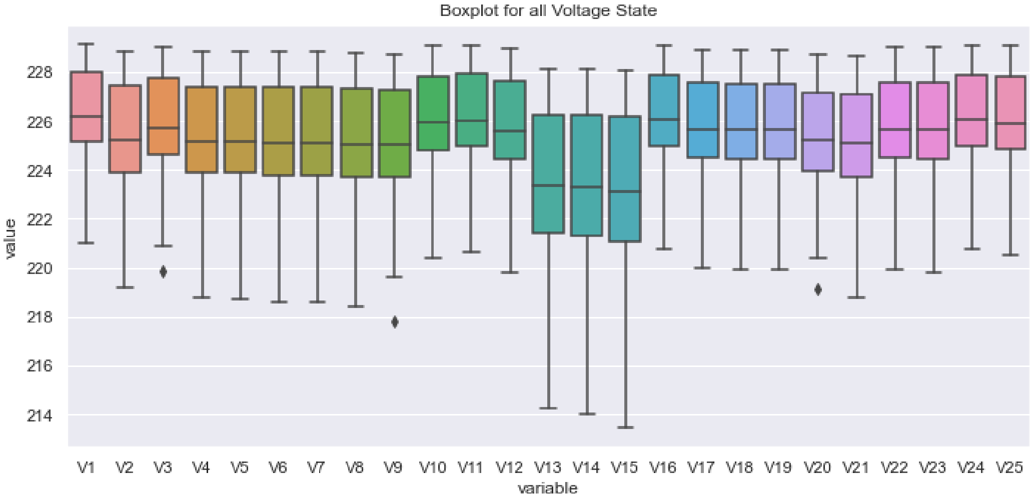


Figure 26: *Boxplot for Voltage Distribution of Each Bus in 25-Nodes Grid.*

All the improvements explored from the previous simple testing cases are inherited to this complex one, which includes the utilization of constrained search, the application of hyperparameters on both AE reconstruction and PSO estimation, and the replacement of $arctan()$ for $tanh()$ as last activation function. Also, various cases of missing all components on a group of different buses are examined for the robust performance of AE-PSO state-estimator on the DSSE problem. Concerning an increasingly complex grid with 25 nodes, the adaptability of different model's complexities is also tested by comparing conventional AE and Deep AE with 1 and 3 hidden layers respectively (***Figure 28***). The dimensions of added layers on both encoder and decoder in the deep structure are defined by the lower bound average of input and latent vector size.

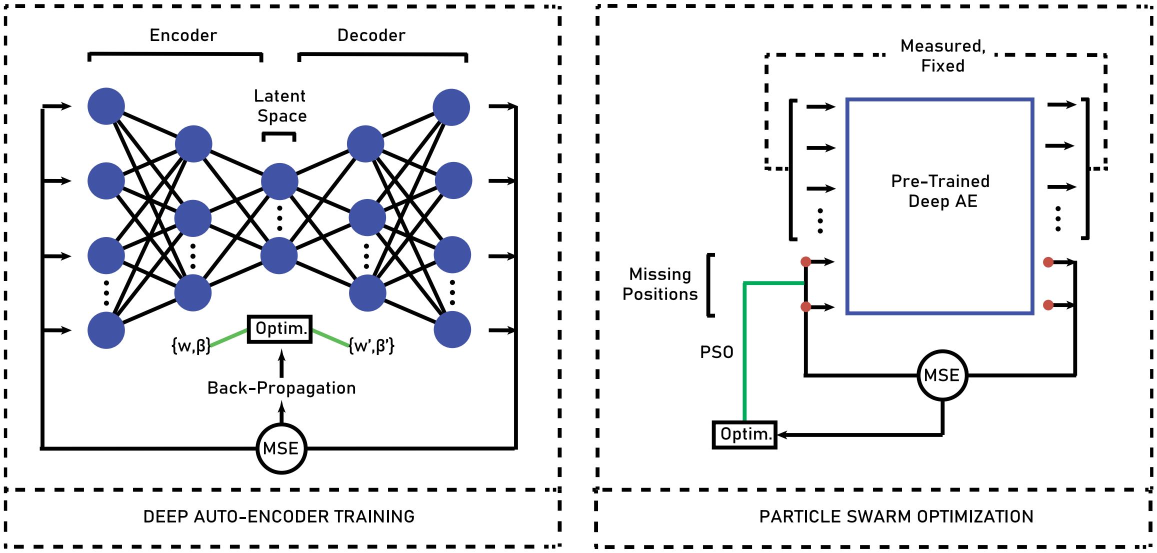


Figure 27: *Proposed Structure & Training Scheme of Deep AE-PSO Estimator.*

The high distribution of voltage drops on V13, V14 and V15 can be clearly seen in ***Figure 30*** with a considerably lower range of correlation with other buses. Moreover, the figure also demonstrates that the buses on the same feeder are highly correlated with each other. Apart from V2, the same pattern is also witnessed with buses connected directly to the source V1 on top of each feeder. Therefore, beside the testing strategy for each bus, the spanning scheme (***Figure 26***) for consecutive and discrete missing relations is also applied to test the adaptability of the model on the increasing number of missing buses. By doing this, different availability of bus measurements is assumed to respect the correlation on each bus of these missing cases.

Furthermore, the impact of critical voltage drop is also verified by considering 2 situations with and without the corruption on V13, V14 or V15.

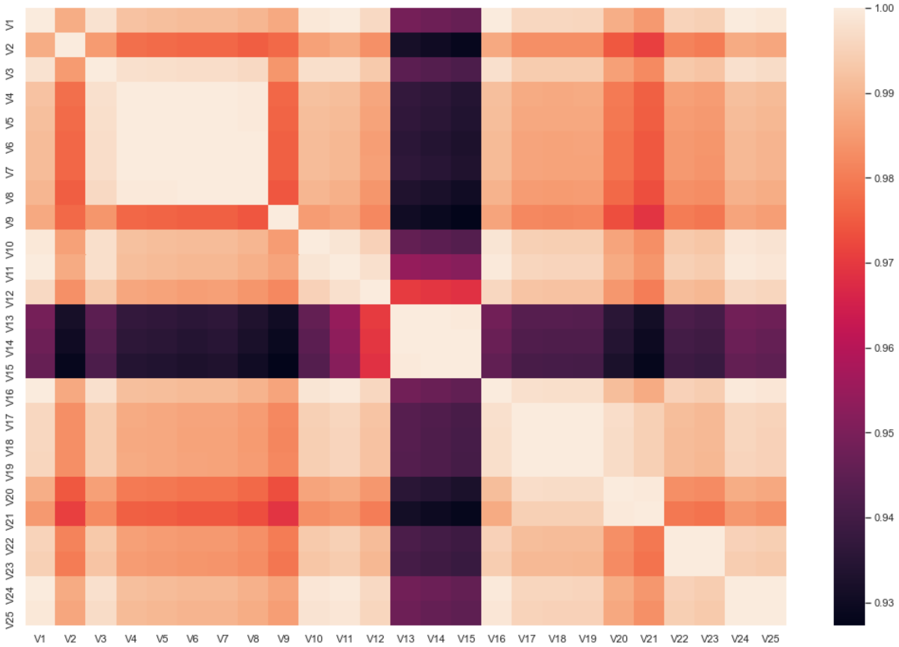


Figure 28: *Heatmap of State Correlation for 25-Bus System.*

CHAPTER 4. PROCEDURES & RESULTS

The sections in this chapter sequentially follows the implementation processes with different testing case scenario aiming to enhance the reliability and adaptability of the model on the DSSE problem. In parallel, the results of the performance based on the predefined training strategies are also demonstrated with PFEM and figures for visualization and error analysis.

4.1 DAE Reconstruction with Random Missing Values & Alpha Weight Training

By being randomly replaced by 0s through binomial rule, each sample of the 5-bus system dataset is corrupted in the preprocessing step for the application of Denoising Auto-Encoder [11]. At this early stage of the test, the states are estimated based on the pure reconstruction of DAE after being learnt through different cases of random missing values. The length of the latent vector is set to around 25 which is slightly greater than the total number of input parameters (15). This overcomplete form is preferred after several testing times, which indicates its possibility of learning greater number of features in this specific context.

Table 15: *Settings of Training Parameters for DAE.*

Batch size	Epochs	Data size	Alpha Weight	Latent Dim	Last Activation
1	8	1500	0.8	25	$arctan()$

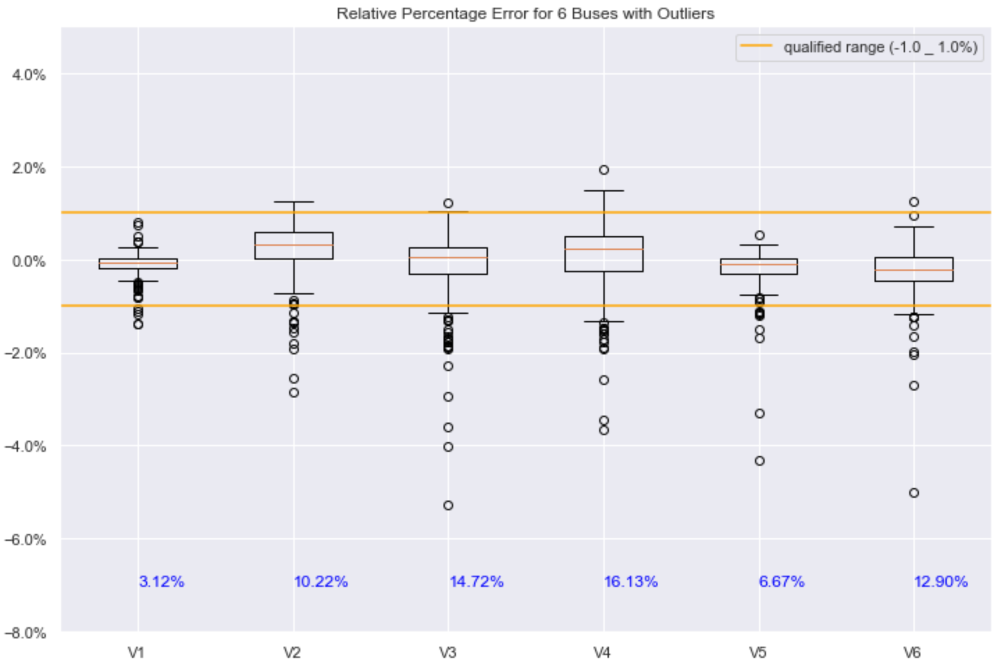


Figure 29: *DSSE boxplot error by DAE for 4-feeder 5-bus system.*

As can be seen in ***Figure 29***, DAE is capable of giving a high proportion of the testing samples within the qualified range and keeping the median error closed to 0%. However, it is witnessed that multiple outliers are distributed unqualifiedly due to the impact of voltage drop, especially for V3, V4 and V6 with 10.22%, 14.72% and 12.90% of out-qualified percentage respectively. Moreover, the ***Figure 30*** also shows a relatively large deviation in a real and random expression of the state prediction compared to that of the ground truth. This poor result can be rooted from the high impact of 0s substitution for missing values, which is also injected into the model and causes the distortion and misleading in the training process. Additionally, the inability to produce all the possible missing cases to train the DAE's adaptability inserts an inadequate learning base and also complexifies the training settings requirement, especially with the increasing number of buses and topology's complexity. Therefore, DAE is evaluated to be unfavorable to be applicable in this DSSE task.

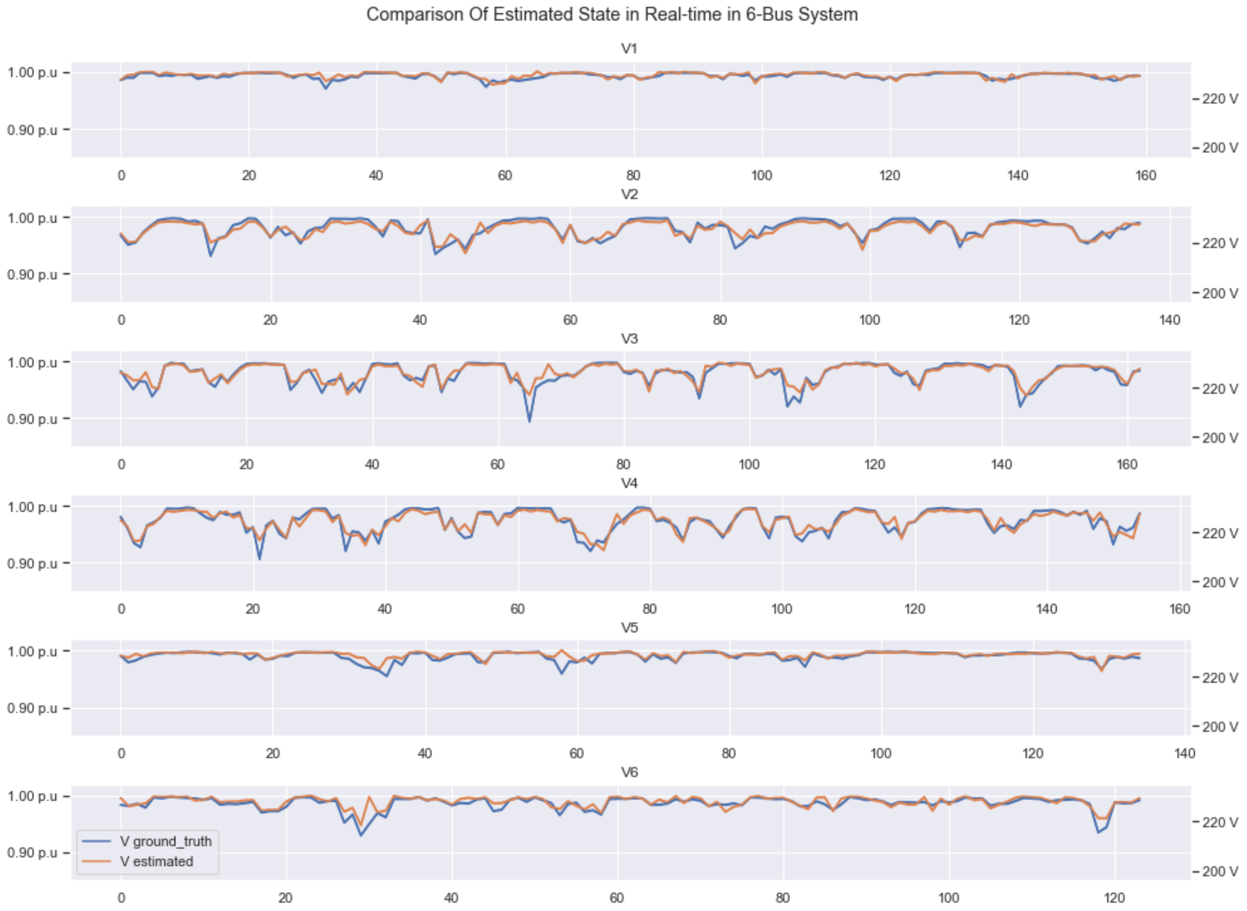


Figure 30: Real-time Comparison between Random Ground-truth & DAE State Estimation.

4.2 Application of Hyperparameters

In order to accelerate the selection process for training the reconstruction task of AE and for PSO estimation separately, HPO method is applied by providing a certain range of searching space for each concerning AE parameters (batch size, number of epochs, data size, learning rate, category of decoder's activation function, latent dimensions and the avalability of ReLU in the encoder) and PSO ones (number of particles and number of iterations). The objective function of the AE optimization is placed on the error of the voltages only while that of PSO is considered according to the minimization on the out-qualified percentage of error. As can be seen, the PSO parameters are set relatively high, increase both the amount of time and the complexity of the search. Regarding the size of latent vectors in the middle layer of AE, the overcomplete form is still recommended by HPO, indicating the capability of this category on capturing multiple features in this context type. More imporantly, $arctan()$ is selected instead of $tanh()$, concerning another manner of adjusting data output's range.

Table 16: *Suggestions of Hyperparameters for Constrained AE-PSO.*

Batch size	Epochs	Data size	Learning Rate	Latent Dim	ReLU	Decoder's Activation	Number of Particles	Number of Iterations
1	15	1390	0.00135862639	25	$True$	$arctan()$	49	111

These modifications are implented on 6-bus system with increasing number of buses on each feeder. Also, the time-series format is expressed instead of random one to clearly analyse the daily changes and voltage drops. In terms of PSO estimation method, constrained format is utilized to only take into account the error of missing parameters, whose operation is facilitated by setting the coefficient of full weighted loss to be equal to 1.

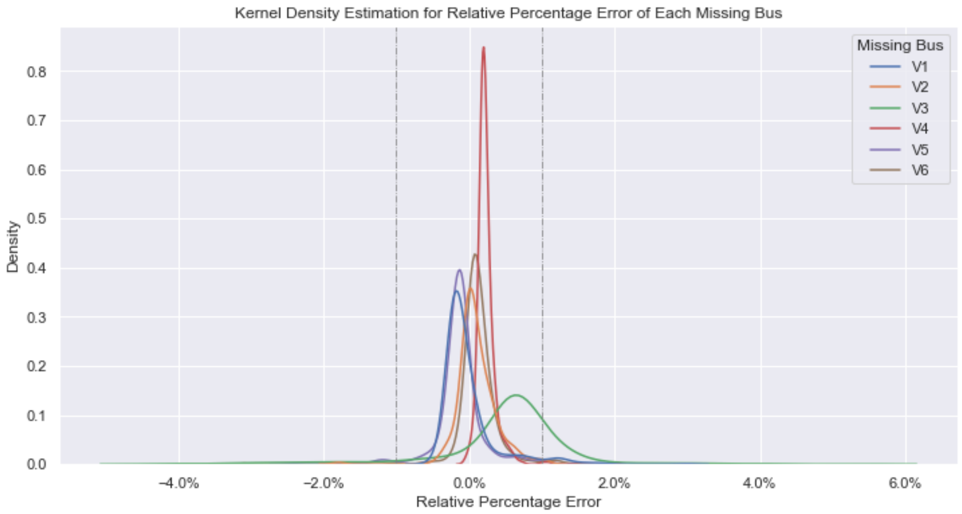


Figure 31: *Real-time Comparison with Constrained AE-PSO Method.*

In general, the allocation of the RPE is substantially compressed towards the qualified range compared to that of DAE. Particularly, the majority of RPE distribution, including interquartile and whiskers range, is covered within the [-1,1] range for 5 over 6 buses. More importantly, the percentage of out-qualified quantity is significantly decreased compared to the previous testing case with approximately 1% for $V2, V4$ and around 3% for $V1, V5, V6$ The lowest and highest relative errors of these buses are covered with the absolute value of 4%, demonstrating an improved efficiency in dealing with outliers of the approach this time.

However, even acquiring the quasi-equivalent voltage drop pattern with V2 and V4, the state estimation for the third bus still introduces a poor performance with deviated interquartile range from 0% and wide spread of outliers along -4% and 5% RPE. This illustrates still an unstable performance of the method. This can be because of the complete consideration on voltage error minimization of the AE reconstruction hyperparameters, which neglects the weights of the other active and reactive power. These can play an important role in being orienting parameters facilitating the estimation of the voltage states.

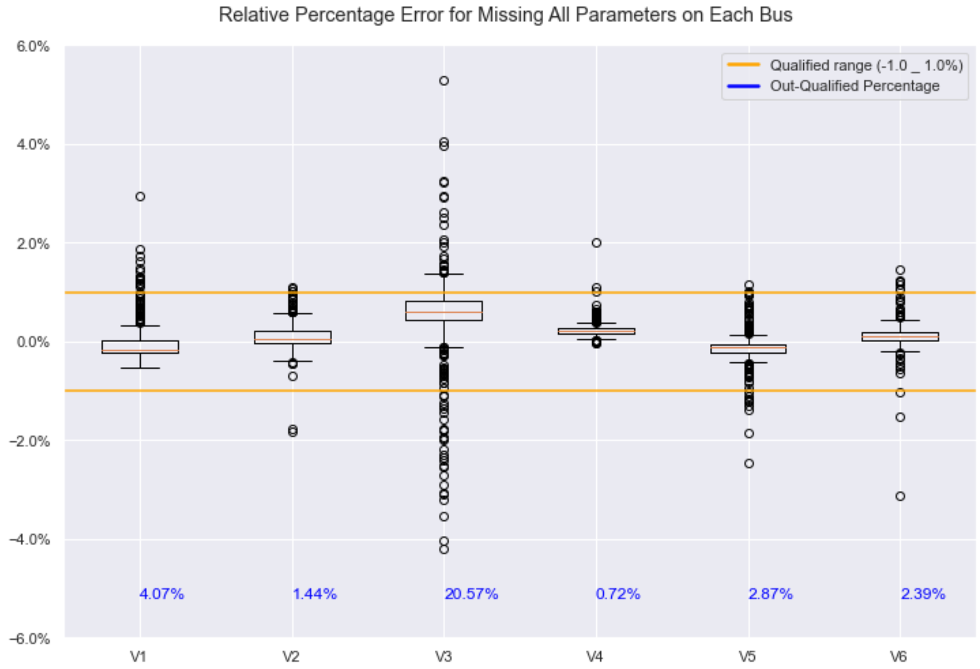


***Figure 32:** Real-time Comparison with Constrained AE-PSO Method.*

As can be seen in the Figure, although the poor performance of $V3$ inserts a high deviation in its real-time expression, the changes of its estimations still follow closely that of the truth values. Remarkably, the other buses obtain quasi-matching voltage level for the 2 concerning attributes, especially with the states highly distributed by a large quantity of outliers like $V2$ and $V4$. More particularly, it is witnessed after multiple training times that there is a strong reduction in the amount of underestimation for the low-level voltage in all the voltage estimations compared to that of the tests with $tanh()$ as an output activation. This strongly indicates the significant modification regarding the activation function of the last layer. By replacing $tanh()$ with $arctan()$, the estimation of the low voltages is improved considerably. This can be explained by the flatter curve and the wider converging range of $arctan()$ with $[-\pi/2\ ,\pi/2]$ compared to that of $arctan()$ with

$[-1\ ,1]$, which allows the higher tendency to differentiate between closed input values. Moreover, containing mainly unit voltage values around 1 after preprocessing and only a low proportion of negative values of P and Q also makes the narrow conversion range of $tanh()$ deviates the low or dropping voltages toward zero, introducing a great deal of unqualified RPE outliers.

In brief, the application of HPO on training parameters selection makes the general errors of state estimations significantly declined. Specially, $arctan()$ appears to be a proper function for the non-linear activation for the decoder's output to deal with the pre-processed power unit of the input data, which plays an important role in the differentiation between closed values, especially in adapting with the wide data range difference between P, Q and V as in the context of this power-related problem. Therefore, AE-PSO estimator is more preferable in terms of both error and time performance to be further tested and improved in the realistic case.

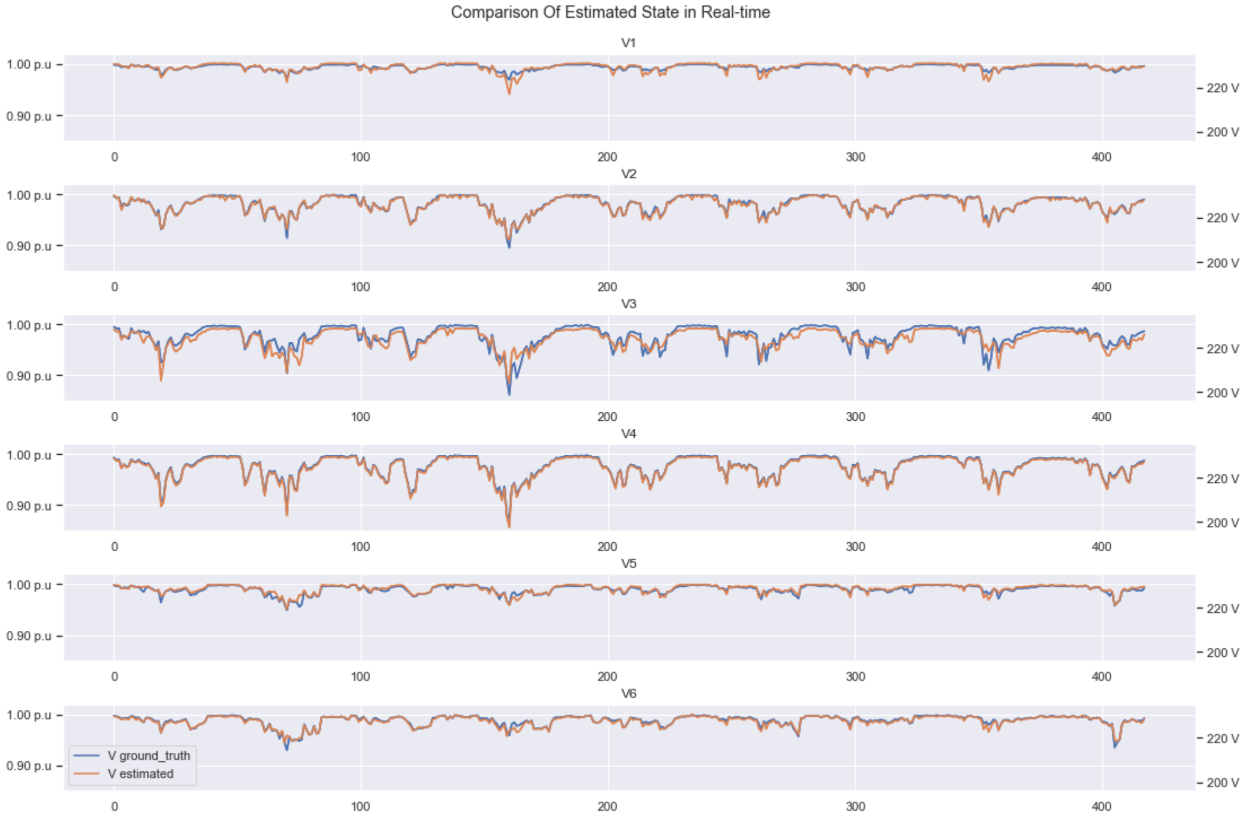


Figure 33: *Real-time Comparison with Constrained AE-PSO Method.*

4.3 Application on Realistic Case with 25-bus System

The dimensions of added layers on both encoder and decoder in the Deep AE are defined by the lower bound average of input and latent vector size. Therefore, the number of parameters for hidden layers' sizes in HPO process is similar to that of the conventional AE by considering the dimension of the latent dim only. In comparison with the input size with 75 parameters, the overcomplete and undercomplete structure can be visualized for these considered approaches.

Table 17: *Suggestions of Hyperparameters for AE and Deep AE.*

Model	Batch size	Epochs	Data size	Learning Rate	Latent Dim	ReLU	Last Activation
AE	1	10	1867	0.00117395	90	$True$	$arctan()$
Deep AE	1	22	1967	0.00089432	62	$True$	$arctan()$

As can be seen in the suggestions of HPO for the two approaches, on a relatively equivalent number of data samples, a similar PSO set of particles (45) and iterations (90); AE requires a larger size of latent representation with overcomplete form while a decreasing dimension in hidden layers of encoder is witnessed in the deep structure. This indicates the ability of the added layers in Deep AE on capturing features instead of increasing the latent dimension in AE. Because of this, the deep format requires approximately doubled training epochs with a slightly slower learning rate. Concerning the amount of time required, it takes approximately only 1 minute and 1 second to pre-train the Deep AE and estimate the state by PSO respectively, demonstrating a high reactivity of the approach on real-time estimation.

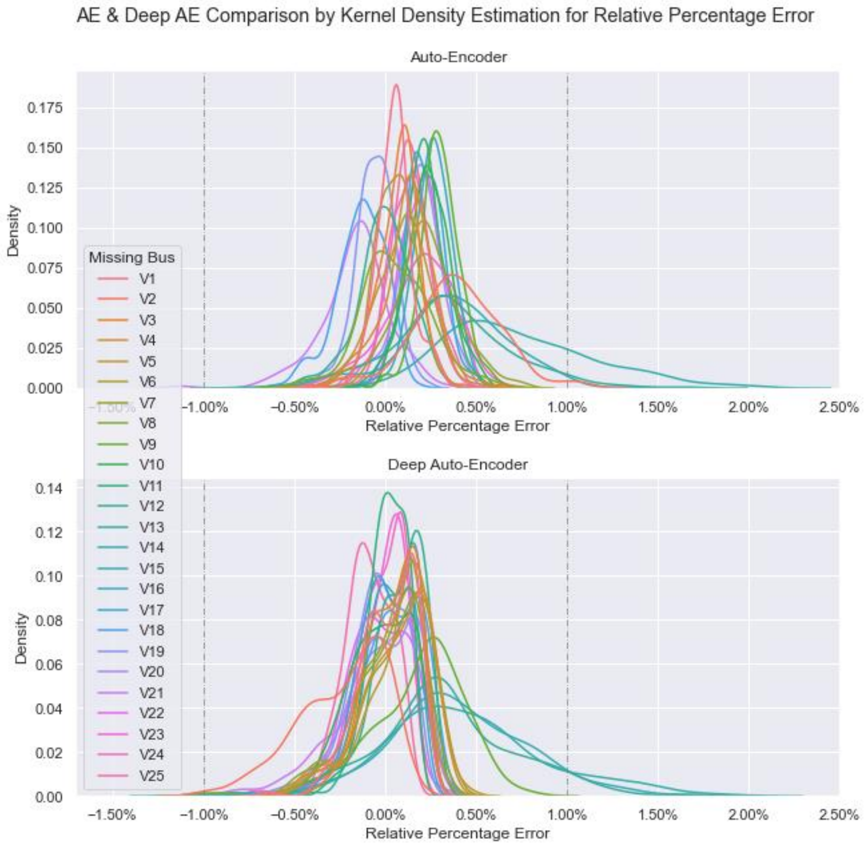


Figure 34: *Comparison by KDE on Relative Percentage Error of AE & Deep AE.*

Similar to the training scheme of the 6-bus system, both of the considered models are tested with missing data on each bus individually. Overall, the KDE plot represents a significantly large distribution of RPE within the qualified range, showing an acceptable result of both methods. Concerning voltages under normal operation, nearly all the errors of them are kept in the qualified range (mainly 100%) as can be particularly seen in ***Figure 35***. Even so, the RPE boxplot of the deep method is witnessed with more stable distribution around 0% while that of the 1-latent vector one contains more fluctuations. This can be clearly observed in ***Figure 34*** with a relatively dense concentration in the $[-0.5\ ,0.5]$ % range, compared to the sparser density of AE method. More importantly, the appearance of outliers performed by Deep AE is significantly reduced with absolutely 0% on the underestimation side. This proves a high adaptability and robustness of deep structure on capturing complex characteristics as in the sudden changes in state.

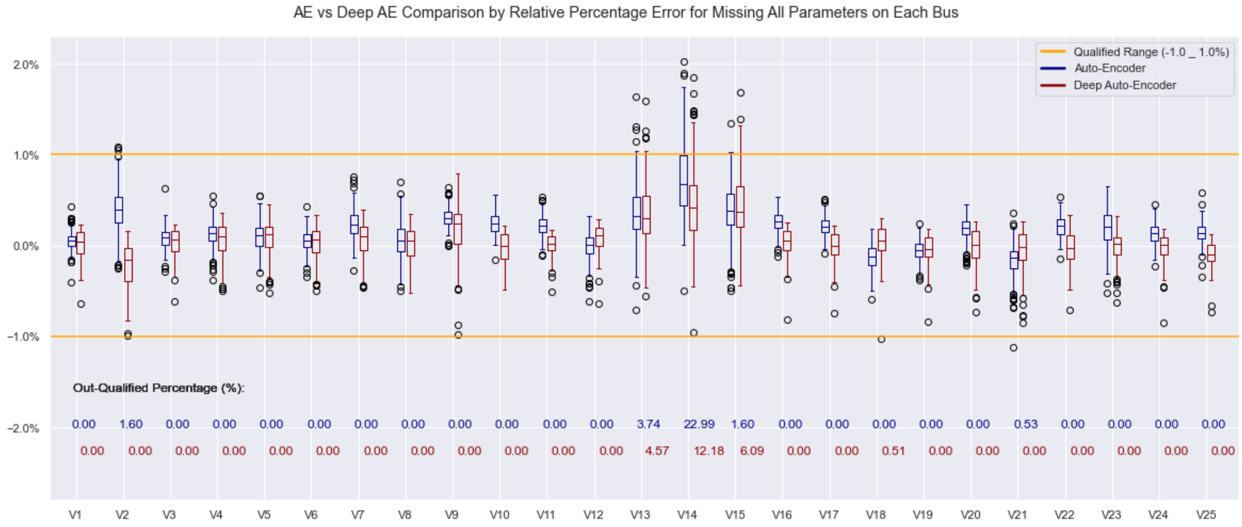
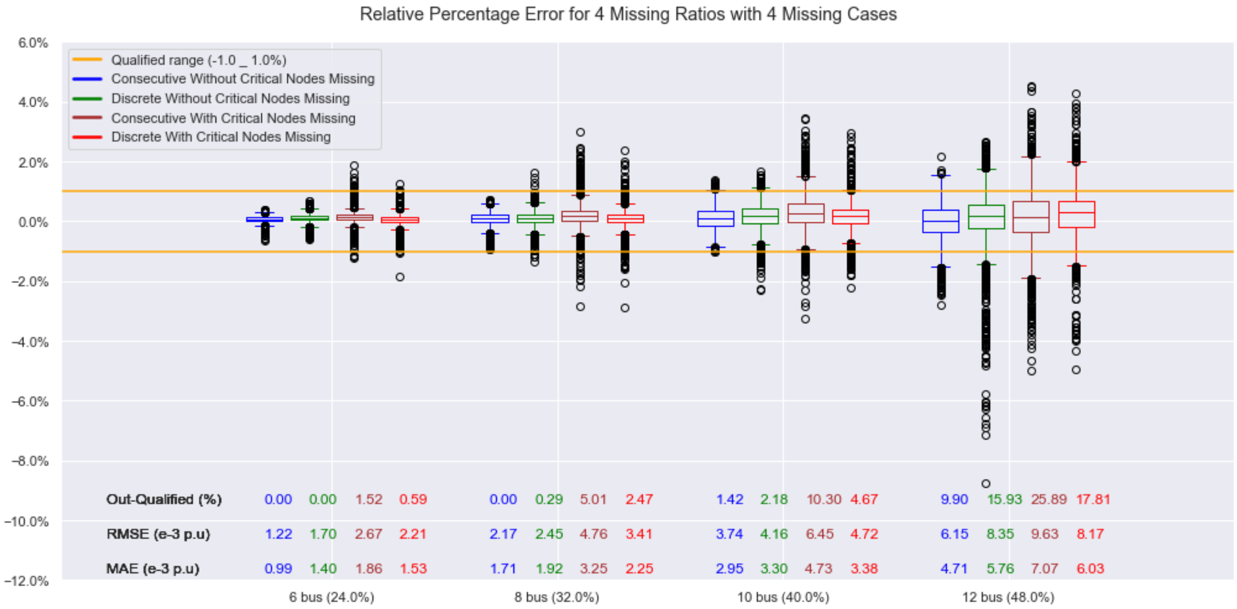


Figure 35: *Comparison by Boxplot on Relative Percentage Error of AE & Deep AE.*

Regarding the performance on critical voltage drops in $V13$, $V14$ and $V15$ however, both methods represent a considerable underestimation result with relatively high deviations from the 0% range. In particular, the conventional approach gives slightly lower percentage of non-qualification for $V13$ and $V15$ with 3.74% and 1.6% orderly whereas nearly one fourth of the relative errors in the middle node of them are witnessed outside the qualified range. While these results by Deep AE acquire a steadier pattern with nearly 2 times lower in estimating $V14$ (12.18%). Also, the RPE of this method is still observed in a relatively low error range (approximately 2% maximum) with an acceptable out-qualified percentage (around $5-10\%$) in dealing with this severe dropping condition.

In brief, even still obtain a certain amount of error in critical voltage drop estimation, the addition by 2 more hidden layers allows the Deep AE method to perform by far a more stable result compared to that of the conventional single latent representation. Therefore, this undercomplete deep structure is selected to be further tested for more intricate missing cases concerning the decreasing proportion of available measurements in this realistic complex 25-node context.



Out-Qualified (%)	0.00	0.00	1.52	0.59	0.00	0.29	5.01	2.47	1.42	2.18	10.30	4.67	9.90	15.93	25.89	17.81
RMSE (e-3 p.u)	1.22	1.70	2.67	2.21	2.17	2.45	4.76	3.41	3.74	4.16	6.45	4.72	6.15	8.35	9.63	8.17
MAE (e-3 p.u)	0.99	1.40	1.86	1.53	1.71	1.92	3.25	2.25	2.95	3.30	4.73	3.38	4.71	5.76	7.07	6.03

Figure 36: *Consecutive & Discrete with & without Critical Nodes Missing in Increasing Percentage of Missing Bus.*

During the course of testing Deep AE-PSO's adaptability on cases with multiple missing nodes, a negligible proportion of out-qualified estimations is witnessed until 5 bus for all 4 tested cases. Therefore, 4 levels of missing percentage are represented to consider the threshold of model's performance from 24% to 48%. In general, the inclusion of critical nodes enlarges both the quantity of non-qualified estimations and average difference in terms of RMSE and MAE. In particular, the main parts of the boxplot including interquartile range and whiskers are kept within the $[-1,1]\%$ range for up to 10-bus missing case. Moreover, the out-qualified percentage is recorded to be from 0% to about 5% for this extent, proving a high DSSE performance of the approach by only requiring at least 60% data availability on the grid. Starting from 11 missing case, the whisker ranges of the RPE's boxplot are extended and produces a dramatical surge in error. This can be clearly seen in the out-qualified percentages of all 4 cases (approximately from 10% to 26%) with only a half number of nodes measurement. This escalation in the frequency of poor estimation can also be observed with nearly doubled RMSE in this 12-bus case compared to the one with only 2 more buses measured. Particularly in the 3 first situations, the records of RMSE are only 1.0×10^{-3} approximately deviated from those of MSE; whereas over 2.0×10^{-3} difference is represented for the last one, illustrating a large influence of outliers detected in this case.

Concerning the testing categories, the model has a lower quantity of errors without taking into account the high voltage fluctuation as can be seen in the state distribution of $V13$, $V14$ and $V15$. Adding these nodes on the missing list, the RMSE can be nearly doubled ($4.76 \times 10^{-3}\ p.u$ and $2.17 \times 10^{-3}\ p.u$ in 8-bus missing case), showing a larger density of deviated estimations. Particularly, producing measurements bounding a series of missing nodes introduces a higher performance with only 1.42% out-qualified and about $3.0 \times 10^{-3}\ p.u$ MAE compared to that of the nonconsecutive case with 2.18% and $3.3 \times 10^{-3}\ p.u$ MAE respectively in the 40% missing case. However, this characteristic is significantly changed when the critical voltage drops are considered with the dramatical increase in both out-qualified percentage and RMSE of the consecutive spanning scheme with 10.3% and $6.45 \times 10^{-3}\ p.u$ compared to only 4.67% and $4.72 \times 10^{-3}\ p.u$ in the discrete one. This demonstrates high correlations of the buses which are closely located and connected within a certain feeder. Because of this, by providing the estimator at least 1 correlated bus, the states on a series of surrounding buses can be potentially estimated with low error. Even so, when there is no measurement on consecutive buses with critical voltage drop, the low correlation with other normal buses can detrimentally impact the performance of state estimation.

CHAPTER 5. CONCLUSIONS & PERSPECTIVES

To conclude, a broad lecture review allows an overview to compare the performance of different current machine learning methods on DSSE task. By doing this, Auto-Encoder approach is selected to apply on the tested simple 6-bus grid before being further improved in the realistic 25-node system. On top of that, the application of hyperparameters optimization provides a list of suggestions for the training section on a highest performance for a certain range of searching range and a quick manner for a certain grid context. More importantly, the extension in the hidden layers appears to be a suitable model construction to adapt more steadily in multiple-nodes complex structure. This is capable of both reducing the density of outliers or out-qualified estimations and stabilizing the deviation with respect to the ground-truth value. Even obtaining a low relative percentage of underestimation, the inclusion of missing critical voltage drops still produces a relatively large estimating deviation. Furthermore, the Deep AE-PSO estimator is able to still acquire a significant adaptability with up to 40% missing bus giving a sufficient number of measurements that respect the consecutive correlation for the normal voltage and the discrete one for the critical nodes.

Regarding the model's cost, the amount of time required to train a Deep AE for each grid context is relatively low (1 minute for the 1-phase 25-node system). More importantly, after this phase, this pre-trained model can be flexibly applied in PSO estimation for every scenario of missing buses. This is because the symmetrical structure of the Auto-Encoder allows an identical number of parameters in the input and output so that the estimation can be implemented on the positions marked by missing values in various types of situations. This is highly beneficial compared to the direct-target approaches whose NNs needs training again for different missing case scenario. In addition, although requiring relatively high number of iterations and particles (90 & 45 in the tested realistic system), it is witnessed that PSO only takes approximately 1 second to give the estimation for each instance. This represents a potentially quick response of the proposed method in real-time operation. In brief, by detaching training and estimating phases, Deep AE-PSO estimator can use the pre-trained AE based on the historical database of a certain grid as an objective function to flexibly estimate the state in various missing contexts of the grid in reality.

In terms of planification for the next stage of the project, it is highly recommended that the adaptability of the Deep AE-PSO method can be examined on the 3-phase consideration of the same realistic 25-bus system so that the performance of this approach can be further verified on a more reality-based complex case,

REFERENCES

[1] A. S. Zamzam, X. Fu and N. D. Sidiropoulos, "Data-Driven Learning-Based Optimization for Distribution System State Estimation," in IEEE Transactions on Power Systems, vol. 34, no. 6, pp. 4796-4805, Nov. 2019, doi: 10.1109/TPWRS.2019.2909150.

[2] Z. Cao, Y. Wang, C. -C. Chu and R. Gadh, "Fast State Estimations for Large Distribution Systems using Deep Neural Networks as Surrogate," 2020 IEEE Power & Energy Society General Meeting (PESGM), 2020, pp. 1-5, doi: 10.1109/PESGM41954.2020.9281827.

[3] G. Hong and Y. -S. Kim, "Supervised Learning Approach for State Estimation of Unmeasured Points of Distribution Network," in IEEE Access, vol. 8, pp. 113918-113931, 2020, doi: 10.1109/ACCESS.2020.3003049.

[4] K. Dehghanpour, Z. Wang, J. Wang, Y. Yuan and F. Bu, "A Survey on State Estimation Techniques and Challenges in Smart Distribution Systems," in IEEE Transactions on Smart Grid, vol. 10, no. 2, pp. 2312-2322, March 2019, doi: 10.1109/TSG.2018.2870600.

[5] P. N. Pereira Barbeiro, H. Teixeira, J. Pereira and R. Bessa, "An ELM-AE State Estimator for real-time monitoring in poorly characterized distribution networks," 2015 IEEE Eindhoven PowerTech, 2015, pp. 1-6, doi: 10.1109/PTC.2015.7232679.

[6] A. S. Zamzam and N. D. Sidiropoulos, "Physics-Aware Neural Networks for Distribution System State Estimation," in IEEE Transactions on Power Systems, vol. 35, no. 6, pp. 4347-4356, Nov. 2020, doi: 10.1109/TPWRS.2020.2988352.

[7] V. Miranda, J. Krstulovic, H. Keko, C. Moreira and J. Pereira, "Reconstructing Missing Data in State Estimation With Autoencoders," in IEEE Transactions on Power Systems, vol. 27, no. 2, pp. 604-611, May 2012, doi: 10.1109/TPWRS.2011.2174810.

[8] S. Chatzivasileiadis, A. Venzke, J. Stiasny and G. Misyris, "Machine Learning in Power Systems: Is It Time to Trust It?," in IEEE Power and Energy Magazine, vol. 20, no. 3, pp. 32-41, May-June 2022, doi: 10.1109/MPE.2022.3150810.

[9] Y. Lin, J. Wang and M. Cui, "Reconstruction of Power System Measurements Based on Enhanced Denoising Autoencoder," 2019 IEEE Power & Energy Society General Meeting (PESGM), 2019, pp. 1-5, doi: 10.1109/PESGM40551.2019.8973925.

[10] Pereira, Ricardo Cardoso & Santos, Miriam & Rodrigues, Pedro & Henriques Abreu, Pedro. (2020). Reviewing Autoencoders for Missing Data Imputation: Technical Trends, Applications and Outcomes. Journal of Artificial Intelligence Research. 69. 1255-1285. 10.1613/jair.1.12312.

[11] S. Ryu, M. Kim and H. Kim, "Denoising Autoencoder-Based Missing Value Imputation for Smart Meters," in IEEE Access, vol. 8, pp. 40656-40666, 2020, doi: 10.1109/ACCESS.2020.2976500.

[12] Sharma, Siddharth & Sharma, Simone & Athaiya, Anidhya. (2020). ACTIVATION FUNCTIONS IN NEURAL NETWORKS. International Journal of Engineering Applied Sciences and Technology. 04. 310-316. 10.33564/IJEAST.2020.v04i12.054.

[13] Ahmad, A. (2017, November 13). An overview of activation functions used in neural networks. GitHub. https://adl1995.github.io/an-overview-of-activation-functions-used-in-neural-networks.html#:%7E:text=ArcTan,0%20for%20large%20input%20values.

[14] GeeksforGeeks. (2020, October 24). *Intuition of Adam Optimizer*. https://www.geeksforgeeks.org/intuition-of-adam-optimizer/#:%7E:text=Adam%20optimizer%20involves%20a%20combination,minima%20in%20a%20faster%20pace.

[15] Miranda, V. & Win Oo, N. New experiments with EPSO – Evolutionary Particle Swarm Optimization. INESC Porto and also FEUP, Faculty of Engineering of the University of Porto, Portugal.

[16] V. Miranda and N. Fonseca, "EPSO-evolutionary particle swarm optimization, a new algorithm with applications in power systems," IEEE/PES Transmission and Distribution Conference and Exhibition, 2002, pp. 745-750 vol.2, doi: 10.1109/TDC.2002.1177567.

[17] Clerc, Maurice. (2010). Particle Swarm Optimization. Particle Swarm Optimization. 10.1002/9780470612163.

[18] Hutter, Frank; Kotthoff, Lars; Vanschoren, Joaquin (2019). [The Springer Series on Challenges in Machine Learning] Automated Machine Learning (Methods, Systems, Challenges), doi:10.1007/978-3-030-05318-5

[19] Winastwan, R. (2022, January 4). *Hyperparameter Tuning of Neural Networks with Optuna and PyTorch*. Medium. https://towardsdatascience.com/hyperparameter-tuning-of-neural-networks-with-optuna-and-pytorch-22e179efc837

[20] García, Salvador; Luengo, Julián; Herrera, Francisco (2015). [Intelligent Systems Reference Library] Data Preprocessing in Data Mining Volume 72, doi:10.1007/978-3-319-10247-4

[21] Kotsiantis, Sotiris & Kanellopoulos, Dimitris & Pintelas, P.. (2006). Data Preprocessing for Supervised Learning. International Journal of Computer Science. 1. 111-117. /hyperparameter-tuning-of-neural-networks-with-optuna-and-pytorch-22e179efc837

[22] Muralidharan, K.. (2010). A Note on Transformation, Standardization and Normalization.

[23] Naser, M. Z., & Alavi, A. (2020). Insights into performance fitness and error metrics for machine learning. *arXiv preprint arXiv:2006.00887*.

[24] Galarnyk, M. (2022, July 25). *Understanding Boxplots - Towards Data Science*. Medium. https://towardsdatascience.com/understanding-boxplots-5e2df7bcbd51

[25] Chen, Yen-Chi (2017). A tutorial on kernel density estimation and recent advances. Biostatistics & Epidemiology, 1(1), 161–187. doi:10.1080/24709360.2017.1396742

[26] J. (2021, January 3). *MAE and RMSE — Which Metric is Better? - Human in a Machine World*. Medium. https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d#:%7E:text=The%20RMSE%20result%20will%20always,from%20a%20single%20test%20sample.

APPENDICES

4.4 Pretrained AE for PSO estimation – Unconstrained & Constrained Search

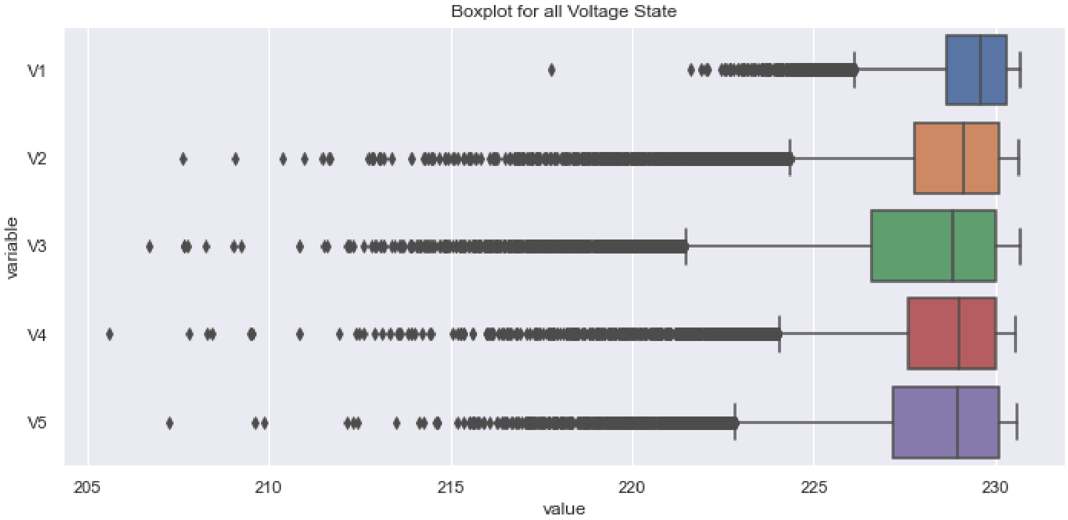
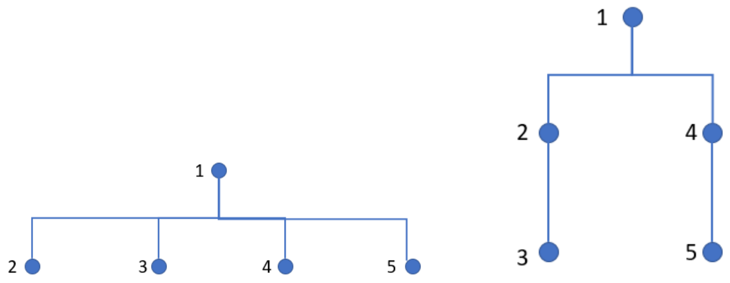


Figure 37: *Boxplot for Simulated Voltage Distribution in Single-phase 5-bus systems.*

In particular, a 5-bus system is examined using a single supplying source and 4 other simple feeders each of which contains only 1 bus ***Figure 24*** *(a)*. As can be seen in ***Figure 23***, the states of these feeders are highly influenced by outliers with voltage drop ranging approximately below 220V. This system is projected to be tested for DAN using random missing data whose values are weighted ($equation$ 19) with high bias ($\alpha = 0.8$) in training section. Also, pretrained-AE for PSO estimation strategy is also applied for this system using $\alpha = 1$ and $\alpha = 0$ for constrained and unconstrained PSO search respectively. The number of batch size, data size, latent dimension and learning rate are chosen based on experience on some certain times of training. Concerning the activation function on the last layer, $tanh()$ is tested initially on this simple case.



(a) *(b)*

Figure 38: *Single-phase 5-bus systems.*

To validate the uniqueness of pretrained-AE model for different topology type, another 5-bus system is considered with 2 buses on each of 2 feeders ***Figure 23*** *(b)*. To achieve this, each AE trained on these topologies is planned to applied for the objective function of the other on PSO estimation to examine the ability of the model to be used on different bus topologies.

Apart from using direct reconstruction for state estimation in DAE with data corruption, the AE is trained with the complete form of each sample before being used as an objective function for PSO estimation. In particular, the algorithm searches for the optimal position of the particles in the missing dimension space to minimize the error they produce at the output. The alpha weight is set to 1 to put the weights equally to all parameters (unconstrained search) and 0 for the missing indices only (constrained search). The number of particles and iterations is selected 6 and 60 respectively.

Table 18: *Settings of Training Parameters for Unconstrained AE-PSO.*

Batch size	Epochs	Data size	Alpha Weight	Latent Dim	Last Activation
1	20	900	1	20	$tanh()$

Table 19: *Settings of Training Parameters for Unconstrained AE-PSO.*

Batch size	Epochs	Data size	Alpha Weight	Latent Dim	Last Activation
1	25	1500	0	18	$tanh()$

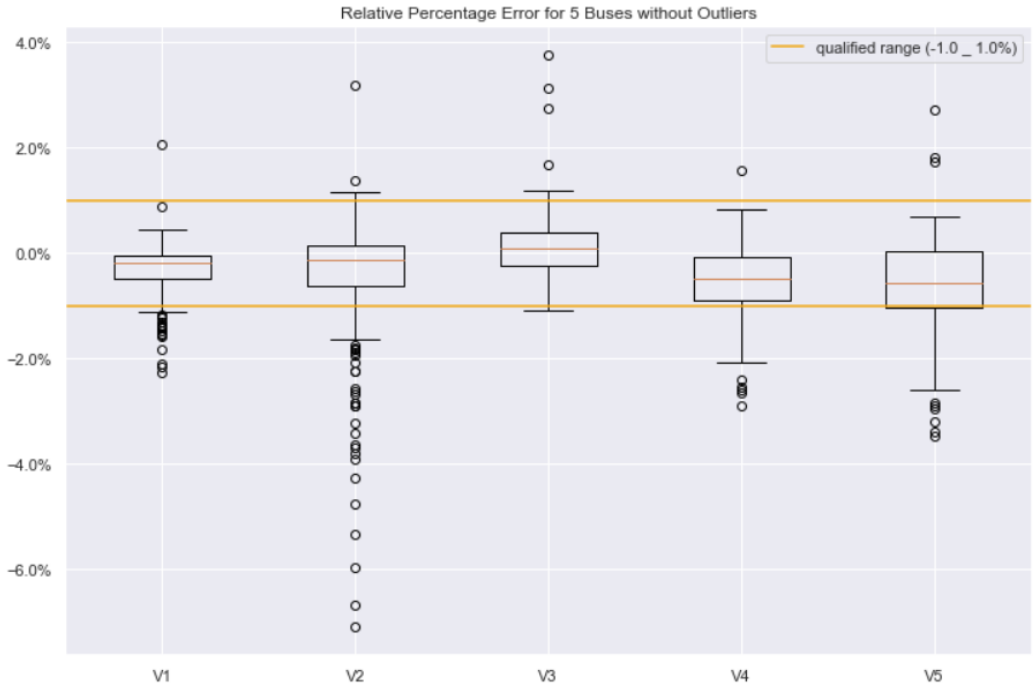


Figure 39: *DSSE boxplot error by Unconstrained Search AE-PSO for 4-feeder 5-bus System.*

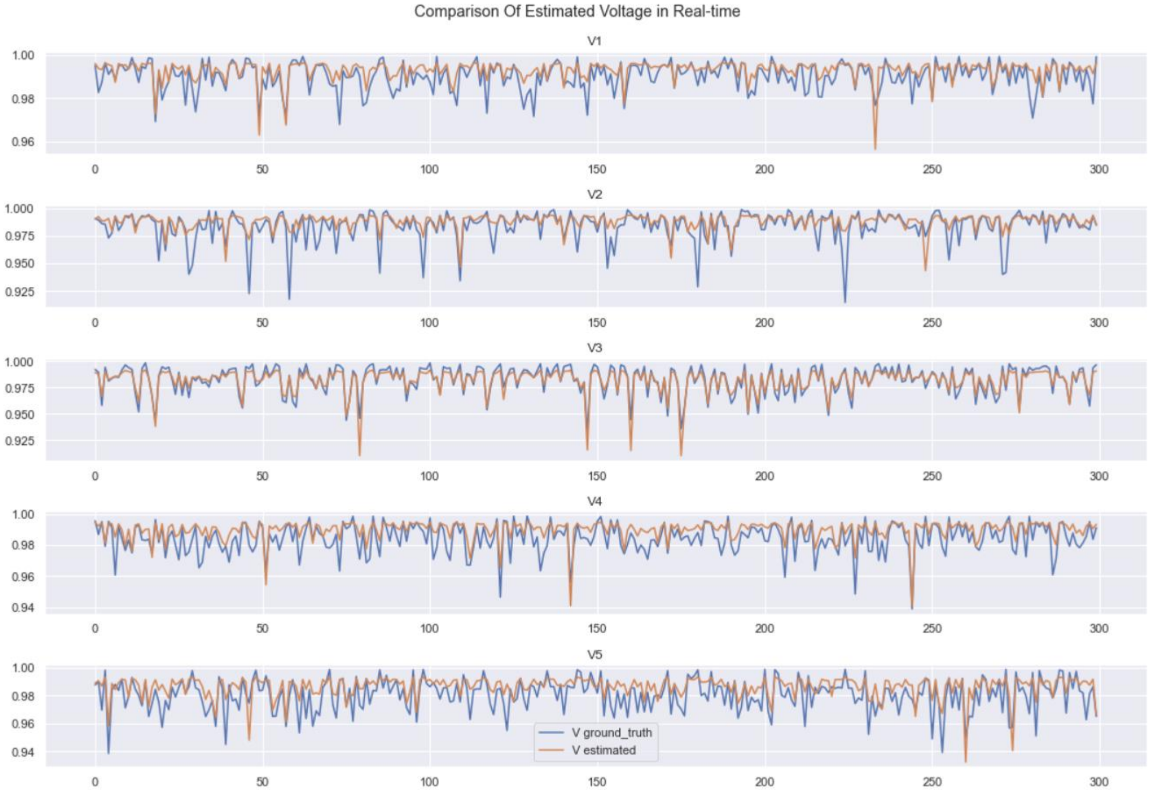


Figure 40: *Real-time Comparison with Unconstrained AE-PSO Method.*

As can be seen in the boxplot of both search types, the constrained search is able to produce nearly all the RPEs distributed between the “minimum” and “maximum” within the qualified range. Moreover, the middle values of them are allocated closed to zero instead of being skewed towards negative side as in the unconstrained category. This also makes the real-time expression of estimated Voltages of the constrained type approaches closer to the ground-truth ones compared to that of the other type with high intensity of deviation for all states. It proves a higher robustness of targeting the optimal positions of the missing values rather than diffusing the weights to other parameters which can be used to direct the converge of missing ones. This creates a concentration on the indices requiring the estimation and facilitates the optimization task. As a result, this constrained method with $\alpha = 1$ is more appropriate to the task and is projected to be selected for other trials. However, strong voltage-drop impact is still witnessed in the error due to high relative percentage of outliers, especially with the states in the feeding branches. As can be particularly seen in the real-time plot of the constrained search, the estimations for voltage drops are pull down toward the negative orientation compared to these of other values, intensifying the distribution of RPE’s outliers.

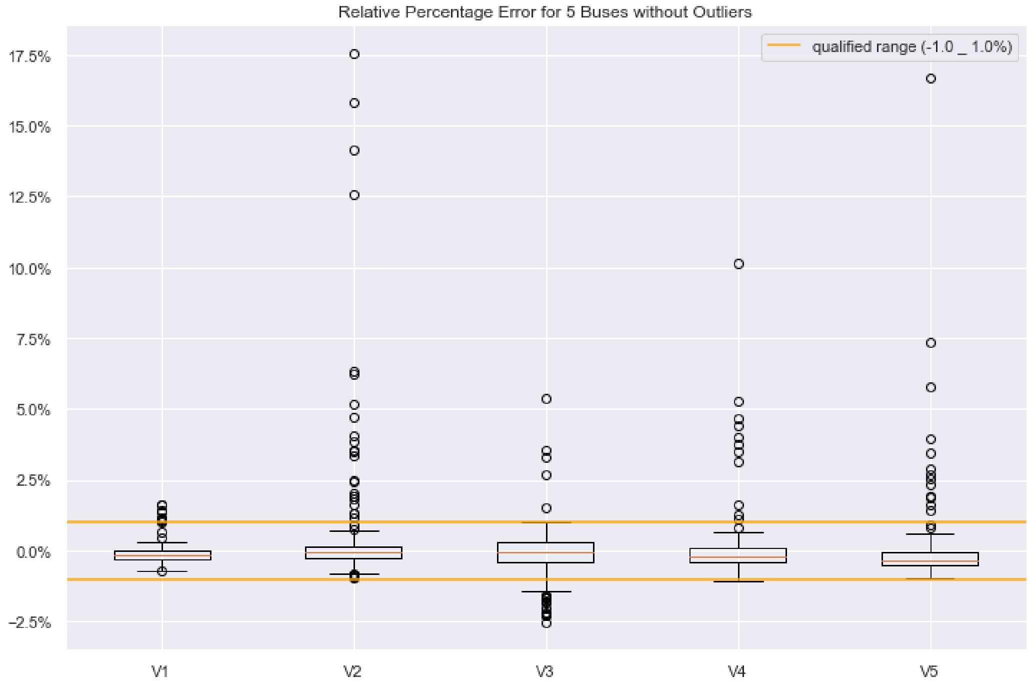


Figure 41: *DSSE boxplot error by Constrained Search AE-PSO for 4-feeder 5-bus System.*

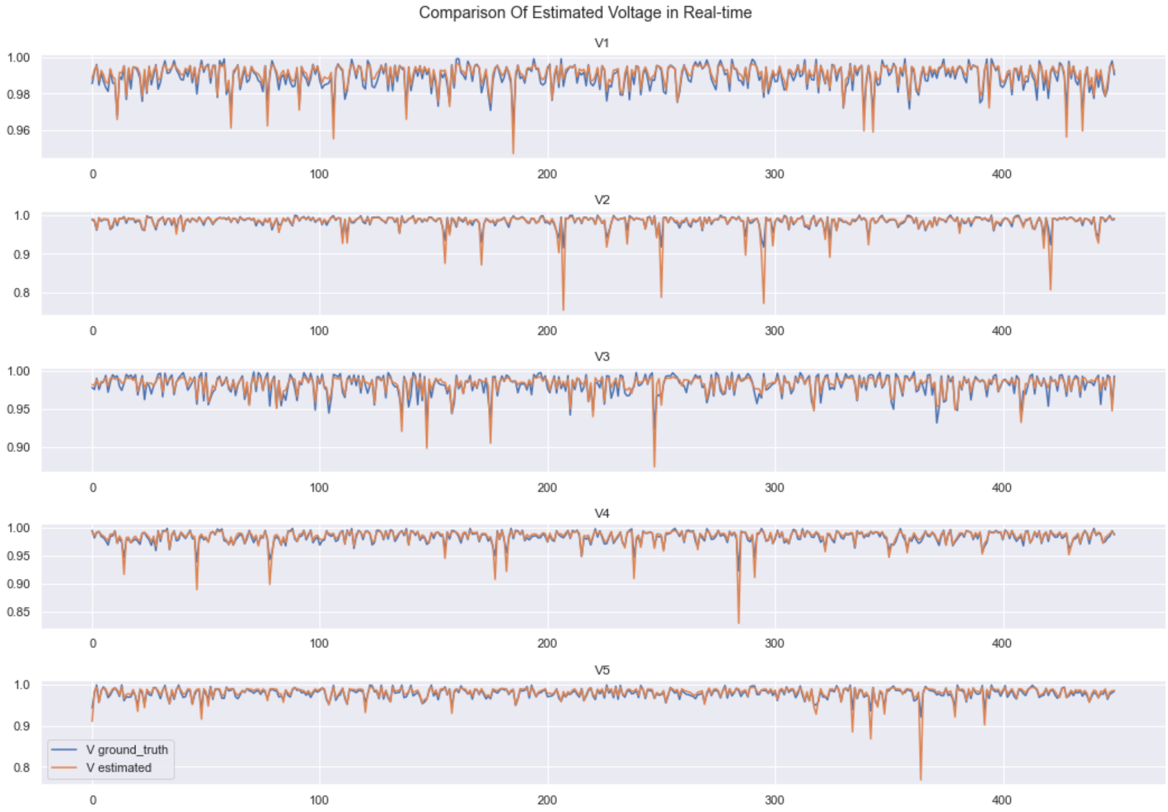


Figure 42: *Real-time Comparison with Constrained AE-PSO Method.*