DATS 6203 Final Project Group-4

Individual Final Report

Yifu Li

December 8, 2020

In this project, I take the role of building the whole structure of the project. The motivation of this project is from my master thesis topic [1], but it is using different neural network architecture with different input dataset.

I proposed the flowchart of the proposed Autoencoder(AE), shown in Figure 1, where the first step is to collect phasor measurement unit (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.

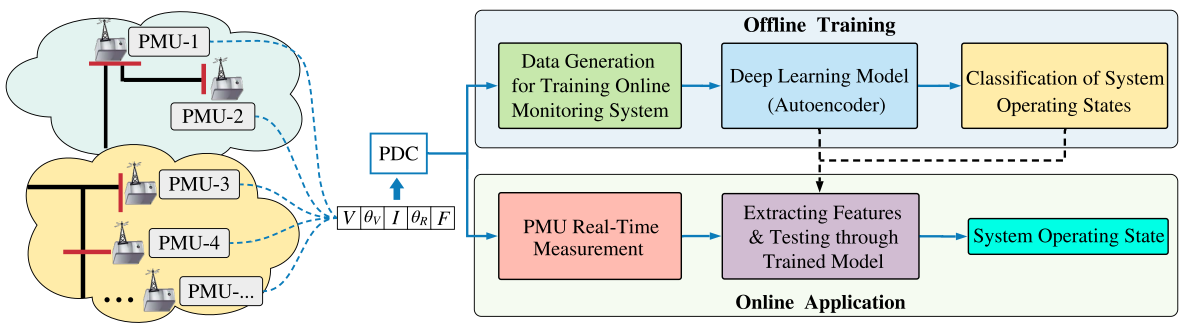

Figure 1: Autoencoder working flowchart.

Once the AE network model was trained well by the off-line dataset, it could identify the topologies of the known power grid even the data is totally new to the model. This Autoencoder architecture will be used as a building block in the proposed framework that identifies the power distribution network topology in real-time.

For the IEEE 34-Node Test Feeder system, I built the model on Matlab. As a dataset generator, I designed 8 topologies by switching 5 breakers on and off. And generate 8 groups of balanced number of data by adjusting 5 loads from 95% to 105%.

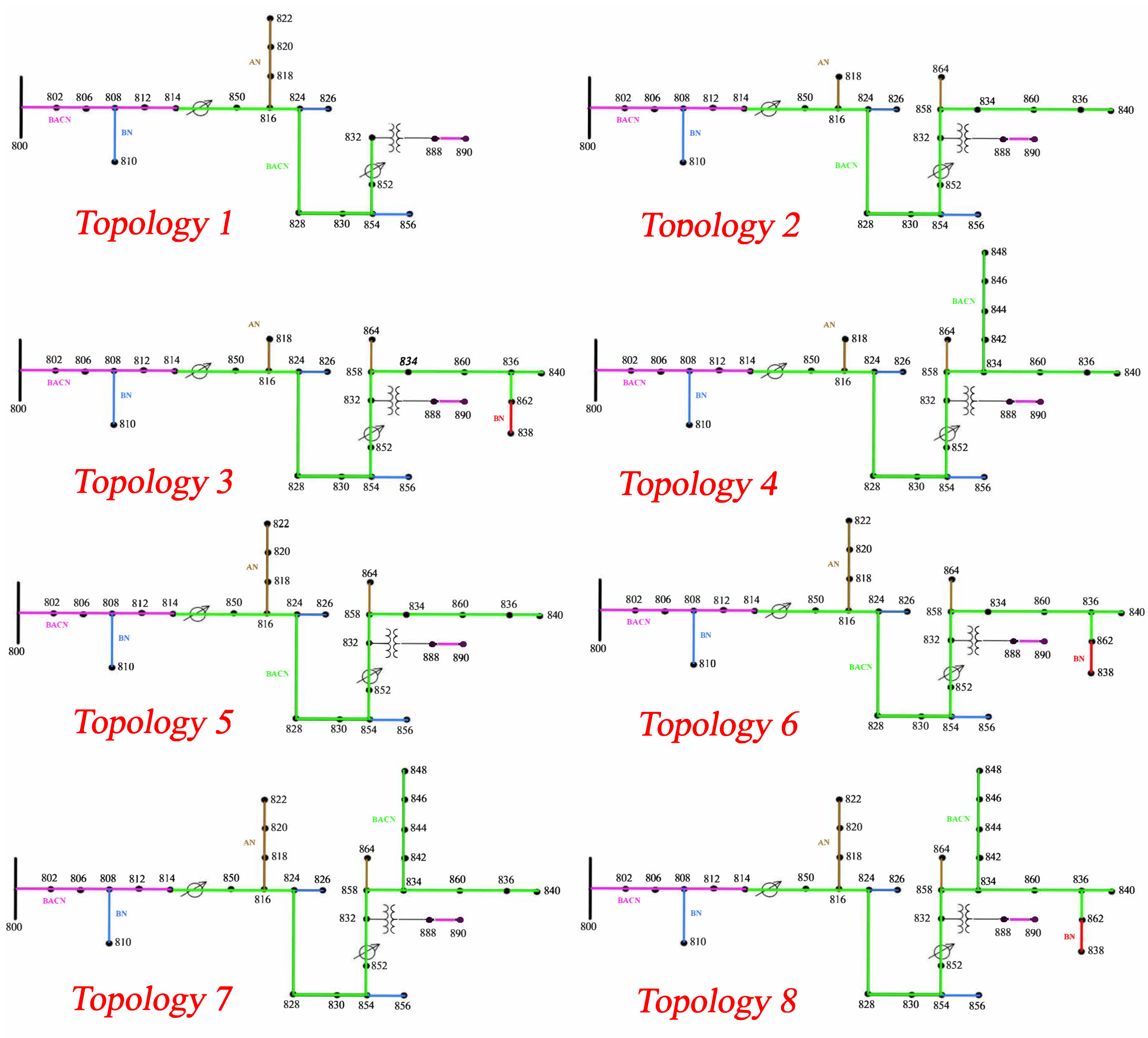

Figure 2: All Studied Network Topologies

I proposed the numerical testing framework. And generated 3 groups of Interfered Data which are totally new to the trained models. The table reults shown below.

Table 1: The prediction accuracy (%) by training and testing the AE in different extents of interferences for **8 topologies**

Interfered Data / **Models**	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)	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

Table 2: The prediction accuracy (%) by training and testing the AE in different extents of interferences for **7 topologies**

Interfered Data / **Models**	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 accuracy from left to right in the table went down which because the validation data getting "worse", and that also fits the common sense of the deep learning. And I found the reason for the low accuracy of the ideal model testing result in **Table 1**, and revised it in **Table 2**. The accuracy of the ideal model grew from around 55% to around 80%.

For the coding part, I totally wrote the code by myself. What's more, I upload two scripts which wrote by myself as the tools in my individual code folder. One "Move_Files.py" file is for cut the raw dataset into "Train", "Test" and "Val" three folders with 80%, 10% and 10%, of course the portion can be changed. Another "compare_file.py" file is for checking the integrity of the converting files. For example, when I convert the .csv files into .png files, there are huge numbers of files under converting and would takes a long time. If somehow I wrong click "stop" or something happened to terminated the converting, this tool script would help you check the unconverted files and saved them into a new folder, so you can easily continue the conversion next time.

References

[1] Y. Li, *Date-Driven Topology Identification in Power Distribution Systems with Machine Learning*. PhD thesis, The George Washington University, 2020.