Deep Reinforcement Learning for Power Grid Control

Final Project Report

S-93G

Power Systems Engineering Research Center
Empowering Minds to Engineer
the Future Electric Energy System

Deep Reinforcement Learning for Power Grid Control

Final Project Report

Project Team
Yang Weng, Project Leader
Arizona State University

Graduate Student
Kishan Prudhvi Guddanti
Arizona State University

PSERC Publication 21-03

January 2021

For information about this project, contact:

Yang Weng
Arizona State University
School of Electrical, Computer and Energy Engineering
Tempe, AZ, 85287
Phone: 650-924-3618
Fax: 480-965-3837
Email: yang.weng@asu.edu

Power Systems Engineering Research Center

The Power Systems Engineering Research Center (PSERC) is a multi-university Center conducting research on challenges facing the electric power industry and educating the next generation of power engineers. More information about PSERC can be found at the Center's website: http://www.pserc.org.

For additional information, contact:

Power Systems Engineering Research Center
Arizona State University
527 Engineering Research Center
Tempe, Arizona 85287-5706
Phone: 480-965-1643
Fax: 480-727-2052

Notice Concerning Copyright Material

Acknowledgements

We would like to thank and acknowledge the technical feedback and helpful suggestions provided by the project advisors, especially:

- Patrick Panciatici, RTE
- Antoine Marot, RTE
- Benjamin Donnot, RTE

Executive Summary

Power grid operation and control research has gained a lot of traction due to the high penetration of distributed energy resources in power grids. These emerging distributed energy resources and their sub-optimal integration planning result in the underutilization of the power grid's transfer capability. An ample amount of research is conducted to maximize the transfer capability by using demand-side management, load shedding, generation redispatch, etc. However, these methods are more costly from a transmission system operator point of view. A more economical and effective method to maximize the transfer capability while reducing both losses and contingencies is by using the topology reconfiguration techniques. These techniques include the switching of transmission lines using the breakers (line switching - changes the total number of edges in a graph) and switching of elements at a substation between two different busbars (bus switching – may change the total number of nodes in a graph).

However, historically, when specific important lines in the power grid, either lightly or heavily loaded lines become out-of-service, it resulted in numerous power grid outage events. This inability of the transmission and generation infrastructure to support the total load on the power grid results in the long-term voltage instability (LTVI) phenomenon. This can ultimately result in a system-level blackout, islanding, or trigger other instability mechanisms. Thus, the power system operators need to assess the occurrence of the LTVI phenomenon when topological actions are performed to mitigate the thermal limit-based contingencies in the power grid.

The topology reconfiguration problem translates to a time-series planning problem for the power grids and it is computationally infeasible to obtain the global solution due to the combinatorial nature of the action space (topological actions and time-dependent loading conditions on the power grid). In addition to the high complexity of the topology reconfiguration problem, one must also track/assess the LTVI phenomenon. Unfortunately, the existing methods to track voltage instability require multiple power flows to be computed and this adds more complexity to the original topology reconfiguration problem. The proposed method helps to convert heavy time intensive deep reinforcement learning (DRL) based topology controllers to that of a moderate one to address both safety and optimal control problems in power grids as a real-time application. Hence, this project focuses on the machine learning that can learn the generalized physical laws such as distance to collapse while using only a limited amount of data.

In this project, a fast and accurate machine learning-based method is provided. This method generalized its learning by observing the small subset of the solution space to a superset of the same solution space. This enables the development of futuristic topology controllers that can assist the power system operators in real-time while also considering the aspect of voltage instability in addition to thermal-based contingencies which cause cascading events. The proposed method uses graph neural networks (GNNs) to predict the distance to LTVI phenomenon given the state of the power grid. The state of the power grid is a function of the loading at each substation as well as the topological actions performed by the operator at that current time step. Therefore, the inputs of the GNNs are the adjacency matrix and loading conditions of the power grid. The outputs of the GNNs predict the distance to LTVI phenomenon given the input information. The training dataset included only a subset of the solution space i.e., topological actions that impact only one substation at a time. However, two testing datasets are used to evaluate the performance of the trained model.

Testing dataset-1 has unseen scenarios from the training dataset. Testing dataset-2 is a superset of the training dataset i.e., topological actions that impact two substations at a time. It is highly desired for the machine learning model to not only be very fast but also highly accurate and generalize its learning to unseen scenarios. Thus, high accuracy on both testing datasets indicates the best performance of the trained machine learning model. The intuition behind using the GNNs for generalization is due to its message-passing feature which uses the computational graph derived based on the neighborhood of a given node in the graph.

The trained GNNs have 94% accuracy on testing dataset-1 and surprisingly it also generalizes well for the unseen superset scenarios i.e., 93% accuracy on the testing dataset-2. It is shown that the proposed model using GNNs is 1440 times faster than the state-of-the-art recently published methods with an accuracy of 94% and especially, it generalizes very well for new topologies. Thus, making the proposed method more attractive to be used in developing the real-time topology reconfiguration controllers for power system operators in control centers. This not only optimizes the transfer capability of the power grid, but it also ensures the safety of the operation of the power grid.

The proposed method is added as a feature to an existing open-source software called Grid2Op, developed by the RTE, a utility based out of France. Grid2Op helps the users to develop both expert systems as well as reinforcement learning-based topology controllers for power grid operation and control. This research-grade software along with the added feature from this project can now enable to development of an AI recommendation system in control centers for topology reconfigurations while also considering the safe operation of the power grid.

Table of Contents

1. Introduction....1

1.1 Background....1

1.2 Report organization....3

2. Overview of the problem....4

2.1 Problem formulation with an example....4

2.1.1 Power grid, action space and model formulation of topology controller....4

2.1.2 Contingencies and their relation to voltage instability mechanics....6

2.1.3 Distance to voltage collapse as an additional constraint and its importance....7

2.1.4 Main issues with addition of distance to voltage collapse (VSI) as a constraint....12

2.2 Goal of the project....13

3. Graph neural networks to predict VSI and generalize mapping rule....14

3.1 Introduction to graph neural networks....14

3.2 Inputs and outputs of the graph neural networks....17

3.3 Datasets used to train and test the graph neural networks....17

3.4 Various graph neural network architectures....18

3.5 Metrics to analyze the performance of trained graph neural networks....20

3.6 Performance analysis of the trained graph neural networks....22

4. Conclusion and future work....27

References....28

List of Figures

Figure 2.1 - IEEE-14 bus grid. Bus 1 and bus 2 in legend indicate the two busbars in a substation. 4

Figure 2.2- A graph representation with four nodes (substation representation). 5

Figure 2.3- Bus splitting action using the busbars in a substation. 5

Figure 2.4- Total nodes in the graph representation change due to bus splitting actions. 5

Figure 2.5- VSI for IEEE-30 system when all consumptions and productions are linearly scaled. 7

Figure 2.6- IEEE- 14 bus system in Grid2Op [29] with the feature to measure distance to collapse. The power grid is operated at base case loading condition. 8

Figure 2.7- IEEE-14 bus system in Grid2Op [29] when the power grid is operated at operated very close to voltage collapse situation. 9

Figure 2.8- IEEE-14 bus system in Grid2Op [29] when the power grid's loading condition is linearly increased from base case until voltage collapse. 9

Figure 2.9- Cascading event in the IEEE-14 bus system. 10

Figure 2.10- The production and consumption profiles of all loads and generators in the IEEE-14 bus system. 11

Figure 2.11- Impact of topological actions on the VSI values. 11

Figure 2.12- Visualization of states and possible actions for each given state. 12

Figure 3.1- High-level model of the problem to estimate VSI at every node in the grid. 14

Figure 3.2- An undirected graph network with nodes and edges. 14

Figure 3.3- Graph convolution or message passing step at node B. 15

Figure 3.4- Graph convolution methodology employed in graph neural networks. 15

Figure 3.5- Intuition behind the generalization capability of GNNs for unseen topologies. 16

Figure 3.6- Inputs and outputs of the graph neural networks. 17

Figure 3.7- Training, testing and generalization testing datasets for the GNNs. 18

Figure 3.8- GNN model with “GCN” convolution layer and pooling layer. 19

Figure 3.9- GNN model with “GraphConv” convolution layer and pooling layer. 19

Figure 3.10- Predicted critical VSI versus actual critical VSI of the grid for various scenarios. 20

Figure 3.11- Cases that are bad predictions versus correct predictions. 21

Figure 3.12- Positive (unsafe) and negative (safe) classes for the VSI estimation problem. 22

Figure 3.13- GCNConv versus GraphConv with pooling layers. 22

Figure 3.14- GCNConv versus GraphConv without pooling layers. 23

Figure 3.15- For testing dataset-1, performance comparison between GCNConv with pooling versus GraphConv without pooling layers. 24

Figure 3.16- For testing dataset-2 (generalization test dataset), performance comparison between GCNConv with pooling versus GraphConv without pooling layers. 25

List of Tables

Table 2.1- Size of the action space 6

Table 2.2- Comparison between existing methods and requirements of an ideal method for DRL.. 13

Table 3.1- Comparison between the existing methods and proposed method.............................. 26

1. Introduction

1.1 Background

Power system operation and control is an important area of power grids research that has gained a great amount of interest lately. Especially, due to the intermittent energy resources on the production side as well as distributed energy resources on the consumer side [1]. This makes the power grid planning and controls more dynamic than ever before by adding uncertainties to it [02, 03]. Not only that the installation of new equipment is costly in terms of investment, but public acceptance is also a growing issue. Thus, instead of traditionally developing the grid infrastructure by increasing its capacity by building new lines, it is important to optimize the operation of existing power grid infrastructure. This optimization can be done by any digital means and every flexibility at the operator's disposal. Once such a cost-effective approach as well as a flexible option used by some utilities would be to reconfigure the grid topology [4].

However, it is not trivial to optimally control the grid topology at a scale beyond the line/branch switching actions. Even though there are more complex topology actions such a bus splitting that may improve the grid capacity, it is not easy to obtain an optimal solution considering the stochastic nature of the intermittent energy sources and complex coordinated decisions to be taken over time. [5] provides an expert system-based topology controller that uses bus splitting actions to control the power grid in acceptable computation times, but it does not consider its decisions over a time horizon as a strategy to manage various cooperative and non-cooperative decisions. [6] provides an optimal control strategy over a time horizon but they are hard to deploy in control rooms because of limited computational budget. Thus, methods need to be not only optimal but also very fast for it to be deployed as a real-world application.

With the latest breakthroughs in Artificial Intelligence (AI) field from AlphaGo at Go [7] and Libratus at Poker [8], it is shown that with enough computing and a good simulator, deep reinforcement learning (DRL) can develop control strategies to solve the most challenging problems with expertise that far beyond the elite human's capability at a scale in real-time. Motivated by these developments, recently a great amount of interest is shown by both academic as well as the industrial community to use a DRL-based topology controller to develop real-time control strategies for power grid operations and control [4]. [4] organized a competition Learning to Run a Power Network (L2RPN) with an emphasis on using the various flexibilities such as generation redispatch, line switching, and bus splitting actions to optimally control the power grid. It provides extensively developed topology controllers by the research and industrial community for larger grids up to 118-bus system [9]. However, the topology reconfiguration problem in [9] has a fundamental assumption that the power grids do not get drastically impacted by the topology actions. There have been many voltage instability incidents that occurred around the world and it was found that the out-of-service of heavily loaded lines and at times lightly loaded lines trigger such events [10-13].

One way to consider the voltage instability phenomenon and topology reconfiguration are by operating the power grid such that the optimal topology actions are decided while ensuring the power grid is operated at a threshold margin set by the operator. This margin set by the operator is defined as a metric that indicates proximity to system voltage collapse and they are known as voltage collapse proximity indices (VCPIs). Typically, for the application of developing real-time topology controller, it is desired to have a method that is accurate, generalizes for the unseen topologies and can compute VCPIs fast enough for real-time. More broadly there are two popular types of methods (VCPIs) in the literature to calculate the proximity to system collapse. It is important to note that these two types of VCPIs require real-time PMU measurements for their computation [14]. The two methods are 1) Sensitivity based VCPIs [15,16] and 2) Thevenin and analytically exact method based VCPIs [17-20].

Sensitivity based VCPIs even though are generally fast (given the Jacobian) occasionally they are prone to errors due to an assumption that the operator must know the trajectory of load change when the power grid blackout occurs [21,22]. [23] and [24] provide a topology reconfiguration algorithm that can learn the optimal topology actions in real-time by considering line switching and bus splitting actions respectively. However, [22-25] assume that the Jacobian near the voltage instability phenomenon is known and sometimes it is not realistic to predict the trajectory of load change where voltage instability could occur. This assumption makes the sensitivity based VCPI methods prone to errors [21].

Thevenin and analytically exact method based VCPIs do not have an assumption that is present in the sensitivity based VCPI methods [21-25] and is more reliable [17-20,26]. However, the analytically exact VCPI method is more accurate when compared to that of the Thevenin based VCPI method [26]. Therefore, in this work, we use an analytically exact method based VCPI known as voltage stability index (VSI) [26] to compute the proximity to system collapse given the state of the power grid. However, a major challenge with using such measurement based methods [17-20,26] to compute the VCPIs is that they require PMU voltage measurements of the given state. Therefore, when the computation of VSI is added as a constraint to the topology reconfiguration optimization problem, the problem becomes not trivial and it is not tractable for deployment as a real-time application. Hence, we need a method that is not only as accurate as VSI [26] but it is also a faster method like sensitivity based VCPIs [22-25]. One potential solution for such a problem could be to learn to predict the VSI given the state of the power grid using machine learning. Due to the combinatorial nature of the load change over time and topology actions over a time horizon, it is computationally infeasible to generate a dataset that contains all possible scenarios that could potential occur. Hence, the proposed method should not only be as accurate as [26] and as fast as [25] but it should also be able to generalize its learning for new unseen topologies [32].

The proposed approach for learning the VSI uses graph neural networks (GNNs) to learn the mapping rule between the state of the power grid and its VSI value. The inputs of the proposed method use state of the power grid that includes the real power and voltage setpoints of all the generators, real and reactive power of all the loads in the power grid, and the adjacency matrix (Laplacian) of the power grid. The output of the proposed method uses the VSI values at all nodes in the power grids. The fundamental intuition behind using the GNNs to predict the VSI value of the power grid is due to its message-passing feature. This message passing feature in GNNs is like a convolution step in convolution neural networks except that this message passing step uses only the information from neighboring nodes to update the main node information. It is like a shared function learnt over nodes and that can hence generalize to any new unseen node (topology) or node attributes (scenario). This computation network obtained using only the neighbors is like how the voltage at a node is the function of neighboring bus parameters in power systems. This work shows that GNNs can learn to generalize the mapping rule for unseen topologies while predicting VSI values with high accuracy. Finally, this newly proposed method can be combined with deep reinforcement learning (DRL) based topology controllers such as [04,09] to address both safety and optimal control problems in power grids as a real-time application.

1.2 Report organization

This report is organized into four chapters. The first chapter discussed the background of the problem, limitations of the existing methods, and objectives for a new machine learning-based method that is fast, accurate, and generalizes its mapping rules. The second chapter describes the overview of the problem with the help of an IEEE-14 bus system example. It also provides the optimization formulation of the problem. This chapter also presents the main issues of the existing state-of-the-art (SOTA) method and desired potential improvements over the SOTA method. The third chapter introduces the graph neural networks (GNNs) and its message-passing feature (graph convolution step). It also contains the inputs, outputs, various datasets used to train the GNNs, and various GNN architectures used in this project. This chapter also compares the results of various techniques used and finally, provides a comprehensive discussion about the best available solution among the results. The fourth chapter contains the conclusion and future scope of the project.

2. Overview of the problem

2.1 Problem formulation with an example

2.1.1 Power grid, action space and model formulation of topology controller

<u>Power grid</u>: In this project, the experiments are conducted on the IEEE-14 bus system shown in Figure 2.1. The generations are indicated by the green circles and the consumptigons are indicated by the yellow circles. Each node (blue circle) represents a substation. Each substation in this grid contains two bus bars. These busbars are names bus 1 and bus 2 for each substation node as shown in the legend of Figure 2.1.

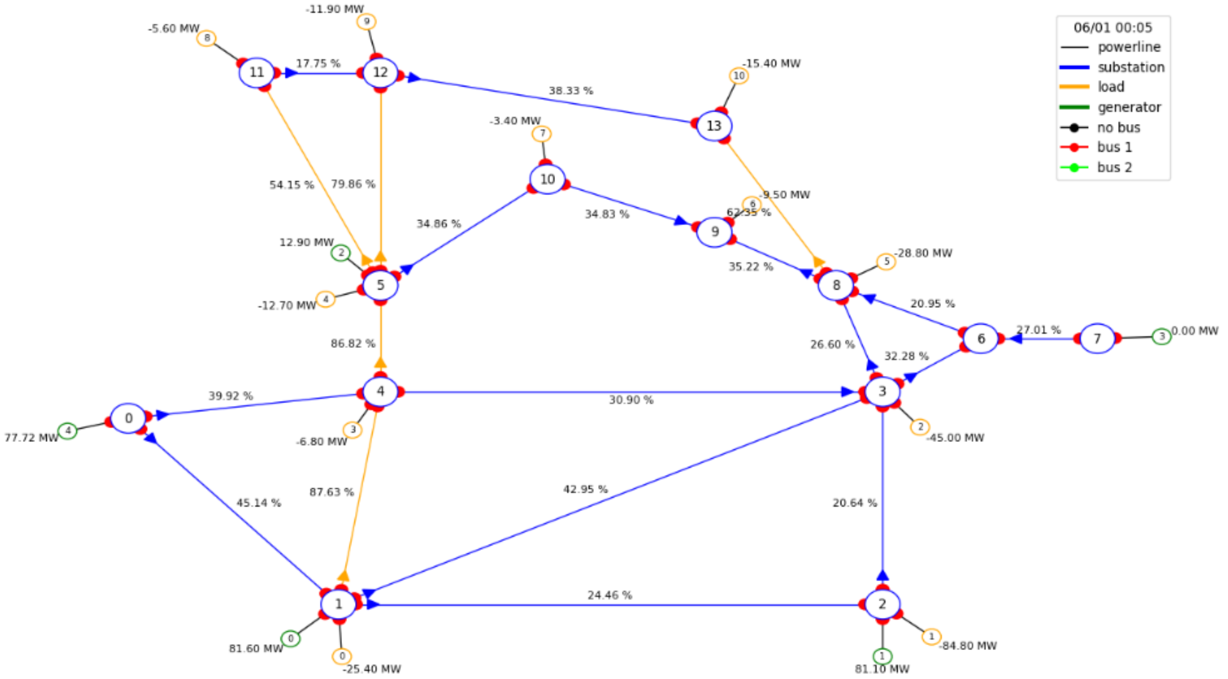


Figure 2.1 - IEEE-14 bus grid. Bus 1 and bus 2 in legend indicate the two busbars in a substation.

<u>Action space</u>: The action space of the topology reconfiguration of the power grid involves two types of actions i.e., “line switching” and “bus splitting” actions. A line switching action decides whether a transmission line should be in-service or out-of-service. This changes the number of edges in the graph representation of the power grid. However, a bus splitting action is more complex, and it can be explained with the help of Figure 2.2. Figure 2.2 shows a graph with four substations and it also shows the inside configuration of a substation with two busbars. Each substation in the power grid is designed to have two busbars in our experimental setup of the power grid. It can be observed from Figure 2.2 that transmission lines from the three substations (nodes) are connected to both busbars (1 & 2) of the fourth substation. An example of a bus splitting action on the graph shown in Figure 2.2 could be that the three incoming transmission lines are separated by connecting two of the lines to busbar 2 and the other one line to busbar 1 located at the substation four. This is presented in Figure 2.3 and it results in a new node in the graph as shown in Figure 2.4. Thus, a bus splitting action may change the number of nodes in the graph representation of a power grid.

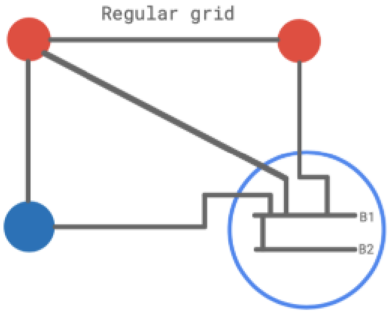


Figure 2.2- A graph representation with four nodes (substation representation).

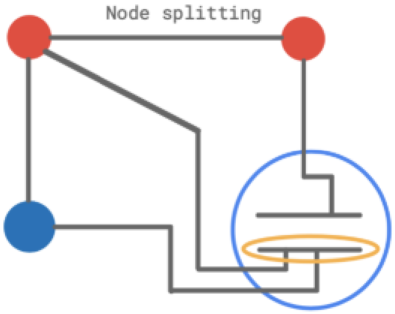


Figure 2.3- Bus splitting action using the busbars in a substation.

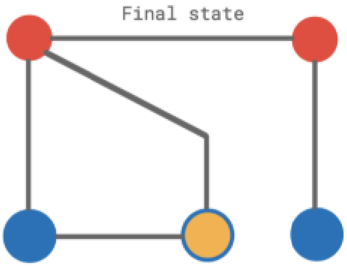


Figure 2.4- Total nodes in the graph representation change due to bus splitting actions.

Size of the action space: The IEEE-14 bus system presented in Figure 2.1 has 14 substations, 20 transmission lines, and 16 injections. The total possible line switching actions for a grid with 20 transmission line is 2^{20}. Similarly, the total bus splitting actions at a substation with k elements is $\sim 2^{k-1}$. Table 2.1 shows the approximate size of the total possible actions available to be selected by the operator at a given time step. Thus, the complexity of selecting an optimal action at a given time step is not trivial let alone computing the strategy over a time horizon.

Table 2.1- Size of the action space

Power network	***Number of substations***	***Number of power lines***	***Number of bus splitting actions***	***Number of line switching actions***
IEEE - 14	*14*	*20*	*1,397,519,564*	*1,048,576*
IEEE - 118	*118*	*186*	*3.88e+76*	*9.81e+55*

Model formulation: The aim of the topology controller is to maximize the total line loading on the grid over a time horizon $t = \{1,2, \ldots, n\}$ by identifying the optimal topology N_t for every time step t with transmission line switching and bus splitting actions. $f_{N_t}(x_t) = 0$ represents the real and reactive power flow equations corresponding to the timestep t with topology N_t and state vector x_t. The constraint 2.2 represents that AC power flow solution must exist for entire time horizon t. The constraint 2.3 represents that the voltage at bus i i.e., $V_{i,t}$ must lie within the specified thresholds over the entire time horizon t where $\forall i \in B$ and B is the set of all buses in the power grid. The constraint 2.4 represents that the loading on a transmission line p i.e., $I_{p,t}$ must be less than its thermal limits $I_{p,max}$ over the entire time horizon t where $\forall p \in E$ and E is the set of all transmission lines in the power grid.

$$\max_{N_t} = \sum_{t=1}^{t=n} \sum_{\forall p \in E} \left(\frac{I_{p,t}}{I_{p,max}} \right) \tag{0.1}$$

$$Sub.to: f_{N_t}\left(x_t\right) = 0 \ \forall t = \{1, 2, \mathrm{L}\ \ , n\}, \tag{0.2}$$

$$0.95 \le V_{i,t} \le 1.05 \ \forall i \in B, \forall t = \{1, \mathrm{L}\ \ , n\}, \tag{0.3}$$

$$I_{p,t} \le I_{p,max} \ \forall p \in E, \forall t = \{1, \mathrm{L}\ \ , n\}. \tag{0.4}$$

2.1.2 Contingencies and their relation to voltage instability mechanics

There have been many voltage instability events occurred around the world and the contingency of transmission lines is one of the major causes. South Florida, May 17, 1985, is one such scenario where a brush fire caused three lightly loaded 500kV lines to trip which resulted in voltage collapse and blackout within a few seconds [27,28]. Interestingly, the dynamic simulations showed that the system would recover but however, that was not the case, resulting in 4,292 MW load loss. There are many such events reported around the world and some of them are briefly reported in [10-13,27,28].

Hence, when the operator uses a topology controller to optimize for maximum utilization of the grid infrastructure, it is also equally important for the operator to know if the selected topology action would make the power grid safe or unsafe to operate. [23-25] presents this problem of maximizing the distance from the voltage collapse point while minimizing the thermal limit-based contingencies on the grid. However, their formulation assumes that the look-ahead consumption projection is the loading condition at which the power grid would collapse. But it is not straightforward whether the selected consumption projection by the operator is correct and thus it may cause inaccurate margin calculation at times. [26] provide a better metric to measure the distance to voltage collapse without the caveat of needing to predict the loading condition at which the system might experience voltage collapse. This method is more accurate than [21-25] and it is also not prone to any inaccuracies due to its assumption-free formulation. Therefore, in this project, the voltage stability index (VSI) from [26] is used to compute the distance to voltage collapse given the state of the power grid as described in the next section

2.1.3 Distance to voltage collapse as an additional constraint and its importance

Let p_d and q_d be the real and reactive power injections at bus d. Let $g_{k,d} + j \cdot b_{k,d}$ be the $(k,d)^{th}$ element of the admittance matrix. $v_{k,r}$ and $v_{k,i}$ be the real and imaginary part of the voltage phasor at bus k. The neighboring bus set of bus d is represented by $\mathrm{N}\ (d)$. The voltage stability index (VSI_d) [26] at a bus d is given by

$$VSI_d = \left(c_p - \frac{\|b_p\|^2}{4}\right) \cdot \left(c_q - \frac{\|b_q\|^2}{4}\right) - \left(\frac{1}{8} \cdot \| b_p - b_q\|^2 - \frac{1}{2} \cdot \left(\frac{\|b_p\|^2}{4} - c_p\right) - \frac{1}{2} \cdot \left(\frac{\|b_q\|^2}{4} - c_q\right)\right)^4, \qquad \textbf{(0.5)}$$

$$\text{where } b_p = \begin{bmatrix} \frac{t_{d,2}}{t_{d,1}} & \frac{t_{d,3}}{t_{d,1}} \end{bmatrix}^T, b_q = \begin{bmatrix} \frac{-t_{d,3}}{t_{d,4}} & \frac{t_{d,2}}{t_{d,4}} \end{bmatrix}^T, c_p = \frac{-p_d}{t_{d,1}}, c_q = \frac{-q_d}{t_{d,4}}, t_{d,1} = -\sum_{k \in \mathrm{N}\ (d)} g_{k,d},$$

$$t_{d,2} = \sum_{k \in \mathrm{N}\ (d)} (v_{k,r} g_{k,d} - v_{k,i} b_{k,d}), t_{d,3} = \sum_{k \in \mathrm{N}\ (d)} (v_{k,r} b_{k,d} + v_{k,i} g_{k,d}), and, t_{d,4} = \sum_{k \in \mathrm{N}\ (d)} b_{k,d}\ .$$

In this section, first, the characteristics of VSI on various grid operating scenarios are explained to understand its resourcefulness in identifying the distance to voltage collapse. Second, an updated model formulation is provided for topology controllers which will have an additional constraint to operate power grids within an operational threshold using the VSI.

<u>Characteristics of VSI</u>:

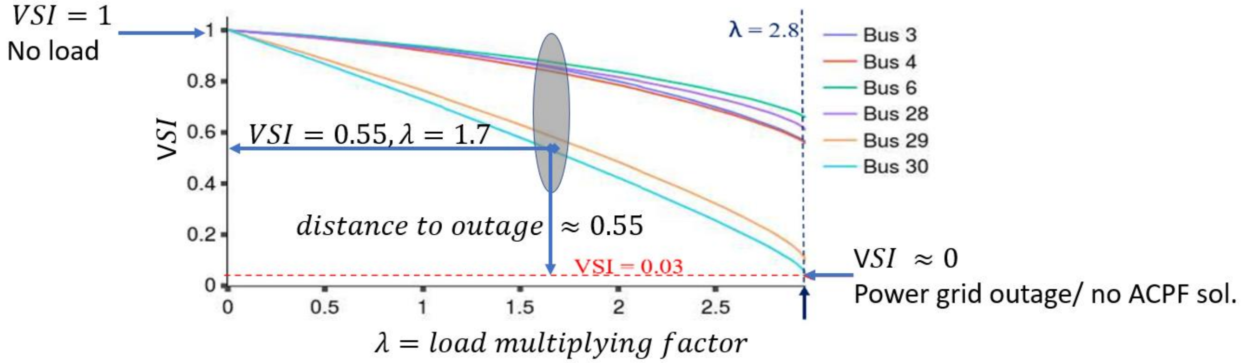


Figure 2.5- VSI for IEEE-30 system when all consumptions and productions are linearly scaled.

Figure 2.5 shows the plot of VSI calculated at buses 3, 4, 6, 28, 29, and 30 of an IEEE-30 bus system under uniformly increasing loading conditions. It can be observed that when VSI = 1, it represents a no-load condition, and when VSI ≈ 0, it represents the voltage instability point. Additionally, when the load multiplying factor $\lambda = 1.7$ the corresponding VSI value at bus 30 is 0.55 which provides a measure of distance to voltage instability condition. Later in this section, it is also shown that VSI value changes dynamically based on the operating/loading condition and it is not necessary to assume a load change trajectory in the simulation unlike [23-25]. However, it always represents the measure of distance to voltage instability phenomenon. An unnormalized VSI does not have an upper bound and in this report the VSI used is the unnormalized version described in Equation (0.5).

<u>VSI indicates closeness to voltage collapse</u>: Let us consider the IEEE-14 bus system presented in Figure 2.6, each substation in the grid is assigned a VSI value when the system is operated at base case loading conditions. It can be observed that the highlighted area in Figure 2.6 represents the area that is farther away from the generators and thus it is reflected in the magnitude of the VSI values. Similarly, when the IEEE-14 bus system is operated at the specific loading condition as shown in Figure 2.7 when all the loads and productions are multiplied by a multiplying factor of 4. When the multiplying factor is 4.1 there is no feasible AC power flow solution. Figure 2.7 indeed shows that buses 9, 13 are critical and their VSI values change from 0.46 to 0.1, 0.65 to 0.19 respectively. The bus 13 value of 0.1 (multiplying factor = 4) indicates that the grid is close to voltage collapse and indeed a slight increase (multiplying factor = 4.1) in the loading conditions result no valid AC power flow solution.

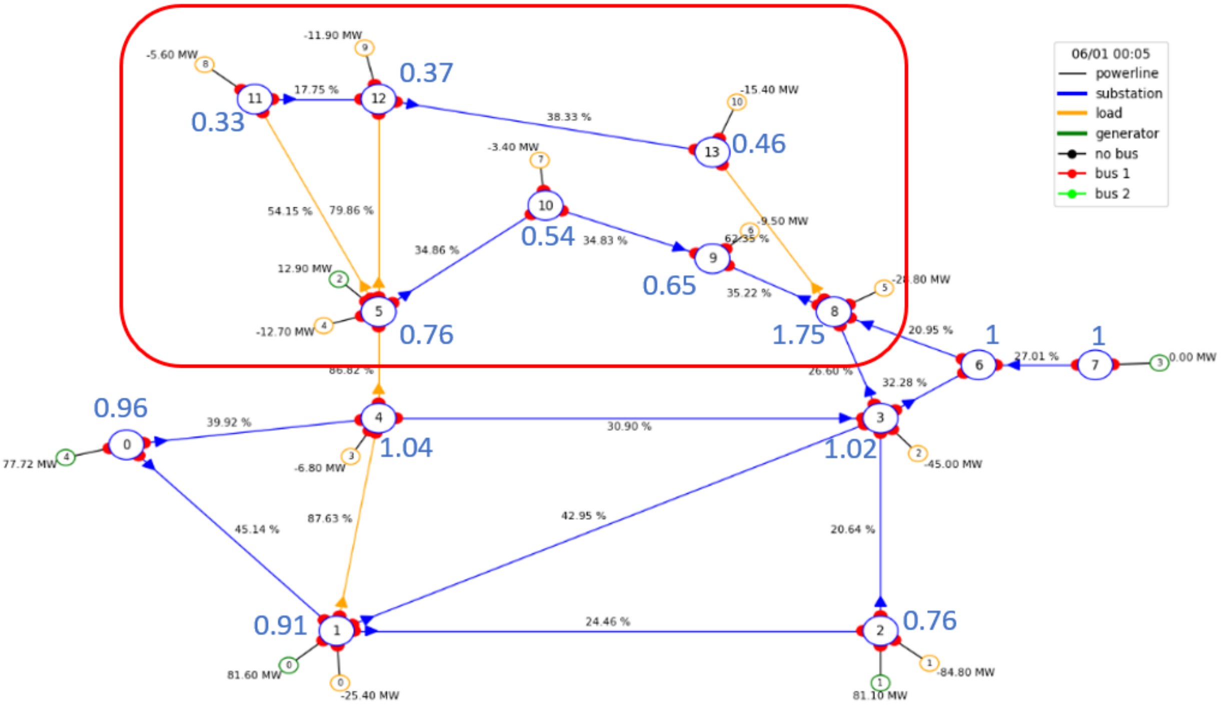


Figure 2.6- IEEE- 14 bus system in Grid2Op [29] with the feature to measure distance to collapse. The power grid is operated at base case loading condition.

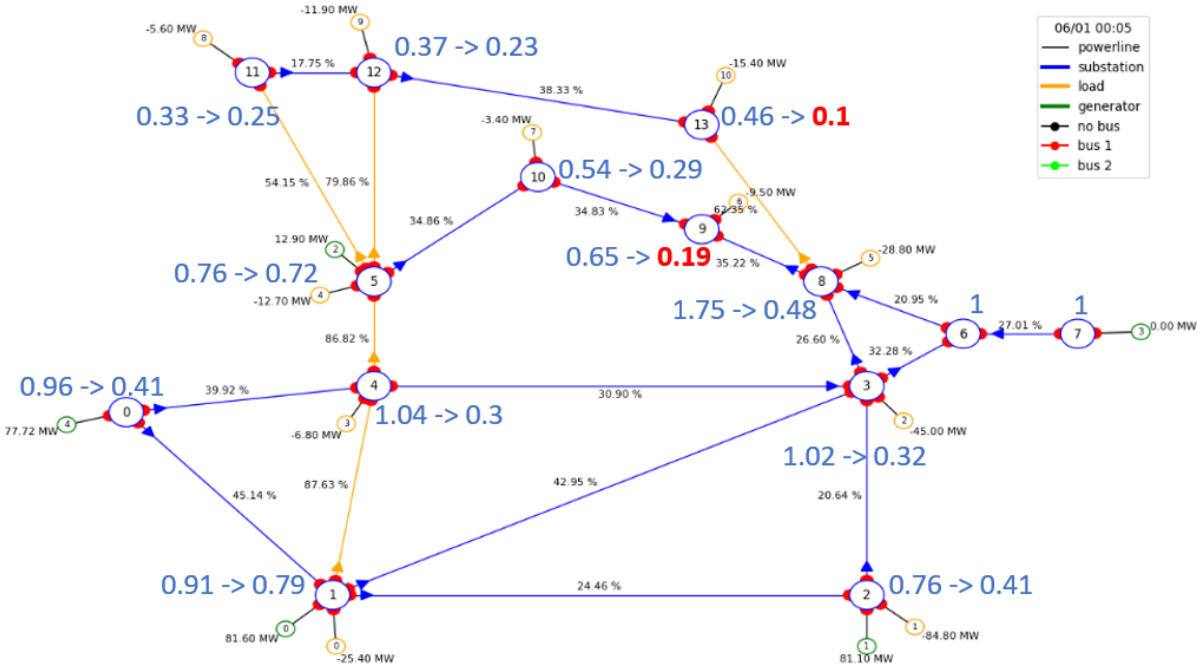


Figure 2.7- IEEE-14 bus system in Grid2Op [29] when the power grid is operated at operated very close to voltage collapse situation.

Monitoring the VSI – tracks the system operation's health:

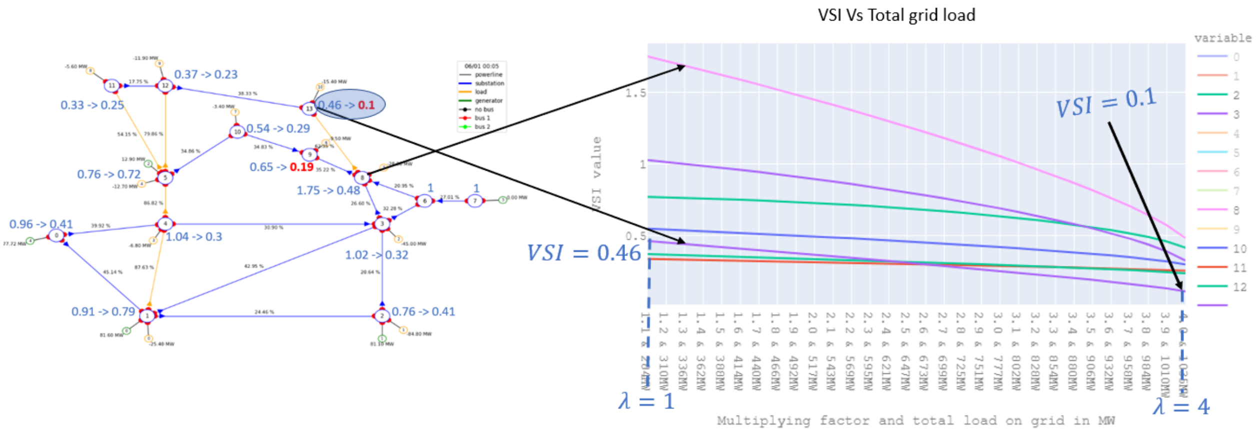


Figure 2.8- IEEE-14 bus system in Grid2Op [29] when the power grid's loading condition is linearly increased from base case until voltage collapse.

Figure 2.8 shows VSI values at every bus in the IEEE-14 bus system when the loading on the grid is linearly increased from the base case until the voltage collapse scenario. In Figure 2.8, the VSI line corresponding to bus 13 is the not most critical (minimum VSI value among all buses in the power grid) until λ = 2.6. This highlights that depending on the loading conditions of the power grid (λ), the VSI only indicates the distance to collapse from the given state of the power grid. Thus, it is important to continuously track the VSI value in real-world situations (by tracking the state of the power grid) where the power grid's operating conditions change based on the daily consumption profile of the users. Mathematically, Equation (0.5) also shows that VSI at bus d is dependent only on the current state information of the power grid and it does not require any information about the prediction of load trajectory unlike [23,24].

Contingencies and topological actions impact the VSI value: In this case study, realistic consumption and load profiles are injected into the power grid. These profiles are modeled using the Chronics2Grid [29] for various generators types present in the grid shown in Figure 2.9. In Figure 2.9, generators 0, 1, 2, 3, 4, 5 are nuclear, thermal, wind, solar, solar, hydro respectively. Figure 2.10 presents the production and consumption profiles in the power grid over a time horizon. The transmission lines in this IEEE-14 bus system are provided with thermal limits and when a transmission line is overloaded, it will disconnect and becomes out-of-service. Without performing any topological actions on the power grid, the injections to the power grid shown in Figure 2.10 cause thermal limit violations in the grid which result in a cascading event that eventually leads to system collapse. The cascading event involves the transmission line connecting substations 1 and 4 disconnects first at time step 1382 followed by another transmission line between substations 4 and 5 being out-of-service at time step 1387. Finally, as shown in Figure 2.9, the transmission lines connecting substations 8, 9 and 8, 13 disconnect due to overloading of 311.72% and 174.11% respectively.

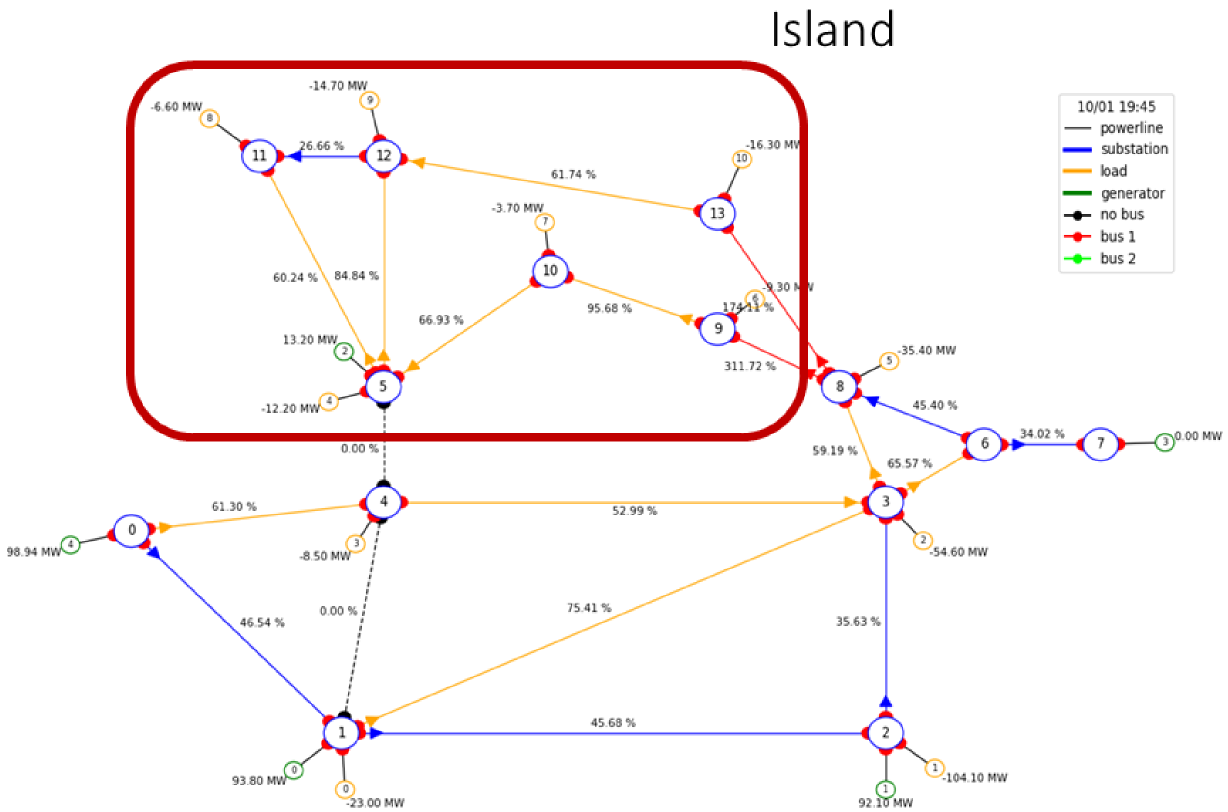


Figure 2.9- Cascading event in the IEEE-14 bus system.

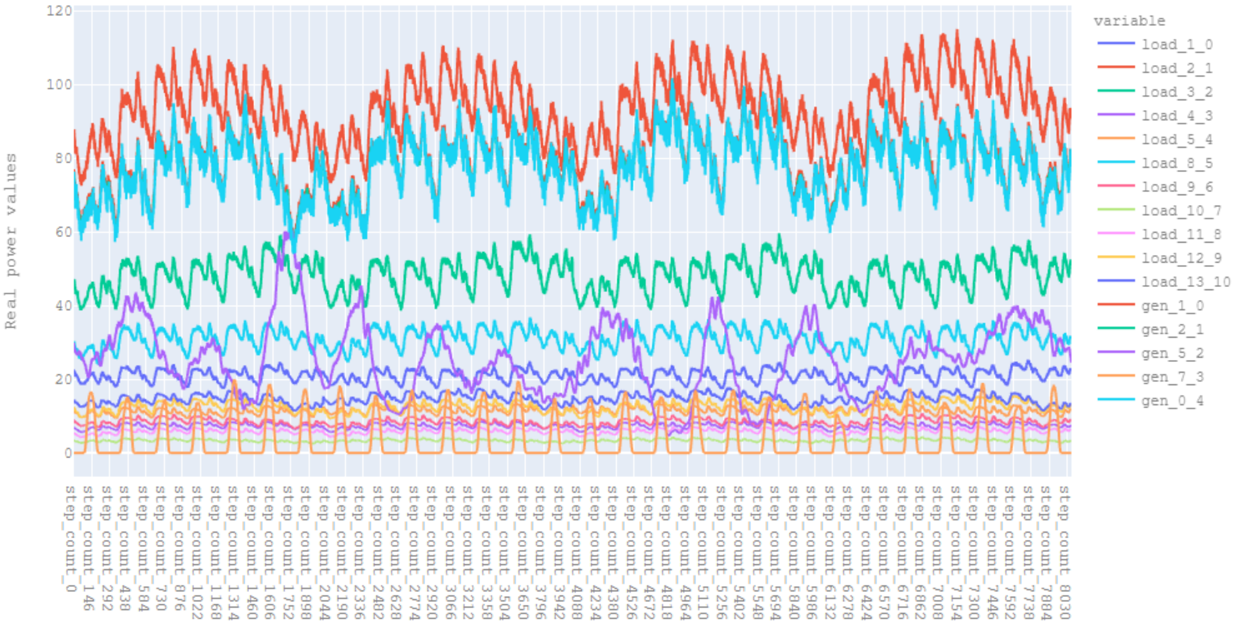


Figure 2.10- The production and consumption profiles of all loads and generators in the IEEE-14 bus system.

Figure 2.11 shows the impact of transmission line disconnection at time steps 1382 and 1387 corresponding to lines 1-4 and 4-5 respectively. In this experiment, it can be observed that the VSI represents the negative impact of losing a transmission line in the power grid by a sudden dip in its magnitude. This indicates that VSI is a good constraint to quantify voltage instability behavior given injections and topology of a power grid for the topology controller model formulation.

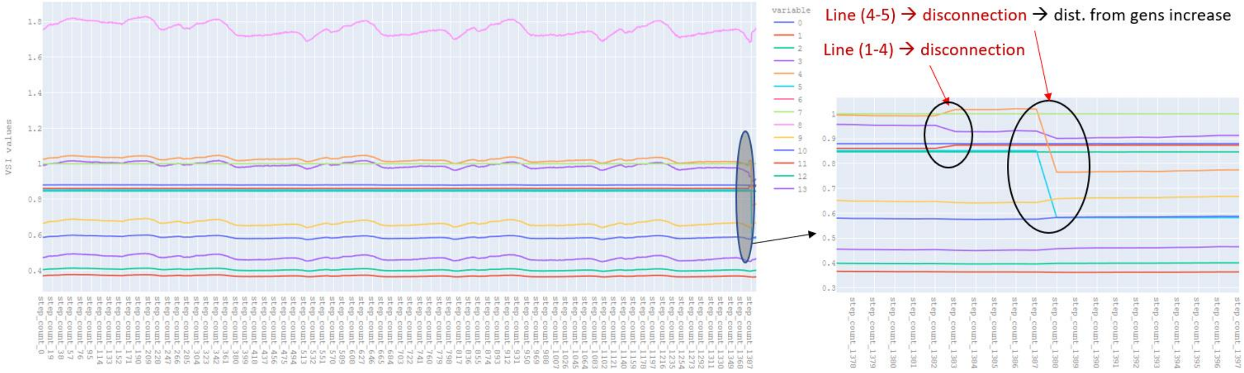


Figure 2.11- Impact of topological actions on the VSI values.

<u>VSI as constraint in the updated topology controller model formulation</u>: The addition of VSI as a constraint enables the power system operators to operate the power grid within the pre-defined safety thresholds. This ensures that the actions performed by the topology controller do not result in an unexpected voltage instability situation. The aim of the topology controller is to maximize the total line loading on the grid over a time horizon $t = \{1,2,\dots,n\}$ by identifying the optimal topology N_t for every time step t with transmission line switching and bus splitting actions.

$f_{N_t}(x_t) = 0$ represents the real and reactive power flow equations corresponding to the timestep t with topology N_t and state vector x_t. The constraint 2.7 represents that AC power flow solution must exist for entire time horizon t. The constraint 2.8 represents that the voltage at bus i i.e., $V_{i,t}$ must lie within the specified thresholds over the entire time horizon t where $\forall i \in B$ and B is the set of all buses in the power grid. The constraint 2.9 represents that the loading on a transmission line p i.e., $I_{p,t}$ must be less than its thermal limits $I_{p,max}$ over the entire time horizon t where $\forall p \in E$ and E is the set of all transmission lines in the power grid. The constraint 2.10 represents that the VSI at every bus in the grid should be greater than the safety margin set by the operate over the entire time horizon.

$$\max_{N_t} = \sum_{t=1}^{t=n} \sum_{\forall p \in E} \left(\frac{I_{p,t}}{I_{p,max}} \right) \tag{0.6}$$

$$Sub.\,to: f_{N_t}\left(x_t\right) = 0 \; \forall t = \{1, 2, \mathrm{L} \;\, , n\}, \tag{0.7}$$

$$0.95 \le V_{i,t} \le 1.05 \; \forall i \in B, \; \forall t = \{1, \mathrm{L} \;\, , n\}, \tag{0.8}$$

$$I_{p,t} \le I_{p,max} \; \forall p \in E, \; \forall t = \{1, \mathrm{L} \;\, , n\}, \tag{0.9}$$

$$VSI_{i,t} \ge \xi, \forall i \in B, \forall t = \{1, 2, ..., n\}. \tag{0.10}$$

2.1.4 Main issues with addition of distance to voltage collapse (VSI) as a constraint

In real-world applications, to monitor (compute) the VSI [26] at the main bus in real-time, PMUs are required on the neighboring buses of the main bus. This is because the PMUs provide the real-time neighboring bus voltage measurements from the field instantaneously. Finally, VSI can be computed immediately using these real-time measurements since it is a function of neighboring bus voltages. However, in the case of the topology controller problem, the optimization model must be solved (either using an optimization or learning algorithm - DRL) in a computer (AC power flows to compute VSI + optimization technique) to obtain an optimal planning strategy over the time horizon. For example, from Table 2.1, for any arbitrary loading condition, there are ≈ 1.3 billion possible one-step look ahead actions to explore for the IEEE-14 bus system. Figure 2.12 shows a topology controller problem over a time horizon of seven timesteps and each time step is a unique state of the power grid based on the user's consumption. To find an optimal action for each unique state (time step) of the power grid there are approximately 1.3 billion actions to explore. For each of these 1.3 billion actions, the optimization model must select an optimal action that ensures that the VSI value after taking a particular action must be within the safety threshold set by the operator. This requires ≈ 1.3 billion AC power flows to be solved in order to compute the VSI which is not realistic, let alone considering the optimal action sequence over the entire time horizon which is a combinatorial problem in nature.

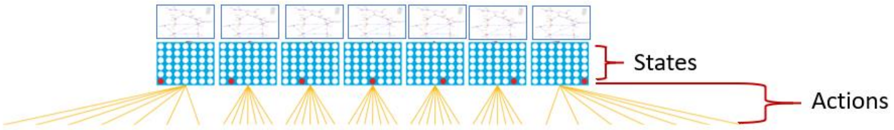


Figure 2.12- Visualization of states and possible actions for each given state.

2.2 Goal of the project

There are two goals for this project. One is to implement the VSI [26] as a feature in the Grid2Op environment [29] and this also serves as a method to generate VSI data on the power grid. The second goal of the project is to overcome the computational infeasibility of requiring solving the power flows in order to compute the VSI during the exploration of a new action when training a deep reinforcement learning (DRL) to solve the topology controller problem.

Table 2.2- Comparison between existing methods and requirements of an ideal method for DRL

Method	***Accuracy in predicted an unsafe operating state***	***Time to compute***	***New unseen topology generalization***
Sensitivity method [23-25]	*Low*	*Fast*	*Yes*
Analytically exact method (VSI) [26]	*High*	*Slow*	*Yes*
Proposed GNN-based method	*??*	*??*	*??*

Table 2.2 shows the comparison of the existing methods, a DRL friendly method to calculate the distance to voltage collapse must be accurate (so no errors in VSI calculation), very fast (DRL can explore and compute VSIs without the need to train for years), and generalizable (accurate VSIs can be computed for any new topology). However, Table 2.2 shows that the sensitivity based methods are fast and generalizable but they are prone to errors as discussed in Section 2.1.1. VSI method is highly accurate and generalizable but it is a very slow method that requires AC power flows to be computed each time. Hence in this project, a graph neural network (GNN) based method is proposed to predict the VSI given the state of the power grid. The goal of this project is to obtain a method that is highly accurate, fast, and generalizable. It is important to note that the generalizability feature is very crucial for a machine learning based method.

3. Graph neural networks to predict VSI and generalize mapping rule

3.1 Introduction to graph neural networks

Graph neural networks (GNNs) are like convolution neural networks (CNNs) except that GNNs can learn non-Euclidean data more efficiently than that of the CNNs. With respect to the power system data, we know that power grid data is graphical data whose mapping rule at each bus relies on the computational graph based on the neighboring buses. Graphical data generally come under the non-Euclidean type and recently GNNs have shown state-of-the-art results in similar problems [30]. In this section, we introduce the GNNs and its computational process to learn. On a high-level, as shown in Figure 3.1, the goal is to estimate VSI for all nodes in the grid given the adjacency/Laplacian matrix and the node features (chronics).

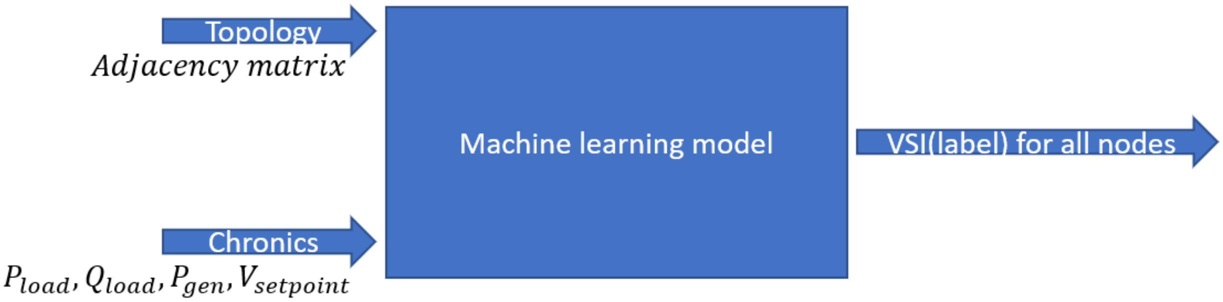


Figure 3.1- High-level model of the problem to estimate VSI at every node in the grid.

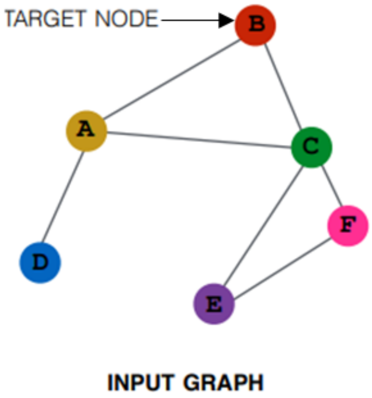


Figure 3.2- An undirected graph network with nodes and edges.

Figure 3.2 shows an undirected graph network with nodes and edges. We will first define few terms associated with a graph network. Given a topology, each node has a neighborhood that contains the neighboring nodes. Each node has a set of node features. Similarly, each edge also has a set of edge features. In GNNs, the basic step is known as graph convolution step. It is also popularly known as message passing step. Equation (2.11) shows the mathematical formulation of a graph convolution step. It mainly has three functions, namely 1) message function, 2) aggregator function and 3) update function. The $\vec{x}_i$ represents the node features of node *i* and $\vec{e}_{i,j}$ represents the edge features of an edge connecting nodes *i and j*.

$$\vec{x}_i' = UPDATE\left(\vec{x}_i, AGGREGATE(\vec{x}_i, \vec{x}_j, \vec{e}_{j,i})\right) \qquad \textbf{(0.11)}$$

Let us consider an example to perform a graph convolution step at node B in Figure 3.2. Figure 3.3 shows that the neighbors of node B are nodes A and C. So, according to the message passing equation from above, first the node features from node A and node C as passed as message-1 and message-2 respectively. The aggregation function applies a mathematical operation on the incoming messages from the neighbors as shown in Equation (0.11) followed by the update function which applies a nonlinear transformation to $\vec{x}_i$ using neuron to get $\vec{x}_i'$ [33]. This process of graph convolution step is applied for every node in the network before proceeding to the next iteration and repeating this process. These iterations are known as epochs and the number of epochs is dependent on the accuracy of the trained GNN model after each epoch/iteration. This process is described in Figure 3.4. Figure 3.5 shows that intuition behind using the GNNs to estimate VSI, it can be seen that the addition of a new node (new topology) would not change the computation graph of the neighborhood drastically and in fact, the new node follows a similar computation graph as it belongs to the same local vicinity in the graph.

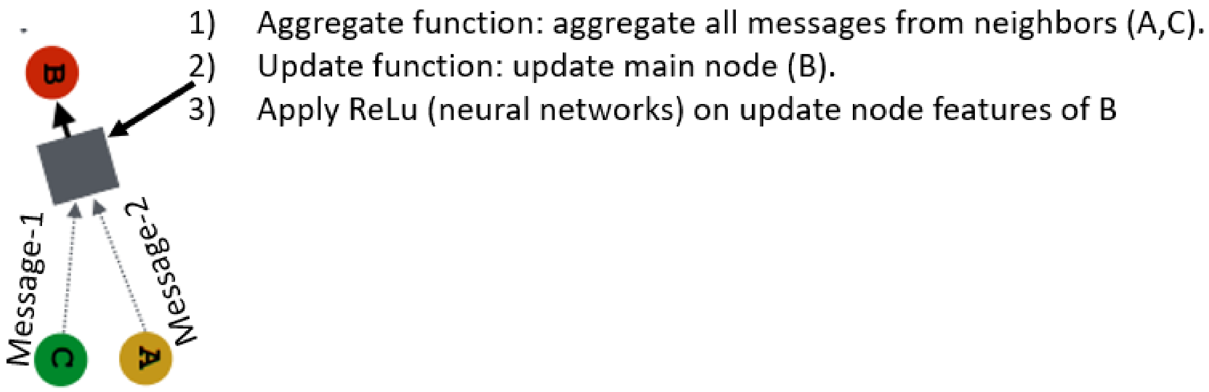

1) Aggregate function: aggregate all messages from neighbors (A,C).
2) Update function: update main node (B).
3) Apply ReLu (neural networks) on update node features of B
B
Message-1
Message-2
C
A

Figure 3.3- Graph convolution or message passing step at node B.

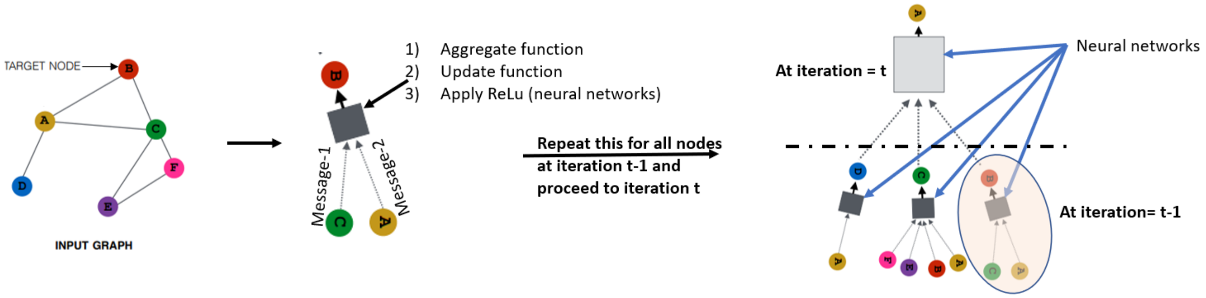


Figure 3.4- Graph convolution methodology employed in graph neural networks.

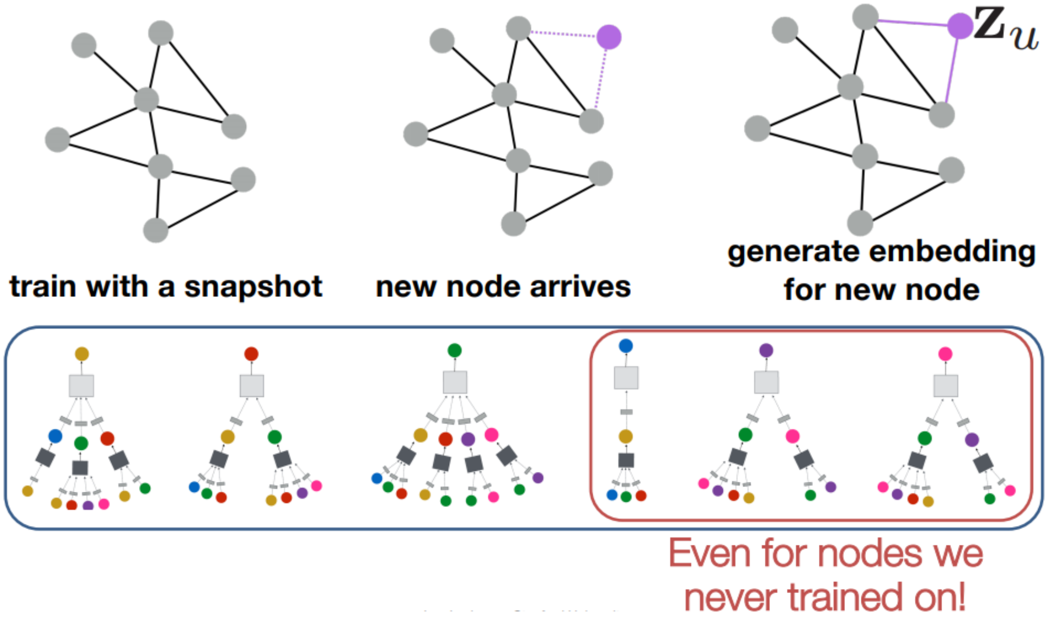
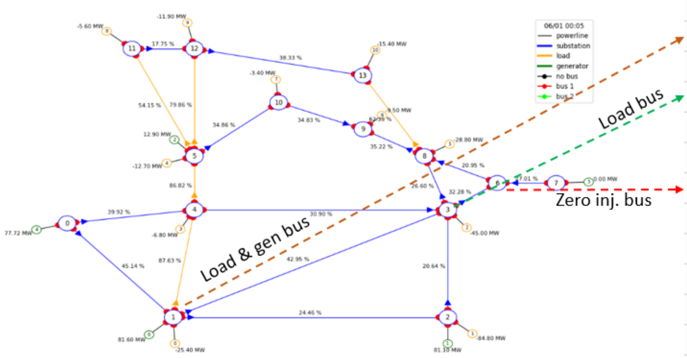

$\mathbf{z}_u$
train with a snapshot
new node arrives
generate embedding for new node
Even for nodes we never trained on!

Figure 3.5- Intuition behind the generalization capability of GNNs for unseen topologies.

3.2 Inputs and outputs of the graph neural networks

Inputs: As discussed in Section 3.1, the GNNs take graphs as input and this means that it uses the Laplacian to understand the connectivity between the nodes in the graph. It also requires the node features and edge features. In this project, the edge features are not chosen to be included in the inputs.

GNN input data representation: node features + adjacency matrix



Node features

Node	P_{load}	Q_{load}	P_{gen}	$V_{setpoint}$
0	0	0	69	1.02
1	-21.7	-15.1	67.2	1.02
2	-74.3	-51.3	66.4	1.02
3	-37.9	-26.7	0	0
4	-5.7	-4	0	0
5	-10.7	-7.5	14.7	1.1
6	0	0	0	0
7	0	0	0	1.1
8	-24.7	-17.3	0	0
9	-8	-5.7	0	0
10	-2.9	-2.1	0	0
11	-4.5	-3.2	0	0
12	-9.5	-6.6	0	0
13	-13.6	-9.6	0	0

GNN output: predict the minimum VSI (distance to voltage collapse) among all node in the grid.

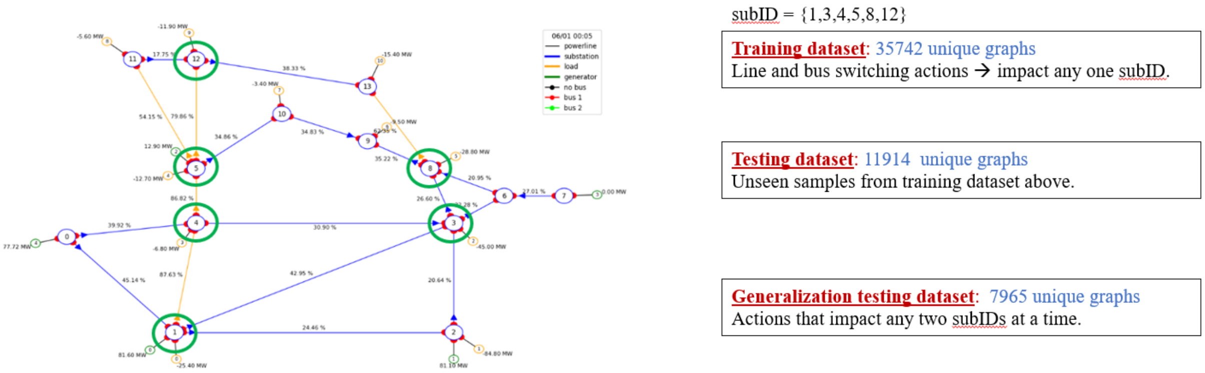
Figure 3.6- Inputs and outputs of the graph neural networks.

Figure 3.6 presents the inputs and outputs to the GNNs at a given snapshot of the IEEE-14 bus system i.e., one data sample. In this project, a realistic problem setting is used to set up the experiment. For example, for a PQ bus, only the real and reactive power information is assumed to be known. Similarly, for the PV bus, only the real power generation and voltage setpoint are assumed to be known. For example, Figure 3.6 shows an IEEE-14 bus system with 14 substations (nodes). Each substation (node) has four node features namely real and reactive powers of any loads, real power generation and voltage setpoint of any generator. Thus, the size of the node feature for an entire graph is NX4 where N is the number of substations (nodes) in the power grid. Additionally, the connectivity between nodes is also provided as input to the GNNs.

Output: The output of the GNNs is to predict the VSI value at each node (substation) in the power grid.

3.3 Datasets used to train and test the graph neural networks

It is also crucial to generate the datasets in a meaningful fashion to train and especially test the GNNs rigorously. The substations with most possible topology configurations are first selected from the IEEE-14 bus system as shown in Figure 3.7 i.e., subID = {1, 3, 4, 5, 8, 12}. The line and bus switching actions that topologically impact only one of the substations in subID set are used to generate the training and testing dataset for various loading conditions. The unique graph instances or the size of the training dataset used to train the GNN model is 35,742. The size of the testing dataset used to evaluate the GNN model is 11,914. However, the trained GNN model must perform well on the new topologies that are very different from the topologies the GNN model is trained upon. Therefore, we generate another set of graphs or data samples but this time, the line and bus switching actions performed on the grid impact two substations from subID set at a time (generalization testing dataset). The goal is that the trained model was able to learn the generalization rule such that it can predict VSI accurately not only the testing dataset but it can predict accurately for new unseen topologies (generalization testing dataset) as shown in Figure 3.7.



subID = {1,3,4,5,8,12}

Training dataset: 35742 unique graphs
Line and bus switching actions → impact any one subID.

Testing dataset: 11914 unique graphs
Unseen samples from training dataset above.

Generalization testing dataset: 7965 unique graphs
Actions that impact any two subIDs at a time.

Figure 3.7- Training, testing and generalization testing datasets for the GNNs.

3.4 Various graph neural network architectures

GCNConv model: There are various architectures for the GNNs depending on the mathematical formulation of the graph convolution or message passing step. Figure 3.8 presents the GNN model namely "GCN model" [34] whose graph convolution step is given by Equation (3.1).

$$\mathbf{x}_i^{'} = \mathbf{\Theta} \sum_{j} \frac{1}{\sqrt{\hat{d}_j \hat{d}_i}} \mathbf{x}_j, \tag{1.1}$$

where $\hat{d}_i = 1 + \sum_{j \in \mathrm{N}\,(i)} e_{j,i}$, $e_{j,i}$ represents the edge weight from source node *i* to target node *j*. x_j represents the neighboring node feature representation and $x_i^{'}$ represents the main node feature representation. In Equation (1.1), the summation and theta corresponds to the aggregate and update function respectively. Sometimes, this GCN model experiences a problem known as over smoothing during which the neighboring bus feature representations dominate the main node feature value and thus it may result is not so good learning process. Hence, in this project another GNN model known as "GraphConv model" is also trained to estimate the VSI. The results for both the models is compared in the later sections.

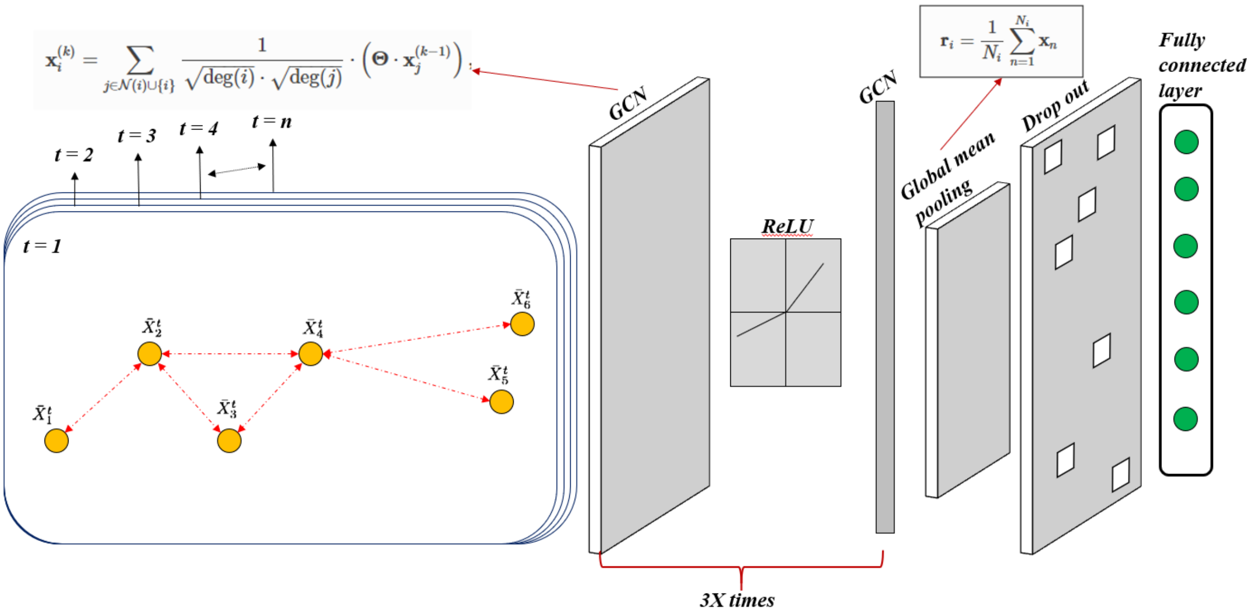

$\mathbf{x}_i^{(k)} = \sum_{j\in\mathcal{N}(i)\cup\{i\}} \frac{1}{\sqrt{\deg(i)}\cdot\sqrt{\deg(j)}} \cdot \left(\Theta \cdot \mathbf{x}_j^{(k-1)}\right)$
$\mathbf{r}_i = \frac{1}{N_i}\sum_{n=1}^{N_i} \mathbf{x}_n$
t = 2
t = 3
t = 4
t = n
t = 1
$\bar{X}_1^t$
$\bar{X}_2^t$
$\bar{X}_3^t$
$\bar{X}_4^t$
$\bar{X}_5^t$
$\bar{X}_6^t$
GCN
ReLU
GCN
Global mean pooling
Drop out
Fully connected layer
3X times

Figure 3.8- GNN model with "GCN" convolution layer and pooling layer.

<u>GraphConv model</u>: Figure 3.9 represents the second GNN model namely "GraphConv model" [31] whose graph convolution step is given by the Equation (3.2).

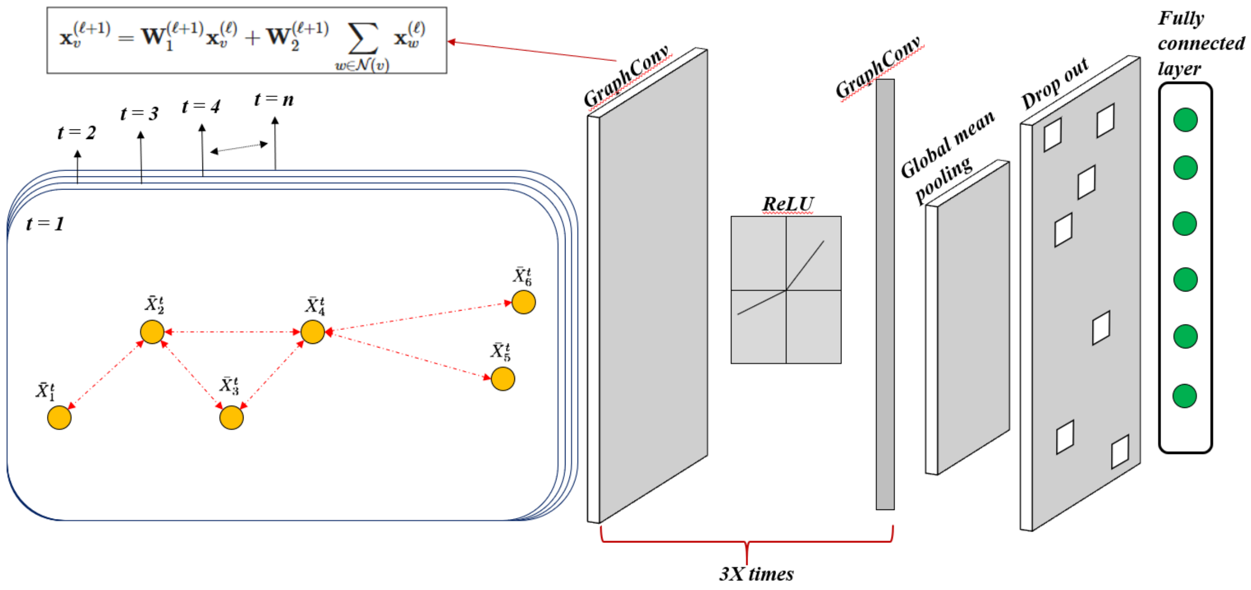


Figure 3.9- GNN model with "GraphConv" convolution layer and pooling layer.

$$\mathbf{x}_i^{'} = \mathbf{\Theta}_1\mathbf{x}_i + \mathbf{\Theta}_2 \sum_{j\in \mathrm{N}\ (i)} e_{j,i}\cdot \mathbf{x}_j \qquad \textbf{(1.2)}$$

where $e_{j,i}$ represents the edge weight from source node i to target node j. x_j represents the neighboring node feature representation from previous iteration, x_i represents the main node feature representation from the previous iteration and $x_i^{'}$ represents the updated main node feature representation in the current iteration. This GraphConv model can overcome the occasional over smoothing problem experienced by the GCN model using a popular concept known as skip-connections [31] where the node representation of neighbors and main node weighted according to some important criteria during the graph convolution step. Θ_1 and Θ_2 represents these weights.

3.5 Metrics to analyze the performance of trained graph neural networks

It is important to not only use the best mathematical representation model for a given problem, but it is also equally important to evaluate the results in a meaningful fashion. In this section, first, various metrics are defined that meaningfully measure the performance of the GNN model for the problem application at hand.

For example, Figure 3.10 presents the predicted critical VSI versus the actual critical VSI of the grid for various scenarios. The critical VSI of a grid/scenario is given by $\min(VSI_i), \forall i \in B$, where B represents set of nodes/substations in the power grid. Ideally, if the data points in Figure 3.10 lie on the line passing through the origin then the model's performance is the best. Let us define two different points of view to analyze the results in Figure 3.10.

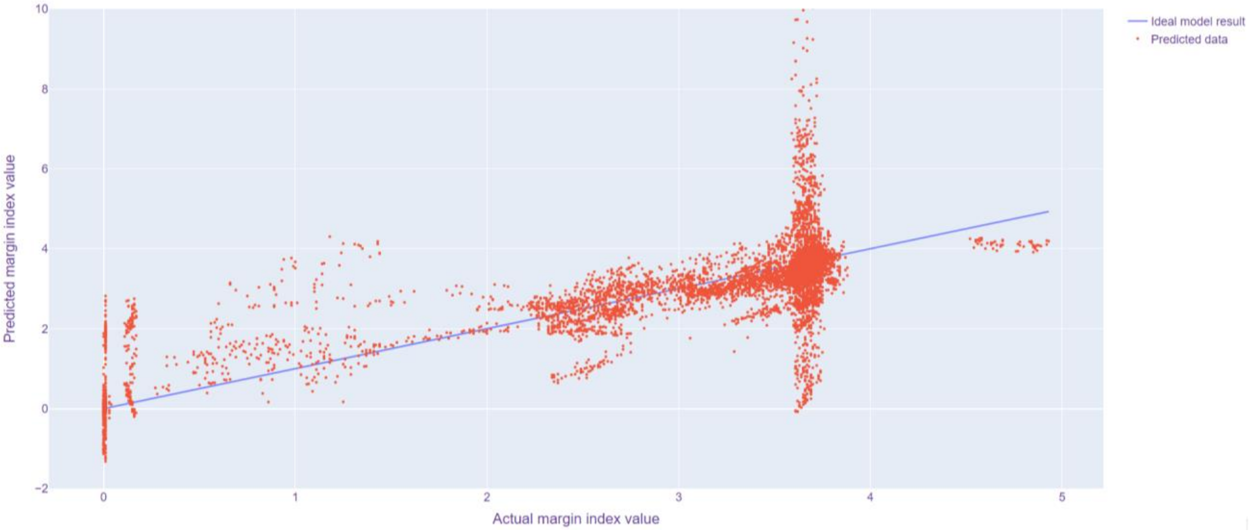


Figure 3.10- Predicted critical VSI versus actual critical VSI of the grid for various scenarios.

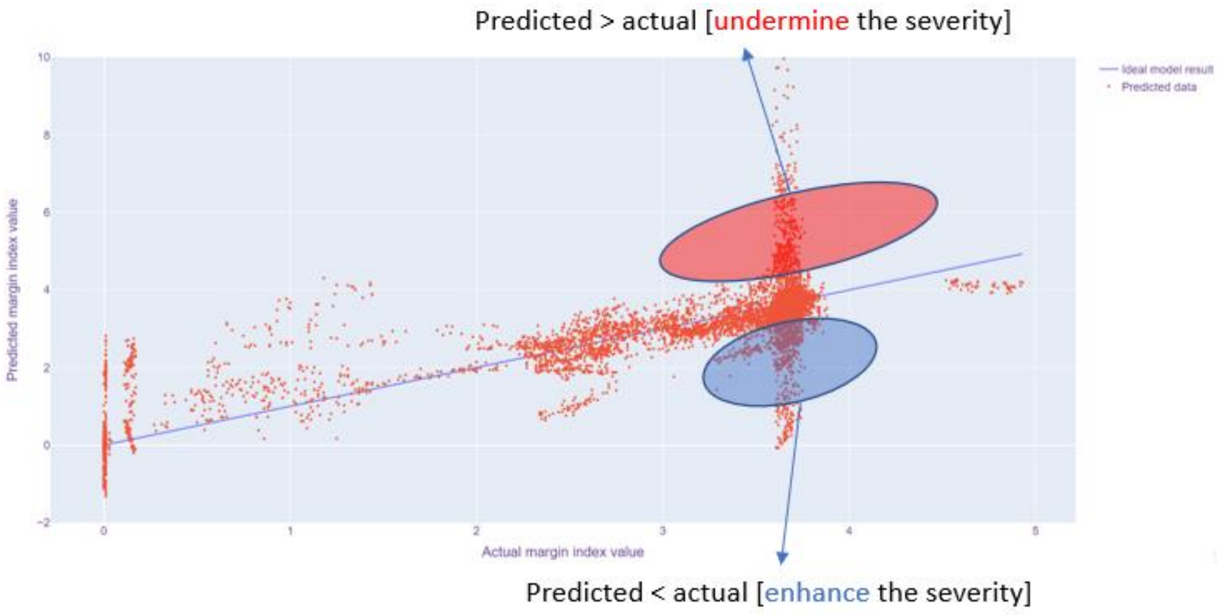
1) **<u>Error rate</u>:** Figure 3.11 visually presents that when the predicted critical VSI of a scenario is greater than the actual critical VSI then the power system operator is misinformed about the criticality of the scenario which can result in taking decisions that may be harmful. Therefore, this undermines the severity of a scenario (bad prediction cases). Similarly, when the predicted VSI of a scenario is smaller than the actual critical VSI then the

prediction model enhances the severity of the scenario which is still acceptable because it is always better to be safe than unsafe. Based on this point of view, the following Equation (3.5), error rate metric is a good indicator to identify the bad prediction cases. N(.) in below equations represent the count of instances that satisfy the condition within N(.).

$$Bad\ predictions\,(BP)\ = N(\,predicted\ critical\,VSI - actual\ critical\,VSI > 0.3) \qquad \textbf{(1.3)}$$

$$Correct\ predictions\,(CP) = N(\,predicted\ critical\,VSI \leq actual\ critical\,VSI\,) \qquad \textbf{(1.4)}$$

$$\therefore Error\ rate = \frac{Bad\ predictions}{Total\ predictions} = \frac{BP}{BP+CP},\ where\ 0 \leq Error\ rate \leq 1. \qquad \textbf{(1.5)}$$


Predicted > actual [undermine the severity]

Predicted < actual [enhance the severity]

Figure 3.11- Cases that are bad predictions versus correct predictions.

2) **<u>Confusion matrix-based metrics</u>:** To compute the confusion matrix for the prediction results of a trained model, the regression model of predicting the VSI must be transformed into a classification model. Interestingly, the topology controller optimization model also only requires the VSI of all buses to be within a specified threshold set by the operator. It is critical that the VSI at all buses i.e., minimum VSI among the VSIs of all buses to be greater than the user-defined threshold for the power grid to operate in a safe region.

Therefore, as shown in Figure 3.12, in this project to analyze the results we assume that the operator has to successfully identify the scenarios where the critical VSI of the grid is lower than 0.25 (scaled to 2.5 in Figure 3.12) and avoid all such scenarios. Thus, Equations (3.6) and (3.7) present the negative and positive classes of the confusion matrix respectively. Various metrics such as false negative rate (FNR), recall, precision, F1-score and MCC are used to analyze the performance of the trained models and present the results in this project.

$$Safe \rightarrow VSI \geq 0.25\,(negative\,class - no\,class) \qquad \textbf{(1.6)}$$

$$Unsafe \rightarrow VSI < 0.25\,(positive\,class - yes\,class) \tag{1.7}$$

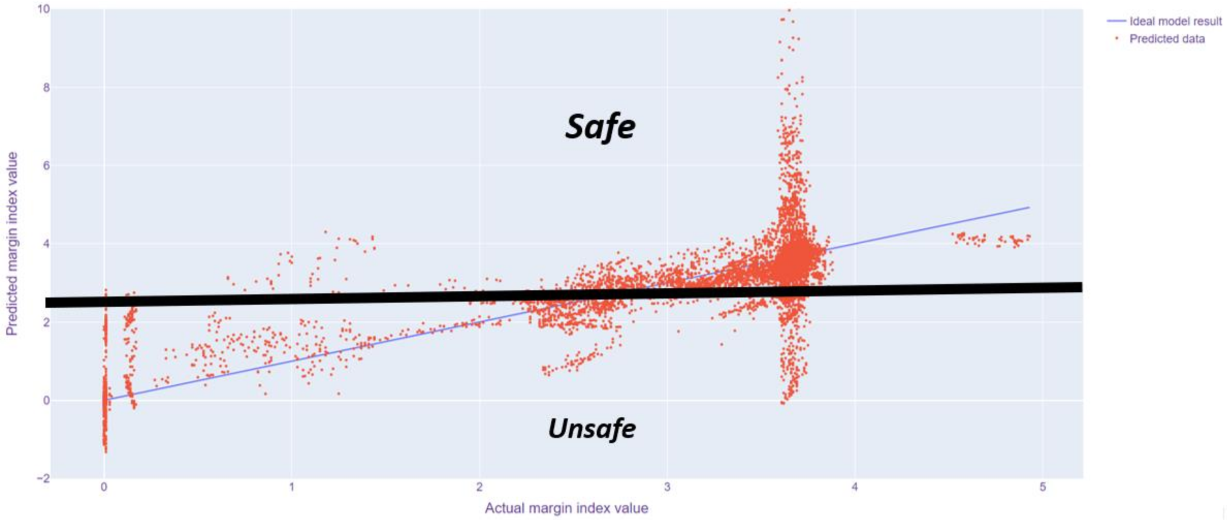


Figure 3.12- Positive (unsafe) and negative (safe) classes for the VSI estimation problem.

3.6 Performance analysis of the trained graph neural networks

The two models i.e., GCN and GraphConv models are compared with each other both with and without pooling layers. The results are as follows:

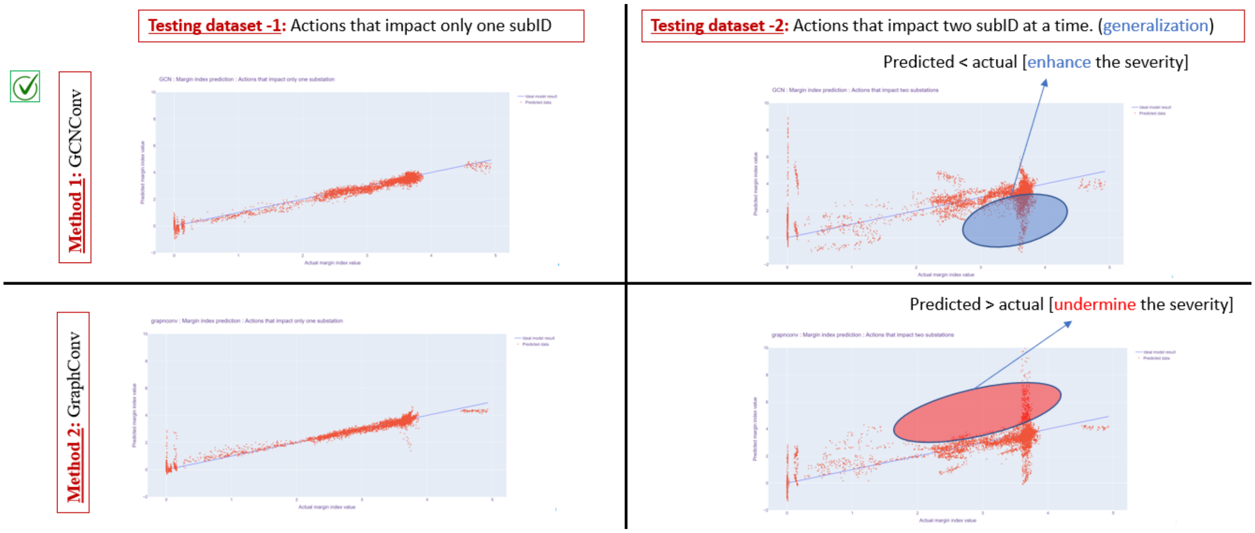


Figure 3.13- GCNConv versus GraphConv with pooling layers.

1) **GCN model versus GraphConv model with pooling layers:** In the case when a pooling layer is present, both models GCN and GraphConv directly predict the critical VSI of a grid/scenario given by $\min(VSI_i), \forall i \in B$, where B represents set of nodes/substations in the power grid. Figure 3.13 shows the actual critical VSI versus predicted VSI plot for

GCNConv and GraphConv models on both the testing dataset (actions that impact only one substation at a time) and the generalization testing dataset (actions that impact two substations at a time). Figure 3.13 here provides a more holistic view of the model performances and a more detailed analysis is carried below in this section using various metrics defined in Section 3.5.

Figure 3.13 shows that both GCNConv and GraphConv predict very well in the case of testing dataset-1 which can be observed by the fact that the majority of the data points are on the line that passes through the origin. However, the performance of GCNConv and GraphConv on the generalization testing dataset (testing dataset 2) is quite different from each other. It can be observed that GraphConv undermines the severity while GCNConv enhances the severity of the scenarios and therefore GCNConv performs better than that of the GraphConv holistically according to Figure 3.13.

2) **<u>GCN model versus GraphConv model without pooling layers</u>:** In the case of without pooling layers, both the models directly VSI at every node in the power grid and the minimum of all the predicted VSI values among all the node is considered as the critical VSI of the grid.

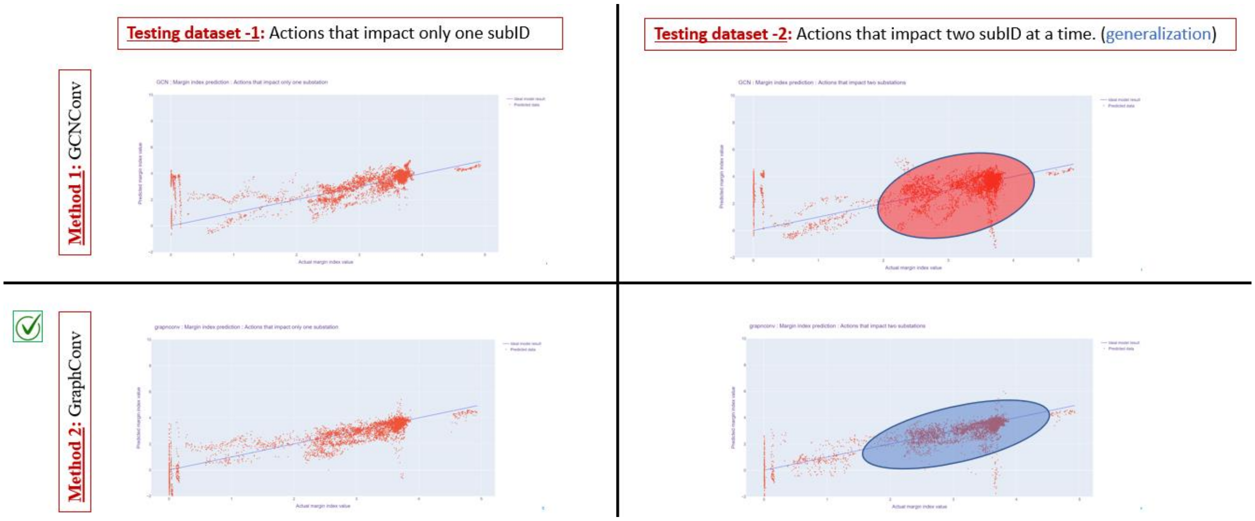


Figure 3.14- GCNConv versus GraphConv without pooling layers.

Figure 3.14 shows the actual critical VSI versus predicted VSI plot for GCNConv and GraphConv models on both the testing dataset-1 (actions that impact only one substation at a time) and generalization testing dataset (actions that impact two substations at a time). Figure 3.14 here provides a more holistic view of the model performances and a more detailed analysis is carried below in this section using various metrics defined in Section 3.5. It can be observed that both the models have a similar performance for testing dataset 1. However, unlike the case study with pooling layers, in this case study, without pooling layers, for generalization testing dataset (testing dataset 2) the GraphConv model performs better than that of the GCNConv by predicting the critical VSI values closer to the line passing through the origin.

Therefore, the detailed discussion and qualitative analysis about the better model among GCNConv with the pooling layer versus GraphConv without the pooling layer are provided below using the metrics defined in Section 3.5.

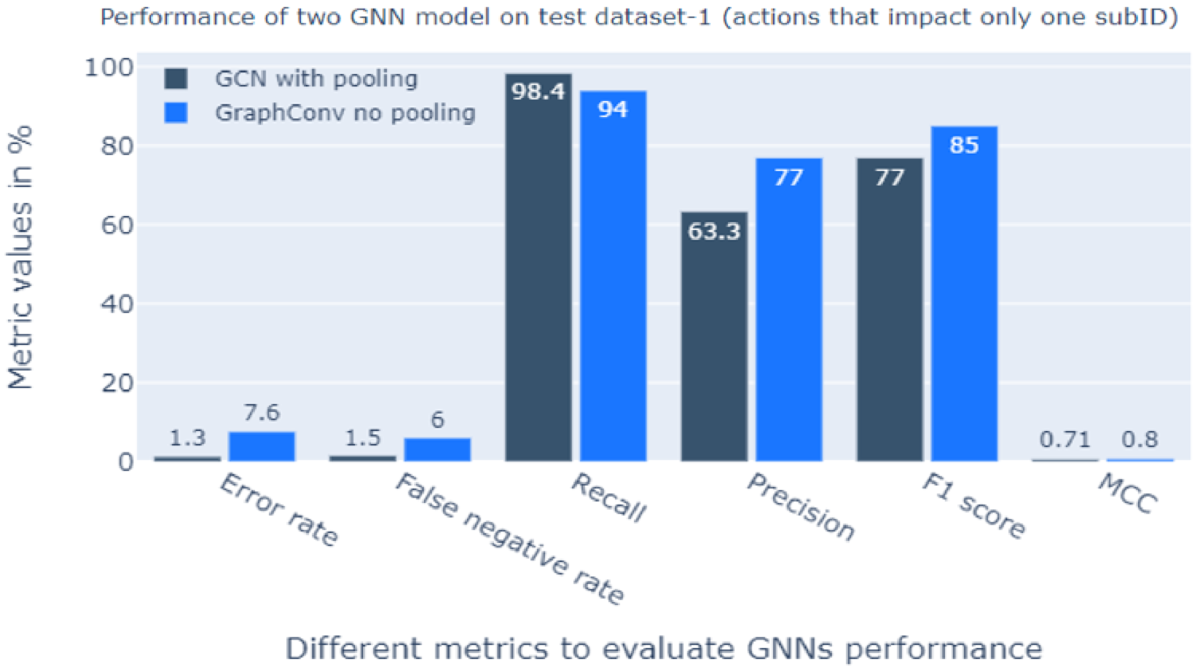


Figure 3.15- For testing dataset-1, performance comparison between GCNConv with pooling versus GraphConv without pooling layers.

Figure 3.15 presents the various metric measures of the selected two models on the testing dataset 1. The smaller the error rate, the fewer the count of bad predictions as presented in Equation (1.5) . False negative rate (FNR) indicates the ratio of cases predicted as safe but they are actually unsafe to the total number of unsafe cases. Hence, a smaller FNR is desirable for a model. Recall is the ratio of cases predicted as unsafe that are actually unsafe to the total number of unsafe cases. Higher the recall the more desirable a model is. Additionally, it is observed that the false positive rate (FPR) which is a ratio of cases predicted as unsafe but are actually safe to the total number of safe cases is ≈ 20% only. Precision is the ratio of correct unsafe predictions to the total unsafe prediction. It is important to note that it is not harmful to predict a safe case to be unsafe when compared to predicting an unsafe scenario to be safe scenario. Hence having a high precision does not reflect the true performance in our model but having a high recall does indicate that the performance of our model is great. Similarly, F1 score is a function of recall and precision and since precision is not a correct metric to analyze the problem, F1 score does not provide any concrete information on the model's performance. Hence, precision and F1 score are not relevant in this problem setting. MCC is correlation coefficient between observed and predicted data. MCC value ranges between -1 and, MCC values closer to one indicate a better model.

From Figure 3.15, it can be observed that GCNConv with pooling has better performance metrics when compared to that of the GraphConv without pooling. GCNConv has a smaller error rate, FNR, and best recall. However, it is also important to analyze the performance of GCNConv on whether it can generalize its learning to the new unseen topologies (testing dataset 2).

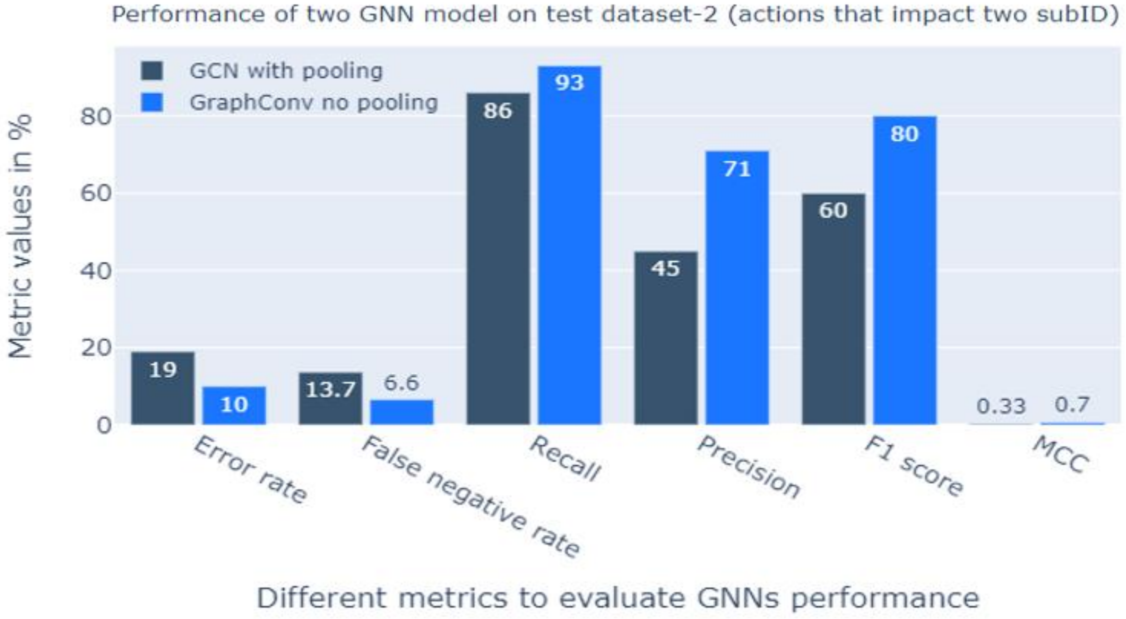


Figure 3.16- For testing dataset-2 (generalization test dataset), performance comparison between GCNConv with pooling versus GraphConv without pooling layers.

Figure 3.16 presents the comparison of GCNConv and GraphConv for the generalization testing dataset. From Figure 3.16, it can be observed that GraphConv has better metric values when compared to that of the GCNConv i.e., smaller error rate, FNR, and better recall values. However, when Figure 3.15 and Figure 3.16 are compared, it can be observed that GraphConv has a 94% recall score for test dataset 1 and 93% for test dataset 2. It also has relatively smaller error rates of 7.6% and 10% on test datasets 1 and 2 respectively. This shows that with a small trade-off in the performance on test dataset 1, the GraphConv model is able to generalize on the new unseen topologies (test dataset 2) very well with a 93% recall score. However, GCNConv performs very on test dataset 1 but fails to generalize for new topologies (test dataset 2). This is also reflected by the smaller MCC score of GCNConv on test dataset 2 in Figure 3.16. Additionally, the false positive rate for GraphConv is only 10.9% on new unseen topologies while that of the GCNConv is 20%. Therefore, GraphConv model without pooling has the best performance i.e., low error rate, low false negative rate, 94% and 93% best recall score on the test and generalization datasets, and, 0.8 and 0.7 MCC score on test and generalization dataset.

Table 3.1 presents an experiment where the analytically exact method [26] is used to compute VSI for 55,637 scenarios by 55,637 AC power flows. This computation took ≈ 48 mins while the same prediction of VSI using the GNN-based method took ≈ 2 seconds only. The proposed GNN-based model is also able to successfully predict the unsafe scenarios 94% which is a great improvement in accuracy when compared to the sensitivity-based method and a great improvement in speed when compared to the analytically exact method [26].

Table 3.1- Comparison between the existing methods and proposed method.

Method	***Accuracy in predicted an unsafe operating state***	***Time to compute***	***New unseen topology generalization***
Sensitivity method [23-25]	*Low*	*Fast*	*Yes*
Analytically exact method (VSI) [26]	*High*	*Slow*	*Yes*
Proposed GNN-based method	*High (94%)*	*Very fast*	*Yes (93%)*

4. Conclusion and future work

With the increase in intermittent energy resources, the operation and control of the power grid become more challenging dynamic. A cost-effective method is to control and operate the grid using the digital topology switching feature of the power grid. However, when the topology of the power grid is modified, its maximum power transfer limits also change. Especially, in presence of intermittent energy sources, the operators must assess whether the power grid would be operating in a safe margin before performing the topology switching action. State-of-the-art methods to perform topology switching actions use deep reinforcement learning since it can provide a strategy for an entire time horizon in real-time.

In this project, to facilitate the deep reinforcement learning-based topology controllers to also learn to operate the grid in a safe margin, a machine learning-based model is proposed which can estimate the distance to voltage collapse given the state of the power grid. This proposed model uses graph neural networks to estimate the distance to voltage collapse with a high degree of accuracy and speed. Additionally, the proposed model also learned the inherent mapping rule of the power system physics, and hence, it can generalize its learning to new unseen topologies.

The future work of this project involves developing a physics-based graph convolution or message-passing step which allows a highly efficient generalized learning of the derived graph neural networks.

References

[1] E. K. Bawan, “Distributed generation impact on power system case study: Losses and voltage profile,” Australasian Universities Power Engineering Conference, pp. 1–6, Sep 2012

[2] P. P. Barker and R. W. D. Mello, “Determining the impact of distributed generation on power systems: part 1 - radial distribution systems,” Power Engineering Society Summer Meeting, pp. 1645– 1656, Jul 2000.

[3] G. K. Ari and Y. Baghzouz, “Impact of high PV penetration on voltage regulation in electrical distribution systems,” International Conference on Clean Electrical Power, pp. 744–748, Jun 2011.

[4] Marot, A., Donnot, B., Romero, C., Donon, B., Lerousseau, M., Veyrin-Forrer, L., & Guyon, I. Learning to run a power network challenge for training topology controllers. *Electric Power Systems Research*, 2020

[5] A. Marot and al. Expert system for topological remedial action discovery in smart grids. Medpower, 2018.

[6] F. M. Fisher E., O’Neill R. Optimal branch switching. IEEE Transactions on Power Systems, 2008.

[7] D. Silver and al. Mastering the game of go with deep neural networks and tree search. Nature 529, 2016.

[8] N. Brown and T. Sandholm. Safe and nested subgame solving for imperfect-information games. NIPS, 2017.

[9] Kelly, A., O'Sullivan, A., de Mars, P., & Marot, A. (2020). Reinforcement Learning for Electricity Network Operation. *arXiv preprint arXiv:2003.07339*.

[10] Taylor, L. Y., & Hsu, S. M. (2000, July). Transmission voltage recovery following a fault event in the Metro Atlanta area. In 2000 Power Engineering Society Summer Meeting (Cat. No. 00CH37134) (Vol. 1, pp. 537-542). IEEE.

[11] North American Electric Reliability Council, Review of Selected Major Bulk Power System Disturbances in North America, 1982.

[12] A Cheimanoff and C. Curroyer, “The Power Failure of December 19 ,1978,” Revue Generale de l’electricite, 1980.

[13] J. A. Casazza, Interim Report on the French Blackout of December 19, 1978, DOE report, 1979.

[14] Simpson-Porco, J. W. & Bullo, F. Distributed Monitoring of Voltage Collapse Sensitivity Indices IEEE Trans. on Smart Grid, 2016

[15] Matavalam, A. R. R. & Ajjarapu, V. Sensitivity Based Thevenin Index With Systematic Inclusion of Reactive Power Limits IEEE Trans. on Power Systems, 2018

[16] Weng, Y.; Rajagopal, R. & Zhang, B. Geometric Understanding of the Stability of Power Flow Solutions IEEE Trans. on Smart Grid, IEEE, 2019

[17] Vu, K.; Begovic, M. M.; Novosel, D. & Saha, M. M. Use of local measurements to estimate voltage-stability margin IEEE Trans. on Power Systems, IEEE, 1999

[18] Gubina, F. & Strmcnik, B. Voltage collapse proximity index determination using voltage phasors approach IEEE Trans. on Power Systems, IEEE, 1995

[19] Milosevic, B. & Begovic, M. Voltage-stability protection and control using a wide-area network of phasor measurements IEEE Trans. on Power Systems, IEEE, 2003

[20] Verbic, G. & Gubina, F. A new concept of voltage-collapse protection based on local phasors IEEE Trans. on Power Delivery, IEEE, 2004

[21] Greene, S., Dobson, I., & Alvarado, F. L. (1997). Sensitivity of the loading margin to voltage collapse with respect to arbitrary parameters. IEEE Transactions on Power Systems, 12(1), 262-272.

[22] Chiang, H. D., Wang, C. S., & Flueck, A. J. (1997). Look-ahead voltage and load margin contingency selection functions for large-scale power systems. IEEE Transactions on Power Systems, 12(1), 173-180.

[23] L. Wang and H. Chiang, "Toward online line switching for increasing load margins to static stability limit,"IEEE Trans. on Power Systems, 2016.

[24] L. Wang and H. Chiang, "Toward online bus-bar splitting for increasing load margins to static stability limit,"IEEE Trans. on Power Systems, 2017

[25] L. Wang and H. Chiang, "Group-based line switching for enhancing contingency-constrained static voltage stability,"IEEE Trans. on Power Systems,2020

[26] Guddanti, K. P., Matavalam, A. R. R., & Weng, Y. (2020). PMU-based Distributed Non-iterative Algorithm for Real-time Voltage Stability Monitoring. IEEE Transactions on Smart Grid, 11(6), 5203-5215.

[27] D. McInnis, "South Florida Blackout", unpublished Florida Power & Light report.

[28] C. W. Taylor, Power System Voltage Stability, McGraw-Hill, 1994.

[29] Marot, A., Guyon, I., Donnot, B., Dulac-Arnold, G., Panciatici, P., Awad, M., ... & Hampel-Arias, Z. (2020). L2RPN: Learning to Run a Power Network in a Sustainable World NeurIPS2020 challenge design.

[30] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Philip, S. Y. (2020). A comprehensive survey on graph neural networks. IEEE Transactions on Neural Networks and Learning Systems.

[31] Morris, C., Ritzert, M., Fey, M., Hamilton, W. L., Lenssen, J. E., Rattan, G., & Grohe, M. (2019, July). Weisfeiler and leman go neural: Higher-order graph neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 33, pp. 4602-4609).

[32] Donnot, B., Donon, B., Guyon, I., Liu, Z., Marot, A., Panciatici, P., & Schoenauer, M. (2019). LEAP nets for power grid perturbations. arXiv preprint arXiv:1908.08314.

[33] Hamilton, W. L. (2020). Graph representation learning. Synthesis Lectures on Artifical Intelligence and Machine Learning, 14(3), 1-159.

[34] Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.