

# DATS 6203 Final Project Group-4 Report

Maeshal Hijazi, Yifu Li, and Jinshun Su

December 8, 2020

# 1 Introduction

This project investigates the effective utilization of the Autoencoder (AE) neural network model for online identification of the distribution system (DS) topology based on a large amount of real-time phasor measurement units (PMUs) detection data. In subsection 1.1, we provide background information on several recent high-impact low-probability (HILP) events that led to prolonged electricity outages and highlight the criticality of DS resiliency and reliability. In subsection 1.2, we discuss the current practice for DS topology identification through PMUs data. Eventually, the report outline is presented in subsection 1.3.

## 1.1 Background

In recent years, there has been an increase in the frequency and magnitude of HILP events—which, according to recent statistics [1], have resulted in excessive equipment damages, prolonged electricity outages, significant economic losses, and disruptions in our modern society. In the USA in particular, a total of seventy weather-driven disasters occurred from 2015 to 2019 and resulted in billions of dollars of costs [2]. Seven major blackouts in the U.S. history lasted between 10 and 50 hours, with the associated costs exponentially increasing as the duration of the outage increases. Example events with major electricity outages are: 1) the 2017 Hurricane Harvey causing substantial electricity outages (around 10,000 MW) and leaving more than 291,000 people without power; 2) the Hurricane Sandy in 2012 resulting in 10 percent of customers in New York and New Jersey without power for 10 days, \$14 to \$26 billion economic losses, and 50 deaths due to the sustained outage of electricity [1].

The power system serves as the backbone for lifeline networks and drives a myriad of interdependent systems and mission-critical services, such as water, communication, transportation, health, military, and government sectors and services. A bulk power system mainly consists of three hierarchical levels including generation system, transmission system, and DS. The generation system owns different types of power plants to produce electric power, mostly located far away from the demand centers. The gener-

ated electricity is transmitted over hundreds of miles through the transmission system from distant power plants to demand centers. In each demand center, a substation is located facilitating the transfer of electricity from the transmission system to the DS, the DS then brings the electricity to individual customers. While all three segments of the bulk power system might be vulnerable to extreme events, the DS provides the last mile electricity connection to the end-consumers and is particularly vulnerable due to its radial topology. Any disruption in the DS may directly and swiftly translate into customer interruptions and power outages. With the increasing frequency and intensity of HILP events, it is primordial to identify and estimate the real-time DS topology following these events, which can contribute to ensure continuous, secure, and reliable supply of electricity to the end-use consumers particularly mission-critical services.

## 1.2 Current Practice for DS Topology Identification by PMUs

In order to generate and dispatch the power grid stably and safely, especially under a HILP event, system operators need to be informed of the DS topology at all time. With increasing complexity in the DS structure growingly reinforced with heterogenous resources and the increasing demand for electricity needed for an electrified economy, PMUs have been introduced and widely deployed to observe the dynamic performance of the power grid with synchronized measurements. A PMU is a device used measure the magnitude and phase angle of an electrical phasor quantities (e.g., voltage and current) in DS applying a common time source for synchronization [3]. Due to the fact that the time is synchronized by the Global Positioning System (GPS), PMUs are able to capture real-time electrical phasor quantities from multiple remote points on the DS, thereby providing a real-time snapshot of the entire system making it possible to approach wide-area monitoring, protection and control.

Several recent researches [4–6] propose mathematical models for topology identification based on PMUs data. The frameworks of their models assume a DS topology first, and then measure the collected data to compare the features and determine the accuracy of the topology of previously assumed DS. Such strategies are time-consuming, less accurate, and with practical limitations. To overcome the limitations of these math-

ematical models, the study [7] presents a training artificial neural network model to automatically recognize the DS topology and solve complex problems. The learning algorithm of this study is based on a nodal voltage graphical model which only uses data of magnitude and phase angle of system's voltage from PMUs. Comparing with [7], our study employs magnitudes and phase angles of both voltage and current collected by PMUs on our proposed AE model, which can enhance accuracy and robustness of the model for DS topology identification.

### 1.3 Report Outline

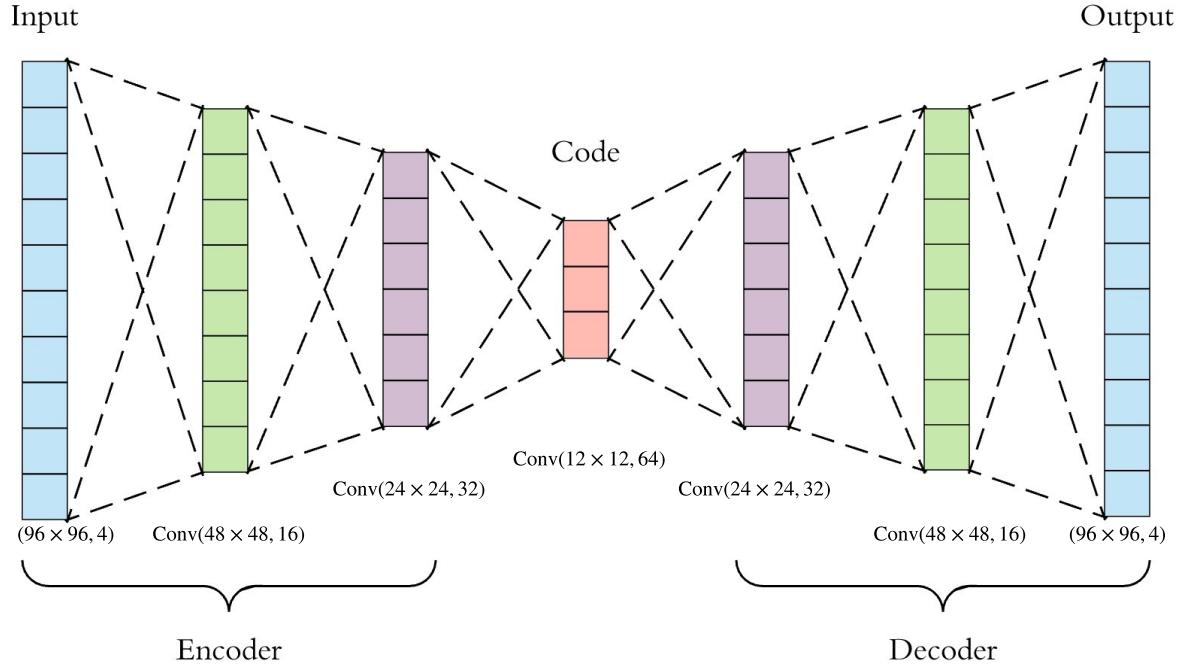
An Autoencoder (AE) is a type of artificial neural network used to learn efficient data coding in an unsupervised manner. It could be used in supervised learning and unsupervised learning. The aim of an AE is to learn a representation (encoding) for a set of data, typically for dimensionality reduction, by training the network to ignore signal “noise” [8].

In this study, the proposed AE for the heatmap classification in the IEEE 34-Node Test Feeder has the following architecture:

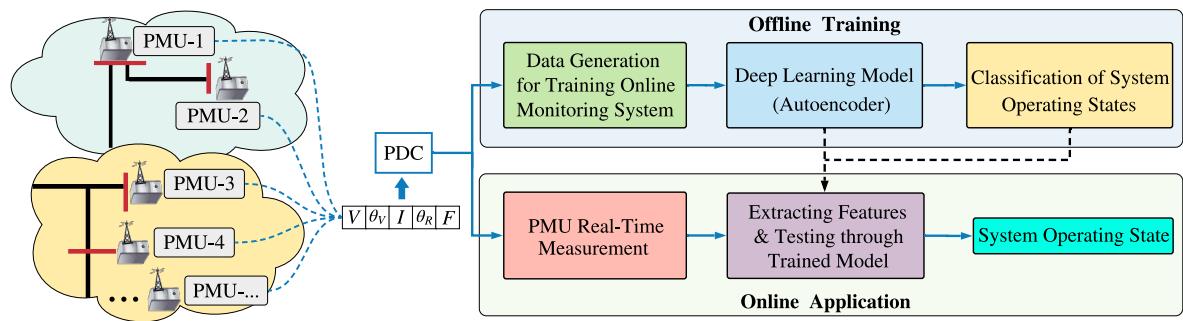
The flowchart of the proposed Autoencoder is shown in Figure 2, where the first step is to collect PMU data and normalize them into per-units. Such data with their corresponding topologies are then inputted into the neural network, and the trained network learns to identify distribution grid topology with PMU measurements. The AE used mean squared as the loss function. Finally, we used additional PMU measurements beyond the training set to verify the model accuracy.

This Autoencoder architecture will be used as a building block in the proposed framework that identifies the power distribution network topology in real-time. The relation between the PMU measurements and the AE module will be detailed in the Section 3.

The rest of this report is organized as follows: Section 2 introduces the source of dataset utilized to this project and presents the preprocessing of dataset. Section 3 describes the proposed AE model for DS topology identification. The numerical results and discussions are provided in Section 4, and finally Section 5 concludes the report.



**Figure 1:** Proposed Autoencoder architecture.

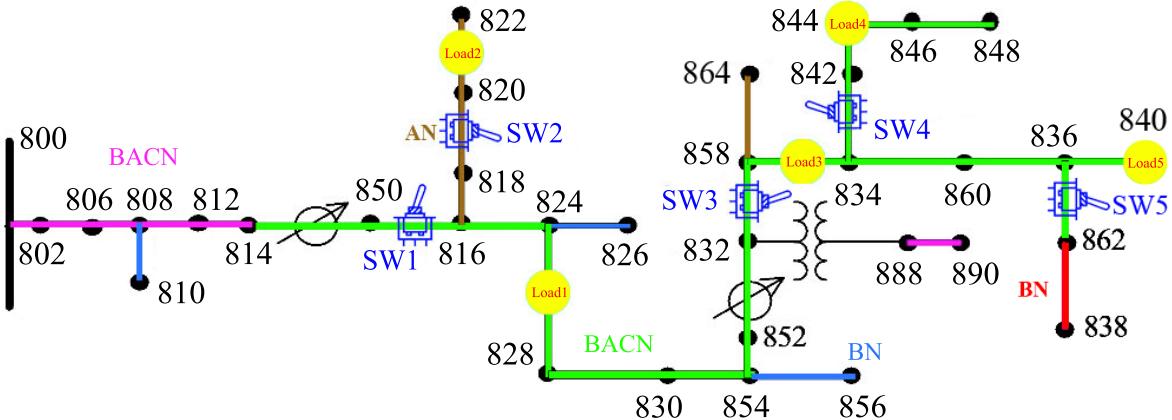


**Figure 2:** Autoencoder working flowchart.

## 2 Data Mining

### 2.1 Dataset Description

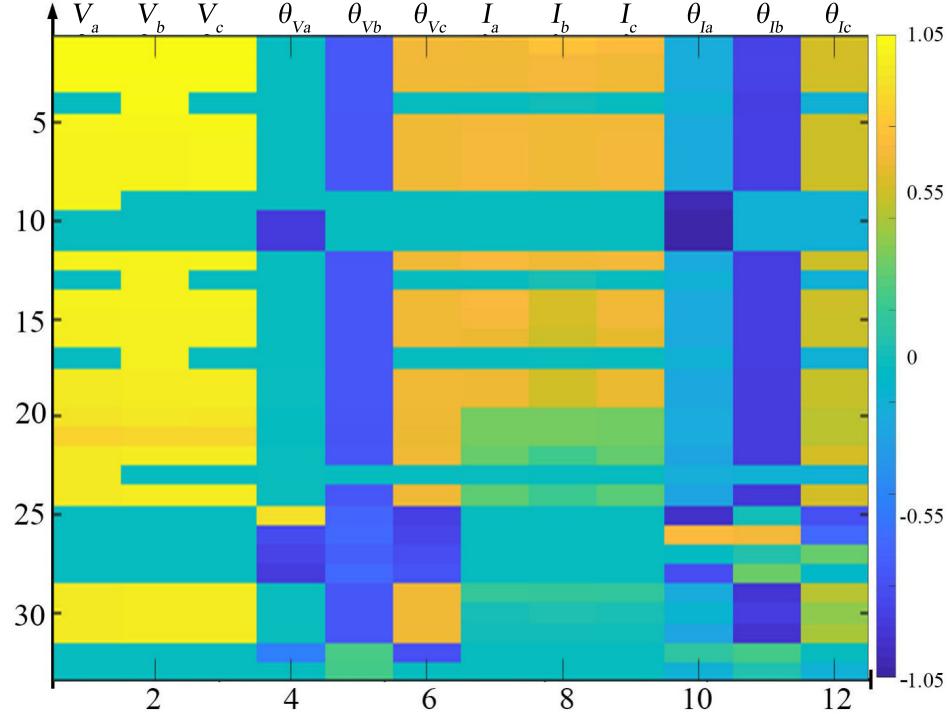
The dataset we used in this project is the PMU data which is generated from IEEE-34 node test feeder [9] by Matlab Simulink. This radial power DS is an actual feeder located in Arizona, and its topology is illustrated in **Figure 3**. The detailed information of IEEE-34 node test feeder is presented in Appendix A. Noted that IEEE-34 node test feeder is an unbalanced network, different colors are here used to mark the phasing status. For example, we used pink to mark the line 800-812 as BACN, indicating that it is a three-phase four-wire segment in DS. To gain a full observation of the IEEE-34 node test feeder, we set 33 PMUs on each node expect node 800 (substation bus). We also added 5 breakers (see **Figure 3**: SW1-SW5) in order to generate different network topologies for the AE training dataset. To generate more scenarios under one topology, we marked 5 loads (see **Figure 3**: Load1-Load5), which allows the PMU data could be varying under different realization of the load demand.



**Figure 3:** IEEE 34-Node Test Feeder Scenarios.

Upon simulating each scenario in Matlab Simulink environment, the resulting PMU data will characterize a 33 by 12 heatmap matrix which contains three-phase voltage, three-phase voltage angle, three-phase current, and three-phase current angles (see **Figure 4** for a heatmap example). For the nodal measurements that contain single-

phase or two-phase data, we let the remaining entries be zero. We line up the data in a heatmap format into eight groups (8 labels), process the data in each group individually, and finally integrate the information in each group together. A partially connected neural network is dedicated to the processing of these groups of heatmaps.



**Figure 4:** A Heatmap Example of the Generated PMU Measurement Data Sample.

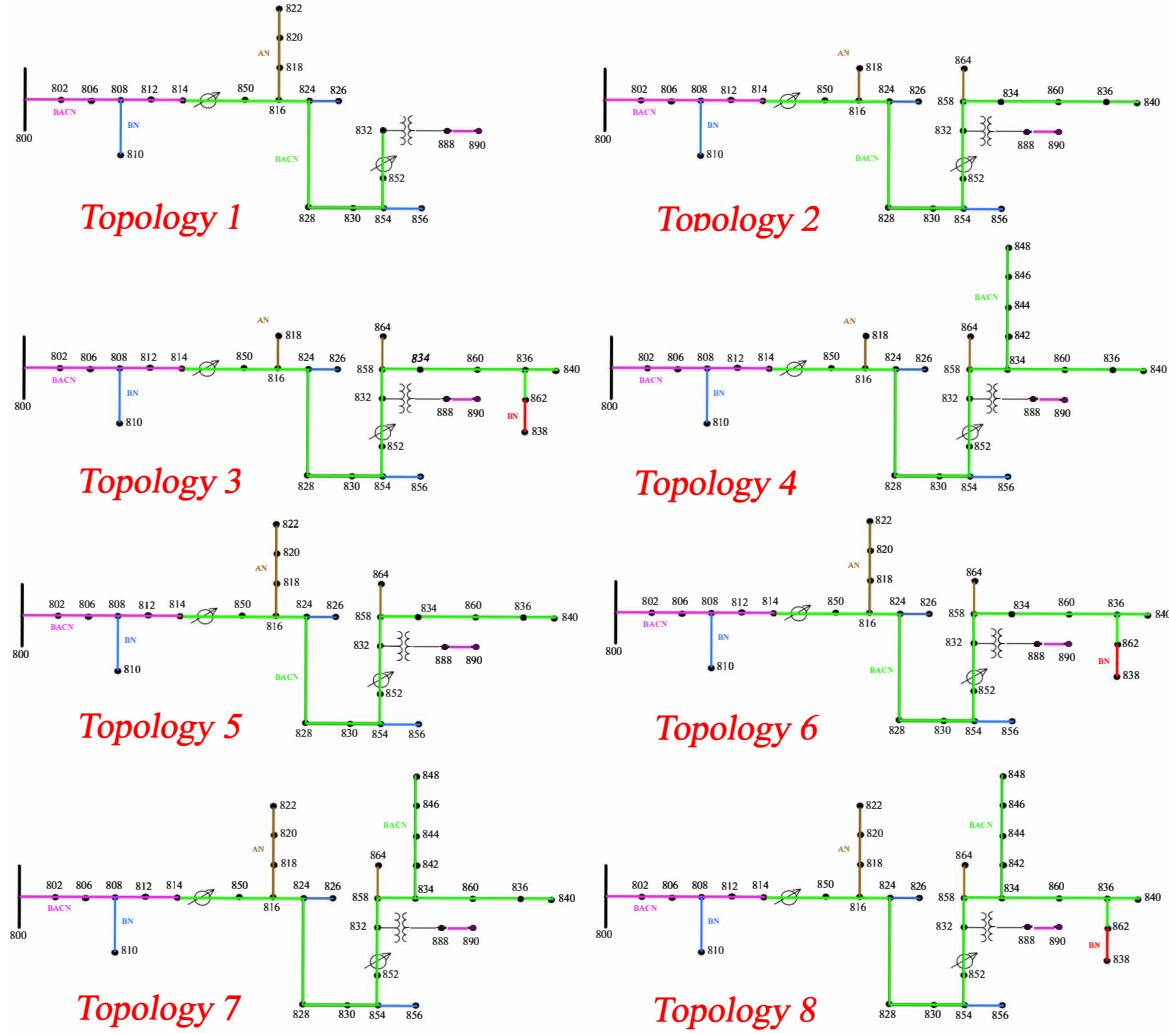
**Table 1** shows the general configuration all simulated scenarios, and the exact topologies of different scenarios are illustrated by **Figure 5**.

Take Topology 1 as an example, Breakers SW1 and SW2 are closed (status=1), the rest of Breakers are opened (status=0). For each network topology configuration, we generate different scenarios by adjusting the loads values. To keep the *balance* of training dataset and avoid generating *redundant* training data for the AE model, keen considerations were taken into account in generating different load scenarios. We let the load change amplitude distributed between 95% to 105% of the rated demand at each load point based on different number of served loads: i) Topology 1, only Load 1 and Load 2 are served through the connected distribution line and it is not necessary to adjust the remaining three load points for scenario generation. Assuming each load has 40 possible amplitudes in the constrained range above, the total number of scenarios is found  $40^2$  (1600) in this case; ii) the network Topology 2, 3 has three loads (i.e., Loads 1, 3 and 5) served, we generated  $13^3$  (2197) scenarios; iii) Topologies 4, 5, and 6 have four loads served, in which  $7^4$  (2401) scenarios are generated; iv) Under the network Topologies 7 and 8, all five loads are being served in the distribution grid, and as each one is characterized with 5 possible amplitudes for the training process, there are  $5^5$  (3125), number of scenarios generated. The total number of generated scenarios that contribute to the training dataset is found 19447. All values (e.g., magnitudes and

**Table 1:** Network Topology Realizations with the Corresponding Number of Generated Scenarios

Topology	SW1	SW2	SW3	SW4	SW5	Number of Scenarios
1	1	1	0	0	0	$1600 (40^2)$
2	1	0	1	0	0	$2197 (13^3)$
3	1	0	1	0	1	$2197 (13^3)$
4	1	0	1	1	1	$2401 (7^4)$
5	1	1	1	0	0	$2401 (7^4)$
6	1	1	1	0	1	$2401 (7^4)$
7	1	1	1	1	0	$3125 (5^5)$
8	1	1	1	1	1	$3125 (5^5)$

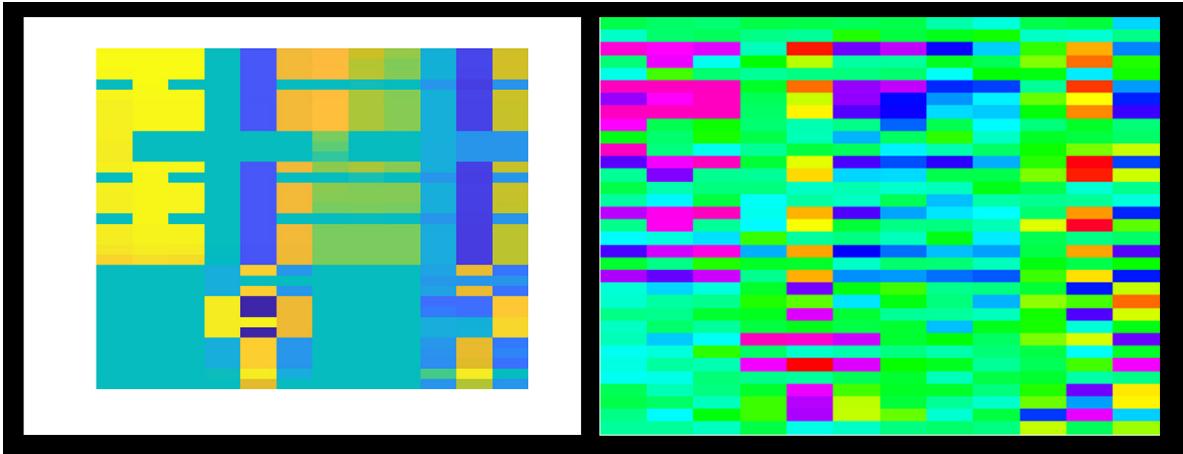
phase angles of voltage and current) obtained from PMUs are normalized based on the rule of power system calculation [10]. Thus, all numbers in this heatmap (see **Figure 4**) is ranged between 1.05 to -1.05.



**Figure 5:** All Studied Network Topologies

## 2.2 Dataset Preprocessing

In **Figure 4**, the heatmap which generated from Matlab is in "parula" format. However, when we batch-produced heatmaps, we found a problem. The white margin of the image are too wide. If it is used as input, it will cause the input data size to be too large and not compressible and affect the prediction accuracy on the neural network. So we use the pure data form .csv format of the PMU saved by Matlab, and generate heatmaps in batches through python instead, shown in the **Figure 6**.



**Figure 6:** The "parula" heatmap (left) and "gist\_rainbow" colormap (right).

So in Python, we used "seaborn.heatmap" library to generate the heatmap [11]. In general format of heatmaps, the single color depth various changes cannot distinguish the data boundary well [12], so we use a colormap format as "gist\_rainbow" in generating images as the input of the neural network.

The original size of one heatmap is 497 by 371 pixels, we compressed it into 96 by 96 pixels in the input layer. The RGB heatmap has 3 channels, and we add an additional dimension to save the corresponding label, so the size for the first layer in AE is (96, 96, 4), as the **Figure 1** shown.

## 3 Model Description

### 3.1 Ideal Model

For this project, we built an ideal model in addition to two non-ideal models. The ideal model was built in a way that the grid is not affected by any type of noise and that all 33 PMUs are available. However, this ideal model does not reflect real life scenarios, where noise is a big factor affecting the grid and that PMUs are expensive to have at each node. The goal of this ideal model was to show that AE can be implemented and can help in classifying topologies. In this model, we generated ideal images based on the IEEE 34-Node Test Feeder, and built AE in Keras [13].

For building this model, we divided our dataset to 70% for training, 20% for testing, and 10% for validation. Before building the classifier to classify the topologies, we had built an AE that can reconstruct our heatmap images. To use the generated heatmaps, we resized all images to  $96 \times 96$  and then normalized them by dividing each image by 255. For our AE, we used minibatches because if we used a full batch, the graphical processing unit (GPU) will not be able to handle it. Therefore, for us to pick the right batch size, we picked a large enough value that our model trains better on it yet be supported by the GPU. The batch size we picked was 512. For determining the learning rate, it depends how complex the data is. Since this was an ideal case, a learning rate of 0.01 was good. To check if our learning rate was good, this can be shown in the training loss vs the validation loss plots of the AE. If both losses are decreasing, then our learning rate is performing well with the batch size we had picked. To prevent over fitting in our AE, we defined an early stopping call back with patience equal to 3 and a model check point to save the lowest loss.

As the **Figure 7** shown, the AE we built has 3 layers for encoding, 3 layers for decoding, and one layer for the output. The code for the AE for all models is shown in Appendix B (see **Figure 22**). We used max pooling layer with each dense layer we add. We also used batch normalization and rectified linear activation function (Relu) for each layer except for the output layer, where we used a sigmoid activation function. We used a sigmoid activation function at the output to get an output between 0 and 1

which what each pixel in our image is.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 96, 96, 4)]	0
conv2d (Conv2D)	(None, 96, 96, 16)	592
batch_normalization (BatchNo)	(None, 96, 96, 16)	64
max_pooling2d (MaxPooling2D)	(None, 48, 48, 16)	0
conv2d_1 (Conv2D)	(None, 48, 48, 32)	4640
batch_normalization_1 (Batch)	(None, 48, 48, 32)	128
max_pooling2d_1 (MaxPooling2	(None, 24, 24, 32)	0
tf_op_layer_Relu (TensorFlow	[(None, 24, 24, 32)]	0
conv2d_2 (Conv2D)	(None, 24, 24, 64)	18496
batch_normalization_2 (Batch)	(None, 24, 24, 64)	256
max_pooling2d_2 (MaxPooling2	(None, 12, 12, 64)	0
conv2d_3 (Conv2D)	(None, 12, 12, 64)	36928
batch_normalization_3 (Batch)	(None, 12, 12, 64)	256
up_sampling2d (UpSampling2D)	(None, 24, 24, 64)	0
conv2d_4 (Conv2D)	(None, 24, 24, 32)	18464
batch_normalization_4 (Batch)	(None, 24, 24, 32)	128
up_sampling2d_1 (UpSampling2	(None, 48, 48, 32)	0
tf_op_layer_Relu_1 (TensorFl	[(None, 48, 48, 32)]	0
conv2d_5 (Conv2D)	(None, 48, 48, 16)	4624
batch_normalization_5 (Batch)	(None, 48, 48, 16)	64
up_sampling2d_2 (UpSampling2	(None, 96, 96, 16)	0
conv2d_6 (Conv2D)	(None, 96, 96, 4)	580
<hr/>		
Total params: 85,220		
Trainable params: 84,772		
Non-trainable params: 448		

**Figure 7:** The layers information of proposed Autoencoder neural network.

Using the AE we built for the ideal model, we were able to construct a topology classifier [14]. To build that classifier, we got the AE’s encoding layers with their weights

and freeze them. To add fully connected layers for classification purposes, we flattened the last layer in the encoder. After that, we added two fully connected layers with 128 and 8 neurons for the 8 classes, respectively. We kept the batch size and the learning rate the same but decrease the epochs from 30 to 10 since the encoding layers are not trainable. To train these classifiers, we used categorical cross-entropy as a loss metric.

### 3.2 Non-Ideal Models

Since the ideal model does not reflect real-world scenarios, we reconstructed some models that can act like a real-world model. To build these models, we introduced white noise, removed some of the PMUs, and removed random data. We constructed two non-ideal models (10dB SNR white noise and missing one data and 10dB SNR white noise), one with keeping 22 PMUs randomly out of 33 PMUs and adding a 10dB SNR white noise, and the other with also keeping only 22 PMUs, adding 10dB SNR white noise, and randomly removing one data from the matrix. For each non-ideal model, we generated cases based on the IEEE 34-node test feeder taking non-ideality into consideration.

For building these AEs, we divided the generated dataset to 80% for training, 10% for validation, and 10% for testing. To use the generated images, we resized them to  $96 \times 96$  and normalized them by dividing them by 255. We kept all hyper parameters the same except for the learning rate. For the non-ideal models, we decreased the learning rate to 0.001 because the models now are much complex due to noise and PMU's availability. To tackle over fitting, we also defined an early stopping call back with patience equal to 3 and a model check point to save the lowest loss. Both non-ideal models have the same AE architecture as the ideal model with 3 layers for encoding, 3 layers for decoding, and one layer for the output. We used the same activation functions for all layers as we did for the AE in the ideal model.

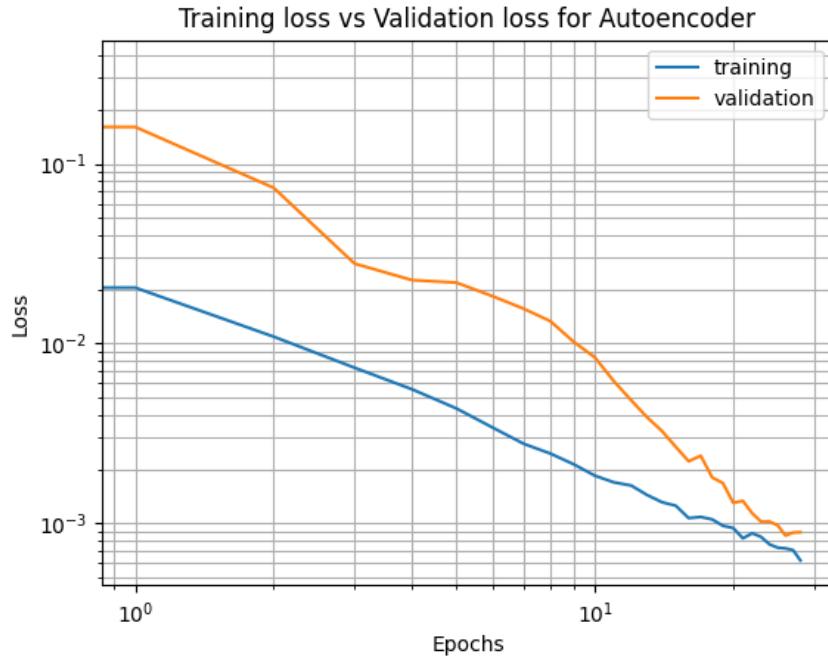
Similarly, as we did with the ideal model, we used the AE's encoding layers to help us classify the topologies. So, we set the weights of the encoding layers to the weights they had at their last epoch and freeze them (not trainable). We then flattened the last encoding layer before adding fully connected layers. Then, we added two fully connected

layers with 512 neurons and 8 neurons for the 8 classes, respectively. However, here we added a dropout layer of 0.5 between the two fully connected layers to tackle over fitting. To train these classifiers, we used categorical cross-entropy as a loss metric. All hyper parameters were the same as in the AE model but decreased the epochs from 30 to 10.

## 4 Numerical Results

### 4.1 Ideal Models

For all models, we trained the AE at the beginning to make sure that the encoding part is working perfectly so we can use it in the classification model. Here, we start with the results of the ideal model. The training loss v.s. the validation loss plot for the AE for the ideal is shown in **Figure 8**.

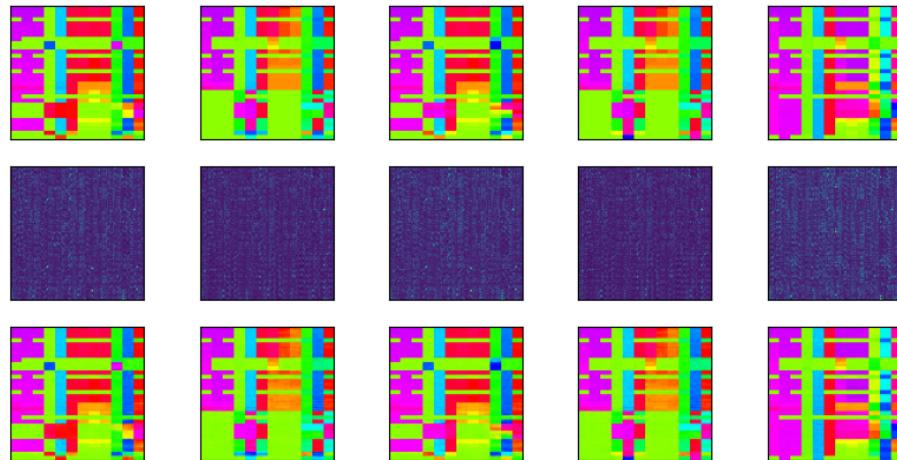


**Figure 8:** Training loss vs validation loss for AE in the ideal model.

It shows that the model is learning and there is no sign of over fitting since both losses are decreasing. The AE for the ideal model has been tested on random test set

images as shown in **Figure 9**.

**Test Images - Encoded Test Images - Reconstructed Test Images**



**Figure 9:** AE prediction on some test set images (Image – Encoding – Decoding).

After we made sure that the classifier is working perfectly, we built the classification model to classify different topologies. The training loss v.s. the validation loss plot for the classifier model is shown in **Figure 10**.



**Figure 10:** Training loss vs validation loss for the classifier in the ideal model.

Also, the classifier was able to predict all the test set correctly with 100% accuracy as shown in **Figure 11**.

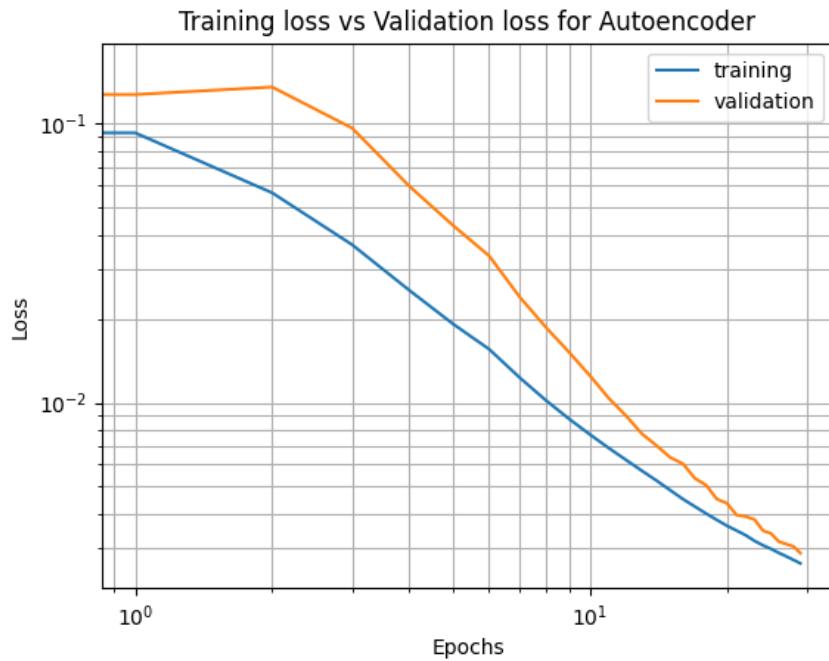
```
Test loss on test set: 4.459287993086036e-06
Test accuracy on test set: 1.0
(3890,) (3890, 8)
Found 3890 correct labels
Found 0 incorrect labels
```

**Figure 11:** Test accuracy on test set for the ideal model classifier.

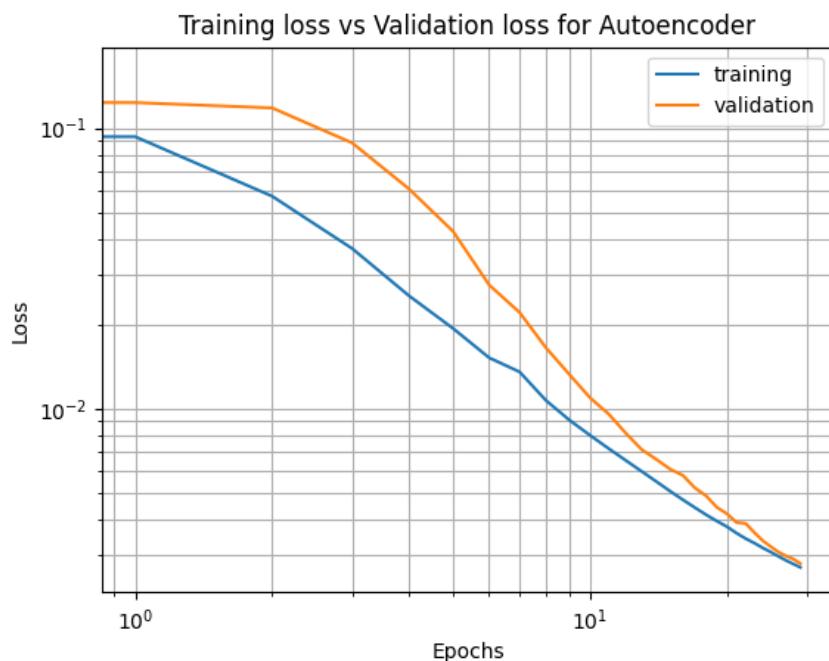
This indicates that the classifier was able to predict all unseen images correctly.

## 4.2 Non-Ideal Models

As we built the ideal model, we built the non-ideal models. We built the AE for each model and made sure that working efficiently. The training loss v.s. validation loss plot for the AE 10dB SNR white noise model is shown in **Figure 12** while the plot for the AE missing one data and 10dB SNR white noise model is shown in **Figure 13**.



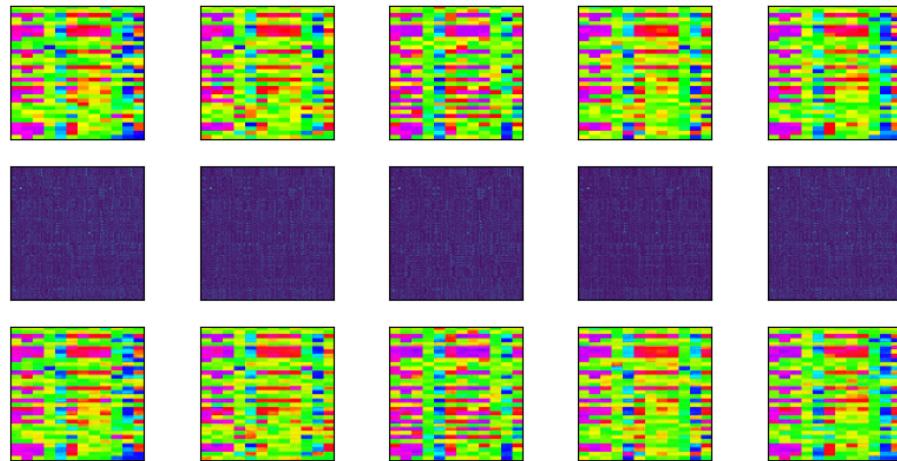
**Figure 12:** Training loss vs validation loss for the AE 10dB SNR white noise model.



**Figure 13:** Training loss vs validation loss for the AE missing one data and 10dB SNR white noise model.

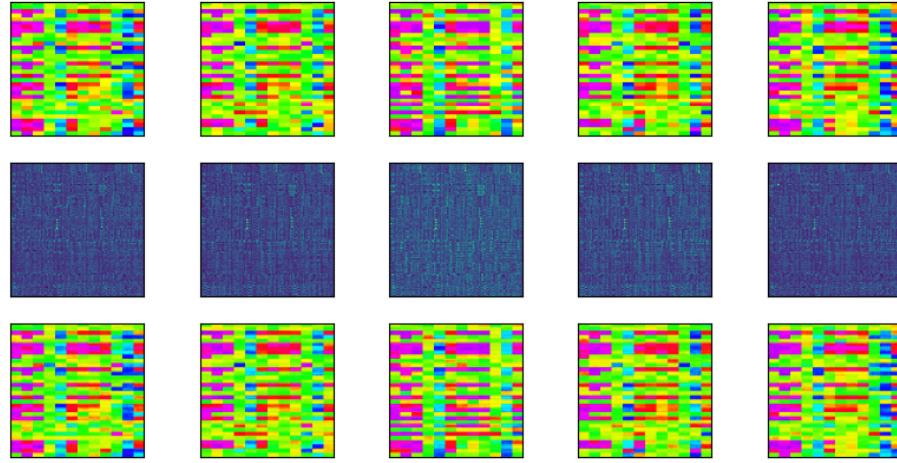
These two figures indicate that both AEs were trained efficiently because all losses were decreasing. In addition, since the validation losses are not increasing then the model is not overfitting. Furthermore, **Figure 14** and **Figure 15** show the test of AE of each model on some test images.

Test Images - Encoded Test Images - Reconstructed Test Images



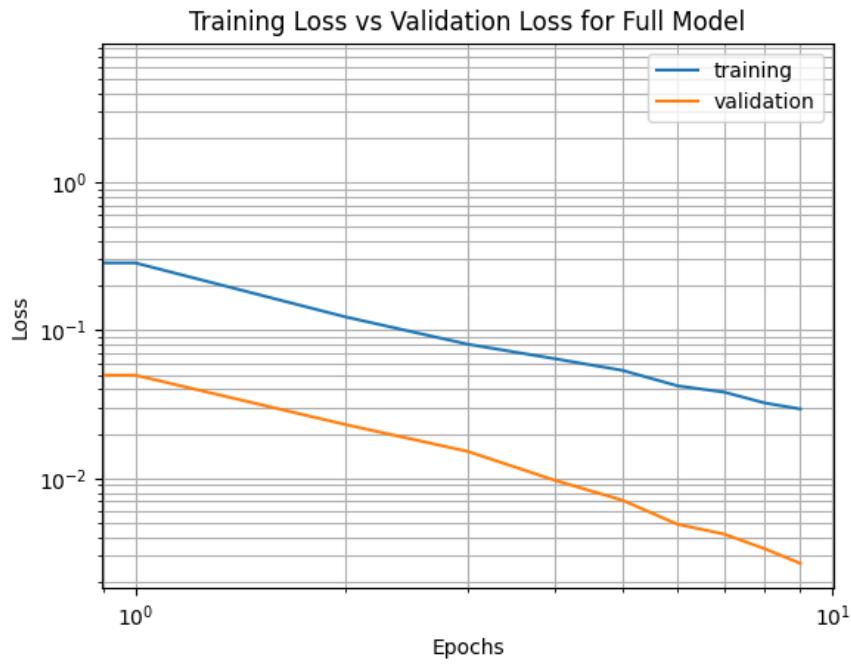
**Figure 14:** AE 10dB SNR white noise model prediction on some test set images (Image – Encoding – Decoding).

Test Images - Encoded Test Images - Reconstructed Test Images

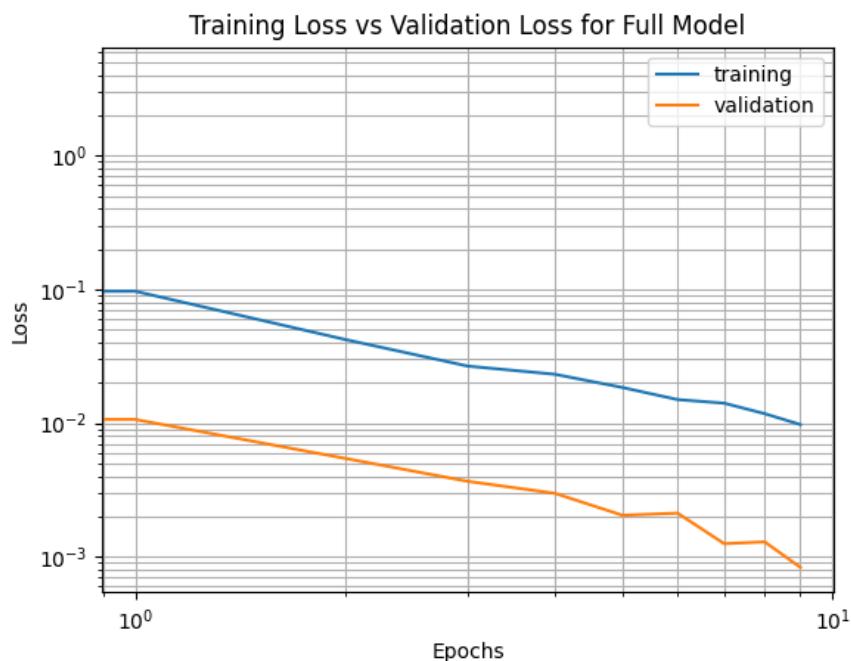


**Figure 15:** AE missing one data and 10dB SNR white nosie model prediction on some test set images (Image – Encoding – Decoding).

Since the AEs for each non-ideal model is working efficiently and perfectly, we built the classifier using the encoding layers and some fully connected layers. The training loss vs the validation loss plots for each classifier is shown in **Figure 16** and **Figure 17**.



**Figure 16:** Training loss vs validation loss for the 10dB SNR white noise model classifier.



**Figure 17:** Training loss vs validation loss for the missing one data and 10dB SNR white noise model classifier.

The classifiers were able to predict all test set images correctly with 100% accuracy as shown in **Figure 18** and **Figure 19**.

```
Test loss on test set: 0.002677673939615488
Test accuracy on test set: 1.0
(1944,) (1944, 8)
Found 1944 correct labels
Found 0 incorrect labels
```

**Figure 18:** Test accuracy on test set for the 10dB SNR white noise model classifier.

```
Test loss on test set: 0.0008325269445776939
Test accuracy on test set: 1.0
(1944,) (1944, 8)
Found 1944 correct labels
Found 0 incorrect labels
```

**Figure 19:** Test accuracy on test set for the missing one data and 10dB SNR white noise model classifier.

This indicates the model has been trained efficiently and that there is no sign of over fitting in these models.

### 4.3 Models Test on Interfered Datasets

The dataset we used to build the ideal and non-ideal models had a load change from 95% to 105%, and these models were tested on 3 new interfered datasets that had a load change from 93% to 95% and 105% to 107%. Furthermore, two of these interfered datasets have missing value/s from their matrices. These interfered datasets were not seen by any of the models (new to the trained models) trained in section 4.1 and section 4.2, thus it is expected to get low accuracy. Since the trained models were not reproducible, even though we have set the seed for cuDNN, random, and TensorFlow, we generated 5 to 10 batches for each model we had. We then took all these batch results and predicted each interfered dataset and average all results to get a value for each model. These results are shown in **Table 2**. The first column in **Table 2** stands for the models we built in section 4.1 and section 4.2 while the first row stands

for the 3 interfered datasets we introduced above. Looking at **Table 2**, no matter which model we pick, the prediction accuracy decreases as we move from left to right because the interfered dataset gets "worse"; It fits the common sense of deep learning. The accuracies of the non-ideal models are acceptable unlike the accuracies of the ideal model. This could be due to that non-ideal models took white noise and missing data in building the model into considerations while ideal model did not. Finally, although the shape of the topology is different, we realized that for "Topology 2" and "Topology 3" switching the breaker SW 5 does not influence the PMU data too much because there is no load on circuit branch node 862 - node 838. Hence, the data of "Topology 2" and "Topology 3" are almost the same.

**Table 2:** The prediction accuracy (%) by training and testing the AE in different extents of interferences for 8 topologies

Models	Interfered Data	40dB SNR	Missing One Data	Missing Two Data
			based on 40dB SNR	based on 40dB SNR
Ideal Model (33 PMU with no Missing and 0dB SNR)		55.357	55.214	55.125
22 PMU with no Missing and 10dB SNR Model		86.571	86.482	86.053
22 PMU with Missing One Data and 10dB SNR Model		77.373	76.998	76.374

## 4.4 For 7 Topologies

As we known the reason for low accuracy, we removed "Topology 3" from the system, in another word, we removed "label 3" in the input and output layers. And now, there are 7 topologies in total, and we tested them again, shown below:

**Table 3:** The prediction accuracy (%) by training and testing the AE in different extents of interferences for 7 topologies

Models \ Interfered Data	40dB SNR	Missing One Data based on 40dB SNR	Missing Two Data based on 40dB SNR
Ideal Model (33 PMU with no Missing and 0dB SNR)	79.571	79.381	79.285
22 PMU with no Missing and 10dB SNR Model	71.476	71.428	71.428

The prediction accuracy increased dramatically to nearly 80% from 55%. And it means our assumption is right, "Topology 3" indeed badly influenced the accuracy. But within the same structure on the non-ideal model "22 PMU with 10dB SNR", the accuracy went down, that is because once the input changed, the neural network also need to adjust to fit that change. However, due to time constraints, this project will stop here. How to improve the neural network structure to adapt to the new input will left to the future research.

## 5 Conclusion

In this project, a deep learning AE framework is proposed for online detection of power distribution system topology. The proposed AE framework can handle missing measurements under unbalanced operating states. The experiments show that the proposed AE not only handles the data with the same level of interference (noise and missing measurements), but also has the capacity of estimating the interfered data which has different distributions from the training examples. Numerical experiments proved

that our trained network can accurately identify the network topology corresponding to the observed data beyond the training dataset.

Future work could be targeted at implementing the proposed framework on a larger real-world power grid, such as the IEEE 123-bus test system, and validating the results' accuracy and computational effectiveness during real-time applications. Moreover, the architecture of the proposed AE neural network could be optimized for increasing the prediction accuracy of general models for improving universality.

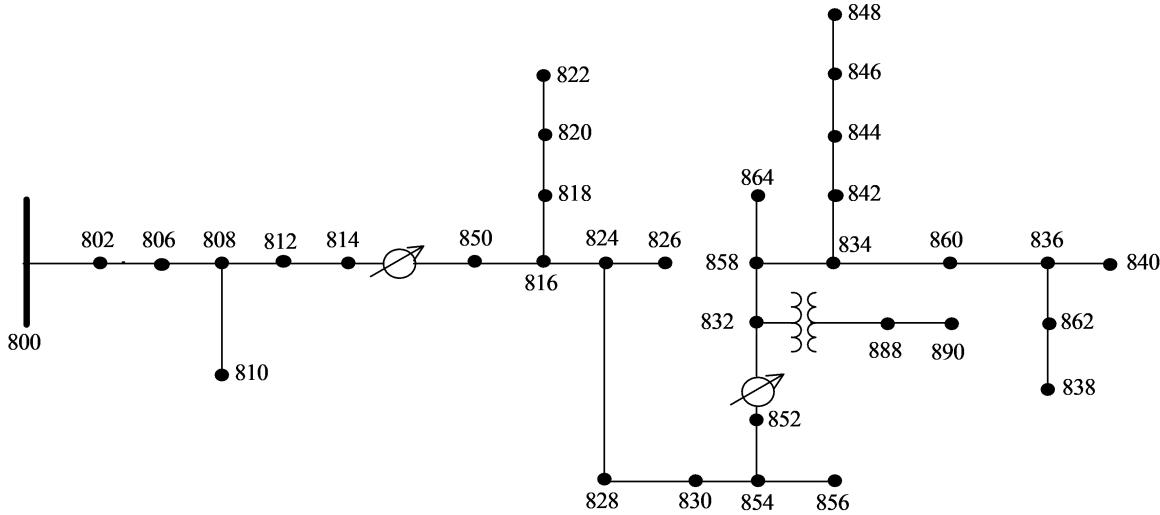
## References

- [1] National Academies of Sciences, *Enhancing the resilience of the nation's electricity system*. National Academies Press, 2017.
- [2] National Centers for Environmental Information, “Billion-dollar weather and climate disasters: Overview,” 2020. [Online] Available at: <https://www.ncdc.noaa.gov/billions/>.
- [3] Wikipedia, “Phasor measure unit,” 2020. [Online] Available at: [https://en.wikipedia.org/wiki/Phasor\\_measurement\\_unit](https://en.wikipedia.org/wiki/Phasor_measurement_unit).
- [4] C. M. Davis, J. E. Tate, and T. J. Overbye, “Wide area phasor data visualization,” in *2007 39th North American Power Symposium*, pp. 246–252, 2007.
- [5] E. Lawrence, S. V. Wiel, and R. Bent, “Model bank state estimation for power grids using importance sampling,” *Technometrics*, vol. 55, no. 4, pp. 426–435, 2013.
- [6] G. Cavraro and R. Arghandeh, “Power distribution network topology detection with time-series signature verification method,” *IEEE Transactions on Power Systems*, vol. 33, no. 4, pp. 3500–3509, 2018.
- [7] D. Deka, S. Talukdar, M. Chertkov, and M. V. Salapaka, “Graphical models in meshed distribution grids: Topology estimation, change detection & limitations,” *IEEE Transactions on Smart Grid*, vol. 11, no. 5, pp. 4299–4310, 2020.
- [8] “Autoencoder.” [Online] Available at: <https://en.wikipedia.org/wiki/Autoencoder>, Accessed:2020.
- [9] IEEE PES, “34-bus feeder case,” 2010. [Online] Available at: <https://site.ieee.org/pes-testfeeders/resources>.
- [10] J. D. Glover, M. S. Sarma, and T. Overbye, *Power system analysis & design, SI version*. Cengage Learning, 2012.
- [11] “seaborn.heatmap.” [Online] Available at: <https://seaborn.pydata.org/generated/seaborn.heatmap.html>, Accessed:2020.

- [12] “How bad is your colormap?.” [Online] Available at: <http://jakevdp.github.io/blog/2014/10/16/how-bad-is-your-colormap/>, Accessed:2020.
- [13] A. Sharma, “Implementing autoencoders in keras: Tutorial.” [Online] Available at: <https://www.datacamp.com/community/tutorials/autoencoder-keras-tutorial#cae>, Accessed:2020.
- [14] A. Sharma, “Autoencoder as a classifier using fashion-mnist dataset.” [Online] Available at: <https://www.datacamp.com/community/tutorials/autoencoder-classifier-python#>, Accessed:2020.
- [15] Y. Li, *Date-Driven Topology Identification in Power Distribution Systems with Machine Learning*. PhD thesis, The George Washington University, 2020.

# Appendix A: IEEE 34 Node Test Feeder

## Introduction



**Figure 20:** IEEE 34-Node Test Feeder [9]

This feeder is an actual feeder located in Arizona. The feeder's nominal voltage is 24.9 kV. It is characterized by:

- (1) Very long and lightly loaded overhead distribution lines
- (2) Two in-line regulators required to maintain a good voltage profile across the network
- (3) A wye-wye grounded transformer reducing the voltage to 4.16 kV for a short section of the feeder
- (4) 24 unbalanced loading with both "spot" and "distributed" loads. Distributed loads are assumed to be evenly distributed on the distribution line.
- (5) Shunt capacitors

## System Data

Here are the data forms originated from the IEEE PES AMPS DSAS test feeder working group [9].

**Table 4:** Overhead Line Configurations [9]

Config.	Phasing	Phase ACSR <sup>1</sup>	Neutral ACSR	Spacing ID
300	BACN	1/0	1/0	500
301	BACN	#2 6/1	#2 6/1	500
302	AN	#4 6/1	#4 6/1	510
303	BN	#4 6/1	#4 6/1	510
304	BN	#2 6/1	#2 6/1	510

<sup>1</sup> ACSR: Aluminum conductor steel reinforced.

**Table 5:** Line Segment Data [9]

Node A	Node B	Length (ft.)	Config.
800	802	2580	300
802	806	1730	300
806	808	32230	300
808	810	5804	303
808	812	37500	300
812	814	29730	300
814	850	10	301
816	818	1710	302
816	824	10210	301
818	820	48150	302
820	822	13740	302
824	826	3030	303
824	828	840	301

828	830	20440	301
830	854	520	301
832	858	4900	301
832	888	0	XFM-1
834	860	2020	301
834	842	280	301
836	840	860	301
836	862	280	301
842	844	1350	301
844	846	3640	301
846	848	530	301
850	816	130	301
852	832	10	301
854	856	23330	303
854	852	36830	301
858	864	1620	302
858	834	5830	301
860	836	2680	301
862	838	4860	304
888	890	10560	300

**Table 6:** Transformer Data [9]

	kVA	kV - high	kV - low	R - %	X - %
<b>Substation</b>	2500	69 - D	24.9 - Gr.W	1	8
<b>XFM-1</b>	500	24.9 - Gr.W	24.9 - Gr.W	1.9	4.08

**Table 7:** Spot Loads [9]

Node	Load	Ph-1	Ph-1	Ph-2	Ph-2	Ph-3	Ph-4
	Model	kW	kVAr	kW	kVAr	kW	kVAr
860	Y-PQ	20	16	20	16	20	16
802	Y-I	9	7	9	7	9	7
806	Y-Z	135	105	135	105	135	105
808	D-PQ	20	16	20	16	20	16
808	D-I	150	75	150	75	150	75
812	D-Z	10	5	10	5	25	10
Total		344	224	344	224	359	229

## Impedances

Configuration 300:

———— Z & B Matrices Before Changes ————

Z (R +jX) in ohms per mile

1.3368	1.3343	0.2101	0.5779	0.2130	0.5015
		1.3238	1.3569	0.2066	0.4591
				1.3294	1.3471

B in micro Siemens per mile

5.3350	-1.5313	-0.9943
5.0979	-0.6212	
	4.8880	

Configuration 301:

Z (R +jX) in ohms per mile

1.9300	1.4115	0.2327	0.6442	0.2359	0.5691
		1.9157	1.4281	0.2288	0.5238
				1.9219	1.4209

B in micro Siemens per mile

5.1207	-1.4364	-0.9402
4.9055	-0.5951	
	4.7154	

Configuration 302:

**Table 8:** Distributed Loads [9]

Node	Node	Load	Ph-1	Ph-1	Ph-2	Ph-2	Ph-3	Ph-3
A	B	Model	kW	kVAr	kW	kVAr	kW	kVAr
802	806	Y-PQ	0	0	30	15	25	14
808	810	Y-I	0	0	16	8	0	0
818	820	Y-Z	34	17	0	0	0	0
820	822	Y-PQ	135	70	0	0	0	0
816	824	D-I	0	0	5	2	0	0
824	826	Y-I	0	0	40	20	0	0
824	828	Y-PQ	0	0	0	0	4	2
828	830	Y-PQ	7	3	0	0	0	0
854	856	Y-PQ	0	0	4	2	0	0
832	858	D-Z	7	3	2	1	6	3
858	864	Y-PQ	2	1	0	0	0	0
858	834	D-PQ	4	2	15	8	13	7
834	860	D-Z	16	8	20	10	110	55
860	836	D-PQ	30	15	10	6	42	22
836	840	D-I	18	9	22	11	0	0
862	838	Y-PQ	0	0	28	14	0	0
842	844	Y-PQ	9	5	0	0	0	0
844	846	Y-PQ	0	0	25	12	20	11
846	848	Y-PQ	0	0	23	11	0	0
Total			262	133	240	120	220	114

**Table 9:** Shunt Capacitors [9]

Node	Ph-A	Ph-B	Ph-C
	kVAr	kVAr	kVAr
844	100	100	100
848	150	150	150
Total	250	250	250

**Table 10:** Regulator Data [9]

<b>Regulator ID</b>	1			2		
<b>Line Segment</b>	814-850			852-832		
<b>Location</b>	814			852		
<b>Phases</b>	A-B-C			A-B-C		
<b>Connection</b>	3-Ph, LG			3-Ph, LG		
<b>Monitoring Phase</b>	A-B-C			A-B-C		
<b>Bandwidth</b>	2.0 volts			2.0 volts		
<b>PT Ratio</b>	120			120		
<b>Primary CT Rating</b>	100			100		
<b>Compensator Settings</b>	Ph-A	Ph-B	Ph-C	Ph-A	Ph-B	Ph-C
<b>R-Setting</b>	2.7	2.7	2.7	2.5	2.5	2.5
<b>X-Setting</b>	1.6	1.6	1.6	1.6	1.6	1.6
<b>Vlotage Level</b>	122	122	122	124	124	124

Z (R +jX) in ohms per mile					
2.7995	1.4855	0.0000	0.0000	0.0000	0.0000
		0.0000	0.0000	0.0000	0.0000
			0.0000	0.0000	
B in micro Siemens per mile					
	4.2251	0.0000	0.0000		
		0.0000	0.0000		
			0.0000		

Configuration 303:

Z (R +jX) in ohms per mile					
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	2.7995	1.4855	0.0000	0.0000	
			0.0000	0.0000	
B in micro Siemens per mile					
	0.0000	0.0000	0.0000		
	4.2251	0.0000			
		0.0000			

Configuration 304:

Z (R +jX) in ohms per mile					
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	1.9217	1.4212	0.0000	0.0000	
			0.0000	0.0000	
B in micro Siemens per mile					
	0.0000	0.0000	0.0000		
	4.3637	0.0000			
		0.0000			

## Power Flow Results

### Radial Flow Summary

— R A D I A L F L O W S U M M A R Y — DATE: 6-24-2004 AT 16:34:11 HOURS —  
 SUBSTATION: IEEE 34; FEEDER: IEEE 34

SYSTEM	PHASE (A)	PHASE (B)	PHASE (C)	TOTAL
INPUT				
kW : 759.136		666.663		617.072   2042.872
kVAr : 171.727		90.137		28.394   290.258
kVA : 778.318		672.729		617.725   2063.389

PF	.9754		.9910		.9989		.9901				
LOAD —(A-N)——(A-B)--- (B-N)——(B-C)--- (C-N)——(C-A)--- WYE——DELTA—											
kW	359.9	246.4		339.3	243.3		221.8	359.0		921.0	848.8
TOT	606.322			582.662			580.840			1769.824	
kVAr	230.9	128.7		216.9	128.7		161.8	184.6		609.6	441.9
TOT	359.531			345.609			346.407			1051.547	
kVA	427.6	278.0		402.7	275.3		274.6	403.7		1104.5	957.0
TOT	704.903			677.452			676.293			2058.647	
PF	.8417	.8864		.8425	.8840		.8078	.8894		.8339	.8870
TOT	.8601			.8601			.8589			.8597	
LOSSES —(A)—— (B)—— (C)——											
kW	114.836			80.389			77.824			273.049	
kVAr	14.200			10.989			9.810			34.999	
kVA	115.711			81.137			78.440			275.283	
CAPAC —(A-N)——(A-B)--- (B-N)——(B-C)--- (C-N)——(C-A)--- WYE——DELTA—											
R-kVA:	250.0	.0		250.0	.0		250.0	.0		750.0	.0
TOT	250.000			250.000			250.000			750.000	
A-kVA:	265.7	.0		264.8	.0		265.9	.0		796.3	.0
TOT	265.658			264.760			265.869			796.287	

## Voltage Profile

— V O L T A G E P R O F I L E —— DATE: 6-24-2004 AT 16:34:18 HOURS ——  
SUBSTATION: IEEE 34; FEEDER: IEEE 34

NODE	MAG	ANGLE	MAG	ANGLE	MAG	ANGLE	mi. to SR
	A-N		B-N		C-N		
800	1.0500 at .00		1.0500 at -120.00		1.0500 at 120.00		.000
802	1.0475 at -.05		1.0484 at -120.07		1.0484 at 119.95		.489
806	1.0457 at -.08		1.0474 at -120.11		1.0474 at 119.92		.816
808	1.0136 at -.75		1.0296 at -120.95		1.0289 at 119.30		6.920
810			1.0294 at -120.95				8.020
812	.9763 at -1.57		1.0100 at -121.92		1.0069 at 118.59		14.023
814	.9467 at -2.26		.9945 at -122.70		.9893 at 118.01		19.653
RG10	1.0177 at -2.26		1.0255 at -122.70		1.0203 at 118.01		19.654
850	1.0176 at -2.26		1.0255 at -122.70		1.0203 at 118.01		19.655
816	1.0172 at -2.26		1.0253 at -122.71		1.0200 at 118.01		19.714

818		1.0163	at	-2.27							20.038			
820		.9926	at	-2.32							29.157			
822		.9895	at	-2.33							31.760			
824		1.0082	at	-2.37		1.0158	at	-122.94		1.0116	at	117.76		21.648
826						1.0156	at	-122.94				22.222		
828		1.0074	at	-2.38		1.0151	at	-122.95		1.0109	at	117.75		21.807
830		.9894	at	-2.63		.9982	at	-123.39		.9938	at	117.25		25.678
854		.9890	at	-2.64		.9978	at	-123.40		.9934	at	117.24		25.777
852		.9581	at	-3.11		.9680	at	-124.18		.9637	at	116.33		32.752
RG11		1.0359	at	-3.11		1.0345	at	-124.18		1.0360	at	116.33		32.752
832		1.0359	at	-3.11		1.0345	at	-124.18		1.0360	at	116.33		32.754
858		1.0336	at	-3.17		1.0322	at	-124.28		1.0338	at	116.22		33.682
834		1.0309	at	-3.24		1.0295	at	-124.39		1.0313	at	116.09		34.786
842		1.0309	at	-3.25		1.0294	at	-124.39		1.0313	at	116.09		34.839
844		1.0307	at	-3.27		1.0291	at	-124.42		1.0311	at	116.06		35.095
846		1.0309	at	-3.32		1.0291	at	-124.46		1.0313	at	116.01		35.784
848		1.0310	at	-3.32		1.0291	at	-124.47		1.0314	at	116.00		35.885
860		1.0305	at	-3.24		1.0291	at	-124.39		1.0310	at	116.09		35.169
836		1.0303	at	-3.23		1.0287	at	-124.39		1.0308	at	116.09		35.677
840		1.0303	at	-3.23		1.0287	at	-124.39		1.0308	at	116.09		35.839
862		1.0303	at	-3.23		1.0287	at	-124.39		1.0308	at	116.09		35.730
838						1.0285	at	-124.39				36.650		
864		1.0336	at	-3.17								33.989		
XF10		.9997	at	-4.63		.9983	at	-125.73		1.0000	at	114.82		32.754
888		.9996	at	-4.64		.9983	at	-125.73		1.0000	at	114.82		32.754
890		.9167	at	-5.19		.9235	at	-126.78		.9177	at	113.98		34.754
856						.9977	at	-123.41				30.195		

### Voltage Regulator Data

VOLTAGE REGULATOR DATA								DATE:	6-24-2004 AT 16:34:22 HOURS	
SUBSTATION: IEEE 34; FEEDER: IEEE 34										
<hr/>										
[NODE]	--[VREG]	-----[SEG]	-----[NODE]	MODEL				OPT	BNDW	
814	RG10	850	850	Phase A & B & C, Wye				RX	2.00	
<hr/>										
PHASE	LDCTR	VOLT HOLD	R-VOLT	X-VOLT	PT RATIO	CT RATE	TAP			
1		122.000	2.700	1.600	120.00	100.00	12			
2		122.000	2.700	1.600	120.00	100.00	5			
3		122.000	2.700	1.600	120.00	100.00	5			
<hr/>										
[NODE]	--[VREG]	-----[SEG]	-----[NODE]	MODEL				OPT	BNDW	
852	RG11	832	832	Phase A & B & C, Wye				RX	2.00	
<hr/>										
PHASE	LDCTR	VOLT HOLD	R-VOLT	X-VOLT	PT RATIO	CT RATE	TAP			

1	124.000	2.500	1.500	120.00	100.00	13
2	124.000	2.500	1.500	120.00	100.00	11
3	124.000	2.500	1.500	120.00	100.00	12

## Radial Power Flow

— R A D I A L P O W E R F L O W — DATE: 6-24-2004 AT 16:34:32 HOURS —						
SUBSTATION: IEEE 34; FEEDER: IEEE 34						
NODE	VALUE	PHASE A (LINE A)	PHASE B (LINE B)	PHASE C (LINE C)	UNT O/L<	60.%
*	*	A	B	C	*	*
NODE: 800	VOLTS:	1.050 .00	1.050 -120.00	1.050 120.00	MAG/ANG	
kVll	24.900	NO LOAD OR CAPACITOR REPRESENTED AT SOURCE NODE				
TO NODE 802 . . . . .	: 51.56 -12.74	44.57 -127.70	40.92 117.37	AMP/DG		
<802 > LOSS= 3.472:	: ( 1.637)	( .978)	( .858)	kW		
*	*	A	B	C	*	*
NODE: 802	VOLTS:	1.047 -.05	1.048 -120.07	1.048 119.95	MAG/ANG	
	-LD:	.00 .00	.00 .00	.00 .00	kW/kVR	
kVll	24.900	CAP:	.00	.00	.00	kVR
FROM NODE 800 . . . . .	: 51.58 -12.80	44.57 -127.76	40.93 117.31	AMP/DG		
<802 > LOSS= 3.472:	: ( 1.637)	( .978)	( .858)	kW		
TO NODE 806 . . . . .	: 51.58 -12.80	44.57 -127.76	40.93 117.31	AMP/DG		
<806 > LOSS= 2.272:	: ( 1.102)	( .618)	( .552)	kW		
*	*	A	B	C	*	*
NODE: 806	VOLTS:	1.046 -.08	1.047 -120.11	1.047 119.92	MAG/ANG	
	-LD:	.00 .00	.00 .00	.00 .00	kW/kVR	
kVll	24.900	CAP:	.00	.00	.00	kVR
FROM NODE 802 . . . . .	: 51.59 -12.83	42.47 -126.83	39.24 118.52	AMP/DG		
<806 > LOSS= 2.272:	: ( 1.102)	( .618)	( .552)	kW		
TO NODE 808 . . . . .	: 51.59 -12.83	42.47 -126.83	39.24 118.52	AMP/DG		
<808 > LOSS= 41.339:	: ( 20.677)	( 10.780)	( 9.882)	kW		
*	*	A	B	C	*	*
NODE: 808	VOLTS:	1.014 -.75	1.030 -120.95	1.029 119.30	MAG/ANG	
	-LD:	.00 .00	.00 .00	.00 .00	kW/kVR	
kVll	24.900	CAP:	.00	.00	.00	kVR
FROM NODE 806 . . . . .	: 51.76 -13.47	42.46 -127.59	39.28 117.76	AMP/DG		
<808 > LOSS= 41.339:	: ( 20.677)	( 10.780)	( 9.882)	kW		
TO NODE 810 . . . . .		1.22 -144.62		AMP/DG		
<810 > LOSS= .002:		( .002)		kW		
TO NODE 812 . . . . .	: 51.76 -13.47	41.30 -127.10	39.28 117.76	AMP/DG		

<812	> LOSS=	47.531:	( 24.126)	( 11.644)	( 11.761)	kW
	*	A	*	B	*	C
NODE: 810	VOLTS:		1.029	-120.95		MAG/ANG
	-LD:		.00	.00		kW/kVR
kVll 24.900	CAP:			.00		kVR
FROM NODE 808	.....:		.00	.00		AMP/DG
<810	> LOSS=	.002:		( .002)		kW
	*	A	*	B	*	C
NODE: 812	VOLTS:	.976	-1.57	1.010	-121.92	1.007 118.59 MAG/ANG
	-LD:	.00	.00	.00	.00	.00 kW/kVR
kVll 24.900	CAP:		.00		.00	.00 kVR
FROM NODE 808	.....:	51.95 -14.18	41.29 -127.99	39.33 116.90 AMP/DG		
<812	> LOSS=	47.531:	( 24.126)	( 11.644)	( 11.761)	kW
TO NODE 814	.....:	51.95 -14.18	41.29 -127.99	39.33 116.90 AMP/DG		
<814	> LOSS=	37.790:	( 19.245)	( 9.140)	( 9.404)	kW
	*	A	*	B	*	C
NODE: 814	VOLTS:	.947	-2.26	.994	-122.70	.989 118.01 MAG/ANG
	-LD:	.00	.00	.00	.00	.00 kW/kVR
kVll 24.900	CAP:		.00		.00	.00 kVR
FROM NODE 812	.....:	52.10 -14.73	41.29 -128.69	39.37 116.23 AMP/DG		
<814	> LOSS=	37.790:	( 19.245)	( 9.140)	( 9.404)	kW
TO NODE RG10 <VRG>.:	.....:	52.10 -14.73	41.29 -128.69	39.37 116.23 AMP/DG		
<RG10	> LOSS=	.000:	( .000)	( .000)	( .000)	kW
	*	A	*	B	*	C
NODE: RG10	VOLTS:	1.018	-2.26	1.026	-122.70	1.020 118.01 MAG/ANG
	-LD:	.00	.00	.00	.00	.00 kW/kVR
kVll 24.900	CAP:		.00		.00	.00 kVR
FROM NODE 814 <VRG>:	48.47 -14.73	40.04 -128.69	38.17 116.23 AMP/DG			
<RG10	> LOSS=	.000:	( .000)	( .000)	( .000)	kW
TO NODE 850 .....	:	48.47 -14.73	40.04 -128.69	38.17 116.23 AMP/DG		
<850	> LOSS=	.017:	( .008)	( .005)	( .005)	kW
	*	A	*	B	*	C
NODE: 850	VOLTS:	1.018	-2.26	1.026	-122.70	1.020 118.01 MAG/ANG
	-LD:	.00	.00	.00	.00	.00 kW/kVR
kVll 24.900	CAP:		.00		.00	.00 kVR
FROM NODE RG10 .....	:	48.47 -14.73	40.04 -128.69	38.17 116.23 AMP/DG		
<850	> LOSS=	.017:	( .008)	( .005)	( .005)	kW
TO NODE 816 .....	:	48.47 -14.73	40.04 -128.69	38.17 116.23 AMP/DG		
<816	> LOSS=	.538:	( .254)	( .145)	( .139)	kW
	*	A	*	B	*	C

NODE: 816	VOLTS:	1.017	-2.26	1.025	-122.71	1.020	118.01	MAG/ANG
	-LD:	.00	.00	.00	.00	.00	.00	kW/kVR
kVll 24.900	CAP:		.00		.00		.00	kVR
<hr/>								
FROM NODE 850	.....:	48.47	-14.74	40.04	-128.70	38.17	116.23	AMP/DG
<816 > LOSS=	.538:	( .254)		( .145)		( .139)		kW
TO NODE 818	.....:	13.02	-26.69					AMP/DG
<818 > LOSS=	.154:	( .154)						kW
TO NODE 824	.....:	35.83	-10.42	40.04	-128.70	38.17	116.23	AMP/DG
<824 > LOSS=	14.181:	( 4.312)		( 5.444)		( 4.425)		kW
<hr/>								
NODE: 818	VOLTS:	1.016	-2.27					MAG/ANG
	-LD:	.00	.00					kW/kVR
kVll 24.900	CAP:		.00					kVR
<hr/>								
FROM NODE 816	.....:	13.03	-26.77					AMP/DG
<818 > LOSS=	.154:	( .154)						kW
TO NODE 820	.....:	13.03	-26.77					AMP/DG
<820 > LOSS=	3.614:	( 3.614)						kW
<hr/>								
NODE: 820	VOLTS:	.993	-2.32					MAG/ANG
	-LD:	.00	.00					kW/kVR
kVll 24.900	CAP:		.00					kVR
<hr/>								
FROM NODE 818	.....:	10.62	-28.98					AMP/DG
<820 > LOSS=	3.614:	( 3.614)						kW
TO NODE 822	.....:	10.62	-28.98					AMP/DG
<822 > LOSS=	.413:	( .413)						kW
<hr/>								
NODE: 822	VOLTS:	.990	-2.33					MAG/ANG
	-LD:	.00	.00					kW/kVR
kVll 24.900	CAP:		.00					kVR
<hr/>								
FROM NODE 820	.....:	.00	.00					AMP/DG
<822 > LOSS=	.413:	( .413)						kW
<hr/>								
NODE: 824	VOLTS:	1.008	-2.37	1.016	-122.94	1.012	117.76	MAG/ANG
	-LD:	.00	.00	.00	.00	.00	.00	kW/kVR
kVll 24.900	CAP:		.00		.00		.00	kVR
<hr/>								
FROM NODE 816	.....:	35.87	-10.70	39.82	-129.02	38.05	116.25	AMP/DG
<824 > LOSS=	14.181:	( 4.312)		( 5.444)		( 4.425)		kW
TO NODE 826	.....:			3.10	-148.92			AMP/DG
<826 > LOSS=	.008:			( .008)				kW
TO NODE 828	.....:	35.87	-10.70	36.93	-127.39	38.05	116.25	AMP/DG

<828	> LOSS=	1.108:	( .361)	( .393)	( .354)	kW
	*		A	B	C	*
NODE: 826	VOLTS:		1.016	-122.94		MAG/ANG
	-LD:		.00	.00		kW/kVR
kVll 24.900	CAP:			.00		kVR
FROM NODE 824	.....:		.00	.00		AMP/DG
<826	> LOSS=	.008:		( .008)		kW
	*		A	B	C	*
NODE: 828	VOLTS:	1.007	-2.38	1.015	-122.95	1.011 117.75 MAG/ANG
	-LD:	.00	.00	.00	.00	.00 kW/kVR
kVll 24.900	CAP:		.00	.00		.00 kVR
FROM NODE 824	.....:	35.87 -10.72	36.93 -127.41	37.77 116.42 AMP/DG		
<828	> LOSS=	1.108:	( .361)	( .393)	( .354)	kW
TO NODE 830	.....:	35.87 -10.72	36.93 -127.41	37.77 116.42 AMP/DG		
<830	> LOSS=	26.587:	( 8.443)	( 9.214)	( 8.930)	kW
	*		A	B	C	*
NODE: 830	VOLTS:	.989	-2.63	.998 -123.39	.994 117.25 MAG/ANG	
	D-LD:	9.95	4.98	9.86 4.93	24.55 9.82 kW/kVR	
kVll 24.900	Y CAP:		.00	.00		.00 kVR
FROM NODE 828	.....:	35.43 -11.06	36.91 -127.92	37.79 115.96 AMP/DG		
<830	> LOSS=	26.587:	( 8.443)	( 9.214)	( 8.930)	kW
TO NODE 854	.....:	34.22 -9.97	36.19 -127.47	36.49 116.26 AMP/DG		
<854	> LOSS=	.635:	( .197)	( .227)	( .211)	kW
	*		A	B	C	*
NODE: 854	VOLTS:	.989	-2.64	.998 -123.40	.993 117.24 MAG/ANG	
	-LD:	.00	.00	.00	.00	.00 kW/kVR
kVll 24.900	CAP:		.00	.00		.00 kVR
FROM NODE 830	.....:	34.23 -9.99	36.19 -127.48	36.49 116.25 AMP/DG		
<854	> LOSS=	.635:	( .197)	( .227)	( .211)	kW
TO NODE 852	.....:	34.23 -9.99	35.93 -127.72	36.49 116.25 AMP/DG		
<852	> LOSS=	44.798:	( 13.996)	( 15.778)	( 15.023)	kW
TO NODE 856	.....:		.31 -98.70			AMP/DG
<856	> LOSS=	.001:		( .001)		kW
	*		A	B	C	*
NODE: 852	VOLTS:	.958	-3.11	.968 -124.18	.964 116.33 MAG/ANG	
	-LD:	.00	.00	.00	.00	.00 kW/kVR
kVll 24.900	CAP:		.00	.00		.00 kVR
FROM NODE 854	.....:	34.35 -11.00	35.90 -128.66	36.52 115.41 AMP/DG		
<852	> LOSS=	44.798:	( 13.996)	( 15.778)	( 15.023)	kW
TO NODE RG11 .<VRG>.:	34.35 -11.00	35.90 -128.66	36.52 115.41 AMP/DG			

<RG11 > LOSS=		.000:	( .000)	( .000)	( .000)	kW		
		*	A	*	B	*	C	*
NODE: RG11	VOLTS:	1.036	-3.11	1.035	-124.18	1.036	116.33	MAG/ANG
	-LD:	.00	.00	.00	.00	.00	.00	kW/kVR
kVll	24.900	CAP:		.00		.00		.00 kVR
FROM NODE 852	<VRG>:	31.77	-11.00	33.59	-128.66	33.98	115.41	AMP/DG
<RG11 > LOSS=		.000:	( .000)	( .000)	( .000)	kW		
TO NODE 832	.....:	31.77	-11.00	33.59	-128.66	33.98	115.41	AMP/DG
<832 > LOSS=		.011:	( .003)	( .004)	( .004)	kW		
		*	A	*	B	*	C	*
NODE: 832	VOLTS:	1.036	-3.11	1.035	-124.18	1.036	116.33	MAG/ANG
	-LD:	.00	.00	.00	.00	.00	.00	kW/kVR
kVll	24.900	CAP:		.00		.00		.00 kVR
FROM NODE RG11	.....:	31.77	-11.00	33.59	-128.66	33.98	115.41	AMP/DG
<832 > LOSS=		.011:	( .003)	( .004)	( .004)	kW		
TO NODE 858	.....:	21.31	.47	23.40	-116.89	24.34	128.36	AMP/DG
<858 > LOSS=		2.467:	( .643)	( .997)	( .827)	kW		
TO NODE XF10	.....:	11.68	-32.29	11.70	-152.73	11.61	87.39	AMP/DG <
<XF10 > LOSS=		9.625:	( 3.196)	( 3.241)	( 3.187)	kW		
		*	A	*	B	*	C	*
NODE: 858	VOLTS:	1.034	-3.17	1.032	-124.28	1.034	116.22	MAG/ANG
	-LD:	.00	.00	.00	.00	.00	.00	kW/kVR
kVll	24.900	CAP:		.00		.00		.00 kVR
FROM NODE 832	.....:	20.86	.86	23.13	-116.39	24.02	128.48	AMP/DG
<858 > LOSS=		2.467:	( .643)	( .997)	( .827)	kW		
TO NODE 834	.....:	20.73	1.01	23.13	-116.39	24.02	128.48	AMP/DG
<834 > LOSS=		2.798:	( .717)	( 1.145)	( .936)	kW		
TO NODE 864	.....:	.14	-22.82					AMP/DG
<864 > LOSS=		.000:	( .000)					kW
		*	A	*	B	*	C	*
NODE: 834	VOLTS:	1.031	-3.24	1.029	-124.39	1.031	116.09	MAG/ANG
	-LD:	.00	.00	.00	.00	.00	.00	kW/kVR
kVll	24.900	CAP:		.00		.00		.00 kVR
FROM NODE 858	.....:	20.29	2.18	22.37	-116.07	23.23	130.06	AMP/DG
<834 > LOSS=		2.798:	( .717)	( 1.145)	( .936)	kW		
TO NODE 842	.....:	14.75	34.68	16.30	-95.63	15.12	151.05	AMP/DG
<842 > LOSS=		.064:	( .015)	( .032)	( .017)	kW		
TO NODE 860	.....:	11.16	-43.05	9.09	-154.82	10.60	99.34	AMP/DG
<860 > LOSS=		.141:	( .021)	( .104)	( .017)	kW		
		*	A	*	B	*	C	*
NODE: 842	VOLTS:	1.031	-3.25	1.029	-124.39	1.031	116.09	MAG/ANG

		-LD:	.00	.00	.00	.00	.00	.00	kW/kVR
kV11	24.900	CAP:		.00		.00		.00	kVR
FROM NODE 834	.....:	14.74	34.67	16.30	-95.64	15.12	151.03	AMP/DG	
<842 > LOSS=	.064:	( .015)		( .032)		( .017)		kW	
TO NODE 844	.....:	14.74	34.67	16.30	-95.64	15.12	151.03	AMP/DG	
<844 > LOSS=	.306:	( .068)		( .156)		( .083)		kW	
		*	A	*	B	*	C	*	
NODE: 844	VOLTS:	1.031	-3.27	1.029	-124.42	1.031	116.06	MAG/ANG	
	Y-LD:	143.41	111.54	142.97	111.20	143.51	111.62	kW/kVR	
kV11	24.900	Y CAP:		106.23		105.90		106.31	kVR
FROM NODE 842	.....:	14.47	37.12	16.29	-95.71	15.11	150.97	AMP/DG	
<844 > LOSS=	.306:	( .068)		( .156)		( .083)		kW	
TO NODE 846	.....:	9.83	78.88	9.40	-63.87	9.40	-170.67	AMP/DG	
<846 > LOSS=	.323:	( .043)		( .212)		( .068)		kW	
		*	A	*	B	*	C	*	
NODE: 846	VOLTS:	1.031	-3.32	1.029	-124.46	1.031	116.01	MAG/ANG	
	-LD:	.00	.00	.00	.00	.00	.00	kW/kVR	
kV11	24.900	CAP:		.00		.00		.00	kVR
FROM NODE 844	.....:	9.76	78.80	9.40	-52.54	9.78	-161.93	AMP/DG	
<846 > LOSS=	.323:	( .043)		( .212)		( .068)		kW	
TO NODE 848	.....:	9.76	78.80	9.40	-52.54	9.78	-161.93	AMP/DG	
<848 > LOSS=	.048:	( .007)		( .031)		( .010)		kW	
		*	A	*	B	*	C	*	
NODE: 848	VOLTS:	1.031	-3.32	1.029	-124.47	1.031	116.00	MAG/ANG	
	D-LD:	20.00	16.00	20.00	16.00	20.00	16.00	kW/kVR	
kV11	24.900	Y CAP:		159.43		158.86		159.56	kVR
FROM NODE 846	.....:	9.76	78.79	9.77	-42.47	9.78	-161.94	AMP/DG	
<848 > LOSS=	.048:	( .007)		( .031)		( .010)		kW	
		*	A	*	B	*	C	*	
NODE: 860	VOLTS:	1.030	-3.24	1.029	-124.39	1.031	116.09	MAG/ANG	
	Y-LD:	20.00	16.00	20.00	16.00	20.00	16.00	kW/kVR	
kV11	24.900	Y CAP:		.00		.00		.00	kVR
FROM NODE 834	.....:	5.87	-33.62	7.68	-156.52	5.29	86.10	AMP/DG	
<860 > LOSS=	.141:	( .021)		( .104)		( .017)		kW	
TO NODE 836	.....:	4.16	-30.19	5.96	-154.63	3.60	90.25	AMP/DG	
<836 > LOSS=	.039:	( -.035)		( .103)		( -.028)		kW	
		*	A	*	B	*	C	*	
NODE: 836	VOLTS:	1.030	-3.23	1.029	-124.39	1.031	116.09	MAG/ANG	
	-LD:	.00	.00	.00	.00	.00	.00	kW/kVR	
kV11	24.900	CAP:		.00		.00		.00	kVR

FROM NODE 860 . . . . .	1.49	-19.83	4.42	-150.74	1.74	68.08	AMP/DG
<836 > LOSS= .039:	( -.035)	( .103)	( -.028)	kW			
TO NODE 840 . . . . .	1.50	-20.01	2.33	-151.97	1.75	68.00	AMP/DG
<840 > LOSS= .002:	( -.014)	( .026)	( -.010)	kW			
TO NODE 862 . . . . .	.00	.00	2.09	-149.38	.00	.00	AMP/DG
<862 > LOSS= .000:	( -.005)	( .009)	( -.004)	kW			
----- * ----- A ----- * ----- B ----- * ----- C ----- *							
NODE: 840 VOLTS:	1.030	-3.23	1.029	-124.39	1.031	116.09	MAG/ANG
Y-LD:	9.27	7.21	9.26	7.20	9.28	7.22	kW/kVR
kVll 24.900 Y CAP:		.00		.00		.00	kVR
----- * ----- A ----- * ----- B ----- * ----- C ----- *							
FROM NODE 836 . . . . .	.79	-41.11	.79	-162.26	.79	78.21	AMP/DG
<840 > LOSS= .002:	( -.014)	( .026)	( -.010)	kW			
NODE: 862 VOLTS:	1.030	-3.23	1.029	-124.39	1.031	116.09	MAG/ANG
-LD:	.00	.00	.00	.00	.00	.00	kW/kVR
kVll 24.900 CAP:		.00		.00		.00	kVR
----- * ----- A ----- * ----- B ----- * ----- C ----- *							
FROM NODE 836 . . . . .	.00	.00	2.09	-149.50	.00	.00	AMP/DG
<862 > LOSS= .000:	( -.005)	( .009)	( -.004)	kW			
TO NODE 838 . . . . .			2.09	-149.50			AMP/DG
<838 > LOSS= .004:			( .004)				kW
----- * ----- A ----- * ----- B ----- * ----- C ----- *							
NODE: 838 VOLTS:			1.029	-124.39			MAG/ANG
-LD:			.00	.00			kW/kVR
kVll 24.900 CAP:				.00			kVR
----- * ----- A ----- * ----- B ----- * ----- C ----- *							
FROM NODE 862 . . . . .			.00	.00			AMP/DG
<838 > LOSS= .004:			( .004)				kW
----- * ----- A ----- * ----- B ----- * ----- C ----- *							
NODE: 864 VOLTS:	1.034	-3.17					MAG/ANG
-LD:	.00	.00					kW/kVR
kVll 24.900 CAP:		.00					kVR
----- * ----- A ----- * ----- B ----- * ----- C ----- *							
FROM NODE 858 . . . . .	.00	.00					AMP/DG
<864 > LOSS= .000:	( .000)						kW
----- * ----- A ----- * ----- B ----- * ----- C ----- *							
NODE: XF10 VOLTS:	1.000	-4.63	.998	-125.73	1.000	114.82	MAG/ANG
-LD:	.00	.00	.00	.00	.00	.00	kW/kVR
kVll 4.160 CAP:		.00		.00		.00	kVR
----- * ----- A ----- * ----- B ----- * ----- C ----- *							
FROM NODE 832 . . . . .	69.90	-32.29	70.04	-152.73	69.50	87.39	AMP/DG <
<XF10 > LOSS= 9.625:	( 3.196)	( 3.241)	( 3.187)	kW			
TO NODE 888 . . . . .	69.90	-32.29	70.04	-152.73	69.50	87.39	AMP/DG

<888 > LOSS= .000:		( .000)	( .000)	( .000)	kW			
		*	A	*	B	*	C	*
NODE: 888	VOLTS:	1.000	-4.64	.998 -125.73	1.000	114.82	MAG/ANG	
	-LD:	.00	.00	.00	.00	.00	kW/kVR	
kVll 4.160	CAP:		.00		.00		.00 kVR	
FROM NODE XF10 . . . . .:		69.90	-32.29	70.04 -152.73	69.50	87.39	AMP/DG	
<888 > LOSS= .000:		( .000)	( .000)	( .000)	kW			
TO NODE 890 . . . . .:		69.90	-32.29	70.04 -152.73	69.50	87.39	AMP/DG	
<890 > LOSS= 32.760:		( 11.638)		( 9.950)	( 11.173)	kW		
		*	A	*	B	*	C	*
NODE: 890	VOLTS:	.917	-5.19	.924 -126.78	.918	113.98	MAG/ANG	
	D-LD:	139.11	69.55	137.56 68.78	137.01	68.50	kW/kVR	
kVll 4.160	Y CAP:		.00		.00		.00 kVR	
FROM NODE 888 . . . . .:		69.91	-32.31	70.05 -152.75	69.51	87.37	AMP/DG	
<890 > LOSS= 32.760:		( 11.638)		( 9.950)	( 11.173)	kW		
		*	A	*	B	*	C	*
NODE: 856	VOLTS:			.998 -123.41			MAG/ANG	
	-LD:			.00	.00		kW/kVR	
kVll 24.900	CAP:				.00		kVR	
FROM NODE 854 . . . . .:				.00	.00		AMP/DG	
<856 > LOSS= .001:				( .001)			kW	

## The structure of 34-Node Test Feeder in Simulink

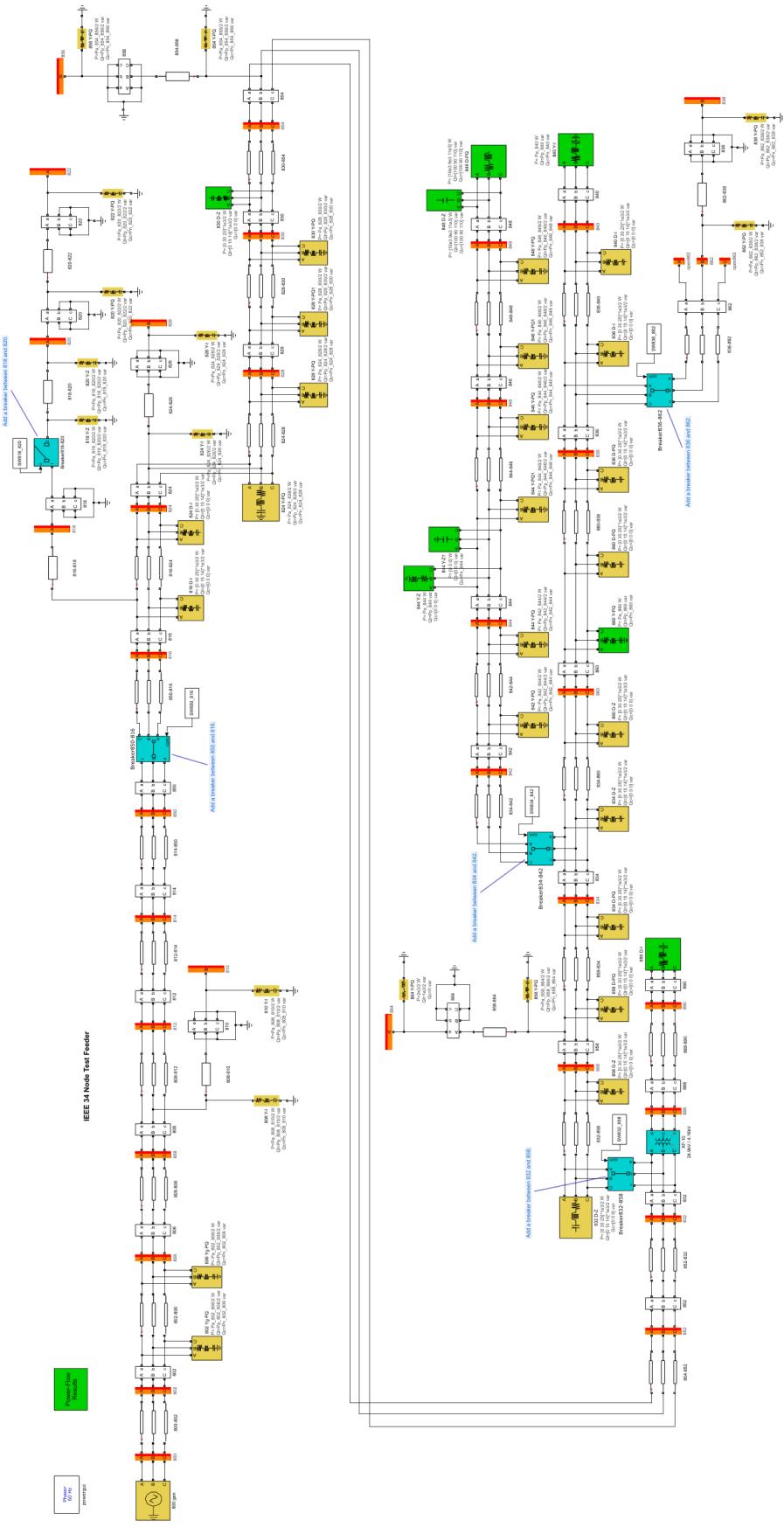


Figure 21: The structure of 34-Node Test Feeder in Simulink. [15]

## Appendix B: Documented Computer Listings

### The coding part of proposed neural network

```
#####
# building the model
input_image = Input(shape=(96, 96, 4))
### Downsampling ---- Encoder
print('-- Encoding --')
z = layers.Conv2D(16, (3,3), padding='same', activation='relu')(input_image) # shape 96 x 96
z = layers.BatchNormalization()(z)
z = layers.MaxPool2D((2,2))(z) # shape 48 x 48

z = layers.Conv2D(32, (3,3), padding='same')(z) # shape 48 x 48
z = layers.BatchNormalization()(z)
z = layers.MaxPool2D((2,2))(z) # shape 24 x 24
z = activations.relu(z)

z = layers.Conv2D(64, (3,3), padding='same', activation='relu')(z) # 24 x 24
z = layers.BatchNormalization()(z)
encoder = layers.MaxPool2D((2,2))(z) # shape 12 x 12

### Upsampling ---- Decoder
print('-- Decoding')
z = layers.Conv2D(64, (3, 3), padding='same', activation='relu')(encoder) # shape 12 x 12
z = layers.BatchNormalization()(z)
z = layers.UpSampling2D((2,2))(z) # shape 24 x 24

z = layers.Conv2D(32, (3, 3), padding='same')(z) # shape 24 x 24
z = layers.BatchNormalization()(z)
z = layers.UpSampling2D((2,2))(z) # shape 48 x 48
z = activations.relu(z)

z = layers.Conv2D(16, (3, 3), padding='same', activation='relu')(z) # shape 96 x 96
z = layers.BatchNormalization()(z)
z = layers.UpSampling2D((2,2))(z) # shape 48 x 48

# 4 channels because we have 4 channels in the input
decoder = layers.Conv2D(4, (3, 3), activation='sigmoid', padding='same')(z) # shape 48 x 48

# Building the model
autoencoder = Model(input_image, decoder)

# Printing the model summary
print(autoencoder.summary())
#####
```

**Figure 22:** Autoencoder Code for all Models