A novel framework for scalable resilience analyses in complex networks

by

Sai Munikoti

B.E., Birsa Institute of Technology Sindri, 2015

M.S., Indian Institute of Technology Gandhinagar, 2018

AN ABSTRACT OF A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Mike Wiegers Department of Electrical and Computer Engineering
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2022

Abstract

Resilience has emerged as a crucial and desirable characteristic of complex systems due to the increasing frequency of cyber intrusions and natural disasters. In systems such as power grids and transportation networks, resilience analysis typically deals with the assessment of system robustness in terms of identifying and safeguarding key system attributes. Robustness evaluation methods can be broadly classified into two types, namely network-based and performance-based. Network-based methodologies involve topological properties of the system, whereas performance-based methods deal with specific performance attributes such as voltage fluctuations in a power distribution network. Existing approaches to evaluate robustness have limitations in terms of (1) inaccurate modeling of the underlying system; (2) high computational complexity; and (3) lack of scalability.

This dissertation addresses these challenges by developing computationally efficient frameworks to identify key entities of the system. First, it develops a probabilistic framework for a performance-based robustness attribute. Specifically, using power grid as a case study, this work focuses on the performance measure of interest, i.e., voltage fluctuations. This work first derives an analytical approximation for voltage change at any node of the network due to a change in power at other nodes of a three-phase unbalanced radial distribution network. Next, the probability distribution of voltage changes at a certain node due to random power changes at multiple locations in the network is derived. Then, these distributions with information theoretic metrics are used to derive a novel voltage influencing score (VIS) that quantifies the voltage influencing capacity of nodes with distributed energy resources (DERs) and active loads. VIS is then employed to identify the dominant voltage influencer nodes. Results demonstrate the high efficacy and low computational complexity of the proposed approach, enabling various future applications (e.g., voltage control).

In the second part, this dissertation emphasizes on network-based robustness measures. Particularly, it focuses on the task of identifying critical nodes in complex systems so that preemptive actions can be taken to improve the system's resilience. Critical nodes represent a set of sub-systems and/or their interconnections whose removal from the graph maximally disconnects the network, and thus severely disrupts the operation of the system. The majority of the critical node identification methods in literature are based on an iterative approach, and thus suffer from high computational complexity and are not scalable to larger networks. Therefore, this work proposes a scalable and generic graph neural network (GNN) based framework for identifying critical nodes in large complex networks. The proposed framework defines a GNN-based model that learns the node criticality score on a small representative subset of nodes and can identify critical nodes in larger networks. Furthermore, the problem of quantifying the uncertainty in GNN predictions is also considered. Essentially, Assumed Density Filtering is used to quantify aleatoric uncertainty and Monte Carlo dropout captures uncertainty in model parameters. Finally, the two sources of uncertainty are aggregated to estimate the total uncertainty in predictions of a GNN. Results in real-world datasets demonstrate that the Bayesian model performs at par with a frequentist model.

Furthermore, the combinatorial case of critical node identification is also addressed in this dissertation, where the node criticality scores would be associated with a set of nodes. This simulates a concurrent scenario where multiple nodes are being disrupted simultaneously. Essentially, this problem falls under the generic category of graph combinatorial problems. This problem is approached through a novel deep reinforcement learning (DRL) based framework. Specifically, GNNs are used for encoding the underlying graph structure and DRL for learning to identify the optimal node sequence. Moreover, the framework is first developed for Influence Maximization (IM), where one is interested in identifying a set of seed nodes, which when activated, will result in the activation of a maximal number of nodes in the graph. This generic framework can be used for various use-cases, including the identification of critical nodes set related to concurrent disruption. The results on real world networks demonstrate the scalability and generalizability of the proposed methodology.

Thirdly, this dissertation presents a comparative study of different performance and network-based robustness metrics in terms of ranking critical nodes of a power distribution network. The efficacy of failure-based metrics in characterizing voltage fluctuations is also investigated. Results show that hybrid failure-based metrics can quantify voltage fluctuations to a reasonable extent. Additionally, several other challenges in existing robustness frameworks are highlighted, including the lack of mechanism to effectively incorporate various performance and network-based resilience factors. Then, a novel modeling framework, namely hetero-functional graph theory (HFGT) is leveraged to model both power distribution networks as well as other dependent infrastructure networks. Results demonstrate that HFGT can address key modeling limitations, and can be used to accurately assess system robustness to failures.

A novel framework for scalable resilience analyses in complex networks

by

Sai Munikoti

B.E., Birsa Institute of Technology Sindri, 2015

M.S., Indian Institute of Technology Gandhinagar, 2018

A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Mike Wiegers Department of Electrical and Computer Engineering
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2022

Approved by:

Major Professor
Dr. Balasubramaniam Natarajan

Copyright

Abstract

Resilience has emerged as a crucial and desirable characteristic of complex systems due to the increasing frequency of cyber intrusions and natural disasters. In systems such as power grids and transportation networks, resilience analysis typically deals with the assessment of system robustness in terms of identifying and safeguarding key system attributes. Robustness evaluation methods can be broadly classified into two types, namely network-based and performance-based. Network-based methodologies involve topological properties of the system, whereas performance-based methods deal with specific performance attributes such as voltage fluctuations in a power distribution network. Existing approaches to evaluate robustness have limitations in terms of (1) inaccurate modeling of the underlying system; (2) high computational complexity; and (3) lack of scalability.

This dissertation addresses these challenges by developing computationally efficient frameworks to identify key entities of the system. First, it develops a probabilistic framework for a performance-based robustness attribute. Specifically, using power grid as a case study, this work focuses on the performance measure of interest, i.e., voltage fluctuations. This work first derives an analytical approximation for voltage change at any node of the network due to a change in power at other nodes of a three-phase unbalanced radial distribution network. Next, the probability distribution of voltage changes at a certain node due to random power changes at multiple locations in the network is derived. Then, these distributions with information theoretic metrics are used to derive a novel voltage influencing score (VIS) that quantifies the voltage influencing capacity of nodes with distributed energy resources (DERs) and active loads. VIS is then employed to identify the dominant voltage influencer nodes. Results demonstrate the high efficacy and low computational complexity of the proposed approach, enabling various future applications (e.g., voltage control).

In the second part, this dissertation emphasizes on network-based robustness measures. Particularly, it focuses on the task of identifying critical nodes in complex systems so that preemptive actions can be taken to improve the system's resilience. Critical nodes represent a set of sub-systems and/or their interconnections whose removal from the graph maximally disconnects the network, and thus severely disrupts the operation of the system. The majority of the critical node identification methods in literature are based on an iterative approach, and thus suffer from high computational complexity and are not scalable to larger networks. Therefore, this work proposes a scalable and generic graph neural network (GNN) based framework for identifying critical nodes in large complex networks. The proposed framework defines a GNN-based model that learns the node criticality score on a small representative subset of nodes and can identify critical nodes in larger networks. Furthermore, the problem of quantifying the uncertainty in GNN predictions is also considered. Essentially, Assumed Density Filtering is used to quantify aleatoric uncertainty and Monte Carlo dropout captures uncertainty in model parameters. Finally, the two sources of uncertainty are aggregated to estimate the total uncertainty in predictions of a GNN. Results in real-world datasets demonstrate that the Bayesian model performs at par with a frequentist model.

Furthermore, the combinatorial case of critical node identification is also addressed in this dissertation, where the node criticality scores would be associated with a set of nodes. This simulates a concurrent scenario where multiple nodes are being disrupted simultaneously. Essentially, this problem falls under the generic category of graph combinatorial problems. This problem is approached through a novel deep reinforcement learning (DRL) based framework. Specifically, GNNs are used for encoding the underlying graph structure and DRL for learning to identify the optimal node sequence. Moreover, the framework is first developed for Influence Maximization (IM), where one is interested in identifying a set of seed nodes, which when activated, will result in the activation of a maximal number of nodes in the graph. This generic framework can be used for various use-cases, including the identification of critical nodes set related to concurrent disruption. The results on real world networks demonstrate the scalability and generalizability of the proposed methodology.

Thirdly, this dissertation presents a comparative study of different performance and network-based robustness metrics in terms of ranking critical nodes of a power distribution network. The efficacy of failure-based metrics in characterizing voltage fluctuations is also investigated. Results show that hybrid failure-based metrics can quantify voltage fluctuations to a reasonable extent. Additionally, several other challenges in existing robustness frameworks are highlighted, including the lack of mechanism to effectively incorporate various performance and network-based resilience factors. Then, a novel modeling framework, namely hetero-functional graph theory (HFGT) is leveraged to model both power distribution networks as well as other dependent infrastructure networks. Results demonstrate that HFGT can address key modeling limitations, and can be used to accurately assess system robustness to failures.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to express my sincere gratitude to my supervisor, Dr. Bala Natarajan, whose guidance, support, and encouragement has been the driving force behind my research. His ability to advise and inspire is truly exceptional. I am very thankful to him for being always available in spite of his busy schedules and for his interest in my research work. I would like to extend my thanks to my committee members, Dr. Anil Pahwa, Dr. Pietro Poggi-Corradini, and Dr. Yingchen Zhang , for their time and comments to improve the quality of my research. I would like to thank Dr. Mahantesh Halappanavar for his continuous support and valuable feedback on my work. I would also like to thank my colleagues Dr. Kumarsinh Jhala, Dr. Laya Das and Amit joshi for working alongside and providing constant motivation.

Finally, I would like to thanks members of the CPSWin group Hazhar Sufi Karimi, Shweta Dahale, Mohammad Abujubbeh, Deepesh Agarwal, Biswajeet Rout, Aabila Tharzeen and Dylan Wheeler for their support, encouragement and guidance.

This thesis is dedicated to my parents Annapurna and Govind Raju for their patience and encouragement during my education. This work is partly supported by the Department of Energy, Office of Energy Efficiency and Renewable Energy (EERE), Solar Energy Technologies Office, under Award DE-EE0008767 and National science foundation under award 1855216.

Preface

This dissertation, "A Novel Framework for Scalable Resilience Analyses in Complex Networks," is submitted for the degree of Doctor of Philosophy in the Mike Weigers Department of Electrical and Computer Engineering at Kansas State University. The research was conducted under the supervision of Professor Balasubramaniam Natarajan. This work is original to the best of my knowledge, except where acknowledgments and references are indicated. Most of the work has been presented in the following published peer-reviewed journals and conferences:

- Munikoti, S., Jhala, K., Lai, K., & Natarajan, B. (2020, August). Analytical voltage sensitivity analysis for unbalanced power distribution system. In 2020 IEEE Power & Energy Society General Meeting (PESGM) (pp. 1-5) IEEE
- Munikoti, S., Natarajan, B., Jhala, K., & Lai, K. (2021). Probabilistic voltage sensitivity analysis to quantify impact of high PV penetration on unbalanced distribution system. IEEE Transactions on Power Systems, 36(4), 3080-3092.
- Munikoti, S., Abujubbeh, M., Jhala, K., & Natarajan, B. (2022). A novel framework for hosting capacity analysis with spatio-temporal probabilistic voltage sensitivity analysis. International Journal of Electrical Power & Energy Systems, 134, 107426.
- Munikoti, S., Abujubbeh, M., Jhala, K., & Natarajan, B. (2022). An Information Theoretic approach to identify Dominant Voltage Influencers for Unbalanced Distribution Systems. IEEE Transactions on Power Systems
- Abujubbeh, M., Munikoti, S., & Natarajan, B. (2021, April). Probabilistic voltage sensitivity based preemptive voltage monitoring in unbalanced distribution networks. In 2020 52nd North American Power Symposium (NAPS) (pp. 1-6). IEEE.

- Das, L., Munikoti, S., Natarajan, B., & Srinivasan, B. (2020). Measuring smart grid resilience: Methods, challenges and opportunities. Renewable and Sustainable Energy Reviews, 130, 109918.

- Munikoti, S., Das, L., & Natarajan, B. (2022). Scalable graph neural network-based framework for identifying critical nodes and links in complex networks. Neurocomputing, 468, 211-221.

- Munikoti, S., Das, L., & Natarajan, B. (2021). Bayesian graph neural network for fast identification of critical nodes in uncertain complex networks. In 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 3245-3251). IEEE.

- Munikoti, S., Agarwal, D., Das, L., & Natarajan, B. (2022). A General Framework for quantifying Aleatoric and Epistemic uncertainty in Graph Neural Networks. arXiv preprint arXiv:2205.09968.

- Munikoti, S., Natarajan, B., & Halappanavar, M. (2022). GraMeR: Graph Meta Reinforcement Learning for Multi-Objective Influence Maximization. arXiv preprint arXiv:2205.14834. (*under review*)

- Munikoti, S., Agarwal, D., Das, L., Halappanavar, M., & Natarajan, B. (2022). Challenges and Opportunities in Deep Reinforcement Learning with Graph Neural Networks: A Comprehensive review of Algorithms and Applications. arXiv preprint arXiv:2206.07922 (*under review*).

- Munikoti, S., Lai, K., & Natarajan, B. (2021). Robustness assessment of hetero-functional graph theory based model of interdependent urban utility networks. Reliability Engineering & System Safety, 212, 107627.

- Munikoti, S., Abujubbeh, M., & Natarajan, B. (2022). Robustness of power distribution system: A comparative study of network and performance based metrics. IEEE Access.

Chapter 1

Introduction

The world population has increased by 113.3% [1] in the past 50 years. More importantly, with massive urbanization, the percentage of global population in urban areas increased from 34% to 52% in the past five decades [2]. The swelling urban population imposes increasingly intense demands over a variety of critical infrastructures such as electricity [3], water [4], heating [5], natural gas [6], road transportation [7], and so on. These networks are inherently integrated and their interdependencies are becoming increasingly tight due to the increased reliance on cyber infrastructure to enable smart and efficient operation.

The concepts of resilience and robustness related to critical infrastructures have attracted tremendous attention in response to the frequent and widespread natural disasters and man-made malicious attacks across the globe. Resilience is a property of the system that describes its response to and recovery from extreme events and has been studied in diverse contexts [8], including psychology [9, 10], ecology [11, 12], sociology [13, 14] and engineering [15–17]. Robustness is a key part of resilience and refers to the system's ability to withstand fluctuations in operating conditions under adverse situations while maintaining proper functionality. Figure 1.1 further clarifies the different components of resilience via the evolution of system functionality under extreme conditions. The slope in the red-colored phase of the trajectory depicts the robustness of the system, i.e., the resistance against degradation. Improving system resilience is becoming increasingly important across several domains. This is primarily

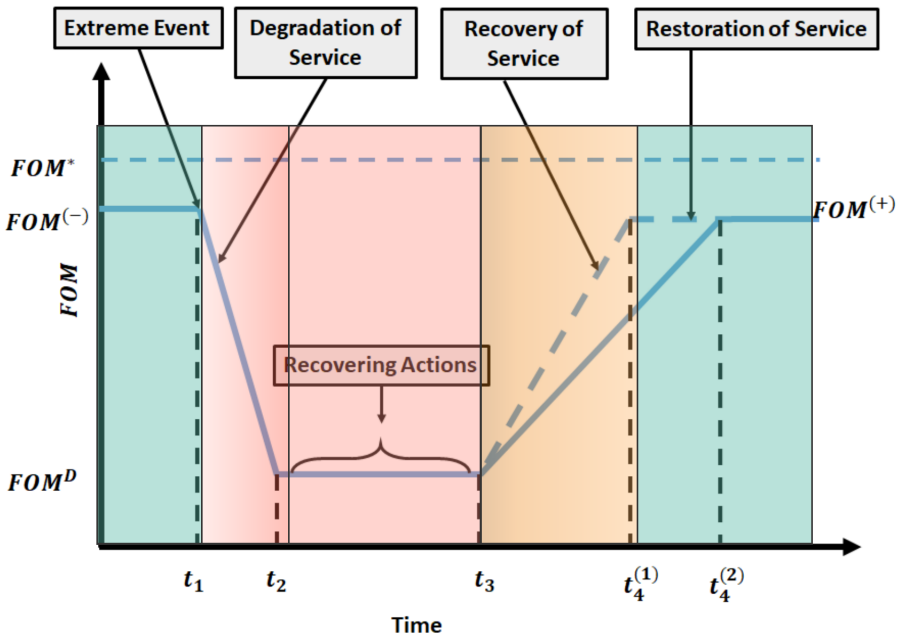


Figure 1.1: Evolution of the performance/service (figure of merit) of a system in the presence of an extreme event and recovering actions

due to two reasons: (1) the increasing frequency of extreme natural and man-made events and (2) growing system complexity and interdependency lead to greater social and economic degradation than ever. Some of the bold examples include power blackouts in the United States (2003) and India (2012), which affected millions and led to financial losses of more than a billion US dollars [18]. A very recent example relates to the rise in the usage of digital infrastructure around the globe because of the ongoing COVID-19 pandemic. This leads to a surge in cyber attacks on several networks [19, 20], making it crucial to identify and reinforce the cyber-security of key data stores/nodes. These and other such events highlight the need for research on resilience and robustness of complex systems against failure.

The first step towards the task of improving resilience is the accurate assessment of system robustness. In this regard, graph theory offers a powerful framework for studying the robustness of such complex systems by representing them as an interconnected network consisting of nodes and edges. Robustness analysis typically deals with identifying key attributes of the system which are majorly responsible for the system's resilience [21]. For

instance, in graph-structured models where nodes represent physical assets like buses in a power grid, these key attributes correspond to critical nodes whose removal/disruption leads to maximum impact on the system functionality (e.g. power flow). Depending upon the factors involved, there are broadly two ways to compute the robustness, i.e., performance-based and network-based. Performance-based methodologies deal with system performance measures such as power flow in the power distribution system, information flow in the internet network, etc. On the other hand, network-based methods solely involve topological characteristics such as node degree and betweenness centrality scores. There are also several works in the literature that take a hybrid approach of fusing these two categories.

Under the performance-based category, voltage fluctuations across the grid is an important factor to consider while assessing power system robustness. This is because, with the increasing penetration of renewable generation and electric vehicles, the distribution grid is witnessing significant new dynamics. Specifically, active consumers with rooftop photovoltaics and distributed generation are expected to alter their generation and usage patterns [22]. This in turn induces frequent power variations. The uncontrolled operations of distributed energy resources (DERs) under this condition lead to voltage fluctuations in the power distribution system (PDS). Although various control algorithms have been developed for regulating voltage changes, they are slow and inadequate to deal with bi-directional power flows and fast dynamics [23]. The voltage fluctuations can have a detrimental impact on the connected devices and customer experience [24, 25]. Therefore, it is important to identify dominant voltage influencer (DVI) nodes that have a relatively high impact on the voltage state of the other nodes in the grid. These DVI nodes allow us to select optimal control nodes for quickly restoring voltage services, thereby improving system robustness against voltage issues.

Several graph-theoretic metrics have been proposed under the network-based robustness characterization category in order to identify critical nodes in complex networks. Essentially, the reliable operation of a complex system depends on proper functioning of its constituent sub-systems (nodes) and their interconnections (links). Typically, there exists a set of critical nodes/links that play a more crucial role in determining the output of the system than other

(non-critical) nodes/links. These nodes/links represent a set of sub-systems and/or their interconnections, whose removal from the graph maximally disconnects the network, and thus severely disrupts the operation of the system. As a result, the identification of such failure-based critical nodes/links in complex networks is very crucial for assuring robustness in complex networks.

The tasks of identifying DVI nodes and failure-based critical nodes have high significance with regard to network and performance-based robustness assessment. The accurate knowledge of these nodes could help us take effective preemptive actions to mitigate the impact of disruptive actions, thereby improving system resilience. Although several methodologies have been continuously developed for accessing the system robustness under these two classes, existing methods suffer from various shortcomings, which are categorized into the following four types:

(1) Computational complexity: Most of the conventional approaches to computing robustness are computationally complex, which hinders their application to large-scale systems. For instance, the identification of DVI nodes in a power grid, involves scenario-based analyses that relies on a large number of Monte-Carlo simulations to account for stochasticity in power changes. Each simulation requires running a load flow, which eventually sums up to a large number of time-consuming processes.

(2) Scalability: The existing frameworks are not scalable to larger networks owing to their large dependency on the global network properties that demand the traversal of the entire network. For example, in the case of identifying failure-based critical nodes with betweenness scores, the entire network needs to be scanned to find all possible node pairs whose paths cross the target node. Such traversal doesn't scale well with the increasing size of network.

(3) Generalizability: Current approaches do not systematically incorporate topological and situational information. Hence, the results are valid for a specific network or scenario. As a result, the computationally expensive processes need to be repeated for every new test case. For instance, in the case of voltage sensitivity analysis in a power distribution network, the Jacobian matrix obtained from the Newton-Raphson method is valid only for a specific state of the system and must be recomputed for any changes in the network state. Thus,

this kind of approach is not generalizable across network types and scenarios.

(4) Modeling inefficacy: Conventional graph-based models of complex infrastructure networks can reveal the topological information and physical relationships among system components, but they do not provide in-depth knowledge about system functionalities and mutual dependencies among different types of resources. This leads to an inaccurate assessment of system robustness. For example, in the case of power distribution networks, existing graph-based test networks cannot explicitly account for the addition of Distributed energy resources (DERs). Although, these new entities play an important role in deciding system robustness, they are not systematically handled by the models.

Based on these perspectives, we seek to address a few fundamental research questions in this dissertation. These questions and the contributions of this dissertation that aim to address them are discussed in the following sections.

1.1 Research Questions

Related to challenges of computational complexity and scalability we ask the following questions:

Question 1: *Performance based robustness analysis: Can analytical and probabilistic approximations of performance measures serve as viable alternatives to scenario based approaches?*

Question 2: *Network based robustness analysis: Is it possible to efficiently identify nodes that are critical for network operation in an efficient and scalable manner ?*

Related to the challenge of generalizability, we focus on the following questions:

Question 3: *For network-based robustness analysis:*

a: *How can we reliably identify critical nodes even in the presence of aleatoric and epistemic uncertainties in the network models ?*

b: *Can we identify a set of nodes critical for a wide variety of networks without explicit training for each ?*

Question 4: *Do performance-based and network-based robustness methods and their hybrid*

variants reveal similar insights on the criticality of nodes ?

Related to the challenge of modeling ineffectiveness, we ask the following question:

Question 5: *Can we create a generic modeling framework that will provide flexibility in integrating network and performance-based analysis for large scale interdependent complex networks?*

1.2 Contributions

To address research question 1, Chapter 3 of this dissertation proposes an analytical approximation of voltage sensitivity analysis and probabilistically models the node voltage changes under stochastic power change scenarios. Then, the probabilistic model of voltage change is used in Chapter 4 to derive a novel performance-based voltage influencing score (VIS). VIS can determine DVI nodes in a very computationally efficient manner. The major contributions of these chapters are listed below.

- An analytical approximation of voltage change due to power change at multiple nodes in an unbalanced distribution network is derived. (Chapter 3)
- To systematically incorporate the stochasticity of power variations, the theoretical probability distribution of voltage change due to random power changes at multiple nodes is derived.
- This work introduces a voltage influencing score (VIS) that quantifies the voltage influencing capacity of nodes with DERs/active loads, and a computationally efficient method to compute it. (Chapter 4)
- The proposed methods are validated against conventional approaches in the standard IEEE 37-node and IEEE 123-node test systems.

More details on voltage influencing metric for effective assessment of grid robustness can be found in Chapters 3 and 4, and in the following published articles:

- Munikoti, S., Jhala, K., Lai, K., & Natarajan, B. (2020, August). Analytical voltage sensitivity analysis for unbalanced power distribution system. In 2020 IEEE Power & Energy Society General Meeting (PESGM) (pp. 1-5) IEEE [26].

- Munikoti, S., Natarajan, B., Jhala, K., & Lai, K. (2021). Probabilistic voltage sensitivity analysis to quantify impact of high PV penetration on unbalanced distribution system. IEEE Transactions on Power Systems, 36(4), 3080-3092 [27].

- Munikoti, S., Abujubbeh, M., Jhala, K., & Natarajan, B. (2022). A novel framework for hosting capacity analysis with spatio-temporal probabilistic voltage sensitivity analysis. International Journal of Electrical Power & Energy Systems, 134, 107426 [28].

- Munikoti, S., Abujubbeh, M., Jhala, K., & Natarajan, B. (2022). An Information Theoretic approach to identify Dominant Voltage Influencers for Unbalanced Distribution Systems. IEEE Transactions on Power Systems [29].

- Abujubbeh, M., Munikoti, S., & Natarajan, B. (2021, April). Probabilistic voltage sensitivity based preemptive voltage monitoring in unbalanced distribution networks. In 2020 52nd North American Power Symposium (NAPS) (pp. 1-6). IEEE. [30]

- Das, L., Munikoti, S., Natarajan, B., & Srinivasan, B. (2020). Measuring smart grid resilience: Methods, challenges and opportunities. Renewable and Sustainable Energy Reviews, 130, 109918 [31].

A novel graph machine learning-based methodology is proposed to address research questions 2 and 3a. Chapter 5 of this dissertation provides a graph neural network (GNN)-driven predictive model to identify critical nodes of the system which are responsible for its robustness against failures. Chapter 6 further increases the usability and generalizability of these predictions by providing a confidence interval around them. The key contributions of these works are as follows:

- The problem of identifying critical nodes/links in a graph is formulated as an inductive machine learning problem for the first time. This formulation allows exploiting the

local neighbourhood information of a node/link and does not require knowledge of the graph's complete topology.

- The proposed framework is used to identify the critical nodes/links in real-life interconnected systems of social networks, biological networks and power grid network.
- The inductive nature of the formulation allows one to train the model on a portion of the graph or a synthetic graph of similar characteristics and apply the trained model to the rest of the graph.
- Quantify total uncertainty due to aleatoric and epistemic sources in a GNN model.
- Efficiently propagate aleaotric uncertainty through layers of a GNN model using Assumed Density Filtering (ADF).
- Generic and computationally efficient uncertainty framework that can be applied at the time of fresh training, as well as to pre-trained networks without the need of a retraining process.

More details related to the GNN-based predictive model can be found in Chapter 5 and Chapter 6, and in the following published articles:

- Munikoti, S., Das, L., & Natarajan, B. (2022). Scalable graph neural network-based framework for identifying critical nodes and links in complex networks. Neurocomputing, 468, 211-221. [32].
- Munikoti, S., Das, L., & Natarajan, B. (2021). Bayesian graph neural network for fast identification of critical nodes in uncertain complex networks. In 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 3245-3251). IEEE [33].
- Munikoti, S., Agarwal, D., Das, L., & Natarajan, B. (2022). A General Framework for quantifying Aleatoric and Epistemic uncertainty in Graph Neural Networks. arXiv preprint arXiv:2205.09968.

To address research question 3b, Chapter 7 of this dissertation fuses Graph Neural Network (GNN) with Deep Reinforcement Learning (DRL) algorithm to develop an efficient framework for graph combinatorial problems in general. Here, critical nodes based on their capability to spread information are investigated. Major contributions of this work are listed below:

- A novel GNN based approach is undertaken first to prune the search space while inspecting for the most influential nodes. This improves the computational efficiency of the DRL module.
- The DRL algorithm of our framework, i.e., double deep Q learning learns the topological patterns which are recurring at every step of the Graph optimization problem. Therefore, once the model is trained, it can predict the appropriate sequence for a new graph in no time.
- Meta learning is achieved by feeding the input graph information along with the state vector while estimating the Q value (long term benefit) of feasible actions. This induces generalization in our framework by allowing predictions across graphs of different families.

More details related to the DRL based node search process can be found in Chapter 7 and in the following articles:

- Munikoti, S., Natarajan, B., & Halappanavar, M. (2022). GraMeR: Graph Meta Reinforcement Learning for Multi-Objective Influence Maximization. arXiv preprint arXiv:2205.14834. (*under review*) [34].
- Munikoti, S., Agarwal, D., Das, L., Halappanavar, M., & Natarajan, B. (2022). Challenges and Opportunities in Deep Reinforcement Learning with Graph Neural Networks: A Comprehensive review of Algorithms and Applications. arXiv preprint arXiv:2206.07922 (*under review*) [35].

A comparative study of performance and network-based robustness metrics are carried out in Chapter 8 to address research question 4. This chapter highlights various limitations of

existing frameworks. Finally, a hetero-functional graph theory (HFGT) is used for modeling interdependent networks in order to address research question 5. The key contributions of these works are as follows:

- Investigate the coherency in rankings of critical nodes using different classes of network failure-based robustness metrics.
- Analyze the expressiveness of failure-based robustness metrics in characterizing the impact of voltage fluctuations.
- Identify key challenges with existing approaches and provide potential research directions related to the design of novel robustness metrics.
- HFGT based model of power distribution network provide more rigorous and detailed framework for accurate assessment of robustness.
- Unlike the original HFGT model [36] that uses binary terms to quantify dependencies, this work proposes edge weights with real numbers in the Weighted HFGT (WHFGT) framework to quantify dependencies. This improvement offers a more generic framework, which enables the simulations of partial attacks (partial loss of functionalities due to attacks).
- With the proposed WHFGT framework, this chapter conducts an in-depth analysis of an interdependent system that reveals system robustness against disruptions from multiple dimensions.

More details related to the comparative study and HFGT based models can be found in Chapter 8 and Chapter 9, and in the following articles:

- Munikoti, S., Lai, K., & Natarajan, B. (2021). Robustness assessment of hetero-functional graph theory based model of interdependent urban utility networks. Reliability Engineering & System Safety, 212, 107627. [34].

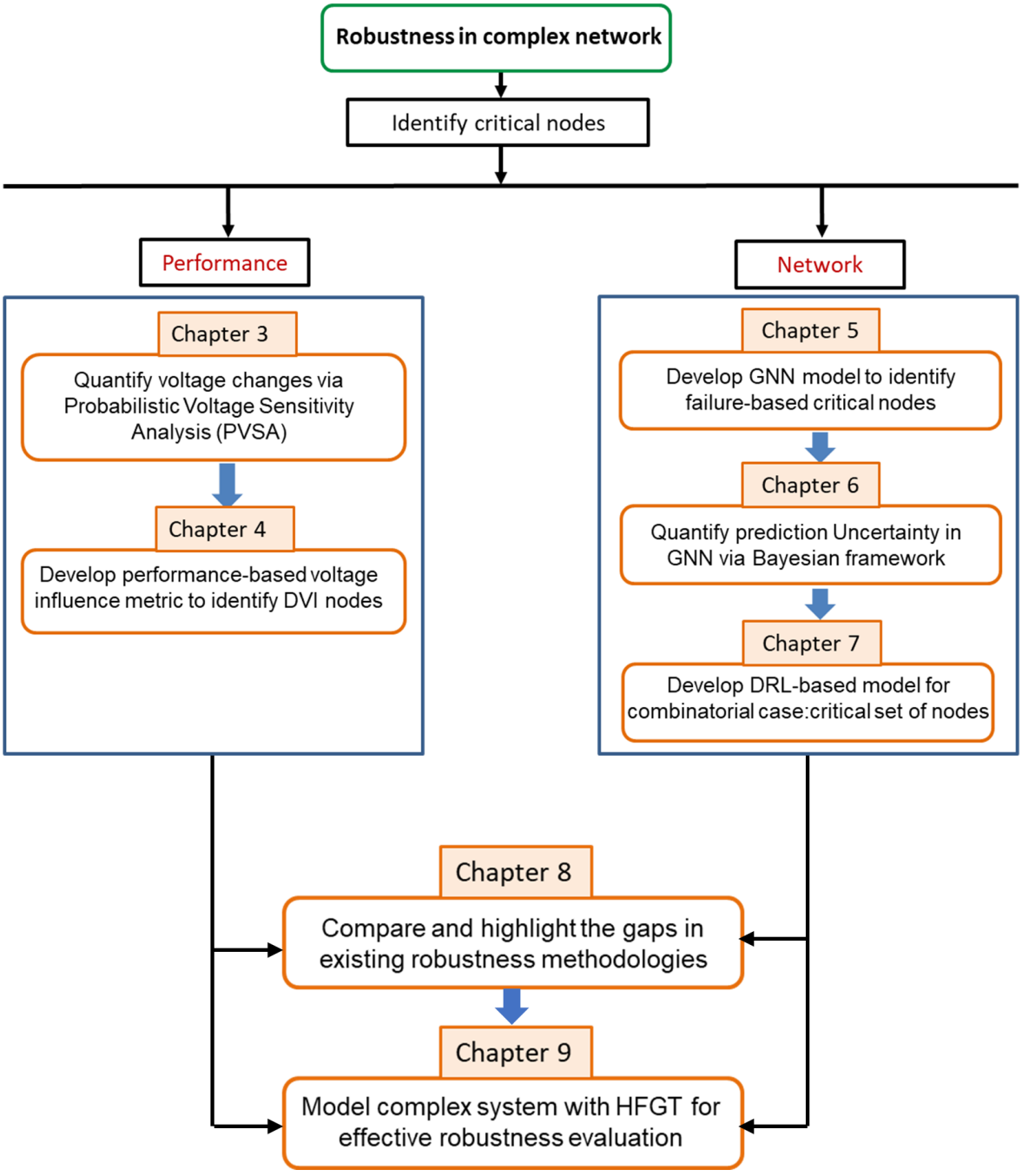


Figure 1.2: Outline of the dissertation

- Munikoti, S., Abujubbeh, M., & Natarajan, B. (2022). Robustness of power distribution system: A comparative study of network and performance based metrics. IEEE Access.

1.3 Organization of the Dissertation

Figure 1.1 outlines the structure of this dissertation. Chapter 2 provides a literature review of related work. Chapter 3, develops an analytical framework for voltage sensitivity analysis which does not rely on simulations for quantifying voltage changes in a power distribution grid. Chapter 4 leverages the foundational work of Chapter 3 to derive a novel voltage influencing metric that can efficiently identify DVI nodes. Chapter 5 proposes a graph machine learning-based novel predictive model to compute robustness scores and consequently identify failure-based critical nodes in complex networks. Chapter 6 further enhances the usability of this predictive model by providing a confidence interval around GNN prediction using a Bayesian framework. Chapter 7 provides a deep reinforcement learning-based framework for solving graph combinatorial problems, including a concurrent attack scenario, where robustness scores would be associated with a group of nodes rather than a single node. Chapter 8 outlines the challenges of existing methodologies for analyzing robustness. The limitation due to modeling inefficacy is overcome through hetero functional graph theory based models in Chapter 9. Finally, Chapter 10 presents key conclusions of this dissertation and suggests avenues for future research.

Chapter 2

Literature review

This chapter highlights the gaps in the existing works, and provides a literature review on the topics covered in this dissertation.

2.1 Voltage sensitivity analysis

VSA quantifies the voltage variation at a given node due to power changes at other locations of the network, can be used as an effective tool to quantify the impacts of PV variations and MTD related intentional perturbations on the voltage stability across the network. Methodologies for VSA can broadly be divided into two categories, namely numerical and analytical. Numerical VSA methods, such as the Newton-Raphson load flow and perturb-and-observe based methods, rely on iterative algorithms to yield an approximate solution. However, these methods suffer from high computational cost and lack of insights on the system states. Prior research efforts have examined the performance of numerical sensitivity analysis methods as it relates to regulating voltage in a power system with distributed generators (DGs)[24, 37, 38], and its drawback in terms of computational efficiency is repeatedly unveiled in these literature. For instance, authors in [37], present a reactive power control method based on voltage sensitivity analysis for mitigating voltage variations in PV integrated distribution systems. Specifically, a new set point for reactive power is computed with varying active power injec-

tion/consumption at other nodes, using the Newton-Raphson load flow. In [24], the authors propose a method for analyzing voltage variations due to PV generation fluctuations in an unbalanced distribution grid, considering a variety of factors. However, its dependency on the inefficient simulation method, limits its applications in large scale distribution networks. In [38], a model predictive control method is proposed to coordinate the active/reactive powers of DGs and on-load tap changing transformers set-points for voltage regulation. However, the high computational burden, associated with the sensitivity matrix updates, limits its application when applied online. In [38], a model predictive controller is used along with the sensitivity matrix to regulate voltages. The authors in [39], propose a centralized coordinated voltage control algorithm for distribution systems with (DGs). Here, the Newton-Raphson method is used to examine DG's effect on the voltage stability of a certain node due to reactive power injection at different nodes across the network. In [40], the authors propose a new reactive power management method for minimizing voltage variation in both steady state and transient conditions due to DER integration. Here, the reactive power of each DER is controlled by exploiting the numerical relationship between variations of voltage and reactive power, based on the traditional VSA method. Authors in [41], develops an optimization model for electric vehicle management based on VSA approaches. Still, the requirements of iterative executions of power flow calculations and optimization models hinder its application in real-world scenarios. Further, an active distribution network management approach is proposed in [42, 43] for maximization of PV hosting capacity. The approach involves adjusting switching capacitors and voltage regulator taps. In this case, thousands of scenarios are incorporated to address the uncertainties that reveal the huge computational burden of VSA in the presence of renewable energy resources. To summarize, most of these numerical approaches involve computationally expensive load flow algorithms or some kind of trade-off that negatively impacts performance, thereby limiting their applicability in large scale distribution systems with uncertainties [44].

To overcome the drawbacks of numerical methods, there are some limited analytical approaches for VSA that have been proposed. In [45], a new sensitivity matrix is derived analytically, relating voltage magnitude with reactive power change. Then, the sensitivity

product is maximized to obtain the optimal generator that has the greatest influence on the voltage of the critical node. Similarly, in [46], an algorithm based on the sensitivity analysis has been designed which optimally manages active and reactive powers of DGs in order to keep the system voltages inside the limits. Here, instead of repeating a load flow calculation to solve the optimization problem, a sensitivity matrix is used to conduct load flow computation in a non-iterative manner, reducing the computational burden significantly. However, the algorithms proposed both in [45] and [46] are not properly validated with standard test systems. Authors in [47], have taken a probabilistic approach where smart meter data is used along with sensitivity analysis to define boundary values of various operation indices. Here, the real and reactive power consumption of houses are assumed to be independent which is not the case in reality and the proposed approach doesn't account for unbalanced load conditions. In [48], authors have computed voltage sensitivities by formulating an over-determined system of linear equations constructed solely using measurements of nodal power injections and voltage magnitudes. Similarly, [49] uses smart meter data with a linear regression model for predicting the voltage change but both [48, 49] rely on the availability of data and monitoring infrastructure. Authors in [50], obtain load dependent voltage sensitivity factors and develop linearized load flow model based on historical smart meter data comprising of load and voltage profiles, without leveraging any grid topology information. This work relies heavily on the availability of smart meter data at the customer level and data needs to be recollected whenever the network gets reconfigured. In a nutshell, existing analytical approaches are not generalized enough for analysis of large scale unbalanced distribution systems with stochastic behavior. Therefore, in our prior work [51], an analytical bound for voltage sensitivity is derived for single phase balanced distribution network. Building off our preliminary work, in this paper, we propose an analytical VSA for a general case of three phase unbalanced distribution system where stochastic power fluctuations can simultaneously occur at multiple nodes of the network. This extension presents many challenges as power change in any one phase impacts the voltage in all the phases.

2.2 Critical node identification

Several methods have been proposed to compute node criticality scores so that critical nodes can be identified [52–54]. However, such approaches typically measure the score of a node/link and repeat the process for all nodes/links in a graph. As a result of this iterative approach, such techniques exhibit computational complexity that increases drastically with the size of the graph, i.e., the number of nodes/links in the graph. For example, the complexity for identifying the optimal link whose removal maximally reduces robustness is of order $O(N^5)$ for a graph with N nodes [53]. As a result, various efforts are directed towards approximating such algorithms and developing a computationally efficient solution [53, 55, 56]. These approaches offer a trade-off between scalable computation and accurate identification of links. However, even with the proposed trade-offs, the lowest achieved complexity for the case of link identification is of order $O(N^2)$. Similarly, the authors in [54] deploy effective graph resistance as a metric to relate the topology of a power grid to its robustness against cascading failures. Specifically, the authors propose various strategies to identify node pairs where the addition of links will optimizes graph robustness. The minimum achieved complexity is of the order $O(N^2 - N + 2L_c)$, where $L_C = \frac{N!}{2!(N-2)!-L}$ with L links and N nodes. However, with an increase in graph size, the accuracy of such approximations decreases, accompanied by a considerable increase in execution time. To overcome computational complexity, the authors in [57], propose a method based on a genetic algorithm to enhance network robustness. Particularly, the authors focus on identifying links whose removal would severely decrease the effective graph resistance of the graph. However, the algorithm requires fine-tuning of various parameters and is not scalable. Moreover, several applications involve dynamically changing network topologies requiring node/link robustness to be dynamically estimated and maintained. The existing approaches being analytical in nature, and not exhibiting an inductive nature, one has to repeat the same procedure whenever the graph structure changes.

In contrast to optimization-based, and related approximation-based approaches, machine learning frameworks that exploit their ability of extract patterns from data and topology of

the system have also been applied for critical node identification. For instance, En-Yu et al. in [58] have proposed the use of convolutional neural networks applied on a feature matrix derived from the adjacency matrix of a graph to identify critical nodes. The neural network model learns the topological structure of the graph and identifies the critical nodes in the system. Sun et al. in [59] formulate a binary classification problem to identify the influential nodes in a graph by making use of the adjacency matrix and the eigen vector matrix of an influence graph derived from the original topology of the system. However, despite the use of machine learning algorithms for automatically learning the topological information from the features, the above approaches lack significantly in their suitability to real-life problems. Specifically, the use of a convolutional neural network with the adjacency matrix as an input restricts the applicability of the trained model to networks of the same size as the trained graphs. Moreover, such an approach always depends on the information of the entire graph to calculate the criticality of a node anywhere in the graph. Furthermore, in these approaches, the fundamental definition of critical nodes involves the use of either a susceptible-infected-recovered (SIR) model, or an influence graph derived from the original graph of the system. These definitions inherently capture the ability of a node to infect/influence a large portion of the graph, a property referred to in the literature as influence of the node. On the other hand, the criticality of a node refers to the importance of the node in proper functioning of the system, and is distinct from its influence. In addition, existing methods are limited to identification of critical (or rather influential) nodes in the network, and do not address the problem of critical link identification. Finally, these methods are not inductive in nature, and thus require fresh training for graphs of different sizes and types.

2.3 Handling Uncertainty in Graph neural network

A variety of Bayesian methods are used in the literature to handle aleatoric uncertainty in deep neural networks [60–63]. However, there are very few works for GNN models. The authors in Zhang et al. [64] propose a Bayesian framework using joint estimation of graph structure and GNN parameters. The authors make use of families of parametric random

graphs to estimate the structure and parameters. This makes the approach sensitive to the choice of the random graph model and the extent to which the random network can accurately represent the characteristics of the true underlying network. As a result, inferences can be inconsistent for different problems and datasets. Another significant drawback of the technique is that the posterior inference of the graph is carried out solely conditioned on the observed graph. As a result, any information provided by the node features and the training labels is completely disregarded. Therefore, Pal et al. [65] proposed an alternative approach which formulates the posterior inference of the graph in a non-parametric fashion, conditioned on the observed graph, features and training labels. Precisely, they obtain MAP estimate of graph, and conducted all the classification/regression tasks on this estimate. It is argued that MAP estimate handles aleatoric uncertainty of the input graph. However, the approach does not systematically define/quantify the sources of uncertainty and their impact on the predictions. Specifically, the uncertainties AU1 and AU2 are not considered in the framework. We address these shortcomings by explicitly incorporating AU1 and AU2 in our framework. Specifically, ADF is leveraged to propagate the aleatoric uncertainty from the input of the GNN to final node predictions through all the intermediate layers

Similar to aleatoric uncertainty, the literature on handling epistemic uncertainty in GNN is limited. Zhang et al.[64] and Pal et al. [65] are some of the few efforts that generated multiple Monte-Carlo samples by using dropout at test time. To address the problem of huge dependency of Zhang et al.[64] on the assumed random graph model, Hasan et al. [66] introduces a stochastic regularization technique for GNN by adaptive connection sampling. Specifically, it adaptively learns the dropout rate for each layer of GNN. Akin to many of the aforementioned methods, we use the dropout-at-test approach to generate Monte-Carlo samples and estimate the epistemic uncertainty in predictions.

2.4 Modeling interdependent systems

The interdependency among infrastructures calls for coordination of the constituent systems. Some examples of integrated infrastructures include (1) electricity and water networks

[67, 68], (2) electricity and transportation networks [69, 70], (3) natural gas and electricity networks [71, 72], (4) heating and electricity networks [73], among others.

To enhance planning, maintenance and emergency decision making, efficient and effective modeling and simulation tools are required. In this regard, Ouyang provides a comprehensive review of a variety of modeling frameworks for interdependent systems in [74]. In particular, graph theory based modeling method is one of the most frequently employed techniques, particularly to understand and quantify the coupling phenomena between different infrastructures [36, 75]. Graph theory has formed the basis of many research works for network based systems, such as transportation systems [76], power grids [77], water networks [78], supply chains [79], and healthcare systems [80]. Conventionally, nodes of a graph representation denote the various infrastructure or service components and edges signify the physical connections among them. To exploit the conventional graph theory based approaches to model interdependent infrastructures, the network science community introduced the concept of "multilayer network". This approach can facilitate modeling of "systems of systems" that crosses entity boundaries. Several prior publications have presented the relevant research works in this field [81–85]. For instance, in [81], the topological characterization of two interdependent small-sized networks is analyzed using multi-layer network concept to investigate the effects of interdependency on network recovery capability. Similarly, Yu et al. address the issue of uncertainties in edge connections across interdependent critical infrastructures via establishing the stochastic block models [82]. In [83, 84], a novel graph model to map the complex interactions among infrastructures and extract interdependencies among the building systems, considering shocks and stresses, is proposed and explored. In [85], a graph based modeling approach is developed to analyze topological factors in a coupled power-law network to characterize network robustness, considering dependency types among networks. However, the conventional underlying modeling philosophy is adopted in the relevant works mentioned earlier, i.e. nodes represent resources and edges denote their physical connections. Although the conventional graph representation of interdependent infrastructure networks can reveal the topological information and fundamental relationship among system components, for complex systems, it does not provide in-depth knowledge

about system processes and mutual dependencies among resources. In other words, questions such as *"What are the dependencies of a particular facility on the others?"*, *"What processes are performed within nodes and edges?"*, *"How important are nodes and edges for system performance?"* cannot be properly addressed by analyzing conventional graph representations [86]. Therefore, Farid et al. [36] introduce a "Hetero-functional Graph Theory" (HFGT) framework to rigorously model interdependent infrastructure networks and tackle all the challenges unaddressed within the conventional graph theory paradigm. The HFGT framework has developed from roots in the Axiomatic Design for Large Flexible Engineering Systems,which provides a rigorous platform for modeling systems of systems [87].

Chapter 3

Performance-based Robustness Analysis: Identification of Dominant Voltage influencer nodes-Foundations

Robustness evaluation of complex systems is either based on topological features of the associated network or system performance measures. Voltage fluctuation is one such performance-based factor to consider while assessing power grid robustness, especially with the increasingly dynamic nature of the grid. In particular, massive deployments of rooftop photovoltaics (PVs) and demand response programs to incentivize consumers for peak load shaving are emerging across communities around the world. PV generation offers reduced operation costs, a low carbon footprint, and other ancillary services. Despite a variety of benefits, high PV penetration imposes significant challenges on the control and operation of distribution systems, in terms of (1) voltage stability affected by the increase in uncertainty due to intermittent power characteristics; (2) complexity of the system associated with bidirectional power flow, and (3) unbalanced characteristics due to a variable number/size of PV installations on the three phases [88]. Therefore, robustness to voltage fluctuations has emerged as a critical need for smart distribution networks with PV penetrations.

Voltage sensitivity analysis (VSA) quantifies the voltage variation at a given node due to

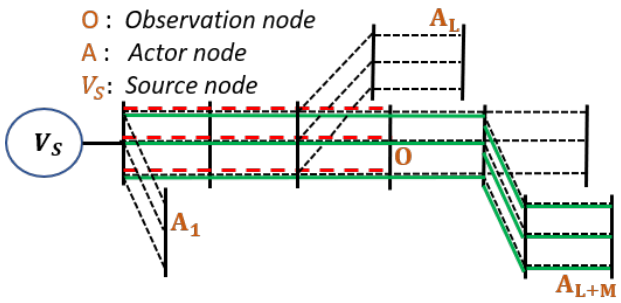


Figure 3.1: Example network with multiple actor nodes

power changes at other locations of the network and can be used as an effective tool to quantify the impacts of PV variations and MTD-related intentional perturbations on the voltage stability across the network. Therefore, this chapter proposes a computationally efficient analytical framework for voltage sensitivity analysis (VSA) that allows for effective robustness analysis. The proposed framework systematically incorporates the stochasticity associated with PV generation, resulting in a probabilistic voltage sensitivity analysis (PVSA) tool. This framework is later used in Chapter 4 to develop a novel performance-based voltage influencing index.

3.1 Analytical approximation of VSA

This section introduces an analytical approach to VSA for three phase unbalanced power distribution system. Changes in real or reactive power at any phase of the bus result in voltage changes at all phases across all nodes of the distribution system. Nodes, where power changes are referred to as actor nodes (A), and the nodes where voltage change is monitored are referred to as observation nodes (O). This work assumes that the source bus is a slack bus and the load is modeled as constant power load with star configuration, which serves as an example for illustration. The analytical approximation for voltage change at an observation node due to the power change at an actor node is stated in Theorem 1.

Theorem 1. *For an unbalanced power distribution system, change in complex voltage (ΔV_{OA}) at an observation node (O) due to change in complex power of an actor node (A) can be*

approximated by,

$$\begin{bmatrix} \Delta V_O^a \\ \Delta V_O^b \\ \Delta V_O^c \end{bmatrix} \approx - \begin{bmatrix} \frac{\Delta S_A^{a\star} Z_{OA}^{aa}}{V_A^{a\star}} + \frac{\Delta S_A^{b\star} Z_{OA}^{ab}}{V_A^{b\star}} + \frac{\Delta S_A^{c\star} Z_{OA}^{ac}}{V_A^{c\star}} \\ \frac{\Delta S_A^{a\star} Z_{OA}^{ba}}{V_A^{a\star}} + \frac{\Delta S_A^{b\star} Z_{OA}^{bb}}{V_A^{b\star}} + \frac{\Delta S_A^{c\star} Z_{OA}^{bc}}{V_A^{c\star}} \\ \frac{\Delta S_A^{a\star} Z_{OA}^{ca}}{V_A^{a\star}} + \frac{\Delta S_A^{b\star} Z_{OA}^{cb}}{V_A^{b\star}} + \frac{\Delta S_A^{c\star} Z_{OA}^{cc}}{V_A^{c\star}} \end{bmatrix}, \tag{3.1}$$

where a, b and c represent the three phases, which are consistent throughout the work. V_A^a and ΔS_A^a represent complex voltage and power changes at the phase a of the actor node A, respectively; Z denotes the impedance matrix including self and mutual line impedance of the shared path between observation node and actor node from the source node. Fig. 3.1 depicts an unbalanced three phase distribution system for illustration. The red lines represent the shared paths between actor node A_{L+M} and observation node O, from the source node.

Proof. Voltage at an observation node can be computed in terms of the difference between voltage at the source node and sum of the voltage drops across all lines/edges between the source node and observation node. Let E_o be set of all edges between the source node and the observation node. Using KVL, voltage at observation node o can be written as:

$$\begin{bmatrix} V_{Oa} \\ V_{Ob} \\ V_{Oc} \end{bmatrix} = \begin{bmatrix} V_{Sa} \\ V_{Sb} \\ V_{Sc} \end{bmatrix} - \sum_{e \epsilon E_o} \begin{bmatrix} V_{ea}^d \\ V_{eb}^d \\ V_{ec}^d \end{bmatrix}, \tag{3.2}$$

where $\boldsymbol{V_O}$, $\boldsymbol{V_S}$, and $\boldsymbol{V_e^d}$ are voltage at observation node, voltage of source node, and the voltage drop across edge e, respectively. Let $\boldsymbol{I_e}$ and $\boldsymbol{Z_e}$ be the current and impedance for edge e. Here, along with self impedance, mutual impedance of the line will also contribute to the voltage drop. In LV distribution network, value of shunt impedance can be ignored. We can represent (3.2) in a form incorporating line current and impedance, denoted by $\boldsymbol{I_e}$ and $\boldsymbol{Z_e}$ as:

$$\boldsymbol{V_O} = \boldsymbol{V_S} - \sum_{e \epsilon E_o} \boldsymbol{Z_e I_e}, \tag{3.3}$$

$$\boldsymbol{V_O} = \begin{bmatrix} V_{Oa} \\ V_{Ob} \\ V_{Oc} \end{bmatrix}, \boldsymbol{I_e} = \begin{bmatrix} I_e^a \\ I_e^b \\ I_e^c \end{bmatrix}, \boldsymbol{Z_e} = \begin{bmatrix} Z_e^{aa} & Z_e^{ab} & Z_e^{ac} \\ Z_e^{ba} & Z_e^{bb} & Z_e^{bc} \\ Z_e^{ca} & Z_e^{cb} & Z_e^{cc} \end{bmatrix}.$$

Let S_n be complex power consumption or injection at node n and V_n^* be the complex conjugate of voltage at node n. The current flowing through a particular phase (say phase a) of edge e can be written as $I_e^a = \sum_{n \epsilon N_e} \frac{S_{na}^*}{V_{na}^\star}$ where N_e is the set of all nodes n for which edge e is between node n and source node. Power from the source node to all the nodes in the set N_e flows through edge e. Therefore, current in edge e will be affected by the power change at nodes $n \epsilon N_e$. Therefore, the voltage at the observation node can be written as:

$$\boldsymbol{V_O} = \boldsymbol{V_S} - \sum_{e \epsilon E_o} \sum_{n \epsilon N_e} \boldsymbol{Z_e} \begin{bmatrix} \frac{S_{na}^\star}{V_{na}^\star} & \frac{S_{nb}^\star}{V_{nb}^\star} & \frac{S_{nc}^\star}{V_{nc}^\star} \end{bmatrix}^{\mathrm{T}} \tag{3.4}$$

When power consumption of node n changes from S_n to $S_n^{'}$, the voltage will change from V_n to $V_n^{'}$ and consequently voltage at observation node will change to $V_o^{'}$. The new voltage at observation node can be written as:

$$\boldsymbol{V_O'} = \boldsymbol{V_S} - \sum_{e \epsilon E_o} \sum_{n \epsilon N_e} \boldsymbol{Z_e} \begin{bmatrix} \frac{S_{na}'^\star}{V_{na}'^\star} & \frac{S_{nb}'^\star}{V_{nb}'^\star} & \frac{S_{nc}'^\star}{V_{nc}'^\star} \end{bmatrix}^{\mathrm{T}} \tag{3.5}$$

where $S_n'^\star = S_n^\star + \Delta S_n^\star$ and $V_n^{'} = V_n + \Delta V_n$. The effective voltage change at observation node can be written as $\boldsymbol{\Delta V_o = V_o - V_o'}$. Using (3.4) and (3.5), change in voltage at observation

node can be expressed as:

$$\begin{aligned}\boldsymbol{\Delta V_O} &= \sum_{e \epsilon E_o} \boldsymbol{Z_e} \left(\sum_{n \epsilon N_e} \begin{bmatrix} \frac{S^\star_{na}}{V^\star_{na}} - \frac{S^\star_{na}+\Delta S^\star_{na}}{V_{na}+\Delta V_{na}} \\ \frac{S^\star_{nb}}{V^\star_{nb}} - \frac{S^\star_{nb}+\Delta S^\star_{nb}}{V_{nb}+\Delta V_{nb}} \\ \frac{S^\star_{nc}}{V^\star_{nc}} - \frac{S^\star_{nc}+\Delta S^\star_{nc}}{V_{nc}+\Delta V_{nc}} \end{bmatrix} \right) \\ &= \sum_{e \epsilon E_o} \boldsymbol{Z_e} \left(\sum_{n \epsilon N_e} \begin{bmatrix} \frac{S^\star_{na}\Delta V^\star_{na} - \Delta S^\star_{na} V_{na}}{V^\star_{na}(V^\star_{na}+\Delta V^*_{na})} \\ \frac{S^\star_{nb}\Delta V^\star_{nb} - \Delta S^\star_{nb} V_{nb}}{V^\star_{nb}(V^\star_{nb}+\Delta V^*_{nb})} \\ \frac{S^\star_{nc}\Delta V^\star_{nc} - \Delta S^\star_{nc} V_{nc}}{V^\star_{nc}(V^\star_{nc}+\Delta V^*_{nc})} \end{bmatrix} \right)\end{aligned} \tag{3.6}$$

In practice, voltage changes are typically small compared to actual node voltage. Hence, it is reasonable to assume that $\Delta V^*_n/(V^*_n + \Delta V^*_n) \to 0$. Thus, (3.6) can be approximated as:

$$\boldsymbol{\Delta V_O} = \sum_{e \epsilon E_o} \boldsymbol{Z_e} \left(\sum_{n \epsilon N_e} \begin{bmatrix} \frac{-\Delta S^\star_{na}}{V^\star_{na}+\Delta V^*_{na}} \\ \frac{-\Delta S^\star_{nb}}{V^\star_{nb}+\Delta V^*_{nb}} \\ \frac{-\Delta S^\star_{nc}}{V^\star_{nc}+\Delta V^*_{nc}} \end{bmatrix} \right) \tag{3.7}$$

Let E_θ be set of all edges between the actor node and source node. When actor node θ changes power consumption, current flowing through the edges changes for all edges of set E_θ. Voltage drop across the edges between the source node and observation node, changes only for edges that belongs to subset $E_\theta \cap E_o$.

$$\boldsymbol{\Delta V_{O\theta}} = \boldsymbol{Z_{O\theta}} \left(\begin{bmatrix} \frac{-\Delta S^\star_{\theta a}}{V^\star_{\theta a}+\Delta V^*_{\theta a}} \\ \frac{-\Delta S^\star_{\theta b}}{V^\star_{\theta b}+\Delta V^*_{\theta b}} \\ \frac{-\Delta S^\star_{\theta c}}{V^\star_{\theta c}+\Delta V^*_{\theta c}} \end{bmatrix} \right) \tag{3.8}$$

where $\boldsymbol{Z_{O\theta}} = \sum_{e \epsilon E_o \cap E_\theta} \boldsymbol{Z_e}$ is the impedance matrix. Here, each component in the summation is the impedance of the shared path between the actor node and observation node from

source node. We decompose (3.8) into real and imaginary components as follows:

$$\begin{aligned}
\boldsymbol{\Delta V^{r}_{O\theta}} &= -\begin{bmatrix} \frac{(\Delta P_{\theta a} R_{o\theta,aa} + \Delta Q_{\theta a} X_{o\theta,aa})(V^{\star r}_{\theta a} + \Delta V^{\star r}_{\theta a})}{(V^{\star r}_{\theta a} + \Delta V^{\star r}_{\theta a})^2 + (V^{\star i}_{\theta a} + \Delta V^{\star i}_{\theta a})^2} - \\ \frac{(\Delta P_{\theta a} X_{o\theta,aa} - \Delta Q_{\theta a} R_{o\theta,aa})(V^{\star i}_{\theta a} + \Delta V^{\star i}_{\theta a})}{(V^{\star r}_{\theta a} + \Delta V^{\star r}_{\theta a})^2 + (V^{\star i}_{\theta a} + \Delta V^{\star i}_{\theta a})^2} + \dots \\ \vdots \end{bmatrix}_{3\times 1} \\
\boldsymbol{\Delta V^{i}_{O\theta}} &= -\begin{bmatrix} \frac{(\Delta P_{\theta a} X_{o\theta,aa} - \Delta Q_{\theta a} R_{o\theta,aa})(V^{\star r}_{\theta a} + \Delta V^{\star r}_{\theta a})}{(V^{\star r}_{\theta a} + \Delta V^{\star r}_{\theta a})^2 + (V^{\star i}_{\theta a} + \Delta V^{\star i}_{\theta a})^2} + \\ \frac{(\Delta P_{\theta a} R_{o\theta,aa} + \Delta Q_{\theta a} X_{o\theta,aa})(V^{\star i}_{\theta a} + \Delta V^{\star i}_{\theta a})}{(V^{\star r}_{\theta a} + \Delta V^{\star r}_{\theta a})^2 + (V^{\star i}_{\theta a} + \Delta V^{\star i}_{\theta a})^2} + \dots \\ \vdots \end{bmatrix}_{3\times 1}
\end{aligned} \tag{3.9}$$

where superscript r and i represents the real and imaginary components, respectively. In a distribution network, voltage angle relative to source node and magnitude of voltage change is usually very small. Under the above assumptions, the real part of voltage change can be approximated as:

$$\boldsymbol{\Delta V^{r}_{O\theta}} \approx -\begin{bmatrix} \frac{(\Delta P_{\theta a} R_{o\theta,aa} + \Delta Q_{\theta a} X_{o\theta,aa})(V^{\star r}_{\theta a})}{(V^{\star r}_{\theta a})^2 + (V^{\star i}_{\theta a})^2} - \\ \frac{(\Delta P_{\theta a} X_{o\theta,aa} - \Delta Q_{\theta a} R_{o\theta,aa})(V^{\star i}_{\theta a})}{(V^{\star r}_{\theta a})^2 + (V^{\star i}_{\theta a})^2} + \dots \\ \vdots \end{bmatrix}_{3\times 1} \tag{3.10}$$

Similarly, with the same arguments, the imaginary part can be approximated. By recombining the real and imaginary parts, the approximate voltage change for all three phases can be written as given in Theorem 1.

$$\boldsymbol{\Delta V_{O\theta}} \approx -\begin{bmatrix} \frac{\Delta S^{\star}_{\theta a} Z_{aa}}{V^{\star}_{\theta a}} + \frac{\Delta S^{\star}_{\theta b} Z_{ab}}{V^{\star}_{\theta b}} + \frac{\Delta S^{\star}_{\theta c} Z_{ac}}{V^{\star}_{\theta c}} \\ \frac{\Delta S^{\star}_{\theta a} Z_{ba}}{V^{\star}_{\theta a}} + \frac{\Delta S^{\star}_{\theta b} Z_{bb}}{V^{\star}_{\theta b}} + \frac{\Delta S^{\star}_{\theta c} Z_{bc}}{V^{\star}_{\theta c}} \\ \frac{\Delta S^{\star}_{\theta a} Z_{ca}}{V^{\star}_{\theta a}} + \frac{\Delta S^{\star}_{\theta b} Z_{cb}}{V^{\star}_{\theta b}} + \frac{\Delta S^{\star}_{\theta c} Z_{cc}}{V^{\star}_{\theta c}} \end{bmatrix} \tag{3.11}$$

□

3.1.1 Multiple actor nodes

With increasing penetration of DERs at different locations across the grid, it is important to analyze the impact of multiple actor nodes on distribution system voltage. Therefore, voltage change expression derived for a single actor node in eqn. (3.11) is extended for multiple actor nodes in the following corollary:

Corollary 2. *For an unbalanced power distribution system, change in complex voltage ΔV_O at an observation node (O) due to change in complex power at multiple actor nodes can be approximated by*

$$\begin{bmatrix} \Delta V_O^a \\ \Delta V_O^b \\ \Delta V_O^c \end{bmatrix} \approx -\sum_{A \epsilon \tilde{A}} \left(\begin{bmatrix} \frac{\Delta S_A^{a\star} Z_{OA}^{aa}}{V_A^{a\star}} + \frac{\Delta S_A^{b\star} Z_{OA}^{ab}}{V_A^{b\star}} + \frac{\Delta S_A^{c\star} Z_{OA}^{ac}}{V_A^{c\star}} \\ \frac{\Delta S_A^{a\star} Z_{OA}^{ba}}{V_A^{a\star}} + \frac{\Delta S_A^{b\star} Z_{OA}^{bb}}{V_A^{b\star}} + \frac{\Delta S_A^{c\star} Z_{OA}^{bc}}{V_A^{c\star}} \\ \frac{\Delta S_A^{a\star} Z_{OA}^{ca}}{V_A^{a\star}} + \frac{\Delta S_A^{b\star} Z_{OA}^{cb}}{V_A^{b\star}} + \frac{\Delta S_A^{c\star} Z_{OA}^{cc}}{V_A^{c\star}} \end{bmatrix} \right) \tag{3.12}$$

where $\tilde{A}$ is the set of all actor nodes.

Proof. Following (3.7), voltage change expression at an observation node due to single actor node can be written as:

$$\begin{aligned} \boldsymbol{\Delta V_O} &= \sum_{e \epsilon E_o} \boldsymbol{Z_e} \left(\sum_{n \epsilon N_e} \begin{bmatrix} \frac{-\Delta S_n^{a\star}}{V_n^{a\star} + \Delta V_n^{a\star}} \\ \frac{-\Delta S_n^{b\star}}{V_n^{b\star} + \Delta V_n^{b\star}} \\ \frac{-\Delta S_n^{c\star}}{V_n^{c\star} + \Delta V_n^{c\star}} \end{bmatrix} \right) \\ &= \sum_{e \epsilon E_o} \boldsymbol{Z_e} \left(\sum_{n \epsilon N_e} \boldsymbol{I_n} \right), \end{aligned} \tag{3.13}$$

where, $\boldsymbol{I_n} = \begin{bmatrix} \frac{-\Delta S_n^{a\star}}{V_n^a + \Delta V_n^{a\star}} & \frac{-\Delta S_n^{b\star}}{V_n^{b\star} + \Delta V_n^{b\star}} & \frac{-\Delta S_n^{c\star}}{V_n^{c\star} + \Delta V_n^{c\star}} \end{bmatrix}^{\mathrm{T}}$. Let us assume a generic network with $L + M$ actor nodes such that there are L nodes between the source node and observation node O and M nodes between observation node and last actor node of the network as shown in the Fig. 1. The nodes are arranged in such a way that the set $E_O \cap E_{A_1}$ has minimum elements

(edges) and the sets $E_O \cap E_{A_{L+1}}$ to $E_O \cap E_{A_{L+M}}$ have same and maximum number of edges. This is represented as,

$$\begin{aligned} &|E_O \cap E_{A_1}| \leq |E_O \cap E_{A_2}| \ldots \leq |E_O \cap E_{A_L}| \\ &\leq |E_O \cap E_{A_{L+1}}| = |E_O \cap E_{A_{L+2}}| \ldots = |E_O \cap E_{A_{L+M}}| \end{aligned} \tag{3.14}$$

where $|E_O \cap E_{A_1}|$ denotes the cardinality of set $E_O \cap E_{A_1}$. On dividing set E_O into $L+1$ subsets as,

$$\begin{aligned} E_O =& |E_O \cap E_{A_1}| \cup |E_O \cap (E_{A_2} - E_{A_1})| \cup \ldots \\ & |E_O \cap (E_{A_{L+1}} - E_{A_L})| \\ =& \bigcup_{l=1}^{A_{L+1}} E_O \cap (E_{A_l} - E_{A_{l-1}}) \end{aligned} \tag{3.15}$$

since $E_O \cap (E_{A_L} - E_{A_{L-1}}) = \phi$ for $A_L = A_{L+2}$ or greater. Using this, (3.13) is expressed as,

$$\begin{aligned} \boldsymbol{\Delta V_O} &= \sum_{l=1}^{L+1} \sum_{e \epsilon E_O \cap E_{A_l} - E_O \cap E_{A_{l-1}}} (\boldsymbol{Z_e I_n}) \\ &= \sum_{n=A_1}^{A_L} \left(\sum_{e \epsilon E_o \cap E_n} \boldsymbol{Z_e} \right) \boldsymbol{I_n}. \end{aligned} \tag{3.16}$$

When power injection/consumption changes at the actor node n, the current flowing through the edges change for all edges of the set E_n. However, the voltage drop across the edges between source node and observation node, changes only for edges that belong to subset $E_n \cap E_o$. Taking the sum of the impedance across all such edges, reduces (3.16) to:

$$\begin{aligned} \boldsymbol{\Delta V_O} &= \sum_{n=A_1}^{A_L} \boldsymbol{Z_{on} I_n}, \\ &= \sum_{n=A_1}^{A_L} \boldsymbol{\Delta V_{On}} \end{aligned} \tag{3.17}$$

where $\boldsymbol{Z_{on}} = \sum_{e \epsilon E_o \cap E_n} \boldsymbol{Z_e}$ is the impedance matrix and its elements are computed by the summation of the impedances of shared paths between all actor nodes and observation node from source node. $\boldsymbol{\Delta V_{On}}$ is the voltage change due to a single actor node n. This proves that the analytical expression in 3.11 exhibits superposition law, thus can be extended to (3.12) for multiple actor nodes. □

3.1.2 Upper bound on approximation error

The proposed analytical method in Corollary 2 approximates the true voltage change for a large range of power variation with very small error magnitude. To further substanticiate the quality of this approximation, Corollary 3 provides an upper bound for the error.

Corollary 3. *For an unbalanced power distribution system, the errors in the real (ΔV_e^r) and imaginary part (ΔV_e^i) of the voltage change approximation are upper bounded by:*

$$\begin{aligned} \Delta V_e^r &\leq \sum_{u \epsilon \tilde{U}} \left(\frac{k_1^u/(1+c_1^u)}{V_A^{a,r}} + \frac{k_2^u/(1+c_2^u)}{V_A^{a,i}} \right) \\ \Delta V_e^i &\leq \sum_{u \epsilon \tilde{U}} \left(\frac{k_2^u/(1+c_1^u)}{V_A^{a,r}} + \frac{k_1^u/(1+c_2^u)}{V_A^{a,i}} \right), \end{aligned} \tag{3.18}$$

where k_1, k_2, c_1, c_2 are parameters dependent on the power change and impedance of the corresponding phases. The set $\tilde{U}$ contains the self and cross phase terms of the phase where error in voltage change is computed (e.g. it is aa, ab, ac for phase a). The voltage change at any phase consist of three components from the three phases. The value of these parameters for phase a are: $k_1 = \Delta P_A^a R_{OA}^{aa} - \Delta Q_A^a X_{OA}^{aa}$, $k_2 = \Delta P_A^a R_{OA}^{aa} - \Delta Q_A^a X_{OA}^{aa}$, $c_1 = (V_A^{a,i}/V_A^{a,r})^2$, $c_2 = c_1^{-1}$.

Proof. Firstly, let us recall that the analytical approximation derived in Theorem 1 is based on a legitimate assumption that the value of voltage change (ΔV_A) can be ignored compared to the rated voltage (V_A). In other words, the terms containing ΔV_A in equation (3.6) are neglected, which leads to the approximation computed in (3.7). Despite the accurate approximation, this simplification incurs the inevitable error. In Corollary 3, we prove that the

incurred error is upper bounded, which ensures the stability of our approximation method. The approximation error in voltage change at any phase consists of three components, corresponding to three phases. The error $\Delta V_e^{aa,r}$ in real part of phase a due to phase a component is the difference of actual voltage change (eqn. (3.18)) and its approximation (eqn. (3.19)) expressed as,

$$\begin{aligned}\Delta V_e^{aa,r} = &\frac{(k_1)(V_A^{a,r}+\Delta V_A^{a,r})}{(V_A^{a,r}+\Delta V_A^{a,r})^2+(V_A^{a,i}+\Delta V_{Aa}^{a,i})^2} - \frac{(k_1)(V_A^{a,r})}{(V_A^{a,r})^2+(V_A^{a,i})^2} \\ &+\frac{(k_2)(V_A^{a,i}+\Delta V_A^{a,i})}{(V_A^{a,r}+\Delta V_A^{a,r})^2+(V_A^{a,i}+\Delta V_A^{a,i})^2} - \frac{(k_2)(V_A^{a,i})}{(V_A^{a,r})^2+(V_A^{a,i})^2}.\end{aligned} \tag{3.19}$$

Similar components from phase b and c exist, which together with (3.19) contribute to the error in phase a. (3.19) can be further simplified as,

$$\begin{aligned}\Delta V_e^{aa,r} &= \frac{(k_1)(\tau^r)(V_A^{a,r})}{(\tau^r)^2(V_A^{a,r})^2+(\tau^i)^2(V_A^{a,i})^2} - \frac{(k_1)(V_A^{a,r})}{(V_A^{a,r})^2+(V_A^{a,i})^2} \\ &+ \frac{(k_2)(\tau^i)(V_{Aa}^{a,i})}{(\tau^r)^2(V_A^{a,r})^2+(\tau^i)^2(V_A^{a,i})^2} - \frac{(k_2)(V_A^{a,i})}{(V_A^{a,r})^2+(V_A^{a,i})^2} \\ &= \Delta V_{e1}^{aa,r} + \Delta V_{e2}^{aa,r}\end{aligned} \tag{3.20}$$

where $\tau^r = 1+\epsilon^r, \tau^i = 1+\epsilon^i, \epsilon^r = (\Delta V_A^{a,r}/V_A^{a,r}), \epsilon^i = (\Delta V_A^{a,i}/V_A^{a,i})$. Here, equation (3.20) consists of two similar error components $\Delta V_{e1}^{aa,r}$ and $\Delta V_{e2}^{aa,r}$, which are evaluated separately:

$$\Delta V_{e1}^{aa,r} = \frac{k_1/(1+c_1)}{V_A^{a,r}}\left(\frac{1}{\tau^r}-1\right) \tag{3.21}$$

where $c_1 = (V_A^{a,i}/V_A^{a,r})^2$. As the ratio of change in voltage and rated voltage, i.e., ϵ^r and ϵ^i are typically very small, we can argue the following inequality:

$$\epsilon^r \le 1-\epsilon^r \implies \frac{\epsilon^r}{1-\epsilon^r} \le 1 \implies \frac{1}{\tau^r}-1 \le 1 \tag{3.22}$$

Then, using (**??**), equation (**??**) can be bounded as,

$$\Delta V_{e1}^{aa,r} = \frac{k_1/(1+c_1)}{V_A^{a,r}}\left(\frac{1}{\tau^r}-1\right) \le \frac{k_1/(1+c_1)}{V_A^{a,r}} \tag{3.23}$$

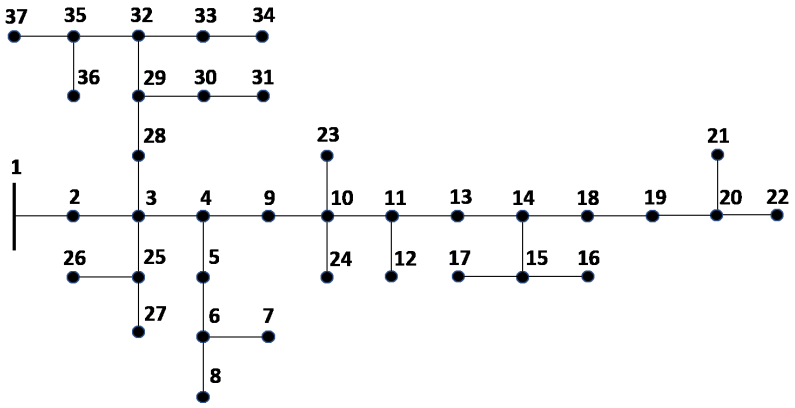


Figure 3.2: Modified IEEE 37 node network

Similarly, with the same arguments, the upper bound can be derived for second part of (3.20) as,

$$\Delta V_{e2}^{aa,r} = \frac{k_2/(1+c_2)}{V_A^{a,i}} \left(\frac{1}{\tau^i} - 1 \right) \leq \frac{k_2/(1+c_2)}{V_A^{a,i}} \tag{3.24}$$

Equations (3.23) and (3.24) are combined to arrive at the upper bound on the specific component of the voltage change, contributed from phase a.

$$\Delta V_e^{aa,r} \leq \frac{k_1/(1+c_1)}{V_A^{a,r}} + \frac{k_2/(1+c_2)}{V_A^{a,i}} \tag{3.25}$$

The bound on the other parts of voltage change, which are contributed from phase b and c, i.e., $\Delta V_e^{ab,r}$ and $\Delta V_e^{ac,r}$, is similar in form to (3.25) except for the constants (k_1, k_2, c_1, c_2) which are dependent on the power and shared path impedance of the corresponding phases. Then, the bound from all the phase terms are added to give the aggregate upper bound on the real part of voltage change in phase a, as stated in Corollary 2. The same procedure can be applied to derive the bounds for the imaginary part of voltage change in phase a. Finally, the bound on the error magnitude can be computed from the bounds on the real and imaginary parts of the error. The upper bound is validated against the actual error in the next sub-section. □

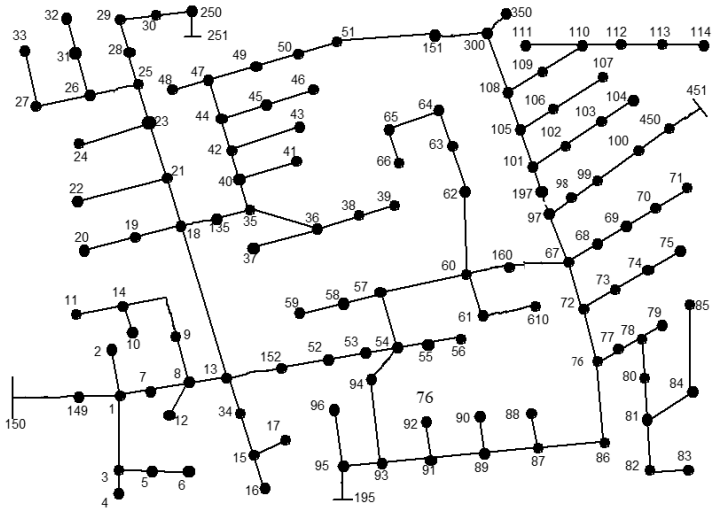


Figure 3.3: Modified IEEE 123 node network

Table 3.1: Power change across different actor nodes

Actor nodes	Phase	Rated power (kVA)	New power (kVA)
22	c	42 + j21	63 + j21
17	b	42 + j21	63 + j31
14	c	84 + j42	126 + j21
8	a	42 + j21	63 + j31
7	c	84 + j42	126 + j21

3.1.3 Validation of VSA

The analytical approximations of VSA need validation via appropriate test networks. Therefore, this section verifies the derived expression using the modified IEEE 37-node and the IEEE 123-node test systems, shown in Fig. 3.2 and 3.3, respectively. These test networks are selected due to their unbalanced characteristics, which include both single and three phase loads. The nominal voltage of the IEEE 37-node test system is 4.8 kV whereas it is 4.16 kV for the IEEE 123-node system. The Classical NR method is used as a baseline method for validating our proposed methods.

The accuracy of the VSA approximation for multiple actor node case, is first evaluated in the 37-node network by simulating the a scenario assuming $22, 17, 14, 8, 7$ as actor nodes. The power changes at these actor nodes will occur simultaneously at different phases, which is about 50% of their rated load as tabulated in Table 3.1.

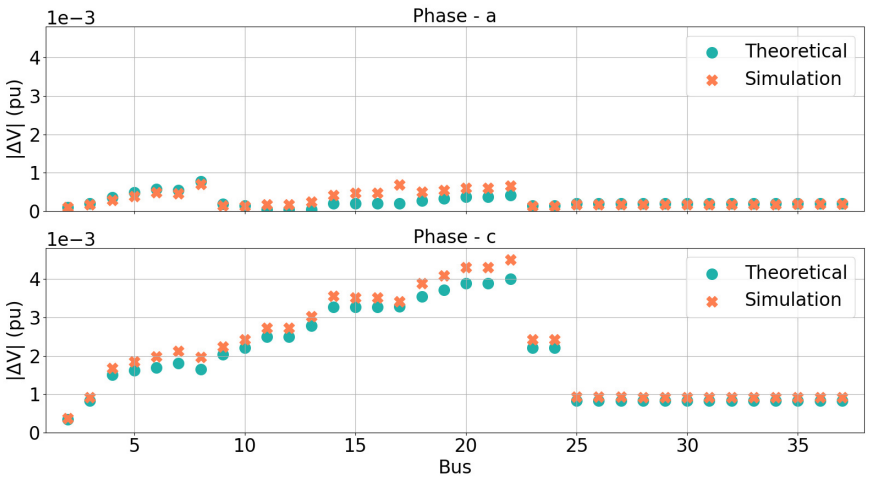


Figure 3.4: Voltage change on all nodes of 37 node network due to multiple actor nodes

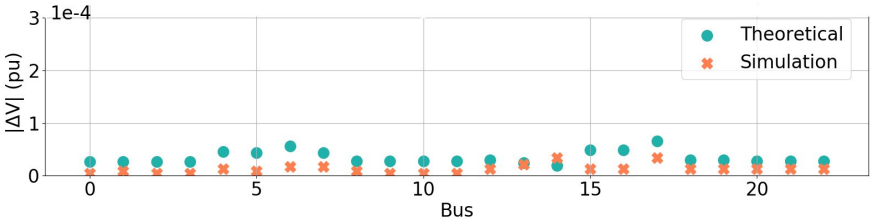


Figure 3.5: Voltage change at phase a of the selected nodes in 123 node test network due to multiple actor nodes

Fig. 3.4 shows the voltage change at various observation nodes. It can be observed that the errors between the theoretical approximation and simulated voltage change are negligible as the maximum deviation is in the range of 10^{-4} pu. The absolute value of error average over all the observation nodes is 0.000196 pu, which is significantly low. As expected, the magnitude of voltage change increases as the observation node moves away from the source node. This is due to the increase in the length of the shared path between the observation node and the actor nodes from the source node. Furthermore, the voltage change remains constant for the range of observation nodes from 25 to 37. This is due to the constant length of the shared paths between the actor nodes and these observation nodes.

Similar to IEEE 37-node test network, the VSA approximation is also tested in the 123-node network with 7 actor nodes, i.e., nodes 7,11,19,28,35,42,68. Like the tests conducted for the IEEE 37-node system, the power changes at these actor nodes occur simultaneously

with a magnitude equal to 50 % of their rated load. Fig. 3.5 shows the voltage change at various observation nodes. Accurate voltage change estimation using the proposed analytical formulation can be observed, as the error is contained within 10^{-4} pu. To further demonstrate the merits of the derived VSA expression, an upper bound on the approximation error is derived in the next section.

3.2 Probabilistic analysis of Voltage sensitivity (PVSA)

Corollary 2 allows us to compute the voltage change at any observation node from known power changes at multiple actor nodes. However, in practice, the power could vary randomly due to intermittent characteristics of PV generation. This stochastic variation in turn introduces randomness in the voltage across the network. Under such stochastic scenarios, the grid operator might be interested in predicting the probability of experiencing a voltage violation, i.e., $P(|\Delta V_O| > 0.05 \text{ p.u.})$ so that corrective actions can be taken beforehand. Therefore, it becomes relevant and necessary to derive the probability distribution of the magnitude of voltage change at certain nodes of the distribution grid due to random fluctuations in power at actor nodes. This result is provided by Theorem 4.

Theorem 4. *For an unbalanced radial power distribution system, the probability distribution of voltage change at an observation node* (ΔV_O) *due to random changes in power consumption/injection of actor nodes, corresponds to Nakagami distribution*

$$|\Delta V_O| \sim Nakagami(m, \omega) \tag{3.26}$$

where, shape parameter $m = (\sigma_r^2 + \sigma_i^2)/\theta$ *and scale parameter* $\omega = \sqrt{\sigma_r^2 + \sigma_i^2}$. *Here,* $\theta = 2(\sigma_r^4 + \sigma_i^4 + 2c^2)/(\sigma_r^2 + \sigma_i^2)$, $\sigma_r^2 = C_R^T \sum_{\Delta S} C_R$, $\sigma_i^2 = C_I^T \sum_{\Delta S} C_I$ *and* c *is the covariance between the real and imaginary part of voltage change.* C_R *and* C_I *are dependent on the shared path impedances and base voltages of the actor nodes, and* $\sum_{\Delta S}$ *is the covariance matrix of complex power change across different actor nodes.*

Proof. The change in complex voltage at any observation node due to change in complex power injection/consumption of an actor node can be expressed in terms of real and imaginary components as,

$$\Delta V_{OA} = \Delta V_{OA}^{r} + j\Delta V_{OA}^{i},$$

where, the real part ($\Delta V_{OA}^{a,r}$) and imaginary part ($\Delta V_{OA}^{a,i}$) of voltage change at any phase (here, we use phase a as an example that can be applied to other phases also) of observation node O can be written as

$$\begin{aligned}\Delta V_{OA}^{a,r} = \sum_{h,u} \frac{-1}{|V_A^h|}[\Delta P_A^h(R_{OA}^u cos(\omega_A) - X_{OA}^u sin(\omega_A)) \\ +\Delta Q_A^h(R_{OA}^u sin(\omega_A) + X_{OA}^u cos(\omega_A))] \\ \Delta V_{OA}^{a,i} = \sum_{h,u} \frac{-1}{|V_A^h|}[\Delta P_A^h(R_{OA}^u sin(\omega_A) + X_{OA}^u cos(\omega_A)) + \\ \Delta Q_A^h(X_{OA}^u sin(\omega_A) - R_{OA}^u cos(\omega_A))]\end{aligned}$$

where $h \,\epsilon\, \tilde{H}$ and $u \,\epsilon\, \tilde{U}$. The sets $\tilde{H}$ and $\tilde{U}$ denote different phases (i.e., a, b, c) and different phase sequences (i.e., aa, ab, ac), respectively. ΔP_A^h and ΔQ_A^h are the active and reactive power changes, respectively. R_{OA}^h, X_{OA}^h are the resistance and reactance of shared path between the observation node O and actor node A from the source node. V_A^h denotes the base voltage of actor node A.

Using the superposition result of Corollary 1, the net voltage change at an observation node due to the aggregate effect of multiple spatially distributed actor nodes can be written as the sum of changes in voltage at the observation node due to every single actor node as,

$$\Delta V_O^a = \sum_A \Delta V_{OA}^{a,r} + \sum_A \Delta V_{OA}^{a,i} \tag{3.27}$$

Intermittent characteristics of PV injection introduces randomness in the power variation. Here, node power change (ΔS) is modeled as zero-mean random vector with covariance matrix $\sum_{\Delta S}$. As shown in (3.28), the notation (ΔS) is a compact vector representing the

power change of phases a (Δs^a), b (Δs^b), and c (Δs^c). In addition, the vector representing power changes of a phase, phase a for e.g., is composed of active and reactive power changes for the corresponding phase of all the nodes.

$$\begin{aligned} \Delta S &= [\Delta s^a \ \Delta s^b \ \Delta s^c]^T \\ \Delta s^a &= [\Delta P_1^a \dots \Delta P_n^a \ \ \Delta Q_1^a \dots \Delta Q_n^a]^T \end{aligned} \tag{3.28}$$

The distribution of the magnitude of voltage change $|\Delta V_O|$ can be computed using the following steps:

Define covariance matrix

The covariance matrix $\sum_{\Delta S}$ of the complex power change is used to quantify the correlation of power changes among various nodes due to geographical proximity. For nodes that do not have PVs, the variance can be set to zero or equal to the nominal load fluctuation variance. In practice, the covariance structure can be learned using historical data.

Compute constant vectors C_R and C_I

In this work, the network topology with meta parameters is assumed to be known. Let us define two vectors C_R and C_I which can be computed using the following equation.

$$C_R = \begin{bmatrix} c_r^{aa} \\ c_r^{ab} \\ c_r^{ac} \end{bmatrix} \qquad c_r^u = \begin{bmatrix} \frac{-(R_{O1}^u cos(\omega_1) - X_{O1}^u sin(\omega_1))}{|V_1^a|} \\ \vdots \\ \frac{-(R_{On}^u cos(\omega_n) - X_{On}^u sin(\omega_n))}{|V_n^a|} \\ \frac{-(R_{O1}^u sin(\omega_1) + X_{O1}^u cos(\omega_1))}{|V_1^a|} \\ \vdots \\ \frac{-(R_{On}^u sin(\omega_n) + X_{On}^u cos(\omega_n))}{|V_n^a|} \end{bmatrix}$$
$$C_I = \begin{bmatrix} c_r^{aa} \\ c_r^{ab} \\ c_r^{ac} \end{bmatrix} \tag{3.29}$$

where c_r^u is the constant matrix for a given set of actor nodes and u denotes the self or

mutual impedance of the phase a line, i.e., aa, ab, ac. A Similar matrix exists for c_i^u but with different values and is omitted for brevity. The compact vectors (C_R) and (C_I) consist of three components corresponding to three phases. Each component (i.e., c_r^u/c_i^u) is composed of ratios between the shared path impedance terms and rated voltage of the corresponding phase for all the nodes.

Compute distribution of ΔV_O^r and ΔV_O^i

Voltage change at an observation node due to multiple actor nodes can be expressed as the weighted sum of elements of vector ΔS as shown by equations (3.30, 3.31). Weights are given from the elements of C_R and C_I, which represent the ratio of shared path impedance and base voltage of the various actor nodes. Invoking the Lindeberg-Feller central limit theorem, it can be shown that the weighted sum of the elements of ΔS converges in distribution to a Gaussian random variable. That is, the distribution of ΔV_O^r and ΔV_O^i can be expressed as,

$$\Delta V_O^{a,r} = \sum_A \Delta V_{OA}^{a,r} = C_R^T \Delta S \overset{D}{\sim} \mathcal{N}(0, C_R^T \textstyle\sum_{\Delta S} C_R) \tag{3.30}$$

$$\Delta V_O^{a,i} = \sum_A \Delta V_{OA}^{a,i} = C_I^T \Delta S \overset{D}{\sim} \mathcal{N}(0, C_I^T \textstyle\sum_{\Delta S} C_I) \tag{3.31}$$

where variances σ_r^2 and σ_i^2 of $\Delta V_O^{a,r}$ and $\Delta V_O^{a,i}$ are $C_R^T \sum_{\Delta S} C_R$ and $C_I^T \sum_{\Delta S} C_I$, respectively.

Compute distribution of $|\Delta V_O|$

After obtaining the voltage change in terms of the real part $\Delta V_O^{a,r}$ and imaginary part $\Delta V_O^{a,i}$, the magnitude of voltage change can be written as

$$|\Delta V_O|^2 = (\Delta V_O^{a,r})^2 + (\Delta V_O^{a,i})^2 \tag{3.32}$$

Square of Gaussian random variables follows a gamma distribution as $(\Delta V_o^{a,r})^2 \sim \Gamma(0.5, 2\sigma_r^2)$ and $(\Delta V_o^{a,i})^2 \sim \Gamma(0.5, 2\sigma_i^2)$ [89]. The shape parameter is 0.5 and scale parameter is twice the

variance of $\Delta V_O^{a,r}$ and $\Delta V_O^{a,i}$ for $(\Delta V_o^{a,r})^2$ and $(\Delta V_o^{a,i})^2$, respectively. The real and imaginary part of voltage change is correlated with $c = C_R^T \sum_{\Delta S} C_I$ as covariance. Then, the covariance between the square terms, i.e., $(\Delta V_o^{a,r})^2$ and $(\Delta V_o^{a,i})^2$ is $2c^2$. Since, the sum of the correlated gamma variable is also a gamma [90], the sum of $(\Delta V_o^{a,r})^2$ and $(\Delta V_o^{a,i})^2$ follows a Gamma distribution

$$|\Delta V_O|^2 = |\Delta V_O^{a,r}|^2 + |\Delta V_O^{a,i}|^2 \sim \Gamma(k, \theta) \tag{3.33}$$

where scale parameter $\theta = 2(\sigma_r^4 + \sigma_i^4 + 2c^2)/(\sigma_r^2 + \sigma_i^2)$ and shape parameter $k = (\sigma_r^2 + \sigma_i^2)/\theta$. The square root of $|\Delta V_O|^2$ which is a random gamma variable, follows a Nakagami distribution [91], and therefore the voltage change magnitude will have the following distribution,

$$|\Delta V_O| \sim Nakagami(m, \omega), \tag{3.34}$$

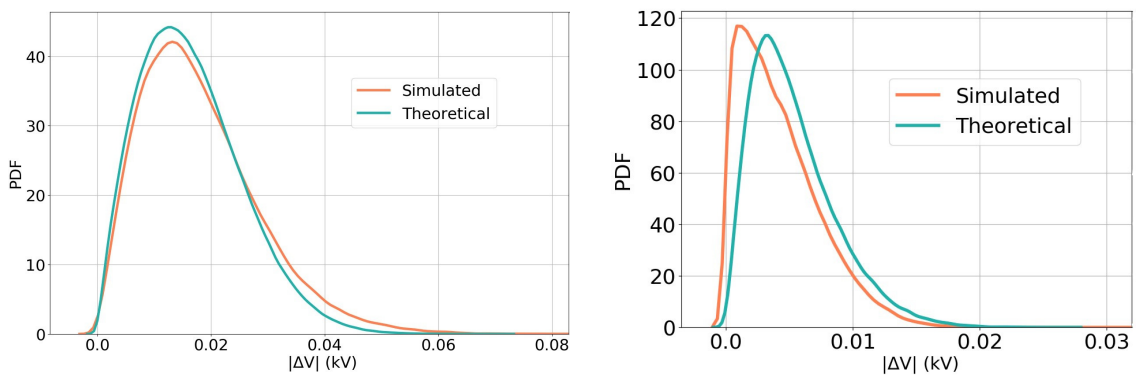
where shape parameter $m = k$, scale parameter $\omega = \sqrt{k\theta}$. □

Theorem 2 is useful in many ways. Using the equation (3.34), the vulnerability of certain observation nodes in terms of voltage violation can be identified quantitatively and efficiently. Furthermore, one can also leverage the probabilistic framework to find dominant nodes, that have maximum influence on the voltage sensitivity of critical nodes such as hospitals, schools, etc,. Later, the power at these dominant nodes can be controlled to mitigate voltage violations at the critical nodes [92]. Specifically, the vulnerability of nodes in terms of voltage violations can be evaluated by using the probability of the voltage change exceeding a certain threshold ($|\Delta V| > 0.05$ p.u.). The proposed probability distribution is validated by comparing it with the simulated distribution in the next sub-section.

3.2.1 Validation of PVSA for three phase system

To evaluate the performance of the proposed theoretical approach, we present two case studies using the same IEEE 37 node test system and IEEE 123 node test network as shown in Fig. 3.2 and Fig. 3.3, respectively. In the first case, power is varied randomly on all odd-numbered nodes, following Gaussian distribution with zero mean. The assumption of Gaussian distribution is considered as a common assumption applied in many prior works [92, 93]. The covariance matrix $\sum_{\Delta S}$ is constructed based on the correlation of power changes on various actor nodes due to their geographical proximities. Note that the proposed approach is quite general and can be applied to PV generation scenarios with different probability distributions. In the first case, power is varied randomly on all odd numbered nodes. Due to their geographical proximity, the power changes of various actor nodes may be correlated. The underlying covariance structure $\sum_{\Delta S}$ can be learned from historical or irradiance related data and its elements are set realistically based on real PV data. The base loads on the test network are the same as that reported in the IEEE PES Distribution system analysis subcommittee report. For nodes with PV's, the variance of change in real power and reactive power for any phase are set to 50 kW and 40 kvar, respectively. The variance of ΔP and ΔQ is set to zero for all non actor nodes. The off-diagonal elements of the covariance matrix capture the covariance between different actor nodes, where correlation coefficient between ΔP's for different actor nodes within the same phase is set to 0.6 and for ΔQ's, it is 0.5. Here, covariance between cross phase terms is assumed to be zero but the proposed approach is quite general to accommodate other covariance structures as well. The correlation coefficient between ΔP's and ΔQ's within the same phase is set as -0.2. For illustration purpose, the variance of all actor nodes is set to same value, but the values can vary with the nodes depending upon the size and location of PVs.

The probability distribution of voltage change at node 9 using two approaches, i.e. the proposed analytical approximation method and the traditional Newton-Raphson based VSA method are plotted in Fig. 3.6a. For computing the actual distribution of the magnitude of



(a) Node 9 of 37-node network

(b) Node 10 of 123-node network

Figure 3.6: Distribution of $|\Delta V|$

voltage change, a scenario is generated where power is varied randomly on all actor nodes using the above described covariance structure. Then, a change in voltage is computed using NR based sensitivity analysis method. On the other hand, for computing theoretical distribution, the value of vectors C_R and C_I are calculated using the network parameters. Then, the variance of real ($\Delta V_O^{a,r}$) and imaginary ($\Delta V_O^{a,i}$) part of voltage change, i.e., σ_r^2 and σ_i^2 are computed by plugging the above defined covariance matrix in equations (3.20, 3.21). Finally, the shape and scale parameter of voltage change magnitude, which is a Nakagami distribution, can be directly computed using equation (3.34). Fig. 3.6a shows the sufficiently high accuracy of the proposed theoretical method particularly the tail probabilities which is our region of interest. The Jensen-Shannon distance between the actual and theoretical distribution is 0.07 where 0 represents identical distribution and 1 denotes maximally different cases[94]. The Jensen-Shannon distance average over all the nodes of the network is 0.06. Furthermore, the PVSA formulation considering randomness is implemented on the modified IEEE 123-node test system for deriving the distribution of the magnitude of voltage change at node 10, assuming nodes 7, 11, 19, 28, 35, 52, 68 as actor nodes. The covariance matrix is developed in an identical way to a 37-node network, with the same parameters as discussed in the above paragraph. Fig. 3.6b depicts the distribution of the magnitude of voltage change computed using the proposed theoretical method and the load flow based numerical approach. High accuracy can be witnessed, as the Jensen-Shannon distance between

the resulting PDFs of two approaches is 0.18. This result demonstrates the scalability of the proposed method and its efficacy in conducting VSA for a larger heterogeneous network that includes both single and three phase loads.

To further validate the effectiveness of the proposed approach, especially in identifying the probability of voltage violations, we conduct comparison studies between the simulated and theoretical approaches with varying percentage of actor nodes in the network. The probability of voltage violation, i.e., $P(|\Delta V| > 0.05 \text{ p.u.})$ is computed from both the methods by varying number of actor nodes from 40% to 80%. The comparison results are tabulated in Table 4.3. It can be observed that the proposed theoretical method approximates the traditional simulation based method with a high mean accuracy of 94.27% for various penetrations of actor nodes, which further demonstrates the scalability of the proposed method. In addition, the complexity of the proposed analytical method is of the order $O(1)$, because the calculation of voltage change in Theorem 4 does not scale with the size of the network (n). While, the complexity in NR method is of order $O(n^3)$, as it involves the inversion operation of the Jacobian matrix.

3.3 Summary

This chapter introduces a novel approach for performance-based robustness analysis of a power distribution system. Specifically, it proposes an analytical approximation of voltage change at any node of the distribution network due to changes in complex power at different actor nodes across a three phase unbalanced distribution network. The approximation error is shown to be tightly upper bounded, illustrating the fidelity of our approach. We also derive the probability distribution of magnitude of voltage change due to random change in power at actor nodes and shown that it can be approximated by a Nakagami distribution. The proposed method can be useful for grid operation and planning as it efficiently allows us to compute the probability of voltage violation at any node in the network. In the next

chapter, this analytical voltage sensitivity analysis approach is used to determine dominant voltage influencer nodes.

Chapter 4

Performance-based Robustness Analysis: Identification of Dominant Voltage influencer nodes with PVSA

This chapter uses probabilistic method of voltage sensitivity analysis developed in Chapter 3 to propose a novel performance-based index for voltage fluctuations, i.e., voltage influence score (VIS). VIS is a node level metric that quantifies the voltage influencing capacity of an actor node on any arbitrary observation node. It can be used to rank the actor nodes and identify dominant voltage influencer (DVI) nodes. DVI nodes denote all those actor nodes that have a relatively high impact on the voltage state of observation nodes compared to the rest of the actor nodes. As a result, altering the actions of DVI nodes result in the highest reduction in voltage issues at the observation nodes. Thus, VIS allow us to identify critical nodes where additional control assets can be deployed to improve grid robustness against voltage fluctuation. Conventional methods of identifying such DVI nodes involve Monte-Carlo simulations using load flow algorithms. These classical methods possess various drawbacks including: (1) high computational complexity, (2) numerical results with no analytical insights, and (3) scenario dependent results with no generalization. These factors

limit the applicability of conventional approaches in modern distribution systems.

Therefore, this chapter uses information theoretic metrics and probability distribution of voltage changes to devise a novel voltage influencing score. VIS enable us to quickly identify DVI nodes without relying on computationally expensive Monte-Carlo simulations thereby, significantly reducing the computation time. Apart from this, there are several other use cases of VIS, including (1) control of DVI nodes in a local area, ensures voltage stability for all nodes within that cluster with minimum control actions [95]; (2) network partitioning where DVI nodes are valuable in defining local clusters for islanding and effective control; (3) DER placement for improving robustness to voltage fluctuations. Later in Chapter 8, we will investigate the efficacy of standard failure-based robustness metrics in approximating this performance-based VIS index.

4.1 Conventional approach to identify DVI nodes

The DVI nodes for an observation node are the nodes that have high impact on the voltage fluctuations at the observation node. An actor node can be a DVI due to the two factors: (1) location of the actor node, i.e. phase/bus of the distribution network, and (2) generation/load capacity of DER/loads connected at the actor node which influences the variance of power change at that node. Generally, simulation-based scenario analysis is used as a major planning tool to identify DVI nodes. A typical approach involves following steps [23]:

Step 1- Compute variance of voltage change at each phase of observation node due to all actor nodes The variance of voltage change at each phase of the observation node is computed by running multiple power flow based Monte-Carlo simulations with varying power, which captures temporal variation of generations.

Step 2- Calculate reduction in variance of voltage change at the observation node due to each actor node by setting power drawn/injected by the actor node as zero This step requires repetition of Step 1 for each actor node after setting the variance of

actor node as zero.

Step 3- Rank actor nodes based on the reduction in variance caused by the removal of the corresponding actor node: Actor nodes are ranked in an ascending order with topmost and bottommost rank assigned to those actor node that causes maximum and minimum reduction in variance of voltage change at the observation node, respectively.

This scenario-based method incurs high computational complexity, which grows with the size of the network. Specifically, the complexity arises at two hierarchical levels. At the core, we have load flow runs, whose computation using NR-based method is of complexity $O(n^3)$ with n being the size of the distribution network [27]. At a top level, multiple power change scenarios are simulated to obtain voltage change distributions in steps 1 and 2 of the approach. Each such scenario involves a load flow computation, thus one ends up running thousands of load flows. Therefore, to counter computational burden, information-theoretic approach is explored to devise a new VIS metric that can identify DVI nodes in a computationally efficient manner.

4.2 Proposed information theoretic framework

This section propose to use information-theoretic-based distance metrics as potential indicators of DVI nodes. Fundamentally, the proposed approach consists of four steps. In the first step, we obtain probability distributions of voltage change at an observation node due to each actor node as well as due to the aggregate presence of all actor nodes. These distributions are derived in a computationally efficient way by employing derived analytical approximations of chapter 3. In the second step, we compute distances between the probability distribution due to each actor node and the distribution due to the aggregate effect of all actor nodes. The Third step utilizes the distances to compute voltage influencing score (VIS) for each pair of observation and actor nodes. Finally, for each observation node, all the actor nodes are ranked based on the computed VIS. The actor node whose voltage change distribution

is nearest to the aggregate voltage change distribution is deduced as the major influencer of voltage change for that particular observation node. Likewise, all actor nodes are ranked based on the ascending order of their distances. The complete procedure is summarized in the Algorithm 1.

4.2.1 Probabilistic model of voltage fluctuations

This sub-section provides a probabilistic model of voltage change in a three-phase unbalanced distribution system, which consist of multiple spatially distributed actor nodes with PVs and active consumers. Random change in power at actor nodes due to intermittent renewable generation causes random voltage fluctuations. Therefore, probability distribution is needed to quantify voltage change under such stochastic scenarios. Here, we implement the first step of the proposed approach, i.e., derive the probability distributions of voltage change at any observation node due to random power change at a single actor node as well as due to the aggregate effect of all actor nodes. Let ΔS_A^a be change in complex power at phase a of the actor node A. Then, using eqn. (4.1), the voltage change at phase-a of an observation node O can be expressed as:

$$\Delta V_{OA}^{a} = \Delta V_{OA}^{a,r} + j\Delta V_{OA}^{a,i}, \tag{4.1}$$

$$\begin{aligned}\Delta V_{OA}^{a,r} = \sum_{h,u} \frac{-1}{|V_A^h|}[\Delta P_A^h(R_{OA}^u cos(\omega_A) - X_{OA}^u sin(\omega_A)) \\ +\Delta Q_A^h(R_{OA}^u sin(\omega_A) + X_{OA}^u cos(\omega_A))], \\ \Delta V_{OA}^{a,i} = \sum_{h,u} \frac{-1}{|V_A^h|}[\Delta P_A^h(R_{OA}^u sin(\omega_A) + X_{OA}^u cos(\omega_A)) + \\ \Delta Q_A^h(X_{OA}^u sin(\omega_A) - R_{OA}^u cos(\omega_A))]\end{aligned} \tag{4.2}$$

where $h \,\epsilon\, \tilde{H}$ and $u \,\epsilon\, \tilde{U}$. The sets $\tilde{H}$ and $\tilde{U}$ denote different phases (i.e., a, b, c) and different phase sequences (i.e., aa, ab, ac), respectively. ΔP_A^h and ΔQ_A^h are the active and reactive power changes, respectively. R_{OA}^h, X_{OA}^h are the resistance and reactance of shared path between the observation node O and actor node A from the source node. ω_A is the impedance

angle for phase-a. V_A^h denotes the base voltage of actor node A. For brevity, the derivation is shown for phase a. However, the same steps can be followed with the corresponding phase quantities to derive expressions for other phases. The real and imaginary parts of voltage change can further be simplified as,

$$\begin{aligned}\Delta V_{OA}^{a,r} = \Big[& C_{OA}^{aa,r}\Delta P_A^a + D_{OA}^{aa,r}\Delta Q_A^a + C_{OA}^{ab,r}\Delta P_b^b + \\ & D_{OA}^{ab,r}\Delta Q_b^b + C_{OA}^{ac,r}\Delta P_c^c + D_{OA}^{ac,r}\Delta Q_c^c \Big], \\ \Delta V_{OA}^{a,i} = \Big[& C_{OA}^{aa,i}\Delta P_A^a + D_{OA}^{aa,i}\Delta Q_A^a + C_{OA}^{ab,i}\Delta P_b^b + \\ & D_{OA}^{ab,i}\Delta Q_b^b + C_{OA}^{ac,i}\Delta P_c^c + D_{OA}^{ac,i}\Delta Q_c^c \Big],\end{aligned} \tag{4.3}$$

Eqn. (4.3) is further written in compact form as,

$$\begin{aligned}\boldsymbol{C_{OA}^{a,r\,T}} &= [C_{OA}^{aa,r}\ D_{OA}^{aa,r}\ C_{OA}^{ab,r}\ D_{OA}^{ab,r}\ C_{OA}^{ac,r}\ D_{OA}^{ac,r}] \\ \boldsymbol{C_{OA}^{a,i\,T}} &= [C_{OA}^{aa,i}\ D_{OA}^{aa,i}\ C_{OA}^{ab,i}\ D_{OA}^{ab,i}\ C_{OA}^{ac,i}\ D_{OA}^{ac,i}]\end{aligned} \tag{4.4}$$

The constants C and D are the functions of the line impedances and base voltages as explained in eqn. (4.4). For brevity, the exact expressions are omitted from here and are provided in the Appendix A. Then, the power change vector can be written as,

$$\boldsymbol{\Delta S_A} = [\Delta P_A^a,\ \Delta Q_A^a\ , \Delta P_A^b\ , \Delta Q_A^b\ , \Delta P_A^c\ , \Delta Q_A^c]^{\mathrm{T}}$$

Thus, because of random power changes at each actor node, the power change vector $\boldsymbol{\Delta S_A}$ is a random vector with mean vector $\boldsymbol{\mu_{\Delta S_A}}$ and covariance matrix $\sum_{\Delta S_A}$. The covariance matrix $\sum_{\Delta S_A}$ quantifies the correlation of power changes among various phases of a particular actor node A. The diagonal elements denote variances of power change at each phase and off-diagonal elements contain the correlation between the power changes. Thus, it can be

seen from (4.3) that the voltage change at an observation node O due to actor node A is the weighted combination of random vector $\boldsymbol{\Delta S_A}$, where weights (i.e., $\boldsymbol{C_{OA}^{a,r\,T}}$ and $\boldsymbol{C_{OA}^{a,i\,T}}$) are constant terms. Invoking the Lindeberg-Feller central limit theorem, it can be shown that the $\Delta V_{OA}^{a,r}$ and $\Delta V_{OA}^{a,i}$ converges in distribution to a Gaussian random variables, i.e.,

$$\begin{aligned}\Delta V_{OA}^{a,r} &\xrightarrow{D} \mathcal{N}(\mu_{OA}^{a,r} = \boldsymbol{C_{OA}^{a,r\,T}}\boldsymbol{\mu_{\Delta S_A}}, \sigma_{OA}^{a,r\,2} = \boldsymbol{C_{OA}^{a,r\,T}}\textstyle\sum_{\Delta S_A}\boldsymbol{C_{OA}^{a,r}})\\ \Delta V_{OA}^{a,i} &\xrightarrow{D} \mathcal{N}(\mu_{OA}^{a,i} = \boldsymbol{C_{OA}^{a,i\,T}}\boldsymbol{\mu_{\Delta S_A}}, \sigma_{OA}^{a,i\,2} = \boldsymbol{C_{OA}^{a,i\,T}}\textstyle\sum_{\Delta S_A}\boldsymbol{C_{OA}^{a,i}}),\end{aligned} \tag{4.5}$$

where, $\boldsymbol{\mu_{OA}^{a,r}}$ and $\boldsymbol{\mu_{OA}^{a,i}}$ are the mean vectors of real and imaginary parts of voltage change, respectively. $\sigma_{OA}^{a,r\,2}$ and $\sigma_{OA}^{a,i\,2}$ are the variances of real and imaginary parts of voltage change, respectively. For investigating the relationship between the real and imaginary parts of the voltage change, a new bivariate random vector is defined,

$$\begin{bmatrix}\Delta V_{OA}^{a,r}\\ \Delta V_{OA}^{a,i}\end{bmatrix} \sim \mathcal{N}\left[\begin{bmatrix}\mu_{OA}^{a,r}\\ \mu_{OA}^{a,i}\end{bmatrix}, \begin{bmatrix}\sigma_{OA}^{a,r\,2} & k_{OA}^{a}\\ k_{OA}^{a} & \sigma_{OA}^{a,i\,2}\end{bmatrix}\right], \tag{4.6}$$

where, $k_{OA}^{a} = \boldsymbol{C_{OA}^{a,r\,T}}\sum_{\Delta S_A}\boldsymbol{C_{OA}^{a,i}}$ is the covariance between the real and imaginary parts of the voltage change due to single actor node A. Eqn. (4.6) provides the probability distribution of voltage change at a particular observation node due to single actor node A. A similar approach can be used to compute individual voltage change distributions due to each actor node in the network. Now, we need to obtain the voltage change distribution due to the aggregate effect of all actor nodes. Using the superposition property in (4.5), the voltage change due to cumulative effect of power changes at multiple actor nodes can be expressed as [27],

where L is the number of actor nodes. By leveraging (4.5), the net voltage change can

be written as,

$$\begin{aligned}\Delta V_O^{a,r} &= \sum_{A=1}^{N} \underset{1\times 6}{\boldsymbol{C_{OA}^{a,rT}}} \underset{6\times 1}{\boldsymbol{\Delta S_A}} \\ &= [\boldsymbol{C_{O1}^{a,rT}} \boldsymbol{C_{O2}^{a,rT}} \dots \boldsymbol{C_{ON}^{a,rT}}][\boldsymbol{\Delta S_1 \Delta S_2} \dots \boldsymbol{\Delta S_N}]^T \\ &= \underset{1\times 6N}{\boldsymbol{C_O^{a,rT}}} \underset{6N\times 1}{\boldsymbol{\Delta S}}\end{aligned} \tag{4.7}$$

where $C_O^{a,rT}$ and ΔS are the long vectors, composed of a constant term and the power change vector corresponding to each actor node, respectively. Similarly, the imaginary part of voltage change can be written as, The equations (4.8) and (4.9) possess a similar form as that of (4.5), i.e., the net voltage change is the weighted combination of power change vector $\boldsymbol{\Delta S}$. Here, weight $\boldsymbol{C_O^{a,r}}$ is a constant vector comprising of line impedances and node base voltages, whereas, $\boldsymbol{\Delta S}$ comprises of power change at all phases of all actor nodes. Now, invoking the same Lindeberg-Feller central limit theorem, the real $\Delta V_O^{a,r}$ and imaginary $\Delta V_O^{a,i}$ part of aggregate voltage change can be shown to converge in distribution to a Gaussian random variables with the following parameters:

$$\begin{aligned}\Delta V_O^{a,r} &\xrightarrow{D} \mathcal{N}(\mu_O^{a,r} = \boldsymbol{C_O^{a,rT}} \boldsymbol{\mu_{\Delta S}}, \sigma_O^{a,r2} = \boldsymbol{C_O^{a,rT}} \textstyle\sum_{\Delta S} \boldsymbol{C_O^{a,r}}) \\ \Delta V_O^{a,i} &\xrightarrow{D} \mathcal{N}(\mu_O^{a,i} = \boldsymbol{C_O^{a,iT}} \boldsymbol{\mu_{\Delta S}}, \sigma_O^{a,i2} = \boldsymbol{C_O^{a,iT}} \textstyle\sum_{\Delta S} \boldsymbol{C_O^{a,i}})\end{aligned} \tag{4.8}$$

Similar to the single actor node case, the correlations between the real and imaginary parts of net voltage change is captured by defining a new random vector as:

$$\begin{bmatrix}\Delta V_O^{a,r} \\ \Delta V_O^{a,i}\end{bmatrix} \sim \mathcal{N}\left[\begin{bmatrix}\mu_O^{a,r} \\ \mu_O^{a,i}\end{bmatrix}, \begin{bmatrix}\sigma_O^{a,r2} & k_O^a \\ k_O^a & \sigma_O^{a,i2}\end{bmatrix}\right] \tag{4.9}$$

where, $k_O^a = \boldsymbol{C_O^{a,rT}} \sum_{\Delta S} \boldsymbol{C_O^{a,i}}$ is the covariance between the real and imaginary parts of net voltage change. Equations (4.8) and (4.9) provide the probability distribution due to single actor node and the aggregation of multiple actor nodes, respectively. It is worth noting that

the voltage change expressions in equation (4.8) are valid for any number of actor nodes. Further, the proposed method to determine voltage change is sufficiently general for PV generation with any arbitrary distribution. This is because the derivation of the voltage change utilizes standard statistical results that are valid for any probability distribution. Precisely, eqn. (4.8) computes voltage change in the real part of phase a ($\Delta V_O^{a,r}$) as the weighted sum of the power changes in the actor nodes. Here, even if each element of ΔS representing PV generation changes is not Gaussian, we can still invoke Lindeberg-Feller central limit theorem [4], to show that the weighted sum of the elements of ΔS converges in distribution to a Gaussian. Therefore, the distributions of the real and imaginary parts of voltage change will always converge to a normal which eventually leads to a Nakagami distribution for the magnitude of voltage change. The next sub-section focuses on computing the statistical distances between these distributions and presents the procedure to rank the actor nodes.

4.2.2 Information theoretic metrics as DVI indicators

This sub-section implements the second step of our proposed approach, i.e., calculate statistical distances between the probability distributions (derived in earlier sub-section), and rank the actor nodes based on the computed distances. The information theoretic distance metrics which are potential indicators of DVI nodes are defined first.

Kullback-Liebler distance

Kullback-Liebler (KL) distance quantifies how much one probability distribution differs from another probability distribution. KL divergence between two multivariate Gaussian distributions ($\mathcal{N}_0$ and $\mathcal{N}_1$) of dimension k with means (μ_0 and μ_1) and covariance matrices ($\boldsymbol{\sum_0}$

and $\boldsymbol{\sum_1}$) can be written as:

$$D_{KL}(\mathcal{N}_0||\mathcal{N}_1) = \frac{1}{2}\Bigg[tr(\boldsymbol{\sum_1^{-1}}\boldsymbol{\sum_0} + (\mu_1-\mu_0)^T\boldsymbol{\sum_1^{-1}}(\mu_1-\mu_0) - k + \ln\frac{|\boldsymbol{\sum_1}|}{|\boldsymbol{\sum_0}|})\Bigg] \tag{4.10}$$

where $tr(.)$ indicates trace of the matrix. Here, the KL distance between the distributions of voltage change at the observation node due to change in power at an actor node A (ΔV_{OA}) and due to change in power at all actor nodes (ΔV_O) is given by,

$$D_{KL}(\Delta V_{OA}||\Delta V_O) = \frac{1}{2}\Bigg[tr(\boldsymbol{\sum_{\Delta V_O^a}^{-1}}\boldsymbol{\sum_{\Delta V_{OA}^a}} + (\mu_O^a-\mu_{OA}^a)^T\boldsymbol{\sum_1^{-1}}(\mu_O^a-\mu_{OA}^a) - 2 + \ln\frac{|\boldsymbol{\sum_{\Delta V_O^a}}|}{|\boldsymbol{\sum_{\Delta V_{OA}^a}}|})\Bigg], \tag{4.11}$$

where $\boldsymbol{\sum_{\Delta V_{OA}^a}}$ and $\boldsymbol{\sum_{\Delta V_O^a}}$ are the covariances of ΔV_{OA} and ΔV_O, respectively.

Bhattacharyya distance

Bhattacharyya (BC) distance measures the similarity of two probability distributions. It is related to the Bhattacharyya coefficient which is a measure of the amount of overlap between two statistical samples. BC distance between the distributions of voltage change at the observation node due to change in power at an actor node A (ΔV_{OA}) and due to change in power at all actor nodes (ΔV_O) can be expressed as:

$$D_{BC}(\Delta V_{OA}||\Delta V_O) = \frac{1}{8}(\mu_O^a-\mu_{OA}^a)^T\boldsymbol{\sum}(\mu_O^a-\mu_{OA}^a) + \tfrac{1}{2}\ln(\frac{|\boldsymbol{\sum}|}{\sqrt{|\boldsymbol{\sum_{\Delta V_O^a}}||\boldsymbol{\sum_{\Delta V_{OA}^a}}|}}), \tag{4.12}$$

where $\boldsymbol{\sum} = \frac{\boldsymbol{\sum_{\Delta V_O^a}} + \boldsymbol{\sum_{\Delta V_{OA}^a}}}{2}$.

4.2.3 Voltage Influencing score (VIS)

The influence of an actor node on an observation node needs to be quantified for identifying its rank among all actor nodes. In this regard, we devise a novel index to quantify voltage

influencing capacity, i.e., Voltage influencing score (VIS). For a given network scenario, i.e., the location of actor nodes with the variance of power change from historical data, the distances between voltage change distributions at an observation node due to each actor node and due to aggregate effect of all actor nodes are used to rank the actor nodes. Here, these distances are employed to compute the VIS between any pair of observation and actor node as:

$$VIS(O, A) = \frac{\frac{1}{D(A,O)} - \frac{1}{D(S,O)}}{\frac{1}{D(A',O)} - \frac{1}{D(S,O)}}, \tag{4.13}$$

where $D(A, O)$ is the statistical distance between the voltage change distribution at an observation node O due to aggregate effect of all actor nodes and when actor node A is solely present in the system. $D(S, O)$ is the statistical distance between source node and observation node. The distance can be computed with any of the metrics described in the earlier subsection. For an observation node O, the lower the distance, the more the actor node A contributes to aggregate voltage change and consequently the more influencing the actor node is and vice-versa. Therefore, the VIS is expressed in terms of inverse of distance. To provide an absolute sense to the score, VIS is normalized with minimum and maximum values. For a particular observation node, the ideal location of actor node A' for minimum distance would be the same observation node location. On the other hand, the maximum distance location would always be the source node as it has minimum influence on voltage fluctuations of any observation node. These minimum and maximum distances are used to normalize VIS as shown in eqn. (4.13).

In the final step of the proposed approach, VIS is leveraged for ranking the actor nodes. The nodes are ranked in an ascending order based on VIS. The topmost (Rank 1) and bottom-most (Rank L for the case of L actor nodes) ranks will be assigned to the actor

nodes in the following way:

$$\begin{aligned} \text{Rank } 1 &: \arg\max_{A} VIS(O, A) \\ \text{Rank } L &: \arg\min_{A} VIS(O, A) \end{aligned} \tag{4.14}$$

It is worth to note that the proposed approach can work with any other information theoretic metric (Frechet distance, Jensen-Shannon distance, etc.), and for illustration purposes, the method is evaluated with two metrics (KL divergence & Bhattacharyya distance) in the results section.

4.3 Experimental Results and Discussion

The efficacy of the proposed method in identifying dominant voltage influencer nodes is evaluated in this section. The baseline method is the conventional Monte-Carlo simulation-based approach, and both algorithms are implemented in the modified IEEE 37-node test system as depicted in chapter 3. This test network is chosen as it denotes a typical unbalanced distribution system and is used by many researchers for illustrating the efficiency of their methods [26, 27]. The nominal voltage of the test system is 4.8 kV. A scenario is generated with 15 actor nodes distributed randomly in the IEEE 37-node network. Change in real and reactive power at 15 actor nodes is modeled as zero-mean Gaussian random vector. It is worth noting that the proposed approach is generic for any choice of actor node and power change distributions, and the simulated case-studies are merely a way to illustrate its performance. Three different PV sizes are considered in this case study. The mean and variance of real and reactive power change of all three kinds of PV capacities along with

their location and phase information are as follows:

$$\begin{aligned}
\boldsymbol{\Delta S_A} &\sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1.5 & -0.05 \\ -0.05 & 0.25 \end{bmatrix} \right), \quad A \,\epsilon\, \{7^c, 8^a, 12^c, 27^b, 28^c\} \\
\boldsymbol{\Delta S_A} &\sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 3 & -0.1 \\ -0.1 & 0.5 \end{bmatrix} \right), \quad A \,\epsilon\, \{7^b, 18^a, 22^c, 34^b, 36^b\} \\
\boldsymbol{\Delta S_A} &\sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 4.5 & -0.2 \\ -0.2 & 0.75 \end{bmatrix} \right), \quad A \,\epsilon\, \{9^c, 14^c, 26^c, 30^a, 31^a\}
\end{aligned} \tag{4.15}$$

where superscript over actor nodes, i.e., $\{a, b, c\}$ represent respective phases of actor nodes at which power is varying. The change in power across different actor nodes can be correlated because of environmental factors. The DERs such as PVs and wind turbines are expected to exhibit a similar generation profile due to their geographical proximity. Further, the real and reactive power of inverter-based DERs is negatively correlated. The underlying covariance structure $\sum_{\Delta S}$ can be estimated based on historical or irradiance-related data. The base loads on the test network are the same as mentioned in the IEEE PES Distribution system analysis subcommittee report. For actor nodes, the variance of change in real power and reactive power are present in the diagonal elements of covariance matrices as shown in eqn. (4.15). Thus, three approximate PV sizes which are simulated in our experiments are 45 kW, 38 kW and 35 kW. The off-diagonal elements of the covariance matrices capture the covariance between real and reactive power. These covariance of each actor nodes are combined to form the covariance matrix of power change vector $\sum_{\Delta S}$ which corresponds to all actor nodes. Here, the correlation coefficient between ΔP's and ΔQ's for different actor nodes within the same phase is kept same and covariance between cross-phase terms is assumed to be zero. However, the proposed approach is quite general to accommodate other covariance structures as well. The variance of nodes other than the actor nodes is set to zero.

Table 4.1: Observation node 7

Node	MC	KL	BC
7	1	1	1
9	0.577	0.163	0.129
12	0.543	0.150	0.117
22	0.530	0.127	0.096
14	0.516	0.137	0.106
26	0.441	0.092	0.067
28	0.350	0.084	0.060
8	0.042	0.065	0.045
17	0.026	0.050	0.033
27	0.001	0.043	0.029

Table 4.2: Observation node 16

Node	MC	KL	BC
14	1	1	1
22	0.919	0.840	0.808
18	0.782	0.170	0.133
12	0.613	0.576	0.533
9	0.426	0.422	0.373
7	0.209	0.290	0.242
17	0.146	0.230	0.186
8	0.020	0.111	0.083
26	0.001	0.176	0.138
28	0.001	0.164	0.128

4.3.1 VIS for ranking DVI nodes

To assess the performance of the proposed approach in identifying DVI nodes, VIS for two arbitrary observation nodes (i.e., 7, 16) are computed using two different distance metrics namely KL divergence (eqn. (4.11)) and Bhattacharyya distance (eqn. (4.12)). The results obtained by using the proposed approach is validated against ground truth values. The baseline approach to compute true DVI nodes is based on Monte-Carlo simulations (MCS) of load flow as explained in Section 2A. Initially, the variance of voltage change magnitude at a given observation node is computed due to the presence of all actor nodes. Multiple power change scenarios are simulated by running 100000 MCS with load flow method. Then, the reduction in the variance of the magnitude of voltage change is determined by setting the variance to zero for each actor node sequentially. Again, for each actor node case, different MCS are executed. Finally, the actor nodes are ranked based on the reduction in variance they bought when the variance of power change at the corresponding actor node is set to zero.

Tables 4.5 and 4.6 tabulates the VIS for observation nodes 7 and 16, respectively. MC, KL and BC refers to baseline MCS approach, proposed KL divergence and Bhattacharyya distance based methods, respectively. Node in the tables refers to actor node. It can be inferred that the VIS for the most dominant actor node is 1 and it decreases as we move to

lower rank nodes. For observation node-7 with KL distance metric, there is a small difference in VIS between nodes 22 and 14 implying that they exhibit almost equal voltage influencing capacity. However, there is a considerable difference when it comes to nodes 14 and 26. Thus, VIS allows us to quantitatively differentiate between dominant influencer nodes.

Table 4.3 tabulate the top 10 dominant voltage influencer nodes for observation nodes 7 and 16, respectively. According to Table 4.3, the MC method indicate that the actor nodes

Table 4.3: DVI nodes for observation nodes 7 and 16

Rank → Metric ↓	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Observation node -7										
MC	7	9	12	22	14	26	28	8	17	27
KL	7	9	12	14	22	26	28	8	17	18
BC	7	9	12	14	22	26	28	8	17	18
Observation node -16										
MC	14	22	18	12	9	7	17	8	26	28
KL	14	22	12	9	7	17	26	18	28	8
BC	14	22	12	9	7	17	26	18	28	8

7 and 14 are Rank-1 nodes for observation node 7 and 16, respectively. The assignment of Rank-1 nodes is carried out correctly by both the metrics of the proposed approach. The reasons for rank-1 allocation are (1) the high variance of power change and (2) proximity of actor nodes to the given observation node in the IEEE 37-node test system. Apart from rank-1, utilities might be interested in identifying other dominant nodes which are next to rank-1. In this regard, we use Top-N accuracy, which can be defined as,

$$\frac{|\{\text{Predicted Top-N nodes}\} \cap \{\text{True Top-N nodes}\}|}{N}, \tag{4.16}$$

where N is the desired number based on the applications. The Top-5 and Top-10 accuracy results are presented for observation nodes 7 and 16 in Table 4.4. The last column of the table denotes the mean accuracy when all the nodes of the IEEE-37 node network act as observation nodes. It can be observed that distance metrics KL and BC have fairly good identification accuracy, and the mean accuracy is more than 90%. Apart from Rank-1 node,

our approach also provides the correct sequence of actor nodes when compared to baseline MC approach. However, in some positions, the order of actor nodes is flipped or offset by one or two units. This is because, the corresponding actor nodes have an equal influencing capacity for that particular observation node. For instance, in the case of observation node 7, the position of actor nodes 22 and 14 are flipped. However, this is not a major concern here, as we are more interested in correctly identifying nodes lying in a particular band (i.e., Top-5, Top-10) rather than their exact order. The power at the Top-5 or Top-10 actor nodes can be efficiently regulated to mitigate voltage violations.

Table 4.4: Identification accuracy of DVI nodes

Node → **Accuracy ↓**	**obs. node-7 (%)**	**obs. node-16 (%)**	**Mean all nodes (%)**
Top-5	100	80	91
Top-10	90	100	93

Furthermore, to get the overall influencing capacity of actor nodes, one can compute the mean VIS. For each actor node, the mean of VIS is taken across all observation nodes of the network. It can be observed From Table **??** that actor nodes 17 and 30 have maximum and minimum influencing capacity among all the actor nodes present in the network. This assignment is because of their topological position in the distribution network and the magnitude of power variance due to associated DER units or load variation. Moreover, apart from correctly identifying the dominant voltage influencing nodes, the proposed approach offers a considerable computational advantage over the conventional approaches. Table **??** reports the execution time of various approaches to identify Top-5 actor nodes for the observation node 7. It can be seen that the proposed approach involving any of the two distance metrics is multiple order faster than that of the conventional approach, which takes around 3.2 hrs.

Table 4.5: Average VIS of actor nodes

Nodes	VIS	Nodes	VIS	Nodes	VIS
7	0.220	17	0.444	28	0.156
8	0.220	18	0.304	30	0.154
9	0.212	22	0.349	31	0.189
12	0.258	26	0.163	34	0.378
14	0.300	27	0.166	36	0.427

Table 4.6: Running time of various approaches

Time Metric	Running time Top-5 (s)
MC	12039
KL	0.92
BC	0.96

4.4 Summary

This chapter highlights an application of the performance-based robustness analysis approach introduced in Chapter 3. Specifically, statistical distance metrics are used to derive a novel voltage influence score (VIS) that quantifies voltage influencing capacity of actor nodes in a power distribution grid. The VIS is then leveraged to identify the DVI nodes. The proposed framework computes VIS not solely on the basis of correlation between change in power at the actor node and change in voltage at the observation node, but it also relies on the magnitude of voltage variation that is being caused due to power change at actor nodes. Effectiveness and computational efficiency of the proposed method are illustrated by comparing the results with conventional method of identifying DVI nodes in multiple test networks. Results demonstrates that the proposed metric effectively predict the DVI nodes while substantially reducing the execution time. Thus, DVI nodes can serve as an optimal control locations for improving system resilience against detrimental voltage fluctuations.

Chapter 5

Network-based robustness analysis: Identification of critical nodes with Graph Neural Network

Until now, Chapters (3 and 4) have focused on developing a computationally efficient framework for a performance-based robustness metric in a power distribution grid. This chapter introduces a novel methodology for analyzing network-based robustness metrics. In particular, it focuses on the important task of identifying critical nodes in a large complex system, where nodes represent some facility or sub-system. In interdependent systems such as smart grid or smart internet, the proper functioning of constituent sub-systems (nodes) and their interconnections (links) are responsible for the reliable operation of the overall system. However, typically there exists a set of *critical nodes/links* that play a more crucial role in determining the output of the system than other (non-critical) nodes/links. These nodes/links represent a set of sub-systems and/or their interconnections, whose removal from the graph maximally disconnects the network, and thus severely disrupts the operation of the system. As a result, identification of critical nodes/links is an important task for analysis and/or design of the underlying systems, and bears significance in several applications in-

cluding social networks analysis, feature expression in biological networks, risk management in telecommunication networks, among others.

Critical nodes/links are typically identified by computing its associated criticality score. Owing to the inherent topological structure of graphs, each node/link contributes differently to the graph robustness, and hence its removal/loss affects the robustness to a different degree. In this regard, the notion of node/link criticality score is introduced, which quantifies the decrease in robustness when the corresponding node/link is removed from the graph. Several methods have been proposed to compute criticality scores based on graph robustness [52, 53]. However, existing approaches suffer from various shortcomings, including high computational cost due to iterative nature of the algorithm and non-scalable to larger networks.

Therefore, this chapter frames the task of identifying critical nodes as an inductive graph machine learning problem. Specifically, it's a graph regression problem where criticality scores are node targets. A graph neural network (GNN) is trained to predict the criticality scores, which are then employed to rank the nodes/links. The topmost rank corresponds to the node/link whose removal maximally decreases the graph robustness and vice-versa. Furthermore, these predictions could be uncertain either due to an uncertain input graph or uncertain GNN parameters. These uncertainties are systematically handled in chapter 6 to further improve the efficacy of the robustness evaluation process.

5.1 Metrics of Graph Robustness

The robustness of a system is its ability to function properly in the presence of disturbances/perturbations, such as failures of components. In a graph theoretic setting, this translates to the ability of the graph to function with loss of nodes/links. Several metrics have been proposed in the literature [96] that quantify the robustness of a graph against random and targeted loss of nodes/links. Effective graph resistance (R_g) is one such widely

used metric to quantify graph robustness [53, 54, 97] and is equal to the sum of the effective resistances over all pairs of nodes [98]. The effective resistance between any two nodes of a graph is computed by performing series and parallel operations on an electrical circuit-equivalent of the graph. R_g considers both the number of paths between nodes and their length (link weight), intuitively measuring the presence and quality of backup possibilities in the graph. The spectral form of R_g can be expressed as:

$$R_g = \frac{2}{N-1} \sum_{i=1}^{N-c} \frac{1}{\lambda_i}, \tag{5.1}$$

where, λ_i, $i = 1, 2, 3, ...N$ are the eigen values of the Laplacian matrix of a graph G with N nodes, and c is the number of connected components in the graph.

Weighted spectrum (W_s) is also a widely used metric for graph robustness [99, 100] and is defined as the normalized sum of n-cycles in a graph [99]. An n-cycle in a graph G is defined to be a sequence of nodes $u_1, u_2, ...u_n$ where u_i is adjacent to u_{i+1} for $i \in [1, n-1]$ and u_n is adjacent to u_1. The weighted spectrum can be expressed as,

$$W_s = \sum_i (1 - \lambda_i)^n, \tag{5.2}$$

where, different values of n correspond to different graph properties. For instance, $n = 3$ denotes the number of triangles in a graph with W_s related to the weighted clustering coefficient. Similarly, with $n = 4$, W_s is proportional to the number of disjoint paths in a graph. Since this work focuses on quantifying the robustness of the graph based on connectivity, $n = 4$ is used in this study. Thus, we first compute the eigen values of the normalized graph Laplacian matrix. Then, we sum the difference of eigenvalues from 1 raised to the power 4 to get the final W_s metric. It must be noted here that although this work uses R_g and W_g as metrics of robustness, the proposed approach is generic enough to be applied for any robustness metric.

5.2 Graph Neural Networks

Graph neural networks are a variant of artificial neural networks that are designed to capture patterns in data that can be represented in a graphical structure. The authors in [101] initially introduced the transformation of the convolutional operations from Euclidean domain to problems involving graphs. The working principle of such models resembles that of convolutional neural networks and can work directly on graphs and exploit their topological information. The standard learning tasks on graph data are node classification, link prediction, graph classification, etc. GNN addresses various learning tasks across domains including computer vision [102, 103], natural language processing [104, 105], bio-chemistry [106], etc. A GNN typically involves learning features called node embedding vectors followed by feedforward layers for regression or classification tasks. However, the algorithm proposed in [101] depends on the size of the graph leading to scalability issues. To address this scalability issue, the authors in [107] propose an inductive learning framework called GraphSAGE, where node embeddings are learned using subgraphs and thus, is independent of graph size. Furthermore, this framework can be leveraged to infer about the unseen/new nodes of graphs belonging to the same family. The standard procedure to learn this embedding vector involves a "message passing mechanism", where the information (node feature) is aggregated from the neighbors of a node and then combined with its own feature to generate a new feature vector. This process is repeated to generate the final embedding for each node of a graph. The GraphSAGE algorithm learns the mapping (aggregator) function instead of learning the embedding vectors. Hence, it can induce the embedding of a new node during training, given its features and neighborhood. The proposed framework makes use of GraphSAGE as it is a state-of-the-art GNN modeling framework and is applicable to graphs of different sizes.

5.3 Proposed ILGR Framework

This paper focuses on identifying critical nodes/links in large graphs. Critical node/link identification is a combinatorial optimization problem introduced by [108]. In the following, we present the problem formulation and algorithmic details of the proposed framework.

5.3.1 Problem formulation

Consider a graph $G = (V, E)$, represented as a tuple of nodes V and links E. The problem of critical node identification is to identify a set of nodes $V_c \subseteq V$, whose deletion from the graph results in a maximal decrease in network functionality measured in terms of a graph robustness metric. Similarly, the problem of link identification is to identify a set of links $E_c \subseteq E$, whose deletion from the graph results in a maximal decrease in the graph robustness metric. It is important to note that in evaluating the decrease in robustness, we remove a single node/link at a time. Assigning criticality based on a sequence of node/link removal is beyond the scope of this article and is kept as a future work.

The above problem is addressed by incorporating local neighborhood information for nodes/links. This approach assumes that there exists a functional relation/mapping that links the local topology and features with the criticality of a node. The approach then makes use of a graph neural network framework, specifically, GraphSAGE to learn this mapping from data via training. Consider a node u, whose criticality score we wish to learn. The first step involves generating the feature vector of the node u (h_u) as well as of the neighboring nodes i.e., $h_v \ \forall v \in N(u)$ where $N(u)$ represents the neighborhood of node u. According to the assumption of the approach, the underlying mapping between these features and the criticality score r_u can be expressed as:

$$r_u = f(h_u, h_v) \qquad v \in N(u) \tag{5.3}$$

The graph neural network framework makes use of the network topology and the feature vec-

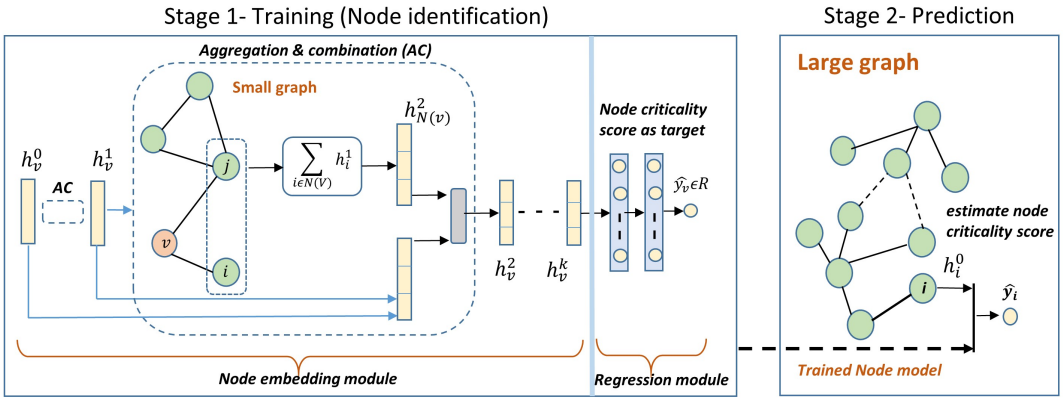


Figure 5.1: Proposed ILGR framework for node identification

tors to learn the underlying mapping for different graphs, and generates an approximation of this mapping $\hat{f}(h_u, h_v, N(u))$. In this regard, this article proposes a two-step approach for inductive learning-based approximation of the criticality scores of nodes/links in a graph, referred to as Inductive Learner for Graph Robustness (ILGR). The framework is first introduced for the identification of critical nodes, and is then extended to link identification problem. The first step involves the use of computationally manageable graphs to learn appropriate node embeddings and subsequently criticality scores with a GNN that allows faster learning. This is followed by the prediction of criticality scores for new nodes or unseen nodes (nodes not used for training) of the graph. It must be noted, however, that in order for the neural network to reliably predict the scores at the time of deployment, the training and testing data (graph) should exhibit some similar properties. In this regard, one can use standard (synthetic) graphs such as power-law graphs and power-law cluster graphs that are known to exhibit properties similar to most real-world networks. However, in certain applications where such synthetic networks with similar properties cannot be obtained, it is possible to use a subset of the nodes of the testing graph and adopt transfer learning to tune the parameters of an already trained neural network model.

The proposed framework is illustrated in Figure 5.1. The GNN model first learns *node embeddings* from node features and neighborhood sub-graph, which are then used to calculate

the criticality scores. The following sections discuss in detail, the steps involved in obtaining the node criticality scores from a set of features and neighborhood.

5.3.2 Node embedding module

The first module of ILGR learns the embedding vector for each node by utilizing the graph structure and node target (criticality) scores. This is achieved in a manner such that nodes that are close in the graph space also lie close in the embedding space. As an initialization, the embedding of each node is composed of only the degree of the node, followed by a predefined number of ones. The node embeddings are learned based on GraphSAGE which is described briefly in the previous section. However, there are some modifications that have been made in the implementation. The node embedding module is further subdivided into two tasks. The first task learns a representation for every node based on a combination of the representation of its neighboring nodes, parametrized by a quantity K, which quantifies the size of the neighborhood of nodes. Specifically, the parameter K controls the number of hops to be considered in the neighborhood. For instance, if $K = 2$, then all the nodes which are 2 hops away from the selected node will be considered as neighbors. This defines the neighborhood of a node v as:

$$N(v) = \{u : \mathrm{D}(u, v) \leq K, \forall u \in G\} \tag{5.4}$$

where $\mathrm{D}(u, v)$ is a function that returns the the smallest distance between nodes u and v. After defining the neighborhood, an aggregator function is employed to associate weights to each neighbor's embedding and create a neighborhood embedding for a selected node. Unlike previous works [107, 109], where weights are pre-defined, this work uses the attention mechanism to automatically learn the weights corresponding to each neighbor node as [110]:

$$h^l_{N(v)} = \mathrm{Attention}(Q^l h^{l-1}_k) \forall\ k \epsilon N(v) \tag{5.5}$$

where, $h^l_{N(v)}$ represents the embedding of neighbourhood of a node v in layer l of the GNN, h^{l-1}_k represents the embedding of k^{th} neighboring node of v in the $l-1$ layer of the GNN. Q^l is the attention weight at layer l of the GNN. Thereafter, for each neighborhood depth until $k = K$, a neighborhood embedding is generated with the aggregator function for each node and concatenated with the existing embedding for node v. However, the existing node embedding is not solely the output from the embedding of previous layer as implemented in various previous works. Rather, for existing node embedding, this article proposes to use the output of node embeddings from the previous two layers. This is similar to skip connections used by various researchers in the past for enhancing model performance for images and speech-related applications [111, 112], and can be expressed as:

$$h^l_v = \text{Relu}(W^l[h^{l-1}_v || h^{l-2}_v || h^l_{N(v)}]) \forall\, k \epsilon N(v) \tag{5.6}$$

where, h^l_v represents the embedding of node v in layer l of the GNN, h^{l-1}_v and h^{l-2}_v denote the embedding of node v in layers $l-1$ and $l-2$ respectively and $h^l_{N(v)}$ is the embedding aggregated from neighbors of v as given in equation (5.5). The operations performed by equations (5) and (6) are known as aggregation and combination, respectively. This aggregation and combination process is repeated for all layers of the model to obtain final node embeddings. The steps involved in the embedding module is summarized in Algorithm 1. The symbol || in step 5 of the Algorithm 1 denotes the concatenation operation.

5.3.3 Regression module

The output of the embedding module is passed through a regression module which is composed of multiple feedforward layers. The feedforward layers transform the embedding nonlinearly and finally generate a scalar that denotes the node criticality score. The output of

Algorithm 1 ILGR embedding module

Input: Graph G, input node features $X_v \forall\, v \epsilon V$, unknown model weights W(combination weights) and Q(aggregation weights).

Output: Nodes embedding vector $z_v \;\forall\, v \epsilon V$.

Initialize: $h_v^0 = X_v \;\forall\, v \epsilon V$
for layer $l = 1$ to $l = L$ **do**
  for node $v = 1$ to $v = V$ **do**
    $h_{N(v)}^l = \text{Attention}(Q^l h_k^{l-1}) \;\forall\, k \epsilon N(v)$;
    $h_v^l = \text{Relu}(W^l [h_v^{l-1} || h_v^{l-2} || h_{N(v)}^l]) \;\forall\, v \epsilon V$;
  end for
end for
return Final embedding vector $z_v = h_v^L \;\forall\, v \epsilon V$;

the m^{th} layer in the regression module can be expressed as:

$$y_v^m = f(W^m y_v^{m-1} + b^m) \tag{5.7}$$

where W^m and b^m represent the weights and biases in the m^{th} layer, f is the activation function such as ReLU, Softmax, etc. y_v^m is the output of m^{th} layer corresponding to node v and $y_v^0 = z_v$ (output of embedding module). The complete framework for node identification is depicted via Fig. 5.1.

The framework for link criticality scores is almost similar to that of the node analysis with small differences. The output of the node embedding module is connected to a link embedding layer which generates link embedding from the associated pair of node embeddings. There are various ways to combine node embeddings including, concatenation, inner product, mean, L2 norm, etc. In this work, the Hadamard product is used to generate link embedding. Thereafter, the link embedding is passed through the regression module to predict link criticality score. The entire framework of ILGR for link prediction is shown in Fig. 5.2. It is worth noting that the benefit of posing the node/link identification task as a regression problem is that once the criticality scores are predicted for nodes/links, then the user can select top-N % of the nodes/links based on the requirement. Hence, the framework is more flexible and does not depend on a specific threshold for determining criticality.

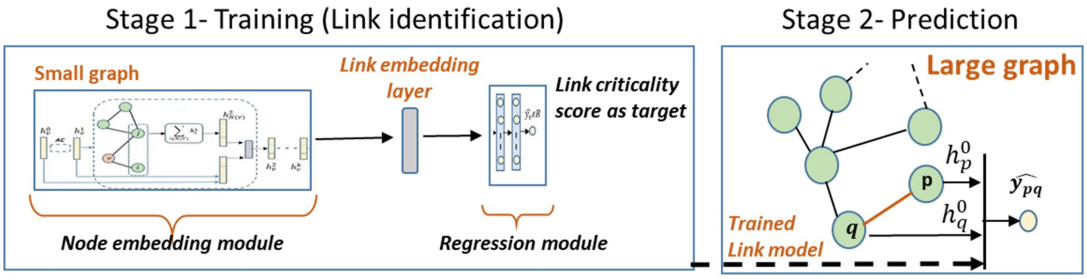


Figure 5.2: Proposed ILGR framework for link identification

The algorithm learns the weights of an aggregator function and feedforward layers by minimizing an appropriate loss function. In the proposed framework, the output of the model is node/link criticality score, which is further employed to rank nodes/links and identify the critical ones. So, instead of exactly learning the criticality values, it is sufficient to learn any real values provided the relative order of nodes remains intact. A suitable loss function for this scenario is the ranking loss [113]. Unlike other loss functions, such as Cross-Entropy Loss or Mean Squared Error Loss, whose objective is to learn to predict a value or a set of values given an input, the objective of ranking loss is to preserve the relative distances between the inputs. Here, the pair-wise ranking loss has been used that looks at a pair of node ranks at a time. The goal of training the model is to minimize the number of inversions in the ranking, i.e., cases where a pair of node ranks is in the wrong order relative to the ground truth. This loss function can be expressed as:

$$L_{ij} = -f(r_{ij}) \log \sigma(\hat{y}_{ij}) - (1 - f(r_{ij})) \log(1 - \sigma(\hat{y}_{ij})) \tag{5.8}$$

where, r_i is the ground truth value of criticality score for node i. $r_{ij} = r_i - r_j$ is the actual rank order, which the model is learning to infer through $\hat{y}_{ij} = \hat{y}_i - \hat{y}_j$ by minimizing the loss L_{ij}. f is a sigmoid function. The loss is aggregated for all the training node/link pairs, and

then optimized to update the model weights. The net training loss can be written as,

$$Loss = \sum_{(i,j) \subset E} L_{ij}$$

where (i, j) denotes an edge pair belonging to graph link set E. Once weights are learned, then an embedding vector and consequently the node/link scores can be predicted for a test node/link given its features and neighboring information.

Algorithm 2 Algorithm of ILGR

Input: Model with unknown weights.
Output: Trained model.
1: Generate ground truth criticality scores of nodes/links based on graph robustness score
2: **for** each iteration **do**
3: Get each node embedding from embedding module.
4: Estimate criticality score of nodes/links through regression module.
5: Update weights of both modules by solving eqn. (7.5)
6: **end for**
7: Predict node/link score on test graph.
8: **return** Top $N\%$ of most critical nodes/links.

5.3.4 Model settings and Training

There are various hyper-parameters in the model that need to be tuned for training the model. The number of node embedding layers, i.e., the depth of the GNN is selected as three. The number of neurons in these layers are 64, 32, and 16 respectively. The regression module consists of three feedforward layers with 12, 8 and 1 neurons respectively. The activation function in all the layers is kept as *relu*. The training of both the embedding and regression modules is conducted end to end with input being a specific node/link along with its neighbor information and output being the corresponding criticality score.

A different model is trained for each family of synthetic graphs. This is because, different families of graphs vary in their overall structure and link connections, i.e., degree distributions, assortativity, average clustering coefficient, etc. For a particular graph family, different

random instances of graph are sampled, and then the ground truth of node/link criticality scores are computed for each sampled graph using a conventional approach as described in Algorithm 3. Fundamentally, it involves an iterative method of removing a node/link from the graph and computing the robustness metric of the residual graph. The term "residual graph" refers to the leftover graph after the removal of node/link. This process is repeated for all the nodes/links of the graph. Thereafter, the nodes/links are ranked based on the computed criticality scores. The ground truth criticality scores are then used to train our GNN model on multiple graphs of the same family. The training steps are outlined in the Algorithm 2. At each iteration of the algorithm, we first compute the embedding vector for nodes via the embedding module. Embedding module primarily takes node features and neighborhood information as input and outputs embedding vectors by passing information across different layers of the graph neural network. Next, the regression module uses these embedding vectors to estimate node/link criticality scores. The trained model is validated to predict criticality score on graphs of higher dimensions. For example, a model can be trained on multiple power-law graphs of dimension $100 - 1000$, and can be evaluated on power-law graphs of dimension $100 - 100000$. The model is trained end to end on the Tensorflow framework with Stellargraph library. For real-world graphs such as US Power grid [114, 115], Wiki-vote, already trained models are employed for predicting node ranks. Additionally, transfer learning is also implemented for a real-world network to demonstrate the efficacy of the framework. The experimental setup and results are shown in the next section.

5.4 Experimental Results

This section discusses results obtained with the proposed framework for node and link criticality score prediction over a wide range of applications. The datasets used in this work to demonstrate the applicability of the proposed framework are first discussed, followed by the evaluation metrics used to report the performance of the framework. A baseline

approach is used to compare the performance of the proposed framework with existing approaches. Codes related to this work can be found at https://github.com/saimunikoti/GraphNeuralNetwork-Resilience-ComplexNetworks

5.4.1 Datasets

We examine the performance of ILGR on both synthetic and real world graphs. The two commonly used synthetic graphs are generated using Python NetworkX library are as follows:

1. **Power law**: Graphs whose degree distribution follow power law (i.e., heavy tailed), and many real networks have shown to be of this family [116]. It is generated through a process of preferential attachment in which probability that a new node N_y connects with an existing node N_x is proportional to fraction of links connected to N_x.

2. **Power law cluster**: Graphs which exhibit both power law degree distribution and clusters, and many real-world networks manifest these properties [117]. As shown in [118], one can construct a PLC graph by following a process of preferential attachment but in some fraction of cases (p), a new node N_y connects to a random selection of the neighbors of the node to which N_y last connected.

The real-world networks that are analyzed in this work are as follows [119]:

1. **Bio-yeast**: It is a protein-protein interaction network for yeast consisting of 1458 nodes and 1948 links. [117].

2. **US-powergrid**: It represents the western US power grid with 4941 nodes representing buses and 6594 links as transmission lines [115].

3. **Wiki-vote**: It contains all the Wikipedia voting data from the inception of Wikipedia till January 2008. 7115 Nodes in the network represent wikipedia users with 103689 links, where each link from node i to node j indicates that user i voted a user j [114].

4. **cit-DBLP**: It is the citation network of DBLP, a database of scientific publications. There are 12591 nodes and 49743 links. Each node in the network is a publication, and each edge represents a citation of a publication by another publication [115].

5.4.2 Evaluation metrics

The trained model predicts the criticality scores of the test nodes/links in a graph, which are then used to identify the most critical nodes/links. In a general setting of robustness analysis, it is more relevant to identify top ranked nodes/links that are most critical, rather than knowing the ranks of all the nodes/links. Therefore, we have used Top-N% accuracy to evaluate the proposed framework against the conventional approach. It is defined as the percentage of overlap between the Top-N% nodes/links as predicted by the proposed method and the Top-N% nodes/links as identified by conventional baseline approach, i.e., Algorithm 3. Top-N% accuracy can be expressed as,

$$\frac{|\{\text{Predicted Top-N\% nodes/links}\} \cap \{\text{True Top-N\% nodes/links}\}|}{|V| \times (N/100)}, \tag{5.9}$$

where, $|V|$ is the number of nodes/links and N is the desired band. In this work, the results are reported for Top-5% accuracy. Further, the computational efficiency of the proposed approach is demonstrated in terms of execution time. More specifically, the execution time is same as wall-clock running time, i.e., actual time the computer takes to process a program.

5.4.3 Baseline approaches

To evaluate the performance of our proposed approach, we compare ILGR with a conventional method of estimating node/link criticality. The classical methodology involves an iterative method of removing a node/link from the graph and computing the robustness metric of the residual graph. The term "residual graph" referred to a leftover graph after the removal of node/link. This process repeats for all the nodes/links of the graph. There-

Algorithm 3 Conventional approach of identifying critical nodes/links on the basis of graph robustness

Input: Graph G with V nodes.
Output: Node/link critical scores

```
for n in V do
    Remove node/link n from graph G
    Compute robustness metric of the residual graph (G − n)
    Assign criticality scores to node/link n
end for
Rank nodes/links on the basis of above computed criticality scores. Top ranks correspond to more critical nodes/links.
return Top N% of most critical nodes/links.
```

after, the computed criticality scores of all the nodes/links are arranged to generate ranks and identify the most critical ones whose removal maximally decreases the graph robustness. Algorithm 3 summarizes the typical conventional approach to identify critical nodes based on graph robustness metrics.

Furthermore, we have also implemented a Graph convolutional algorithm (GCN) which is used in many recent works related to critical node/link identification problem [58, 59, 101]. Mathematically, GCN can be written as:

$$h^{l+1} = \sigma(\tilde{D}^{-0.5}\tilde{A}\tilde{D}^{0.5}h^{l}W^{l}) \tag{5.10}$$

where, h^l denotes the l^{th} layer of the neural network, σ is the non-linearity function (e.g., ReLU), and W is the weight matrix for this layer. $\tilde{D}$ and $\tilde{A}$ are the degree and adjacency matrices for the graph with the superscript referring to the additional link between each node and itself. The model settings (number of layers, node samples, activation functions, etc.) of GCN is kept identical to our model for fair comparison. From equation (5.10), it can be inferred that GCN is dependent on the input dimension (i.e., adjacency matrix of input graph $\tilde{A}$) for prediction. Thus, a model can only be trained on graphs of identical node dimension and consequently can solely be used for node/link identification in graphs of similar dimension. This reduces the scalability and generalizability of the framework which

is collectively addressed in our approach.

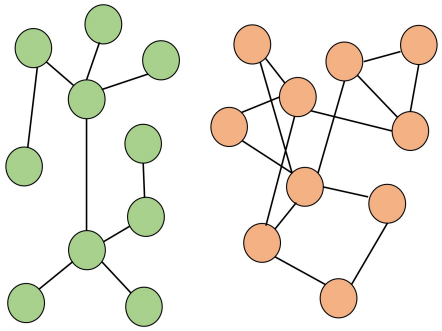


Figure 5.3: Left: Power law; Right: Power law cluster graph

5.4.4 Results and Discussion for node identification

This subsection access the performance of the proposed ILGR framework for nodes, in terms of Top-5% accuracy and algorithm execution time. Different types of a graph with varying dimensions are considered for assessment. Here, we study two different types of synthetic graphs, i.e., power-law and power-law cluster. Further, with respect to the robustness index, we analyze two different metrics namely effective graph resistance and weighted spectrum, which are described in the earlier section. Therefore, for each type of graph, the proposed method is evaluated for two different metrics leading to four different scenarios. Table 5.1 tabulates the Top-5% accuracy of each such scenario. The models are trained on 30 graphs of dimension varying from 100 to 1000 nodes. The trained model is then employed to predict node criticality scores in power-law and power-law cluster graphs of higher dimensions, i.e., 500, 1000, 5000, and 10000 nodes.

It can be inferred from the Table 5.1 that in the process of identifying Top-5% of the most critical nodes, the mean accuracy of the PL model for robustness metrics R_g and W_s is 94.8% and 95.6%, respectively. The mean is taken across graphs of different node counts. Similarly, the accuracy of PLC model for R_g and W_s is 93.3% and 94.5%, respectively. The scalability of the framework is depicted via the model's high performance in graphs of

Table 5.1: Accuracy of node identification task in synthetic graphs.
PL: Power-law; PLC-Power-law cluster; Ntest: # test nodes

Test graph size	Ntest 500		Ntest 1000		Ntest 5000		Ntest 10000	
graph/scores	Rg	Ws	Rg	Ws	Rg	Ws	Rg	Ws
PL (100-1000): **ILGR**	0.972	0.965	0.960	0.960	0.939	0.958	0.924	0.942
PLC (100-1000): **ILGR**	0.943	0.952	0.937	0.950	0.935	0.948	0.919	0.930
PL (1000): GCN	-	-	0.942	0.938	-	-	-	-
PLC (1000): GCN	-	-	0.927	0.935	-	-	-	-

increasing node dimensions. PL models perform relatively better than PLC models because the inherent topology is comparatively simpler in PL than that of PLC. More specifically, the learning mechanism of the ILGR is based on nodes sub-graphs, which generates embedding vectors by combining aggregation and combination operations. In PL graphs, there are very few nodes with a high degree while most of them bear small degrees. Therefore, node sub-graphs are more or less similar for most of the nodes. On the other hand, PLC graphs have a high clustering coefficient which introduces variability and complexity in sub-graphs, thereby makes the learning more challenging. This can be seen from the sample graphs in Fig. **??**. Nevertheless, our framework attains sufficiently high accuracy for both the models, which demonstrates the accuracy of the proposed framework in the task of estimating node criticality scores. Table 5.1 also shows the comparison of the proposed ILGR with the GCN algorithm trained for networks with 1000 nodes. It can be observed that the proposed approach outperforms GCN model for both PL and PLC networks. Furthermore, it must be noted that while it is required to train different GCN models for different sizes of networks, the same ILGR model can be used to predict the criticality for networks of any dimension.

The generalizability and the scalability of the proposed approach are further reinforced through Table 5.2, which reports the Top-5% accuracy in real-world networks, predicted through models trained on PL and PLC graphs. It can be observed that the PL model has a mean accuracy of 87.4% and 89.5 % for R_g and W_s, respectively, where the mean is taken across four different real graphs. Similarly, the mean accuracy of PLC model is 90.0% and 89.8 % for R_g and W_s, respectively. The proposed framework has sufficiently

Table 5.2: Accuracy of node identification task in real-world graphs. Ntest: # test nodes

Robustness metric	Rg			Ws		
graph/scores	Ntest	Model	Top-5%	Ntest	Model	Top-5%
bio-yeast	1500	Rg-pl	0.895	1500	ws-pl	0.877
bio-yeast	1500	Rg-plc	0.912	1500	ws-plc	0.898
US powergrid	4941	Rg-pl	0.86	4941	ws-pl	0.923
US powergrid	4941	Rg-plc	0.928	4941	ws-plc	0.914
Wiki-vote	7115	Rg-pl	0.865	7115	ws-pl	0.893
Wiki-vote	7115	Rg-plc	0.892	7115	ws-plc	0.887
cit-DBLP	12591	Rg-pl	0.875	7115	ws-pl	0.889
cit-DBLP	12591	Rg-plc	0.878	7115	ws-plc	0.895

high accuracy in detecting critical nodes with both the robustness metrics, even though the model has never seen the real-world graph during the training period. This works because the nodes sub-graphs of real-world networks could match with that of synthetic graphs and therefore the model might have counter that type of sub-graphs during the training period. Furthermore, the models trained on PLC graphs perform better than that of PL graphs. The reason is that the PLC graphs manifest both power-law degree distribution and clusters, hence, are more accurate depiction of real-world networks compared to PL graphs. In addition, the model performance for any alternate graph family or robustness metrics can be enhanced via efficient re-tuning. In this regard, we implement transfer learning by tuning the model trained on PLC graph for estimating criticality scores in real-world bio-yeast network. Although we have used 150 nodes for tuning, the model performance has been increased by 2.7% compared to the scenario when the model trained solely on PLC graphs is used for estimation. Thus, ILGR can accurately identify critical nodes for any large network in a very efficient manner which further strengthens its scalability.

Along with the accurate identification of nodes, the proposed framework provides an appreciable advantage in execution time which is indicated by the running times in Table 5.3. The proposed method is multiple orders faster than the conventional approach, and this gap will increase as the network size grows. All the training and experiments are conducted on a system with an Intel i9 processor running at 3.4 GHz with 6 GB Nvidia RTX 2070 GPU.

The time reported for the proposed approach only includes the prediction time as training is done offline. Even the training time of the proposed method is relatively less than that of the conventional approach.

Table 5.3: Running time of node identification task

graphs/specs	Ntest	Model	Time: proposed (s)	Time: conventional(s)
PL	5000	Rg-pl	17	64600
PLC	5000	Rg-plc	17	64600
US Powergrid	4941	Rg-plc	16	64212
Wiki-Vote	7115	Rg-plc	23	86420

5.4.5 Results and Discussion for link identification

This subsection assesses the performance of the proposed ILGR framework in the task of critical link identification. The validation is done across different graph sizes and graph types. Separate models are trained for all possible combinations of graph type and metric type, which results in four distinct models. For each model, 30 different random graphs of dimension varying from 1000 to 2000 links are generated for training. The ground truth for the selected links is computed with Algorithm 3. The training model is similar to that of node except that a layer is added that generates a link embedding vector from an associated pair of node embeddings. Consequently, the link embedding is stacked with a regression module to predict link criticality score. The trained model is then used to predict link scores in graphs of higher dimensions, i.e., 1000, 2000, 10000, and 20000 links. Table 5.4 reports accuracy for R_g and W_s in PL and PLC graphs. It can be observed that the mean accuracy of PL model in detecting Top-5% of the critical links is 91% and 94% for R_g and W_s both, respectively. Similarly, the mean accuracy of PLC model is 97.5% and 96.1% for R_g and W_s, respectively. The model has fairly high identification accuracy even though the mean is taken across graphs of higher dimensions than that of training. There is a very nominal fall of accuracy with increasing size although the graph size scales in the order of two. Furthermore, Table 5.4 depicts the comparison of the proposed ILGR with the GCN algorithm trained for

Table 5.4: Accuracy of link identification task in synthetic graphs.

Test graph size	Ntest 1000		Ntest 2000		Ntest 10000		Ntest 20000	
graph/scores	Rg	Ws	Rg	Ws	Rg	Ws	Rg	Ws
PL (200-2000): **ILGR**	0.979	0.991	0.970	0.985	0.963	0.982	0.953	0.97
PLC (200-2000): **ILGR**	0.981	0.968	0.98	0.961	0.972	0.96	0.967	0.957
PLC (1000): GCN	0.975	0.987	-	-	-	-	-	-
PLC (1000): GCN	0.989	0.974	-	-	-	-	-	-

Table 5.5: Accuracy of link identification task in real-world graphs. Ntest: # test links

Robustness metric	Rg			Ws		
graph/scores	Ntest	Model	Top-5%	Ntest	Model	Top-5%
bio-yeast	1948	Rg-pl	0.920	1948	ws-pl	0.904
bio-yeast	1948	Rg-plc	0.926	1948	ws-plc	0.952
US powergrid	6594	Rg-pl	0.946	6594	ws-pl	0.91
US powergrid	6594	Rg-plc	0.931	6594	ws-plc	0.946
Wiki-vote	30000	Rg-pl	0.887	30000	ws-pl	0.871
Wiki-vote	30000	Rg-plc	0.896	30000	ws-plc	0.925
cit-DBLP	49743	Rg-pl	0.870	49743	ws-pl	0.831
cit-DBLP	49743	Rg-plc	0.872	49743	ws-plc	0.892

networks with 1000 links. It can be inferred that the proposed approach outperforms the GCN model for both PL and PLC networks as well as allows identification of critical links in networks of different sizes.

The scalability and generalizability of our approach in the task of link identification are further supported by evaluating model performance on real-world networks. Table 5.5 tabulates the Top-5% accuracy in four real-world networks. The accuracy is reported for all the four different models that have been trained on synthetic graphs. It can be inferred that the model has sufficiently high accuracy in identifying critical nodes for both robustness metrics, even though the model has never seen the real-world graph during the training period. Compared to PL graphs, models trained on PLC graphs seem to have high accuracy for real-world networks and the reason is the same as discussed earlier. Further, the execution times in Table 5.6 demonstrate the computational efficiency of the proposed method.

Although the performance of ILGR in the tasks of node identification and link identification from the estimated criticality scores are similar to a large extent (due to common

Table 5.6: Running time of link identification task

graph/specs	Ntest	Model	Time: proposed (s)	Time: conventional (s)
PL	5000	Rg-pl	19	64830
PLC	5000	Rg-plc	19	64830
US Powergrid	6594	Rg-plc	21	65470
Wiki-Vote	10000	Rg-plc	25	88231

framework, i.e., node embeddings of sub-graphs), there are some differences. First of all, the overall performance of ILGR for link identification is better than that of node identification as seen from the Tables 5.2 to 5.5. This is because, a link embedding vector is dependent on two nodes' embedding vectors, thereby including more information compared to the node case which solely uses single node embeddings. The second difference can be seen in their execution times. Link identification takes more time than that of a node as it involves an extra operation (i.e., the combination of node pair embeddings to generate link embedding) apart from common executions.

5.4.6 Computational complexity of algorithms

The proposed ILGR framework consists of an embedding module and a regression module. The embedding module is based on GraphSAGE algorithm which involves aggregation and combination operation, and basically it is a matrix multiplication of node feature vector with weight matrix. Similarly, the regression module consists of feed forward layers which require matrix multiplication of weights with node embedding vectors. However, these operations are not dependent on the graph size (i.e., number of nodes/links) and are rather dependent on the size of the node feature/embedding vector and number of layers as discussed in [107]. Therefore, with respect to graph size (N), the overall time complexity of proposed ILGR framework is constant which in terms of Big O can be written as $O(1)$. On the other hand, the time complexity of conventional optimization or analytical based approaches is dependent on the number of nodes/links of the graph.

5.5 Summary

This chapter proposes a GNN based inductive learning framework for fast identification of critical nodes and links in large complex networks. Criticality score is defined based on two graph robustness metrics, i.e., effective graph resistance and weighted spectrum. The framework consists of two parts, where in the first part, a GNN based embedding and regression model are trained end to end on synthetic graphs with a small subset of nodes/links. The second part deals with the prediction of scores for unseen nodes/links of the graph. The Top-5% identification accuracy of the model is more than 90% for both the robustness metrics. Further, the scalability of the model is shown by identifying critical nodes/links on real-world networks. The proposed approach is multiple orders faster compared to the conventional method. In the next chapter, different sources of uncertainty in GNN are incorporated to improve the usability of the predictions.

Chapter 6

Handling Uncertainty in Graph Neural Network models

This chapter presents a generalized framework to quantify uncertainties in graph neural network predictions. It further enhances the usability of criticality score predictions in Chapter 5 by providing a confidence interval around mean scores. Uncertainty in measured quantities and imprecise information about the underlying structure and features of a network can pose a serious impediment to the efficiency of the learning process and quality of the resulting models. Uncertainty in estimated parameters and structure of trained model is a fundamental modeling challenge that imposes restrictions on the confidence of predictions. Learning representations in an uncertainty-aware manner is fundamental to producing robust models and reliable predictions. Models that do not account for these sources of uncertainty can be over-confident in their predictions [120]. These factors can pose serious problems to effective utilization of the available information in the model building process as well as reliable interpretation of the model predictions under adverse situations [32, 121].

Uncertainty quantification ubiquitously arises in modeling, and has been extensively addressed in the context of deep neural networks in computer vision [122], natural language processing [123] and robotics [124]. However, in the context of GNN, uncertainty quantifi-

cation and incorporation has received relatively less attention. Existing methods adopt a Bayesian approach to mitigate the effect of uncertainty on the predictions [64, 65]. However, they consider the links and features of nodes in the input graph to be deterministic and thus, do not consider the measurement uncertainty therein. In addition, an explicit and systematic quantification of the uncertainty in predictions is also not provided.

This chapter addresses the lack of systematic and explicit incorporation of different sources of uncertainty in GNNs within a Bayesian framework. Different sources of aleatoric and epistemic uncertainty in GNNs are formally defined. Specifically, we consider the aleatoric uncertainty arising from (i) imprecise information about the graph structure via probabilistic links and (ii) measurement noise in feature vectors of nodes. We propagate the aleatoric uncertainty through the node embedding layers and classification layers of the GNN model via Assumed Density Filtering (ADF). We quantify the epistemic uncertainty arising from the probabilistic parameters of the GNN model with Monte-Carlo sampling. This framework complements the GNN predictions and also enables us to conduct a fair comparison of various robustness metrics in Chapter 8.

6.1 Aleatoric Uncertainty in Graph Neural Networks

Consider a graph G represented as a tuple $G = (V, E)$ of a set of n_V vertices/nodes V and n_E edges/links E. Each node u_i, $i \in [1, n_V]$ of the graph consists of d features, represented as a vector $h_i = [h_{ij}]$, $j = 1, 2, 3, ..., d$. Each link e_{ij}, between the i^{th} and j^{th} nodes in the graph is associated with a weight p_{ij}, which signifies the strength of the link. We consider the weight to be normalised between 0 and 1 and interpret the weight as the probability of the corresponding link e_{ij}. In the following subsections, we discuss the different types and sources of uncertainty. Aleatoric uncertainty refers to intrinsic randomness of the data due to noisy or inaccurate measurements. In the case of a GNN, the input data is in the form of graphs that model a real-world network. This data consists of the feature vectors

h_i of the nodes and probabilities/weights p_{ij} of the links. The different sources of *aleatoric uncertainty* in GNN are as follows:

AU1 Measurement uncertainty associated with feature vectors of nodes h_i, i.e., the measured feature vectors are considered as being the sum of true feature vectors (h_i^*) and measurement noise (ϵ_i) as: $h_i = x_i + \epsilon_i$, with $\epsilon_i \sim p(\epsilon)$.

AU2 Structural uncertainty of the graph captured via probabilities of links p_{ij}, $i = 1, 2, 3, ..., n_V$, $j = 1, 2, 3, ..., n_V$.

Uncertainty AU1 refers to the uncertainty in features of the nodes. The feature vectors can represent physical quantities such as the coverage of a cell tower in a communication network, or the functionalities of a protein in a biological network.

Uncertainty AU2, on the other hand, represents the uncertainty in the existence of links. In the protein network scenario, the knowledge of interactions between different proteins and protein complexes is also highly uncertain, which results in probabilistic links between different nodes. Since the interactions are derived through noisy and error-prone lab experiments, each link is associated with an uncertainty value [125]. For instance, a graph with 5 nodes and binary weights of links has $2^5 = 32$ possible configurations. In a graph with continuous valued weights, these weights influence the extent to which information between any two nodes is exchanged and assimilated. This type of uncertainty therefore results in fundamentally different information exchange and processing through the graph.

6.2 Epistemic Uncertainty in Graph Neural Networks

Epistemic uncertainty is the scientific uncertainty in the model that exists because of model in-competency to completely explain the underlying process. A GNN model $\Xi(\Theta)$ typically consists of several layers of complex aggregation and combination operations followed by feedforward processing. A single type of epistemic uncertainty can be defined as:

EU1 Parametric uncertainty in the GNN model, i.e., the parameters Θ of the GNN are assumed to be probabilistic with a probability density function $p(\Theta)$

Uncertainty EU1 represents the uncertainty of the learnable parameters Θ of the GNN model, and is represented by by placing a distribution over the neural network weights. However, estimation of the posterior density function of the parameters $p(\Theta|D)$ given the data D is mathematically intractable to compute for deep neural networks and is approximated by different methods. Among these methods, variational inference [126], and sampling-based approaches [127] are the most effective ones. Monte-Carlo sampling methods involve sampling the parameters from a distribution and are generally obtained using an ensemble of neural network predictions. The prediction ensemble could either be generated by differently trained networks [128], or by using dropout at test-time [127].

6.3 Bayesian framework to incorporate uncertainties

A GNN typically involves two modules - node embedding and feedforward modules. The node embedding module performs aggregation and combination operations in the embedding layers of the model and produces a vector of *node embeddings*. These operations capture the information propagation and processing phenomena in networked data. The feedforward module processes the node embeddings with non-linear transformations via feedforward layers and produces the final output. We next describe the proposed approach for quantification of the total uncertainty involved in GNN models.

6.3.1 Problem definition

We consider a network $G = (V, E)$ as described in Section 2 where nodes are associated with noisy feature vectors (AU1) and links are associated with probabilities (AU2) as follows:

1. The noise in the features is considered to be Gaussian with zero mean and known

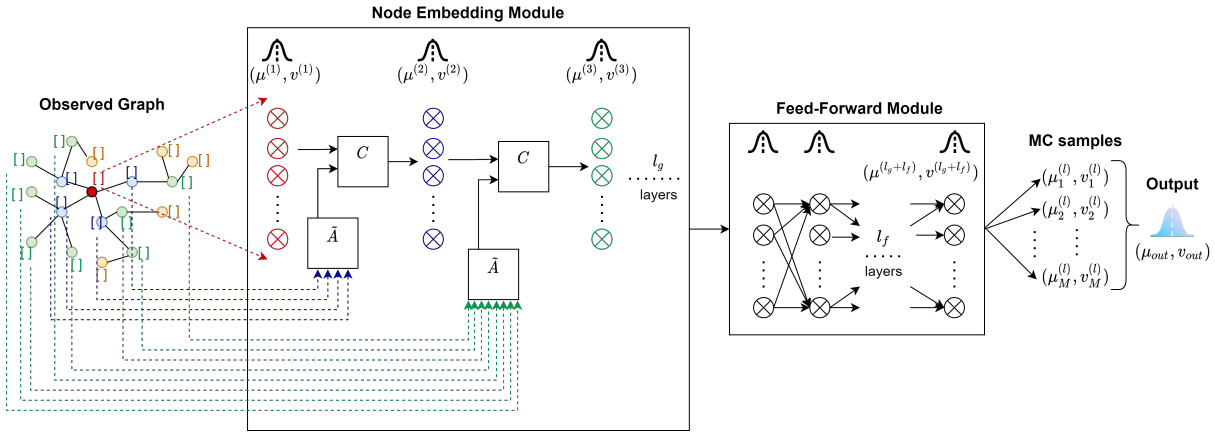


Figure 6.1: Proposed BGNN architecture for incorporating aleatoric and epistemic uncertainty in a GNN.

variance. For a node u in the network, we have:

$$h_u = h_u^* + \epsilon_u, \qquad \epsilon_u \sim \mathcal{N}(0, \Sigma_u) \tag{6.1}$$

where h and h^* are the measured and true feature vector respectively, ϵ_u is the noise in feature vector and Σ_u is a diagonal matrix consisting of the known variances of noise in individual features. The noise in the features of different nodes are assumed to be uncorrelated, i.e., for any two nodes u and ν in the network, we have: $\mathbb{E}[\epsilon_u \epsilon_\nu] = 0$

2. The probabilities $p_{u\nu}$ of links are assumed to be known *a priori*.

3. The learnable parameters Θ of the GNN are assumed to be random variables with an unknown probability density function: $\Theta \sim p_\Theta(\psi)$

These sources of uncertainty result in a probabilistic propagation of the feature vectors through the model, and thus result in probabilistic outputs, i.e., $\hat{y} \sim p(y|h, p, \Theta)$. Obtaining the exact distribution of $\hat{y}$ is mathematically intractable. The problem of uncertainty quantification considered here is to systematically obtain the variance in the predictions because of the different sources of uncertainty. In the following subsections, we discuss how these effects are quantified in the proposed framework.

6.3.2 Propagation of Aleatoric Uncertainty in GNN

We propose a Bayesian approach to propagate the uncertainty in feature vectors through a GNN while also explicitly incorporating probabilistic links in the system. We achieve this with Assumed Density Filtering (ADF) and moment matching. We consider propagating and matching the mean and variance of the probability density function of outputs of all node embedding and feedforward layers of the model. While this can be achieved with existing approaches for feedforward layers, the following result formalises the result for node embedding layers.

Theorem 5. *The expected value (μ_u) and variance (v_u) of the node embedding for node u, accounting for aleatoric uncertainty AU1 and AU2 with mean aggregation and linear activation functions are:*

$$\mu_u^{(i)} = \theta_C^{(i)} \mu_u^{(i-1)} + \theta_A^{(i)} \frac{1}{|N(u)|} \sum_{\nu \in N(u)} p_{u\nu} \mu_\nu^{(i-1)} \tag{6.2}$$

$$v_u^{(i)} = {\theta_C^{(i)}}^2 v_u^{(i-1)} + {\theta_A^{(i)}}^2 \frac{1}{|N(u)|D(u)} \sum_{\nu \in N(u)} p_{u\nu}^2 v_\nu^{(i-1)} \tag{6.3}$$

where the superscript (i) represents the corresponding quantities of the i^{th} node embedding layer; θ_C and θ_A represent the parameters of the combination and aggregation operations of GNN respectively; $N(u)$ and $D(u)$ represent the neighbourhood and degree of node u, respectively; $p_{u\nu}$ represents the probability of the link between nodes u and ν.

Proof. Consider the noisy feature vector for a node u of the observed graph. This feature vector, along with the probabilistic graph structure are fed as input to the GNN. According to Eq. (6.1), feature vector can be expressed as:

$$h_u \sim \mathcal{N}(h_u^*, \Sigma) \tag{6.4}$$

This random variable is processed by the node embedding layers of the GNN model via ag-

gregation and combination operations. The aggregation operation in the i^{th} layer aggregates the embeddings of the neighbouring nodes $h_{N(u)}^{(i-1)}$ in the $(i-1)^{th}$ embedding layer and is equivalent to information collection operation from neighbours in the network. The combination operation combines the aggregated embeddings with the node embedding of the node u and is equivalent to assimilating information from a network. The operation performed by the i^{th} node embedding layer can be expressed as [107]:

$$\begin{aligned} h_u^{(i)} &= f^{(i)}\left(h_u^{(i-1)}, h_{N(u)}^{(i-1)}\right) \\ &= g\left[\theta_C^{(i)} h_u^{(i-1)} + \theta_A^{(i)} \tilde{A}\left(h_{N(u)}^{(i-1)}\right)\right] \end{aligned} \tag{6.5}$$

where $g[\cdot]$ represents the activation function and $\tilde{A}(\cdot)$ denotes the aggregation operation. This operation is performed recursively l_g number of times for a GNN with l_g node embedding layers. The embeddings generated at the i^{th} layer are dependent solely on the embeddings of the $(i-1)^{th}$ layer. As a result, joint density of all embeddings generated for a node u, i.e., $p(h_u^{(0:l_g)})$ can be expressed as:

$$\begin{aligned} p\left(h_u^{(0:l_g)}\right) &= p\left(h_u^{(0)}\right) \prod_{i=1}^{l_g} p\left(h_u^{(i)} | h_u^{(i-1)}\right) \\ p\left(h_u^{(i)} | h_u^{(i-1)}\right) &= \delta\left[h_u^{(i)} - f^i\left(h_u^{(i-1)}, h_{N(u)}^{(i-1)}\right)\right] \end{aligned} \tag{6.6}$$

where, $\delta[\cdot]$ is the Dirac delta function. This process is shown in Figure 6.1. Propagating the uncertainty through the node embedding layers requires obtaining the joint density function described in Eq. (6.6), which is mathematically intractable.

We employ ADF to approximate the joint density function. We choose ADF because of its low computational demand for a systematic propagation of uncertainty through all layers of a neural network [129, 130]. ADF approximates this intractable distribution as follows

(we remove the subscript u from the feature vector for the sake of brevity):

$$p\left(h^{(0:l_g)}\right) \approx q\left(h^{(0:l_g)}\right) = q\left(h^{(0)}\right) \prod_{i=1}^{l_g} q\left(h^{(i)}\right) \tag{6.7}$$

ADF makes the first approximation by assuming that the probability density of the embeddings in the different layers are independent of each other. Furthermore, $q\left(h^{(i)}\right)$ is assumed to be Gaussian, so that we have:

$$\begin{aligned} q\left(h^{(0)}\right) &= p\left(h^{(0)}\right) \\ q\left(h^{(i)}\right) &= \prod_{j=1}^{m_i} \mathcal{N}\left(\mu_j^{(i)}, v_j^{(i)}\right) \end{aligned} \tag{6.8}$$

where m_i represents the size of the embedding vector at i^{th} layer of the model, $\mu_j^{(i)}$ and $v_j^{(i)}$ are the mean and variance of the j^{th} element of the embedding vector $h^{(i)}$. The approximate joint density function of all node embeddings till the i^{th} layer can be expressed as:

$$\tilde{p}\left(h^{(0:i)}\right) = p\left(h^{(i)}|h^{(i-1)}\right) \prod_{j=0}^{i-1} q\left(h^{(j)}\right) \tag{6.9}$$

This step replaces the conditionals in Eq. (6.6) by the corresponding approximations from Eq. (6.8) to obtain an approximate density $\tilde{p}\left(h^{(0:i)}\right)$. ADF then finds the best approximate distribution $q\left(h^{(0:i)}\right)$ by minimizing the KL divergence with $\tilde{p}\left(h^{(0:i)}\right)$ as:

$$q\left(h^{(0:i)}\right) = \underset{\tilde{q}(h^{(0:i)})}{\arg\min}\, KL\left(\tilde{q}\left(h^{(0:i)}\right) || \tilde{p}\left(h^{(0:i)}\right)\right) \tag{6.10}$$

This can be solved by matching the moments between the two distributions [131]. Thus, any layer $h_u^{(i)} = f^{(i)}\left(h_u^{(i-1)}, h_{N(u)}^{(i-1)}\right)$ can be converted into an uncertainty propagation layer by matching first two moments as:

$$\mu_u^{(i)} = \mathbb{E}_{q\left(h_u^{(i-1)}\right)}\left[f^{(i)}\left(h_u^{(i-1)}, h_{N(u)}^{(i-1)}\right)\right] \tag{6.11}$$

$$v_u^{(i)} = var_{q\left(h_u^{(i-1)}\right)}\left[f^{(i)}\left(h_u^{(i-1)}, h_{N(u)}^{(i-1)}\right)\right] \tag{6.12}$$

where, $\mathbb{E}$ and var are the expectation and variance operators respectively. When the aggregation operation $\tilde{A}$ is the mean operator, and the activation function $g(\cdot)$ is linear, substituting Eq. (6.5) in Eqs. (6.11) and (6.12) yields Eqs. (6.2) and (6.3), and hence proves Theorem 5.

This makes use of the following two identities: (1) Expectation of mean is equivalent to mean of expectations; and (2) Variance of means is the normalized form of mean of variances.

□

Equations (6.11) and (6.12) can be determined analytically for most of the functions used in neural network such as ReLu, sigmoid, convolution, etc. For instance if the function g is ReLu, then the modified mean and variance are [132]:

$$\hat{\mu_u}^{(i)}\left(\mu_u^{(i)}, v_u^{(i)}\right) = \mu_u^{(i)}\Phi\left(\frac{\mu_u^{(i)}}{\sigma_u^{(i)}}\right) + \sigma_u^{(i)}\phi\left(\frac{\mu_u^{(i)}}{\sigma_u^{(i)}}\right) \tag{6.13}$$

$$\hat{v_u}^{(i)}\left(v_u^{(i)}, v_u^{(i)}\right) = \left(\mu_u^{(i)} + v_u^{(i)}\right)\Phi\left(\frac{\mu_u^{(i)}}{\sigma_u^{(i)}}\right) + \quad \sigma_u^{(i)}\mu_u^{(i)}\phi\left(\frac{\mu_u^{(i)}}{\sigma_u^{(i)}}\right) - \left(\hat{\mu_u}^{(i)}\right)^2 \tag{6.14}$$

where, $\sigma_u^{(i)} = \sqrt{\left(v_u^{(i)}\right)}$, Φ and ϕ are the cumulative normal and standard normal distributions, respectively. Basically, Eqs. (6.13) and (6.14) are recursive formulae to compute mean $\left(\mu_u^{(i)}\right)$ and uncertainty $\left(v_u^{(i)}\right)$ of the embeddings, given the parameters of the embedding distribution $q\left(h^{(i-1)}\right)$ in previous layer.

Typically, node embeddings from GNN are fed to feed-forward layers for classification/regression task. Therefore, $\mu_u^{(l_g)}$, $v_u^{(l_g)}$ serve as an input to feed-forward layers, and mean and variance is propagated in a similar way as shown in Gast & Roth [130], Loquercio et al. [124]. In a nutshell, ADF reshapes the forward pass of a GNN to generate not only output predictions $\mu_u^{(l)}$, but also their respective aleatoric uncertainties $v_u^{(l)}$. This is achieved by considering two values per dimension of both embeddings in GNN layers as well as neural units in feed-forward layers.

6.3.3 Propagation of Epistemic uncertainty in GNN

Epistemic uncertainty, also known as model uncertainty refers to the model confidence on its prediction. This uncertainty arises because of the single adoption of weights out of many combinations that can attain same loss values on training data. This is usually captured by assuming a probability distribution for neural network weights rather than a scalar value. However, computation of this distribution $p(\omega|X, y)$ is usually intractable. Therefore, MC based approaches have been used to obtain different weight samples by using dropout at test time [127, 133, 134]. Specifically, in our case, epistemic uncertainty is the variance of M MC samples obtained via different dropout masks as shown below:

$$\begin{aligned} p(\omega|X, y) \approx q(\Theta; \phi) = \text{Bernoulli}(\Theta; \phi) \\ \sigma^2_{\text{model}} = \frac{1}{T}\sum_{t=1}^{M}(y_t - \hat{y})^2 \end{aligned} \tag{6.15}$$

where, $\{y_t\}_t^M$ is a set of M sampled outputs for different weight instances from the distribution $\omega^t \sim q(\omega, \phi)$ and $\hat{y} = \frac{1}{T}\sum_t y_t$. Authors in Gal et al.[127] have shown that the optimal dropout rate ϕ for the computation of σ_{model} is same as training dropout rate.

6.3.4 Total uncertainty in GNN

Total variance of GNN predictions y for a sample node with feature vector X corrupted by noise variance v^0 can be written as:

$$\begin{aligned} \sigma_{tot} = \frac{1}{T}\sum_{t=1}^{T} v_t^{(L)} + \left(\mu_t^{(L)} - \hat{\mu}\right)^2, \\ \text{where,} \quad \hat{\mu} = \frac{1}{T}\sum_{t=1}^{T}\mu_t^{(L)}. \end{aligned} \tag{6.16}$$

The first term $\left(v_t^{(L)}\right)$ in denotes aleatoric variance and the second term $\left(\mu_t^{(L)} - \hat{\mu}\right)^2$ represents the model uncertainty from M MC predictions. $L = l_g + l_f$ is the total number

of layers in GNN. Thus, the first part of total variance captures ensembles of propagated variance and the second part handles the ensembles of mean prediction, thereby, addressing both aleatoric and epistemic uncertainty. The overall algorithm to compute total uncertainty can be summarized in following steps: (i) Transform GNN into a bayesian network by associating mean and variance to each embedding vector and neuron unit; (ii) Obtain M mean and variance predictions by forwarding (X, v^0) to network with weights ω^t sampled from $q(\omega, \phi)$; (iii) Compute output predictions and its variances according to Eq. (6.16).

6.4 Experimental Results and Discussion

We apply our method to three standard datasets, namely, Cora, Amazon Computers and PubMed, with varying number of nodes, links, features and classes. The details of these datasets can be referred in Chapter 5. In order to test the effectiveness and generalizability of the method, we address the node classification task in the three networks with 3, 7 and 10 classes. We compare the method with the state-of-the-art in the literature [65] and highlight the stark contrast in computational efficiency and quantification of uncertainty. We also demonstrate this ability of the model to capture the uncertainty via a sensitivity study. It is important to note that the training of GNN is accomplished with standard cross entropy loss function that solely involves the mean prediction. As a part of future work, both mean and variance will be incorporated in the loss function via conditional log likelihood. This will allow using the information about estimated uncertainties for improving model performance/robustness, rather than just quantifying it.

6.4.1 Baseline

We compare the performance of the proposed method with Bayesian Graph Convolutional Network (BGCN) proposed by [65]. This work captures the aleatoric uncertainty through MAP estimation of the network structure and quantifies the epistemic uncertainty with

Monte-Carlo sampling. However, uncertainties AU1 and AU2 are not explicitly incorporated. Moreover, the MAP estimation is dependent on accurate knowledge of node features, and hence the method is incapable of handling AU1.

6.4.2 Experimental Setup

All evaluations are repeated 100 times and average of metrics are reported for each dataset described in Section 6.4.4. GNN is trained with the GraphSAGE algorithm [107]. The detailed architecture of the GNN is as follows: Depth i.e., no. of node embedding modules: 2; no. of neurons in 2 layers: 64, 32; no. of Multi-layer perceptron (MLP) layers: 3; no. of neurons in MLP layers: 12,8,1; Activation function: Linear (except last layer with softmax); Aggregation function: Mean. Since the core task of the GNN is node classification, the node embeddings are concatenated with feedforward layers to provide class probabilities. The loss function is categorical cross entropy with ADAM optimizer. The training is carried out in a mini-batch manner. The batch size is set to 50, the learning rate is set to 0.001 with a dropout rate of 0.1. The models are trained 50 epochs in total.

6.4.3 Sources of Uncertainty

We introduce uncertainty in nodes feature (AU1) by adding Gaussian white noise to the true feature values with zero mean and a known variance as shown in Eq. (6.1). The proposed method is compared with the baseline with different levels of noise. The variance of noise is also varied to highlight the ability of the proposed method to capture the impact of this noise as it propagates through the model. We introduce uncertainty in links (AU2) with probability of nodes. These probabilities are not available in the datasets for all links in the networks. We therefore perform link prediction in a supervised manner following the approach presented by Zhang & Chen [135] and obtain the probabilities of links. These predicted probabilities are then used for training Bayesian models in the proposed method. We introduce uncertainty in parameters of GNN (EU1) with a Bernoulli distribution of

parameters according to Eq. (6.15).

6.4.4 Results

The results demonstrate the computational efficiency of the proposed approach, the ability to reflect different levels of uncertainty in predictions and generalizability of the approach. The source code to regenerate all the results can be accessed at this link.

Adequacy of BGNN

We compare the proposed approach in this paper with Pal et al. [65]. To the best of authors' knowledge, Pal et al. [65] is the only work in the literature that deals with aleatoric uncertainty in GNN. A summary of results evaluated on Cora dataset based on 100 MC runs obtained via different dropout masks is presented in Table 6.1. The results in Table 6.1 show that the proposed method yields higher classification accuracy as compared to the baseline BGCN in all the cases of input variance. In this work, we specify input variance as the percentage of mean features across all nodes in the dataset. The similar trends were observed in PubMed and Amazon Computers datasets as well. This demonstrates that learning MAP estimate of the input graph does not add much value in quantifying uncertainty related to noisy node feature vectors and link weights. On the other hand, the proposed method systematically propagates uncertainties through all the layers of GNN, as discussed in the forthcoming subsections.

Table 6.1: Performance comparison of BGCN and BGNN on Cora dataset (average of 100 MC runs). Standard deviation is shown underneath the average classification accuracy.

Input Variance	**Classification Accuracy (%)**	
	BGCN	**BGNN (ours)**
0.0%	97.71 ± 0.0	97.96 ± 0.0
2.5%	89.25 ± 0.00357	90.21 ± 0.00311
5.0%	76.16 ± 0.00966	78.11 ± 0.0088
12.0%	58.60 ± 0.01153	61.22 ± 0.0108

Effectiveness of BGNN

We demonstrate the effectiveness of the proposed approach by selecting a few samples from the datasets randomly, and examining the variations in class probabilities with changes in input variance. Table 6.2 presents a summary of the results for three random samples, one from each class of the PubMed dataset. Firstly, it can be seen that the class probabilities corresponding to the true class decreases with increase in the levels of input variance. This is intuitive as a higher amount of input variance will introduce more uncertainty in the network, specifically in the node embeddings of GNN layers, thereby leading to reduction in class probabilities (i.e., moving towards a more uniform distribution). Secondly, in some cases like Sample ID 28, increasing levels of input variance may lead to mis-classification. This is because the prediction probability corresponding to the true class in no variance case is much lesser as compared to other samples. Finally, the total propagated variance at the output is observed to increase with increase in input variance across all samples. These examples illustrate the systematic propagation of uncertainties across all layers of GNN for different cases of input variance.

Table 6.2: Longitudinal analysis of a few test samples selected at random from PubMed dataset. Prediction probabilities correspond to the class probabilities of true class. Bold values indicate the misclassification at corresponding variance levels. Total propagated variance are the mean values across all classes.

Sample ID	**True Class**	**Metrics across different levels of input variance**							
		Prediction Probabilities				**Total propagated variance**			
		0.0%	**2.5%**	**5.0%**	**12.0%**	**0.0%**	**2.5%**	**5.0%**	**12.0%**
12	1	0.923	0.835	0.767	0.578	0.0003	0.0031	0.0062	0.0166
13	2	0.775	0.716	0.689	0.635	0.0031	0.0081	0.0119	0.0218
28	3	0.551	**0.353**	**0.262**	**0.131**	0.0023	0.0034	0.0044	0.0063

Generalizability of BGNN

The generalizability of proposed approach is examined by performing experiments over graphs of different sizes and characteristics. The model performance is evaluated using

classification accuracy. Owing to the absence of ground truth for variance assessment, it is evaluated using average per-class negative log likelihood (NLL) [124, 130, 134]. The value of NLL for a specific class is evaluated as:

$$NLL = \frac{1}{2}\log(\sigma_{\text{tot}}) + \frac{1}{2\sigma_{\text{tot}}}(y - \hat{y})^2 \tag{6.17}$$

where, $\hat{y}$ is the mean prediction of class probabilities across 100 MC runs and y is a 0/1 value indicating whether the given node belongs to a specific class. σ_{tot} is the total variance comprising of propagated input variance and that due to model uncertainty. Tables 6.3 and 6.4 depict the metrics values for Cora and PubMed datasets, respectively. These values are obtained based on the average of 100 MC runs, and mean of per-class NLL is reported. It can be observed for all the datasets that the mean classification accuracy of the model decreases with the increase in the input variance. It is intuitive in a sense that as variance in the input feature vector increases, it consistently becomes hard for the model to uniquely identify nodes with the node embeddings and thereby their labels. This idea is also reinforced by the increasing values of prediction loss observed with increase in input variance.

The total variance propagated at output (σ_{tot}) in all the cases is also indicated in Tables 6.3 and 6.4. It can be seen that σ_{tot} lies between 0 and 1 in all the cases of datasets considered in this work. Therefore, the first term in right hand side (RHS) of eq. (6.17) will always be negative. If the values of σ_{tot} are relatively higher, as in Cora (Table 6.3), the values of first term in eq. (6.17) dominate, the second term will not be positive enough and consequently, the overall NLL values turn out to be negative. In these cases, the NLL values increase with increase in input variance as $\log(\sigma_{\text{tot}})$ is a monotonically increasing function. This is clearly evident from the NLL values in Tables 6.3. On the other hand, if the values of σ_{tot} are relatively lower, as in PubMed (Table 6.4) dataset, the values of second term in eq. (6.17) dominate and the overall NLL values are positive. In such cases, the NLL values decrease with increase in input variance, as observed in Table 6.4. Thus, NLL demonstrates the high

quality estimates of uncertainty without changing or re-training the GNN.

Table 6.3: Results for Cora dataset (average of 100 MC runs). Input variance is specified as percentage of mean features across all nodes in the dataset.

Input Variance	**Classification Accuracy**	**Prediction Loss**	**Avg. per class NLL**	**Variance propagated at Output**
0.0%	97.96%	0.19	-	-
2.5%	90.21%	0.45	-0.98	0.12
5.0%	78.11%	0.76	-0.65	0.24
12.0%	61.22%	1.64	-0.23	0.57

Table 6.4: Results for PubMed dataset.

Input Variance	**Classification Accuracy**	**Prediction Loss**	**Avg. per class NLL**	**Variance propagated at Output**
0.0%	84.00%	0.40	-	-
2.5%	82.83%	0.46	10.75	0.0029
5.0%	80.70%	0.52	6.83	0.0046
12.0%	76.03%	0.70	3.90	0.0092

6.5 Summary

This chapter introduces a generic framework for incorporating aleatoric and epistemic uncertainty in GNN. The aleatoric uncertainty arising from imprecise information about graph structure (probabilistic links) and node features is propagated via ADF. On the other hand, epistemic uncertainty arising from the probabilistic parameters of GNN model is quantified through MC sampling. The proposed method, BGNN, systematically propagates these uncertainties through the layers of GNN to final predictions without the need of retraining. Furthermore, this method is agnostic to network architecture, algorithm and the learning tasks. Experimental results show that BGNN achieves superior performance in quantifying uncertainties for different levels of input noise across several types of graphs. Next chapter presents a deep reinforcement learning based framework for addressing concurrent case,

where multiple nodes/links can be disrupted at a time.

Chapter 7

Network-based robustness analysis: Identification of concurrent critical nodes with Deep Reinforcement learning

The graph neural network-based approach introduced in Chapter 5 to identify critical nodes belongs to a supervised learning setting. The ground truth node criticality scores are obtained by simulating a single attack at a time, i.e., one node is being impacted at a time. However, there could be a scenario where multiple nodes could be disrupted simultaneously. Then, the research question modifies to "what is the optimal set of nodes whose removal leads to maximum degradation in graph robustness given some budget". Under this scenario, the sequence of selected nodes plays a crucial role in determining the final desired set, unlike the previous case of Chapter 5. Essentially, this problem falls under the generic category of graph combinatorial problems, having use-cases in a variety of applications, including attack graphs [136], influence maximization [137], etc. Existing graph-theoretic techniques to solve this kind of combinatorial problem suffer from various challenges in terms of computational

inefficiency, low scalability, and lack of generalizability. To address these shortcomings, this chapter fuses Graph Neural Network (GNN) with Deep Reinforcement Learning (DRL) to develop an efficient and generic framework for graph combinatorial problems. Specifically, GNNs are used for encoding the underlying graph structure and DRL for learning to identify the optimal node sequence. Moreover, the framework is first developed for Influence Maximization (IM), where one is interested in identifying a set of seed nodes, which when activated, will result in the activation of a maximal number of nodes in the graph. IM is a standard combinatorial problem, having applications in several real-world problems related to social network analysis and epidemiology. Conceptually, IM seed nodes and DVI nodes of Chapter 4 show a lot of resemblance. Nodes in IM are based on their information spread capability, whereas, nodes in DVI are based on their voltage influence scores. Nevertheless, this framework can be tuned for other use-cases, including the identification of critical node set with regard to robustness scores. The proposed DRL-based method is generic and highly scalable for networks of large sizes.

7.1 Background

Consider a directed graph $G = (V, E, \omega)$, where V is a set of vertices, E is a set of edges (pairwise relationships on vertices), and ω is a set of edge weights; a diffusion model, and a budget k. Influence maximization (IM) is the problem of identifying a set of k seed nodes, which when activated, will result in the activation of a maximal number of nodes in G, for the given diffusion model of influence. IM has applications in various domains ranging from viral marketing in social networks to influential proteins in biological networks. For instance, one of the reasons behind the tremendous success of social media platforms is the quality of content the users create or generate via sharing. These actions can be attributed to influence dynamics in the social network. IM was first introduced in 2001 and was formulated as a combinatorial optimization problem by [138]. Majority of the IM algorithms focus on

settings where seed nodes are activated deterministically and then neighborhood nodes are activated via influence. However, there exists two types of activation in real-life scenarios, namely intrinsic and influenced [137]. In case of social media, intrinsic refers to users who post content whereas sharing, retweeting, commenting constitute influenced activation. To this end, the authors in [139] recognize that the events on social media can be categorized as exogenous and endogenous and model the overall diffusion through a multivariate Hawke's process to address activity shaping in social networks.

7.2 Formulation of Activation informed IM (AIM)

The overall architecture of the proposed framework (GraMeR) is depicted in Fig. 7.1, and consists of three main modules. Each of these modules is explained in forthcoming sections. In this section, we describe the IM model considered in our work. It is reasonably different and generic relative to typical IM models explored in related recent work. In a social network, users propagate their views or opinions while simultaneously consuming and reacting to content created by friends, people, and organizations they follow. Thus, there are two ways of activation, namely *intrinsic* (content creation) and *spread of influence* (content spreading). Conventional IM problem solely considers influence activation and overlooks intrinsic activation. However, in practice, the role of content creators has gained significant importance due to massive digitization. Therefore, in this work, we are considering a generic formulation of IM that incorporates both types of activations and is referred to here as activation-informed influence maximization (AIM).

Enabled by the digital revolution, most users now are both content creators and content spreaders at the same time. This is probabilistically modeled through parameters p_s and p_f which represent the probability of intrinsic and influence activation, respectively. Intrinsic activation for a user u is based on its own activities, and user u is assumed to be directly influenced by its 1-hop neighbors. Therefore, the influence part of the probability for activa-

tion is comprised of the activation probabilities due to the 1-hop neighbors of user u. Thus, similar to the IC model, the probability of user u being activated via influence of an user v is written as:

$$p_{uv} = w_{uv} p_f(u), \tag{7.1}$$

where the weights w_{uv} $(0 \leq w_{uv} \leq 1)$ can be determined from the user interactions in a network. The described probabilistic formulation has similarities to the Friedkin-Johnsen social influence model for opinion change [140], where the authors recognize that the dynamics of opinion change are governed by two mechanisms: intrinsic opinion and influenced opinion. Furthermore, by assuming that the nodes are not lazy and are activated by either of the two mechanisms that we outline, we set $P_f(u) = (1 - P_s(u))$. This renders the overall IC probability between nodes v and u to be:

$$p_{uv} = w_{uv}(1 - p_s(u)). \tag{7.2}$$

Note that all the parameters discussed can be efficiently determined either by a maximum-likelihood-based approach or expectation-maximization (EM) approach as followed in [141]. For example, in the Twitter network, the proportion of tweets by a user i that are intrinsic in nature can quantify $P_s(u)$, while a particular weight w_{uv} can be determined by the proportion of user $u's$ retweets (or influenced activity) having their origin in the activity of user v that user u follows.

7.3 Prediction of Candidate Nodes

A fundamental aspect of our framework is the fact that only a certain fraction of nodes are likely to contribute to the AIM solution set. These nodes are referred to as "candidate nodes" in this work. Hence, it is desirable that instead of focusing on all nodes, one should attempt all computationally expensive predictions for the candidate nodes. This section

describes a novel GNN based node classifier that leverages GNN based classifier to identify candidate nodes for AIM. This classifier precedes our primary DRL algorithm (GraMeR) as shown in Fig. **??**, and thereby eases the computational burden.

The task of identifying candidate nodes is framed as a binary node classification problem where the two classes denote "candidate" and "non-candidate" nodes, respectively. The ground truth labels can either be generated via standard greedy hill climbing algorithm or novel centrality metrics recently being proposed [142, 143]. This work uses Influence capacity metric as it is computationally easy to compute compared to other approaches. Influence capacity (IFC) is a novel centrality metric to identify influential nodes. The influence of any node depends on its neighbor connection (local influence) and its own location in the graph (global influence). Local influence of a node u ($I_L(u)$) can be estimated as [142]:

$$I_L(u) = 1 + \sum_{v \in N(u)} P(u,v) + \sum_{v \in N(u)} \sum_{z \in N(v)} P(u,v)P(v,z), \tag{7.3}$$

where, P is the influence probability associated with links, and the operator $N(.)$ denotes the neighbors. Likewise, the global influence score ($I_G(u)$) can be expressed as:

$$I_G(u) = k_c(u)\big(1 + \frac{D(u)}{D_N}\big), \tag{7.4}$$

where, $k_c(u)$ and $D(u)$ represent coreness score and degree of node u, respectively. Notation D_N denotes maximum node degree in the graph. The overall influence capacity $I(u)$ of a node u can then be written as:

$$I(u) = \frac{I_L(u)}{\max_{v \in V} I_L(v)} \times \frac{I_G(u)}{\max_{v \in V} I_G(v)}. \tag{7.5}$$

Nodes retaining extreme values of IFC are labeled as "candidate nodes" while the rest of the nodes are marked as "noncandidate". The threshold of approx 20% seems to work in our case and it is determined based on several experiments, where the candidacy of nodes is

validated with the standard Greedy hill-climbing algorithm. The ground truths from IFC are then leveraged to train the node classifier without relying on a computationally exhaustive Greedy hill-climbing approach [144]. It is important to note that there are various other algorithms to generate groundtruth and one can take any such approach for training the node classifier.

The GNN based node classification model consists of two hidden layers with GraphSAGE as the message-passing algorithm. We selected GraphSAGE since the computation graph for any node u only depends on the induced subgraph up to k-hop neighbors of u. This allows training/prediction across different graph sizes which is most desirable for GraMeR. The parameters of GNN are trained by minimizing the categorical cross-entropy loss function. The overall candidate node prediction task is summarized in Algorithm A.1 in the appendix. It is interesting to note that if this GNN classifier can provide a confidence interval around its class predictions as shown in [145], then the succeeding DRL engine can utilize that interval while searching for optimal seed nodes. This further strengthens the overall framework and is kept as future work.

7.4 Meta Reinforcement Learner

Graph Meta Reinforcement Learner (**GraMeR**) is a deep reinforcement learning (DRL) based framework that learns to identify optimal seed set for AIM. There are various novel aspects of our framework in terms of: (1) Meta learning to enable prediction across different graph types and sizes; (2) double Q learning to estimate sequence of seed nodes without solving a computationally intensive optimization problem at every test time and (3) single policy multi-objective reward formulation for systematic balancing of multiple AIM objectives. This section describes these novel aspects of GraMeR, and architecture is shown in Fig. **??**.

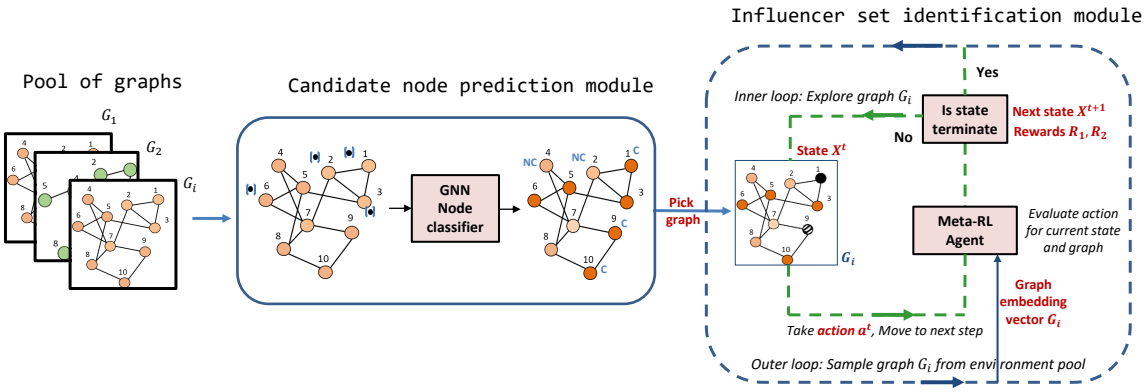


Figure 7.1: Basic architecture of GraMeR. Graphs of different types and sizes are provided to Candidate node prediction module for identifying candidate nodes (C: Candidate; NC: Non-candidate). The Influencer set identification module uses Meta DRL algorithm to train on the candidate nodes of a graph.

7.4.1 Finite MDP

The task of identifying an optimal set of AIM seed nodes is a sequential process where nodes are added one at a time. More importantly, the factors determining the selection of node at any step of the process solely depends on the last node added to the sequence (solution set). Thus, this process follows the Markov property and is therefore formulated as a Markov decision process (MDP). Further, at any step of the process, the action space is finite, i.e., a node has to be selected from a finite set of nodes. So, more precisely, the process can be termed as a finite MDP. The key ingredients to define any finite MDP in the context of DRL are:

State represents the current solution set where nodes are appended in sequence to form the final AIM solution set. Thus, cardinality of the state keeps increasing with the process. Therefore, a state representing vector ($\mathbf{X^t}$) of fixed dimension is needed. $\mathbf{X^t}$ should characterize the state of the system at any time step t in terms of nodes being selected. Therefore, $\mathbf{X^t}$ can be expressed as:

$$\mathbf{X^t} = f(S^t) \tag{7.6}$$

where, S^t is the partial AIM solution set at time t and f is the transformation operator. As nodes in the state are sampled from a graph, an appropriate choice for f operator would be a Graph neural network based transformation which will be discussed in detail in the forthcoming sub-section.

Action refers to the process of adding a new node u to a partial solution set S^t.

Reward quantifies the benefit of taking an action. There are two objectives in our AIM formulation which leads to two reward functions. The first reward (R_1) belongs to the marginal gain in influence spread when a particular node has been added to the solution set. The second reward (R_2) corresponds to the intrinsic probability of node being added to the solution set. The rewards can be written as:

$$\begin{aligned} R_1(X, a) &= \mathcal{I}(G, S \cup \{a\}) - \mathcal{I}(G, S) \\ R_2(X, a) &= p_s(a) \end{aligned} \tag{7.7}$$

where, the operator $\mathcal{I}$ computes the influence spread of solution set S in graph G under the independent cascade model.

Environment: It's an agent world with which it interacts and comprises of everything outside the agent. Here, the environment is a graph. These interactions occur continually, i.e., agent selects node and the graph environment responds to those actions and present new situations to the agent.

Policy: The policy is a strategy or suggested actions that the DRL agent should take at every state of the environment so as to pursue the goal of the learning. It is a probability distribution over feasible nodes that could be added to partial solution set S^t to move the state from X^t to X^{t+1}. Hence, policy $\pi(a|X^t)$ selects the node that yields highest cumulative reward at any arbitrary state X^t.

Termination: At every episode, the search starts with a random node from the candidate set, and the estimated nodes are appended to the partial solution set, one at a time (one at each step of the episode). The episode is terminated when the cardinality of the solution

set S^t attains the search budget b.

7.4.2 Algorithm

GraMeR consists of three modules as illustrated in Fig. 7.1. The first module acts as an environment pool containing training graphs from different families and sizes. The second module provides a set of candidate nodes (described in Section 4) on which the AIM search algorithm will be implemented. Finally, the third module deals with the DRL agent. The process of training the agent starts by randomly selecting a graph G_i from the pool and passing through second module to generate a candidate node set. Each training graph serves as an environment with which our agent interacts via MDP. An episode starts with a random node from the candidate set of the sampled graph and it continues until the budget is consumed. At each step of the episode, the agent will select the next node based on its current policy. Thereafter, the agent updates its current policy by training a Q network with a sampled batch of data from the buffer. Replay buffer stores the state, action and rewards from all the past steps of the process across episodes and environments (graphs), allowing Q network to exploit known information. Once the episode meets termination criterion, the agent samples a new graph and the training iterates until the policy converges.

Vanilla Q Learning: The agent in GraMeR trains via double Q-learning as it is a discrete finite MDP. The vanilla Q-learning maximizes a cumulative reward of actions taken during the interactions of agent with environment [146]. Reward at future times depends on actions taken at current time. The optimal value of an action (i.e., Q-value) corresponds to the optimal policy that maximizes the Q-value. Therefore, Q-value is iteratively updated according to Bellman equation as,

$$\begin{aligned} Q(X^t, a^t) &= Q(X^t, a^t) + \\ \theta * [r^t + \gamma \max_{a'} Q(X^{t+1}, a') - Q(X^t, a')] \end{aligned} \tag{7.8}$$

Q learning using Eq. (7.8) usually suffers from overestimation in practice due to the use of single estimator (Q-network) that determines the best action at next state with highest Q-value as well as the Q-value of that best action [147]. To avoid overestimation, double Q-leaning is proposed in [148], that uses two different estimators. One estimator (Local network Q^L) determines the best possible action for the next state and the other (target network Q^T) provides the Q-value of the selected action. The modified update equation of $Q^L(X^t, a^t)$ is,

$$\begin{aligned} Q^L(X^t, a^t) + \theta * [r^t + \gamma Q^T(X^{t+1}, a^*) - Q^L(X^t, a^t)], \\ a^* = \arg\max_{a'} Q^L(X^{t+1}, a'), \end{aligned} \tag{7.9}$$

and θ is the tuning parameter. Local network (Q^L) is trained at every step of the episode by sampling a batch of data from replay buffer. Mean squared error loss between the predicted Q value (i.e., $Q^L(X^t, a^t)$) and the desired Q value from the bellman equation ($r^t + \gamma Q^T(X^{t+1}, a^*)$) is minimized to update the parameters of the local Q^L network. While, the target network Q^T is not explicitly trained at every step, it is continuously updated with the weights of entire Q^L network after a certain number of episodes.

Meta Q Learning: A meta-learning attribute is introduced in the GraMeR to solve unseen tasks fast and efficiently. Here, the agent is expected to generalize to new graph types that have never been encountered during training. Typically, the Meta reinforcement learning approach contains two optimizer loops [149]. The outer optimizer samples a new environment in every iteration and adjusts parameters that determine agent behavior. In the inner loop, the agent interacts with the environment and optimizes for maximum reward. As in most of the environments (such as mazes, self driving car, etc.), it is not feasible to obtain a representing vector for entire environment. Therefore, learning across environments is captured via an outer optimizer. However, in the case of AIM, the environment is a graph, and it can be represented very accurately by a single graph embedding vector [150]. Hence, we skip the outer optimizer and feed the entire environment (graph) information (graph

embedding vector) along with state and actions to the training algorithm. This enables Q learning to capture the variation of environment via a single optimizer. Then the exact update equations of our agent turns out to be,

$$\begin{aligned} a^* &= \arg\max_{a'} Q^L(X^{t+1}, a', G_i) \\ Q^L(X^t, a^t, G_i) &= Q^L(X^t, a^t, G_i) + \\ \theta * [r^t + \gamma Q^T(X^{t+1}, a^*, G_i) - Q^L(X^t, a^t, G_i)] \end{aligned} \tag{7.10}$$

where, G_i corresponds to the sampled graph for the i^{th} episode. It is worth noting that graphs changes with episodes but for a particular episode, a single graph is explored.

GNN encoding: Similar to the graph, we also need representing vectors for the state (partial solution set) as well as action (node). In this regard, we leverage GraphSAGE [107] to estimate node embeddings. Thereafter, the embedding vectors of nodes in the state $\mathbf{S^t}$ are aggregated via mean/max operation to obtain a single representing vector $\mathbf{X^t}$ for the entire state. As action corresponds to a single node, it is represented by the corresponding node embedding vector. GraphSAGE is selected over conventional GCN [101] due to the factor that the candidacy of a particular node to be a part of AIM solution set depends mostly on its sub-graph. Therefore, GraphSAGE being an indutive sub-graph based learning approach is an appropriate choice. Further, GraphSAGE does not demand entire graph information unlike GCN, which requires the entire adjacency matrix and thus does not scale well with the size of the graph.

Multi-objective shaping: Along with states and action, the reward function needs to be explicitly designed for AIM as it comprises of multiple objectives. The first objective is related to maximizing influence spread and the other goal corresponds to maximizing intrinsic probability of seed nodes. Therefore, GraMeR belongs to the category of multi-objective DRL [151]. We take a single policy approach to learn one optimal policy by combining the two objectives with a known preference weight. Precisely, at each step of the episode, rewards

(Eq. (7.7)) are computed individually for each objective and accumulated in the buffer along with state and actions. Then, Q values of two objectives are combined using linear scalarization technique to generate a single Q value that is used to select an action. This is different than a typical approach of combining rewards into a single value and consequently learning single Q value and it is shown to be a stable and efficient [152]. Algorithms 1 and 2 summarizes the entire GraMeR involving GNN based state/environment representation, meta learning across different graph types and multi-objective rewards shaping.

Algorithm 4 Pseudocode of GraMeR

Input: set of training graphs G, input node features X, set of candidate nodes C.
Output: Trained GraMeR agent.

Initialize: State $X \leftarrow$ Max $\{h_u \forall u \in S\}$
Initialize: Graph embedding to $G_i \leftarrow \text{Max}\{h_u \forall u \in V\}$
Initialize: Action a corresponds to selection of node $u \leftarrow h_u$ where h_u denotes node embedding from GraphSAGE.
Initialize: Some arbitrary values to $Q_1(X, a)$, $Q_2(X, a)$ for all state-action pairs.
for Loop for each episode **do**
  Initialize: X^0 to initial state (random node from C)
  for Loop for each step of episode until state is terminal **do**
    Select node a^* from *scalarized action selection strategy*
    Append node a to partial solution set S.
    Obtain rewards $r1, r2$ from both objectives.
    Move to next step $X^{'}$.
    Update Q^L of both objectives.
    $Q_1^L(X, a, G_i) = Q_1^L(X, a, G_i) + \theta * [r + \gamma Q_1^T(X, a^*, G_i) - Q_1^L(X, a, G_i)]$
    $Q_2^L(X, a, G_i) = Q_2^L(X, a, G_i) + \theta * [r + \gamma Q_2^T(X, a^*, G_i) - Q_2^L(X, a, G_i)]$
  end for
end for
update Q_1^L and Q_2^L parameters by minimizing cross entropy loss between target values from bellman equation and actual prediction from networks.
update Q^T network by copying Q^L parameters to Q^T at every U steps.
return Trained models Q_1^L and Q_2^L

7.5 Experimental Results

This section validates the proposed framework against modified greedy hill-climbing algorithm (MGHC) and S2VDQN. GraMeR offers improved performance with more flexibility

Algorithm 5 Scalarized action selection

Input: Q values of both objectives for current state-action pair (X, a).
Output: action a^*.
Qnew $= W_1 * Q_1(X, a) + W_1 * Q_2(X, a)$
Greedy action selection

$$a^* = \begin{cases} \arg\max A_{a'} & \text{probability } 1 - \epsilon \\ \text{random selection from set C} & \text{probability } \epsilon \end{cases}$$

3: **return** action a^*

while being orders of magnitude faster. The performance is examined on standard networks, namely Barabasi Albert, Power law cluster, stochastic block models. As per the architecture, 1^{st} phase of the local Q-network (Q^L) comprises of 3 GNN layers for generating node embeddings. The number of neurons in these layers are 64, 32 and 16, respectively. This phase generates state, action and graph embedding vectors which are concatenated into a vector and passed through a regression phase of the Q^L network. Regression phase consists of 2 feedforward layers with 16 and 1 neurons respectively. All other settings have been discussed in the appendix.

7.5.1 Baselines

We have first validated the performance of the proposed candidate node prediction model with Greedy Hill-climbing (GHC) algorithm that selects seed nodes based on the marginal gain in influence spread. Then, GraMeR (core module) is compared with modified greedy Hill-climbing (MGHC) algorithm [137] that sample nodes based on intrinsic probability and select seed nodes which provide high gain in spread. Although this algorithm strives for two objectives (i.e, high intrinsic probability and high marginal gain in influence spread) of AIM, there is no inbuilt mechanism to control their priorities. In fact, none of the recent data-driven work addresses this type of multi-objective formulation of IM [153, 154]. In contrast, we have systematically incorporated the multi-objective formulation with a controlled weighing factor α. Furthermore, the baseline also includes modified S2V-DQN (MS2V-DQN)

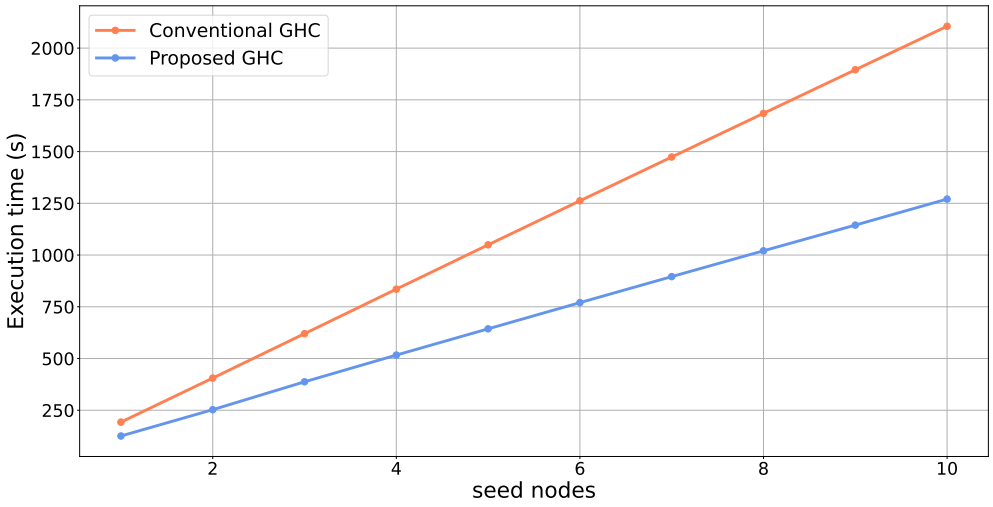


Figure 7.2: Time in identifying seed nodes with and without candidate nodes

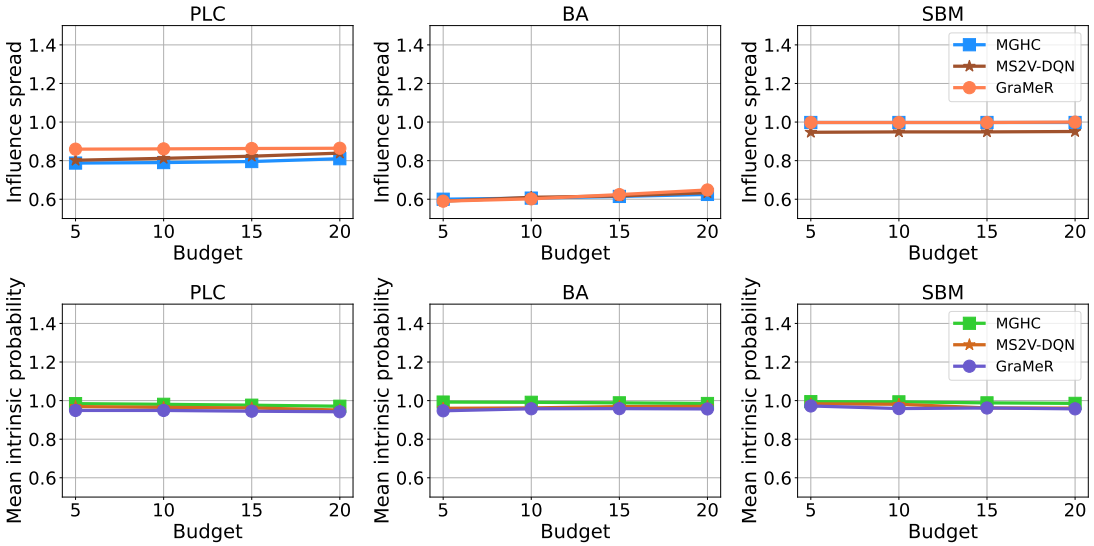


Figure 7.3: Spread/Intrinsic probability vs Budget. PLC: Power law cluster; BA- Barabasi-Albert; SBM-Stochastic block model

[153] that combines RL with graph representation. The baselines implementation is further discussed in the Appendix.

7.5.2 Results

This section demonstrates the performance of proposed framework against baselines via performance metrics and execution times. The evaluation is repeated 100 times and aver-

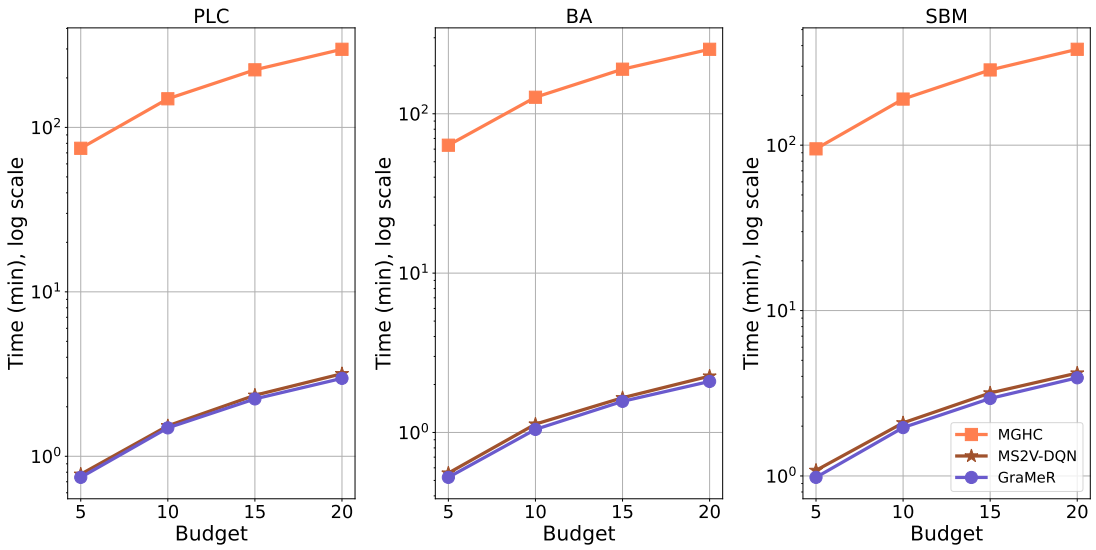


Figure 7.4: Running time vs Budget. PLC: Power law cluster; BA- Barabasi-Albert; SBM- Stochastic block model

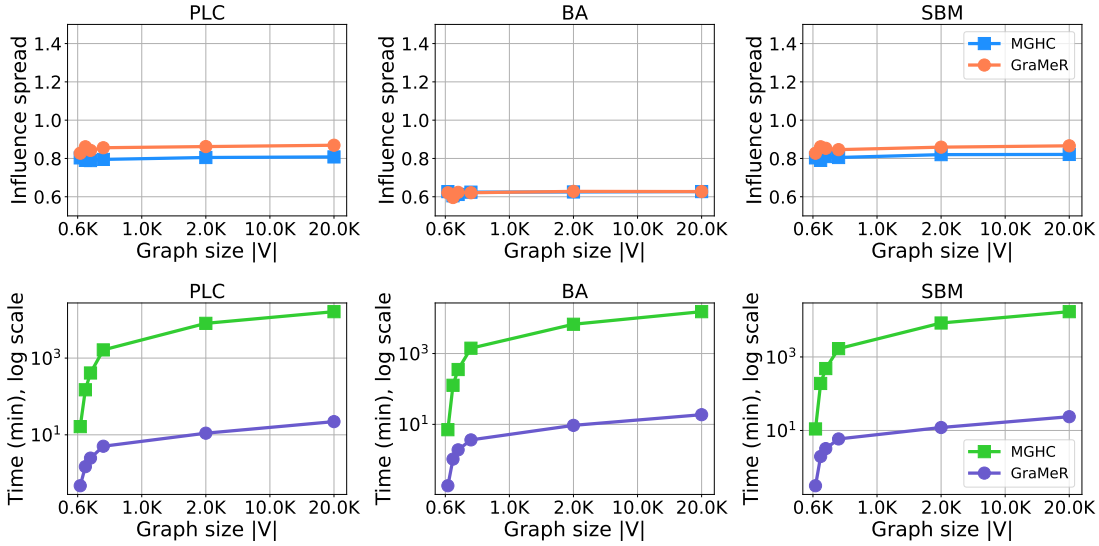


Figure 7.5: Mean Spread and Running time vs graph size for three graph types

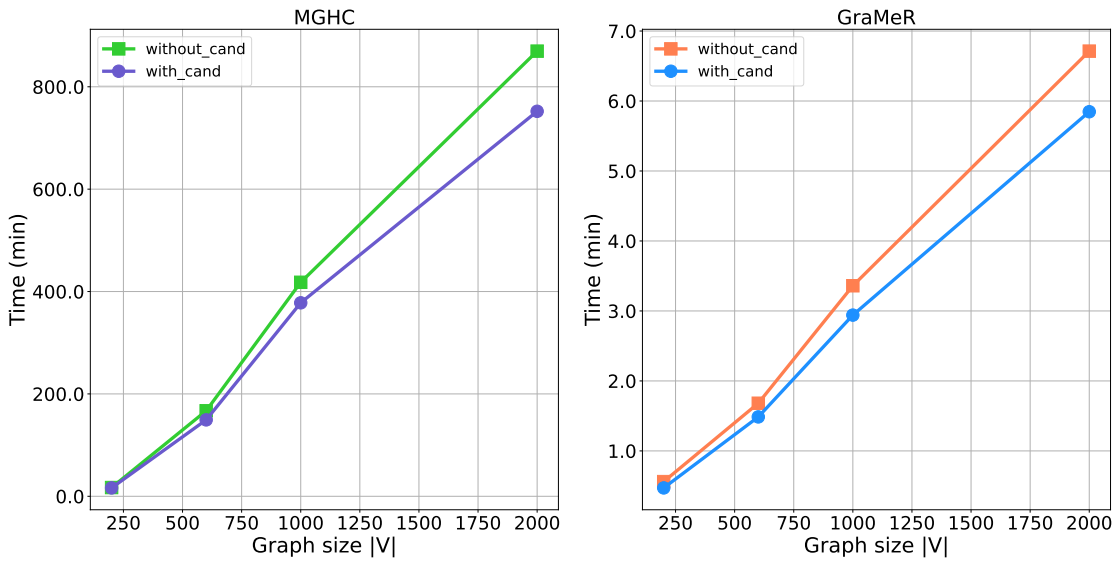


Figure 7.6: Running time vs graph size of GraMeR and MGHC for PLC graph

age scores are reported for each test scenario. The source code can be found in the following anonymized github link https://anonymous.4open.science/r/InfluenceMaximization-Deep-QLearning-B20B.

Performance of Candidate node predictor

The candidate node prediction module is trained on 6 graphs of varying dimension (200 to 400 nodes) and type (BA, PLC, BP). The ground truth value of node classes is obtained using influence capacity metric as discussed in Eq.(7.5). The model is evaluated on 3 test graphs of 1000 nodes, one from each graph family. The mean classification accuracy in detecting candidate and non-candidate nodes is around 96.24 % with a recall score (class-1 accuracy) of 97.95 %. Further, it has been shown via Fig. 7.2 that there is a noticeable reduction of 40% in algorithm running time compared to the greedy hill-climbing approach. This is due to a significant reduction in search space which can immensely speedup the training of GraMeR agent in later part of the learning pipeline.

Accuracy of GraMeR

The performance of GraMeR for activation aware influence maximization can be evaluated via two metrics namely influence spread and node intrinsic probability. Expected influence spread provides the mean number of nodes getting influenced in the network if seed nodes in the solution set are activated with corresponding intrinsic probabilities and information is spread via IC diffusion model. Influence spread is normalized between 0 to 1 for comparison across different graph sizes. The second metric (i.e., intrinsic probability) represents the probability of seeds nodes being activated on their own. The mean probabilities for all the seed nodes in the estimated solution set are reported.

Fig. 7.3 compares expected influence spread and mean intrinsic probability across different graph types and budget. It can be observed that the proposed GraMeR is at par with the baseline MGHC with much lower computational effort as demonstrated in the next subsection. Specifically, GraMeR consistently outperforms MGHC and MS2V-DQN in terms of spread whereas the reverse phenomenon can be seen in case of intrinsic probability. This is because the MGHC and MS2V-DQN does not have a mechanism to balance between influence spread and intrinsic probability. Therefore, they always provides the seed nodes having high intrinsic probability but sub-optimal in terms of influence spread. The scale of the plots is kept same for a fair comparison across different graph types. Further, it can be seen that for a similar graph size (600 nodes) and diffusion model, the influence spread is maximum in case of SBM graph and minimum in BA. This is due to the fact that SBM have community structures where subsets of nodes are connected with each other through a large link densities. AIM led to the selection of seed nodes from different clusters which results in a high influence spread compared to BA graph that misses such clusters.

Computational gain of GraMeR

One of the key advantages of the proposed approach is the computational efficiency which is demonstrated via algorithm running times i.e. wall clock time. Fig. 7.4 depicts the

running times across different budgets and graphs. It can be inferred that as the budget increases, the search time increases which is very intuitive. However, the increase in time is very sharp (linear for the experimented budgets) in the case of baseline MGHC, while it is almost constant for the proposed GraMeR. This is because the trained deep Q networks in GraMeR computes the solution set via forward propogation which mainly involves matrix operations. On the other hand, the MGHC searches for increasing number of nodes as the budget increases which eventually demands the computation of influence spread for large number of sets. Fig. 7.4 also illustrates that search algorithms for AIM are fastest in BA and slowest in SBM. This observation could be attributed to the high clustering coefficient in SBM leading to a large time cost associated with shifting from one cluster to an other while searching for an optimum seed set.

Generalizability of GraMeR

The core theme of our framework reinforces the property of generalizability since several types of real-world and synthetic networks exist in the literature. Many of these graph types have only slight variation in their topological property. Thus, a separate DRL model to identify AIM seed nodes for each of these graph types is not needed rather, a meta learning can serve the purpose by learning across different environments (graph types). This fact is demonstrated by training on two graph types (i.e., PLC and BA) while validating on all the three graph families. The performance in terms of influence spread, intrinsic probability and running time is consistent across all three graphs as shown in Figs. 7.3 to 7.6.

Scalability of GraMeR

Scalability of GraMeR is summarized in Fig. 7.5, which presents the compute time for different graph sizes. Here, the model is trained on 8 graphs (4 from PLC and BA) each having 400 nodes. Then, it is tested on graphs of sizes varying from 200 to 20000. It can be inferred that GraMeR outperforms the baseline without any significant impact on

computational effort. Specifically, running time for MGHC scales by 1500 for $10x$ increase in graph size whereas, it remains almost constant for GraMeR. Further, to reinforce the scalability benefit, the training of GraMeR is carried for a fixed budget of 10 but prediction is carried out for multiple budgets from 5 to 20. The scalability can further be enhanced through distributed computing as shown by [155]. This work will address scaling to larger graphs on a wide variety of platforms (GPUs), which we plan to pursue in our future work.

Ablation study

The proposed GraMeR has computational supremacy over conventional methods because of two factors: (1) Deep Q networks based meta reinforcement learning approach to identify AIM solution set; and (2) candidate node predictor that reduces the search space. Fig. 7.6 depicts the ablation study results where running time is monitored for GraMeR and MGHC with and without the candidate node prediction module. The budget is fixed as 10 and prediction is done for the PLC graph across different sizes. It can be seen that the time gap between the two cases increases with the graph size with nearly 1 minute for GraMeR and 110 minute for MGHC. This gap will further increase as the network size grows. Further, apart from prediction, noticeable gap is also seen in training time. This demonstrates the importance of the node prediction module in GraMeR.

7.6 Summary

This chapter presents a GNN fused meta reinforcement learning framework (GraMeR) for identifying influential nodes in a network. Firstly, the search space of IM is reduced via GNN based candidate node predictor. Then deep Q learning is employed to learn to identify IM seed nodes with GNN as environment encoders. The unique aspects of GraMer lies in its computational efficiency and generalizability. Next chapter presents a comparative study of various performance and network-based robustness metrics.

Chapter 8

Network vs Performance-based Robustness metrics - Smart Grid Case study

This chapter presents a comparative study of various performance and network-based robustness metrics that have been explored till now. Chapter 5 presents a scalable graph neural network-based framework for computing network-based failure metrics. Authors in [96, 156] have conducted a comparative study of several robustness metrics, but they are analyzed from the perspective of general complex networks. Though plenty of robustness metrics are defined for power distribution systems (PDS), there is no systematic study on their coherency and applicability. By coherency, we mean the similarity or dissimilarity (consistency) in the rankings of critical nodes from various robustness indices.

Furthermore, Chapter 4 develops a computationally efficient analytical framework for determining a performance-based voltage fluctuation metric. In general, very little attention has been devoted to defining node-level metrics for accurately quantifying voltage fluctuations, primarily due to their complex computation process. This necessity the need for more metrics in this area. Hence, it is worthwhile to explore the potential of readily available

robustness metrics against system failure to quantify voltage fluctuations.

Therefore, this chapter first conducts a comparative study of various robustness metrics to analyze their coherency, and then investigate their efficacy in characterizing the impact of voltage fluctuations. This comparative study enables us to highlight the benefits and limitations of existing methodologies. Lack of efficacy in robustness evaluation, particularly due to modeling inefficiency, is addressed through a novel modeling framework in Chapter 9.

8.1 Robustness metrics

The term "robustness" in this work is primarily concerned with the drop in performance of a power grid when a disruption occurs [157]. Several metrics have been proposed in the literature to study and improve the robustness against system failures [31, 157–160]. These approaches can be divided into three classes, namely network (topology) based, performance (power flow, system dynamics) based and hybrid which combines topology with electrical properties of the network. Further, this section also describes a voltage variation metric, which would later be compared with the robustness metrics to system failure. Figure 8.1 depicts the the taxonomy of robustness metrics studied in this work.

8.1.1 Robustness to System failure

The failure of the system corresponds to the loss of connectivity which further leads to the loss of power flow to the end users. Some of the widely used metrics to determine system robustness to failures include effective graph resistance, flow robustness, etc. More importantly, different nodes contribute differently to the overall robustness. Hence, some nodes could be more critical compared to others in terms of inducing cascading failures once they are affected. Relevant metrics from each of the three introduced earlier classes are summarized below.

Network-based metrics

- **Network efficiency (NEF):** It is the communication effectiveness of a networked system. That is,

$$E = \frac{1}{N(N-1)} \sum_{u \neq v} \frac{1}{d_{uv}}, \tag{8.1}$$

is a measure of the network performance under the assumption that the efficiency for sending load (electricity, information, packets, whatsoever) between two nodes u and v is proportional to the reciprocal of their distance. Based on this definition, the robustness of a network can be defined as the drop in the efficiency when a node u is removed from the network, i.e.,

$$R_e(u) = \frac{E - E_u}{E}, \tag{8.2}$$

where E_u is the score after removing node u from the network.

- **Betweenness centrality:** It quantifies how often a node u occurs in the paths linking other pairs of nodes. That is,

$$C_B(u) = \sum_{s=1}^{n} \sum_{t=1}^{n} \frac{\sigma_{st}(u)}{\sigma_{st}}, s \neq t \neq u \in V. \tag{8.3}$$

where, σ_{st} is the total number of shortest paths from node s to t, and $\sigma_{st}(u)$ is the number of those paths that pass through u.

- **Effective node resistance (ENR):** It is related to the eigenvalues of the graph Laplacian matrix and corresponds to [53]

$$R(u) = \frac{2}{N-1} \sum_{u=1}^{N-1} \frac{1}{\lambda_u}, \tag{8.4}$$

where λ are the non zero eigen values of graph Laplacian matrix. Recently, authors in

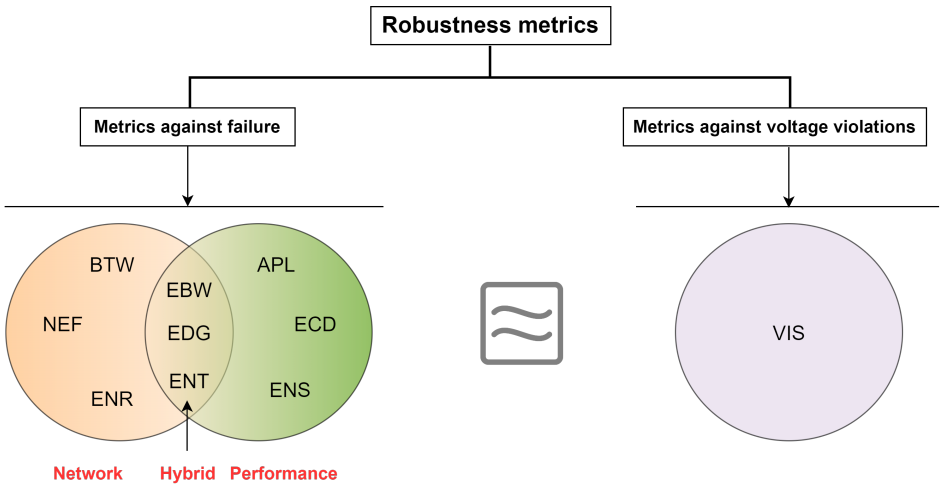


Figure 8.1: A taxonomy of Robustness metrics for power distribution network

[161] introduce the metric Effective Node Resistance by extending the notion of EGR from graph level to node level. Essentially, it is defined as the sum of all node-to-node effective resistances (R_{uv}) with all other nodes. It can be expressed as:

$$R_e(u) = \sum_{v \in V} R_{uv}. \tag{8.5}$$

It has been shown that ENR correlates well with other standard robustness metrics against node and link failures [161].

Performance-based metrics

The performance of the network with respect to robustness is typically measured via loss of active power due to disruption. Some of the relevant metrics under this category are:

- **Electrical coupling connection degree (ECD):** For a power network with total N nodes, the electrical coupling connection degree D_u for the node u is defined as [162]:

$$D_u = \left| \frac{1}{\sum_{v=1, u \neq v}^{N} Z_{uv}^{equ}} \right| \tag{8.6}$$

where, Z_{uv}^{equ} is the electrical distance, and can be described as the equivalent impedance

between two nodes (similar to the concept of distance in graph theory). It is computed for each node through entries in the node impedance matrix as [162]:

$$Z_{uv}^{equ} = (Z_{uu} - Z_{uv}) - (Z_{uv} - Z_{vv}), \tag{8.7}$$

where, Z_{uv} is the branch impedance between node pair u and v. ECD can be used to rank the nodes based on electrical attribute. The shorter the electrical distance, greater the dependence will be. A node with a high electrical coupling connection degree has a high current transmission capability and a high electrical dependence on the other nodes. When this node fails, it will soon induce power flow changes among numerous nodes, resulting in a cascade of failure.

- **Active power flow loss (APL)**: The power supplied (PS) is an essential measurement in DC power flow model. Its loss can be evaluated as [160]:

$$P_l = 1 - \frac{\sum_{u=1}^{N} P_u^d}{\sum_{u=1}^{N} P_u^n}, \tag{8.8}$$

where P_u^n is the active power of the node u under normal operating conditions of the grid, and P_u^d is the power after failure.

- **Electrical node significance (ENS)**: Typically in power grids, some nodes serve as hubs distributing a large amount of power while others distribute very little. A considerable quantity of power is exposed to the rest of the network when a link from one of the hub nodes fails. Redistributing this extra power to neighboring components gradually leads to more link overload failures, which can lead to a large-scale power outage. However, if a link connected to a less critical node fails, the power is instantly re-routed to surrounding components, and the disruption is usually mitigated. This implies that nodes have varied effects on the cascading failure robustness, and that this impact is dependent on the quantity of power distributed by the associated node.

Therefore, ENS is introduced in [163] to determine the impact of a node as:

$$\delta_u = \frac{P_u}{\sum_{v=1}^{N} P_v}, \tag{8.9}$$

where, P_u denotes the total power distributed by node u, and N is the number of nodes in the network.

Hybrid metrics

The unexpected behavior in the power system is related to both the topological (location of events, interconnection of components) and the operative state (flow distribution, demand level, etc.) of the system. Therefore, there is a growing interest in combining both the factors to obtain novel metrics that can better capture network robustness to failures [160]. Some of the hybrid metrics that have been found to work well are presented below:

- **Electrical betweenness (EBW)**: The electrical betweenness centrality of a node u in a network of N nodes is defined as [164];

$$C_B^E(u) = \sum_{s=1}^{n} \sum_{t=1}^{n} \frac{P_{st}(u)}{P_{st}}, s \neq t \neq u \in V, \tag{8.10}$$

where, P_{st} is the maximum power flowing in the shortest electrical path between nodes s and t, and $P_{st}(u)$ is the maximum of inflow and outflow at bus k within the shortest electrical path between nodes s and t.

- **Electrical degree (EDG):** The power flowing in the adjacent links of the target node can be considered as electrical degree of the node and can be written as [164],

$$C_D^E(u) = \frac{\sum_{u \sim v} P_{uv}}{N-1} \tag{8.11}$$

where, $u \sim v$ indicates that node u and v are connected. P_{uv} represents power flowing in line connected in between nodes u and v.

- **Electrical node robustness (ENT):** It quantifies the ability of a node to resist cascades of link overload failures. It incorporates both flow dynamics and network topology. ENT of a node u (i.e. $R_n(u)$) can be expressed as [163],

$$R_n(u) = -\sum_{l=1}^{L} \alpha_l P_l log P_l, \tag{8.12}$$

where, L refers to the out-degree of the corresponding node, α denotes the loading level of the line, and P_l corresponds to normalised flow values on the out-going links given as,

$$P_l = \frac{f_l}{\sum_{l=1}^{L} f_l}, \tag{8.13}$$

where, f_l refers to the power flow in line l. The formulation is similar to entropy because entropy of a load distribution of a node increases as flows over lines are distributed more homogeneously and the node out-degree increases. The loading level of the line has inverse relationship on robustness, therefore α is inversely proportional to loading level. Thus, higher the value of nodal robustness, the more robust the node is.

8.1.2 Robustness to voltage fluctuations

Apart from the robustness metrics to node failures, it is also important to investigate metrics to quantify voltage variations since we are interested in comparing the efficacy of these metrics in characterizing robustness to voltage fluctuations. In this regard, one of the relevant metrics for voltage variations is the Voltage influencing score (VIS) which can be expressed as [29],

$$VIS(O, A) = \frac{\frac{1}{D(A,O)} - \frac{1}{D(S,O)}}{\frac{1}{D(A',O)} - \frac{1}{D(S,O)}}, \tag{8.14}$$

where, $D(A, O)$ is the statistical distance between the voltage change distribution at a target node O due to aggregate effect of all actor nodes (actor node refers to node where power varies) and when actor node A is solely present in the system. $D(S, O)$ is the statistical

distance between source node and observation node. The distance can be computed with any of the information theoretic metrics such as KL divergence, Frechet distance, among others. To provide an absolute value to the score, VIS is normalized with minimum and maximum values. As VIS is defined for a pair of nodes, the net influencing capacity of a particular node can be determined by averaging its score across all the other nodes of the network. The lower the distance, the more the actor node A contributes to aggregate voltage change and consequently the more influencing the actor node is and vice-versa.

Table 8.1: Robustness metrics and their relationship with critical node ranking. Up-arrow denotes increasing value and down-arrow for decreasing value.

Metric	**Class**	**Criticality ranking**
Betweenness (BTW) ↑	Network	↑
Network efficiency (NEF) ↑	Network	↑
Effective node resistance (ENR) ↑	Network	↓
Active power flow loss (APL) ↑	Performance	↑
Electrical coupling connection degree (ECD) ↑	Performance	↑
Electrical node significance (ENS) ↑	Performance	↑
Electrical betweenness (EBW) ↑	Hybrid	↑
Electrical degree (EDG) ↑	Hybrid	↑
Electrical node robustness (ENT) ↑	Hybrid	↓
Voltage influencing score (VIS) ↑	Hybrid	↑

8.2 Comparative results and Discussion

The metrics that have been developed under the network and performance-based robustness analysis paradigms can be used to identify critical nodes for enhancing network resilience against failures. This section presents a comparative analysis of metrics introduced in the previous section. To this end, this study first investigates the coherency of these metrics in explaining the system's robustness. Then, we analyze the efficacy of the methods in representing cross-domain metrics, i.e., leveraging robustness metrics for system failure to quantify voltage variations.

Table 8.1 reports the metrics studied in this work and their relationship with criticality

ranking. Criticality ranking denotes the ranking of nodes based on the robustness metric scores. The lower the rank, the more important the node is with respect to network robustness. Depending upon the metric formulation, increasing metric values have a positive or negative impact on the ranking. For example, a higher value of the node betweenness score implies a better node criticality ranking, whereas the opposite effect can be seen in the case of electrical node resistance. Furthermore, this paper considers the IEEE-37 node network as a test system.

Table 8.2: Ranking of nodes in regard to network-based metrics

Metrics	**BTW**		**NEF**		**ENR**	
Rank	Node	C_b	Node	E	Node	R
1	14	0.050	9	0.067	9	0.627
2	9	0.047	28	0.056	14	0.635
3	18	0.029	14	0.052	28	0.734
4	28	0.021	18	0.026	18	0.796
5	30	0.004	30	0.009	30	0.958
6	7	0.0	22	0.004	22	0.969
7	8	0.0	34	0.003	34	0.981
8	12	0.0	17	0.003	36	0.982
9	17	0.0	36	0.003	17	0.983
10	22	0.0	31	0.002	31	0.987
11	26	0.0	7	0.002	7	0.991
12	27	0.0	8	0.002	8	0.991
13	31	0.0	26	0.001	26	0.997
14	34	0.0	27	0.001	27	0.997
15	36	0.0	12	0.001	12	0.998

8.2.1 Network based metrics:

For network-based analysis, the IEEE-37 test network is abstracted as a directed weighted graph, with nodes representing buses (nodes) of the distribution system and links corresponding to physical connections between buses. The test network under this scenario is solely analyzed from the perspective of topological (i.e., network) features. Radial distribution networks like the IEEE-37 node network are not scale-free like other engineered

networks, since their degree distribution follows a uniform distribution. We have selected three widely used effective metrics under this category, namely betweenness (BTW), network efficiency (NEF) and effective node resistance (ENR). Further, there are 15 actor nodes in the test network, where power can vary. These 15 actor nodes are as follows: $7, 8, 9, 12, 14, 17, 18, 22, 26, 27, 28, 30, 31, 34, 36$.

Table 8.2 tabulates the criticality ranking of 15 actor nodes with the three selected metrics. Although power measurements are not utilized to determine the ranking in this case, they are provided for these 15 nodes to allow for a fair comparison to other approaches. Table 8.2 also includes metric values in addition to node numbers, denoted by "Node". It can be seen that the betweenness values are non-zero only for the top 5 nodes, indicating that only the rank of the Top 5 nodes makes sense. All the remaining nodes can be arranged in any manner, and the presented rank in Table 8.2 is merely one among several ways. Since none of the shortest paths pass through leaf nodes (nodes in the periphery of the distribution network), their betweenness scores are zero, making this metric less effective in distinguishing nodes. This shortcoming is addressed in both NEF and ENR. They provide differentiating values to different nodes as much as possible. However, nodes lying in the lower score band are less distinguishable compared to those in the upper band.

It is also evident from Table 8.2 that the rankings of Top-1 or Top-5 nodes are different with different metrics. For instance, BTW assigns the top position to node 14, whereas the other two metrics place node 9 in position one. Furthermore, the correlation between the rankings of NEF and ENR is high compared to that of BTW. However, it is interesting to note that although different metrics offer different ranks, there is a consistency in the Top-5 nodes when seen as a set rather than ranked entries, i.e., all three metrics assign nodes $9, 14, 18, 28, 30$ in Top-5 positions. Thus, in applications where node selection has major computational and financial implications, i.e., high discriminatory scores are necessary, the priority of metrics usage should be ENR followed by NEF and BTW.

8.2.2 Performance based metrics:

The term "performance" in this category of metrics refers to the electrical attributes of the distribution network, such as power flow in lines, voltage quality, etc. The nominal voltage of the test system is 4.8 kV, and the base load is kept the same as reported in the IEEE PES distribution system subcommittee report. Furthermore, for better generalizability, different load scenarios are simulated by varying the base load at 15 actor nodes. The mean results are reported from over $100,000$ different simulations.

Table 8.3: Ranking of nodes in regard to performance-based metrics

Metrics	**APL**		**ECD**		**ENS**	
Rank	Node	P_l	Node	D	Node	δ
1	9	0.506	28	0.267	36	0.210
2	14	0.331	9	0.258	18	0.209
3	28	0.245	26	0.217	26	0.130
4	18	0.232	7	0.187	9	0.130
5	36	0.083	12	0.208	28	0.127
6	30	0.063	30	0.202	27	0.127
7	26	0.050	14	0.187	31	0.064
8	31	0.050	27	0.170	14	0.064
9	27	0.050	8	0.159	22	0.064
10	8	0.024	31	0.145	12	0.064
11	22	0.024	34	0.141	7	0.063
12	17	0.024	22	0.140	17	0.063
13	34	0.024	18	0.140	34	0.063
14	12	0.024	36	0.134	8	0.063
15	7	0.024	17	0.105	30	0.031

Table 8.3 illustrates the ranking of 15 actor nodes for the three electrical-based robustness metrics, namely active power flow loss (APL), electrical coupling connection degree (ECD), and electrical node significance (ENS). It can be observed from the very first glance that the ECD has the largest discriminatory power compared to the other two. In comparison to network-based metrics, here the ranking among the three metrics is relatively less consistent, likely due to the diverse electrical attributes employed to compute these ranks. However, considering the Top-5 node set, APL and ENS appear to match to a greater extent compared

to ECD. This is because, both APL and ENS involve node power in their formulations, unlike ECD, which deals with the branch impedances. Overall, the performance-based rankings are noticeably different than those of network one, except for a few nodes such as node 9, which is consistently present in the Top-5 nodes in all the cases. In a nutshell, one must experiment with different metrics before determining the relevant metric for a specific use case.

Table 8.4: Ranking of nodes in regard to hybrid metrics

Metrics	**EBW**		**EDG**		**ENT**	
Rank	Node	C_b^e	Node	C_d^e	Node	R_n
1	14	0.682	9	0.99	14	6.365
2	9	0672	14	0.879	7	0.000
3	18	0.404	28	0.698	8	0.000
4	28	0.252	18	0.558	9	0.000
5	30	0.034	30	0.266	12	0.000
6	7	0.000	36	0.199	17	0.000
7	8	0.000	26	0.121	18	0.000
8	12	0.000	27	0.121	22	0.000
9	17	0.000	31	0.121	26	0.000
10	22	0.000	8	0.06	27	0.000
11	26	0.000	7	0.06	28	0.000
12	27	0.000	34	0.06	30	0.000
13	31	0.000	12	0.06	31	0.000
14	34	0.000	22	0.06	34	0.000
15	36	0.000	17	0.06	36	0.000

8.2.3 Hybrid metrics:

Hybrid approaches consider both topological and electrical attributes of the network. The base loads, actor nodes, and other settings are kept the same as in the previous two cases.

Table 8.4 presents the ranking of nodes based on three powerful hybrid metrics, namely electrical betweenness (EBW), electrical degree (EDG), and electrical node robustness (ENT). Similar to the performance-based metric case, there is no consistency in ranking among the three approaches. However, there is noticeable consistency with respect to the Top-5 nodes set. In particular, the rankings of EBW and EDG have a high correlation compared to that

of ENT, although all three involve power flows in their formulations. Another distinguishing characteristic of hybrid metrics is their substantially lower discriminatory power compared to performance and network-based methods. In fact, only the top nodes have distinct values, and the remaining bottom nodes have almost zero values in EBW and ENT. Specifically for ENT, most of the nodes have zero values, either because of no outgoing links (as appears to be the case in leaf nodes) or because of a single outgoing link. Thus, although these hybrid approaches seem to be more elegant in their formulation due to the incorporation of both electrical and topological features, they are not very effective in identifying critical nodes, as evident via entries in Table 8.4.

Overall, when comparing the rankings of three approaches, node 9 appears to be in the first position in a majority of cases, followed by node 14. For network-based metrics, it is the central position of nodes that makes them critical. As for hybrid cases, both power flow and central position make them strong candidates. Nevertheless, node 36 lies in the bottommost position for most of the metrics, followed by 34. These nodes are leaf nodes whose loss would have a minimal impact on the electrical connectivity of the major part of the network.

8.2.4 Robustness metrics for voltage fluctuations

It is interesting to study whether robustness metrics to system failure can capture voltage variations. To this end, we must first rank the actor nodes based on their voltage influencing capacity (VIC). VIC refers to a node's capability of inducing voltage variations in other nodes of the network, and it depends on the node's position as well as its power variation. The experimental setup for this case study is kept the same as in the case of previous metrics. The only extra factor to consider here is the probability distribution of power changes with which different power change scenarios are simulated. This distribution is needed since we rely on [29] for determining the VIC of nodes, which essentially utilizes power change distribution and topological information such as shared path impedances. The change in real and reactive power at 15 actor nodes is modeled as a zero-mean Gaussian random variable.

The Gaussian distribution is commonly used to validate statistical frameworks, and it has been considered a common assumption in many prior works related to distribution systems [92, 165]. Furthermore, a covariance matrix consisting of power change variances and covariances is required which can be learned from the historical data as illustrated in [29]. In this regard, three different power change scenarios are considered with different means and variances similar to eqn. (4.15) in Chapter 4. where ΔS signifies the power change vector across all actor nodes, A denotes the actor node set, and superscript over actor nodes, i.e., $\{a, b, c\}$ represent respective phases at which power is varying. Changes in power across different actor nodes can be correlated because of the geographical proximity of DERs (PV and wind turbines). The diagonal elements of covariance matrices represent the variance of change in real and reactive power at actor nodes, while off-diagonal elements reflect the covariance between real and reactive power change as shown in eqn. (**??**).

Table 8.5: Ranking of nodes based on voltage influencing capacity. Obs: Observation nodes

Node Rank	**Obs Node7** Node	**Obs node-16** Node	**All nodes** Node
1	7	14	7
2	9	22	8
3	12	12	9
4	14	9	12
5	22	7	14
6	26	17	18
7	8	26	22
8	8	18	26
9	17	28	28
10	18	8	30
11	27	27	31
12	30	34	17
13	31	31	27
14	34	34	34
15	36	36	36

Table 8.5 depicts the ranking of 15 actor nodes for various scenarios. The voltage influence score is computed in a pair-wise manner, with one node being an actor where power varies and the other being an observation node where voltage change is monitored [29]. For illustration,

the rankings in Table 8.5 are shown for two observation nodes, 7 and 16. Furthermore, one can compute the mean of each actor node's voltage influencing score across all observation nodes to determine its overall influencing capacity. The last column of Table 8.5 shows the average ranking of the actor nodes.

If we strictly compare the top positions of voltage influencing nodes with those of robustness metrics to system failure in tables 8.2, 8.3, 8.4, none of them seems to match. It is not even fair to make such a strict comparison since process of computing these two classes of metrics are quite different. In fact, it makes sense only to compare the hybrid robustness metrics with voltage influencing scores since both involve power flow and topological related characteristics, unlike network and performance-based metrics that only consider one of the two aspects at a time. EBW and EDG appear to match VIF ranking to a certain extent since they capture two of the Top-5 nodes, i.e., nodes 9 and 14. So, hybrid metrics can be safely employed in applications that require a set of critical nodes to monitor/control voltage fluctuations. There are numerous advantages to using these kinds of hybrid metrics for voltage fluctuations, including (1) light and easy computation; (2) no need to rely on computationally expensive simulations and other system states that are difficult to obtain.

8.3 Challenges and opportunities

The metrics studied in this work, reveal the importance of different factors in determining the system's robustness and the advantage of fusing performance and network-based methodologies. However, to enable widespread adoption, there is scope for improvement in robustness metric design. This section identifies key challenges that lie ahead and suggests future research directions in effective metric formulation.

8.3.1 Generic formulation

There are plenty of metrics developed for exploring various aspects of robustness in PDS. However, there is no clear consensus or analysis regarding their applicability, which ultimately leads users to experiment with different metrics before finding the most effective ones for their use case. In addition, most of the metrics are designed to work for a specific use case, rendering them useless for other applications. Thus, there is a pressing need to design robustness metrics with a generic formulation that can work for multiple applications.

One way to approach the generic paradigm is by incorporating all relevant aspects into the metric formulation itself, i.e., all the factors that contribute to the target objectives should be included in the metric formula. For example, in the ENT metric, the loading level of the line is explicitly added to the actual power flow to effectively account for the link overload failure. Similarly, the voltage influencing score of the studied node can be merged into the current ENT formulation to account for the voltage fluctuations. This will allow the metric to simultaneously account for two objectives, i.e., link overload failure and voltage fluctuations.

8.3.2 Scalable computation

Existing methods for computing robustness metrics, especially those involving electrical characteristics, are computationally intensive and do not scale well with the network size. As a result, robustness analysis on large test networks such as the IEEE 8500 node feeder or 10477 bus system is difficult and time-consuming [166–168]. Furthermore, for every small change in network configuration, the process of computing the metric needs to be repeated without leveraging any past solutions. Thus, a more elegant framework is required that can tackle these issues.

Data-driven models could be a potential candidate to address some of the computational shortcomings of the existing approaches. Deep neural network-based models can be trained to estimate robustness scores at the node level as well as at the graph level. For example,

some inspirations can be drawn from [32], where graph neural network is utilized to obtain critical nodes by training a node classifier. These models are easily scalable to larger networks since they only rely on the sub-graph or a relevant part of the network for any node/link level prediction rather than relying on entire network. Furthermore, this type of model can make predictions across networks of varying sizes. There are numerous other advantages compared to conventional approaches, such as the ability to capture a node's non-linear relationships with the robustness of the entire network, high expressive power due to a large number of model parameters, etc.

8.3.3 Holistic modeling

The majority of current modeling frameworks for studying robustness analyze the electrical network in isolation, without taking into account its interdependency with other coexisting critical infrastructures such as water, transportation, etc. In addition, disruptive events disproportionately impact the low-income and socially vulnerable communities underscoring the need for incorporating social equality via true assessment of community resilience. However, the robustness of complex systems is typically evaluated solely in terms of engineering attributes, such as network topology or electrical parameters, and tends to ignore social factors that are equally important. As a result, robustness evaluations of such decoupled and partially informed systems are sub-optimal. Thus, there is a need to develop a holistic modeling framework for accurate robustness assessment.

Stochastic hetero-functional graph theory (SHFGT), inspired by [157, 169] is one potential modeling framework to incorporate the above-discussed factors. SHFGT can effectively model complex interdependent systems, including electrical, power, and transport networks, via a set of graphs corresponding to different activities. Essentially, these graph-structured models leverage functionality as building blocks, unlike conventional frameworks that only describe physical attributes in terms of nodes. This allows them to efficiently integrate various aspects, including engineering and social robustness factors. Robustness assessment

within this kind of modeling framework would be more realistic and thus enable one to take effective decisions in improving system resilience.

8.4 Summary

This chapter presents a systematic study of different robustness metrics to system failure by comparing their similarity and dissimilarity in ranking critical nodes in a power distribution network. Then, the efficacy of these metrics in characterizing voltage fluctuations is accessed by comparing their rankings with that of voltage influencing scores. From experimental results, it appears that the hybrid robustness metrics can express voltage fluctuations to a reasonable extent. In the next chapter, a new modeling framework is introduced that further enhances the robustness evaluation process.

Chapter 9

Integrated Robustness Analysis via Hetero-Functional Graph Theory

The comparative study in the Chapter 8 reveals that the majority of current modeling frameworks for studying robustness analyze the electrical network in isolation, without taking into account its interdependency with other coexisting critical infrastructures such as water, transportation, etc. In addition, disruptive events disproportionately impact the low-income and socially vulnerable communities underscoring the need for incorporating social equality via true assessment of community resilience. However, the robustness of complex systems is typically evaluated solely in terms of engineering attributes, such as network topology or electrical parameters, and tends to ignore social factors that are equally important. As a result, robustness evaluations of such decoupled and partially informed systems are sub-optimal. Thus, there is a need to develop a holistic modeling framework for accurate robustness assessment. Hetero-functional graph theory (HFGT), inspired by [169] is one potential modeling framework to incorporate the above-discussed factors. HFGT can effectively model complex interdependent systems, including electrical, power, and transport networks, via a set of graphs corresponding to different activities. Essentially, these graph-structured models leverage functionality as building blocks, unlike conventional frameworks

that only describe physical attributes in terms of nodes. This allows them to reveal various functionalities and their dependencies within the system as well as integrate various aspects, including engineering and social factors. Robustness assessment within this kind of modeling framework would be more realistic and thus enable one to take effective decisions in improving system resilience.

This chapter first utilizes HFGT framework for analyzing robustness in the standard IEEE 37-node power distribution network. Specifically, we highlight the various advantages of HFGT, including the feasibility of combining performance and network based robustness measures. Then in the second part, we extend our study to interdependent urban infrastructure networks (IUN), comprising of electricity, water, heating, natural gas and road transportation networks (along with infrastructure repair services). Unlike original HFGT [36] that uses binary terms to quantify dependencies among functionalities, we introduces weighted hetero functional graph theory (WHFGT) framework, where real numbers are used to accurately quantify dependencies. Then a comprehensive and in-depth robustness evaluation of an IUN is carried out via a failure based analysis. WHFGT offers a more generic framework, including the simulations of partial attacks (partial loss of functionalities due to attacks), which is not feasible with conventional HFGT models.

9.1 Background

HFGT framework [36] can be viewed as an intellectual fusion of model-based systems engineering and network science, and has been applied in transportation networks [170], production systems [171], power grids [172], among others. It aims to merge the fields of system engineering and network science for holistic modeling of interdependent engineering systems. Primarily, it is comprised of seven graph-based models that are related to their counterparts in model based system engineering. These graph-based models include (1) *System concept*; (2) *Hetero-functional adjacency graph*; (3) *The controller agency graph*; (4) *Controller*

adjacency graph; (5)*Service as operand behavior*; (6) *Service feasibility graph*; (7) *System adjacency matrix*. The first two models are structural models and can be applied to many types of engineering systems. Due to explicit differentiation of system functionalities in the system concept model, the networks with unlike functions can be integrated into a single mathematical model via models (1) and (2). Unlike the conventional graphic representation, where nodes and edges represent system elements, nodes of the hetero-functional graph represent system functionalities and edges represent the logical sequences or dependencies among them. The models (3) and (4) constitute the system control models. Together, they differentiate systems based upon the structure of their control and decision-making architectures. The models (5) and (6) constitute the service models, which differentiate systems based upon the behaviors of their operands. These models are then ultimately coupled together for a holistic modeling of interdependent engineering systems in a matrix form that is defined as the model (7) (i.e. system adjacency matrix).

9.2 HFGT model of Power distribution network

This section analyzes a standard IEEE-37 node power distribution network with HFGT-based models. We focus solely on the system concept model of the HFGT, since it is a fundamental building block for other models. Essentially, the system concept maps system functionalities to its physical form (resources). To this end, resources can be divided into transformation (capable of transforming operand states), transportation (capable of transporting operands), and buffer (capable of storing operands). Similarly, functionalities can be categorized into transformation (transforming properties of the operands) and transportation (transporting operands between the resources/buffer). Transformation knowledge base maps transformation resources and transformation processes, transportation knowledge base maps transformation resources and transformation processes. System knowledge base connects all processes with their associated resources. The hetero-functional adjacency matrix

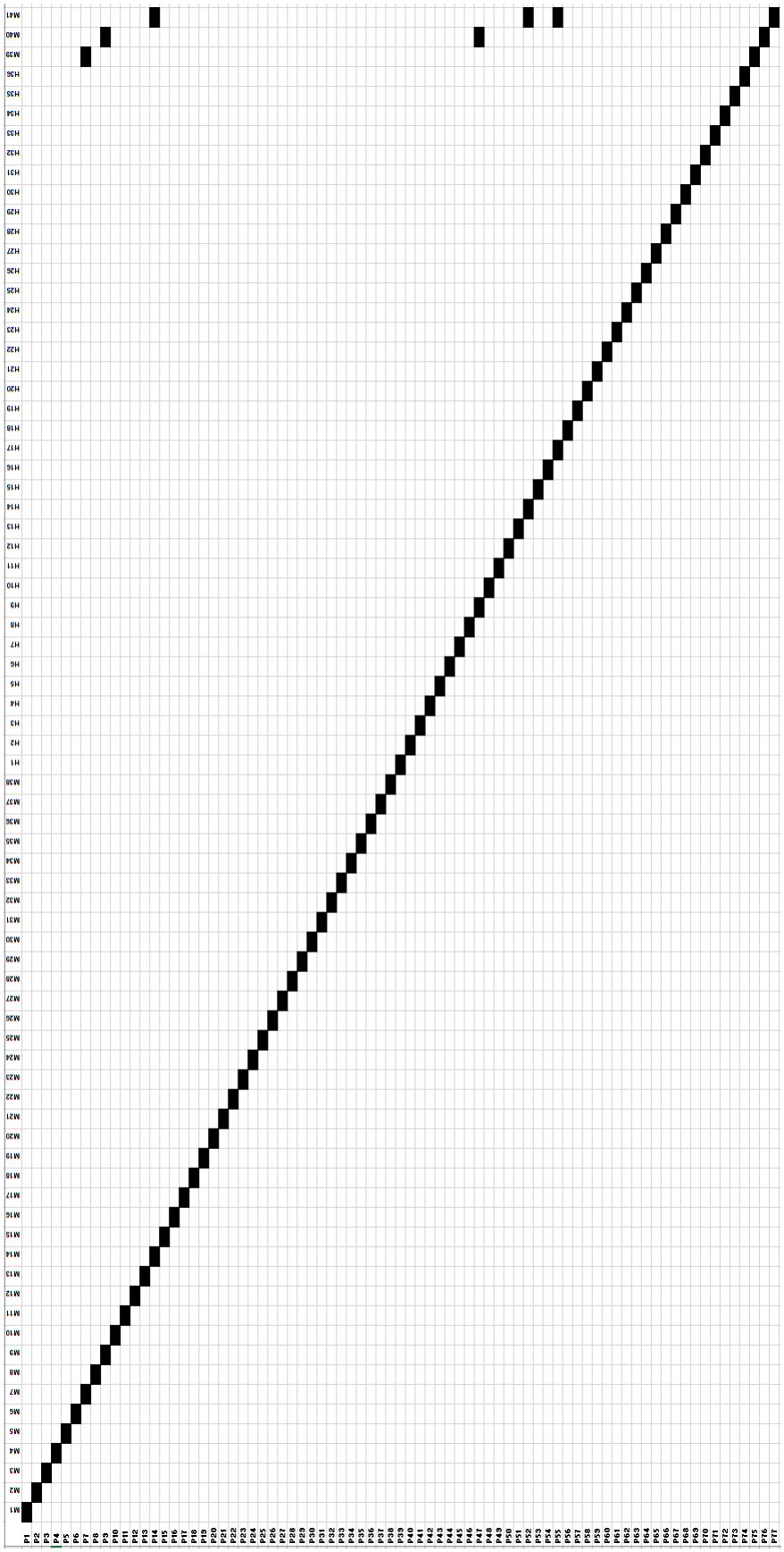


Figure 9.1: System knowledge base of IEEE-37 node system

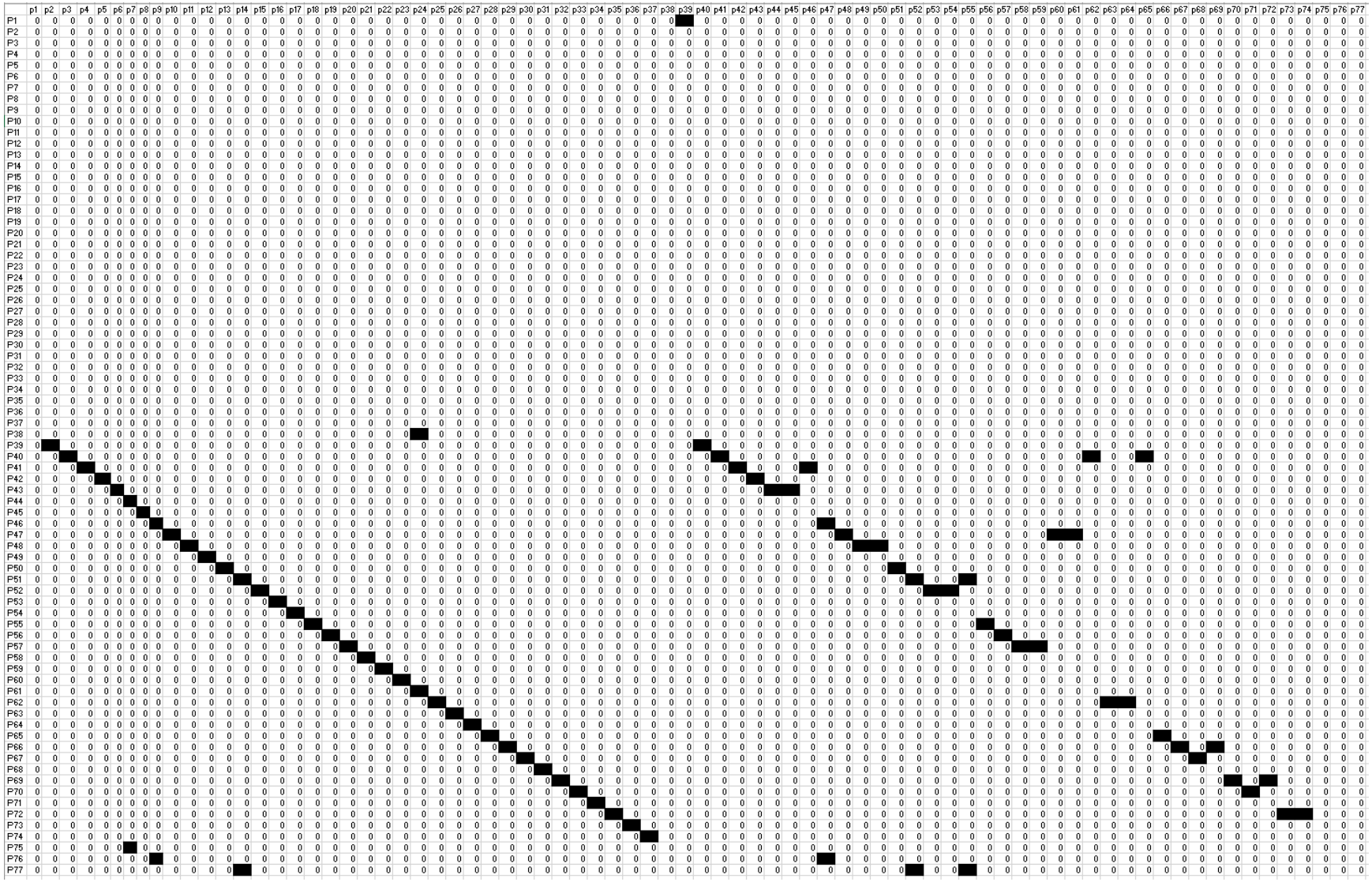


Figure 9.2: Hetero functional adjacency matrix of IEEE-37 node system

(HFAM) provides information related to the sequence of processes.

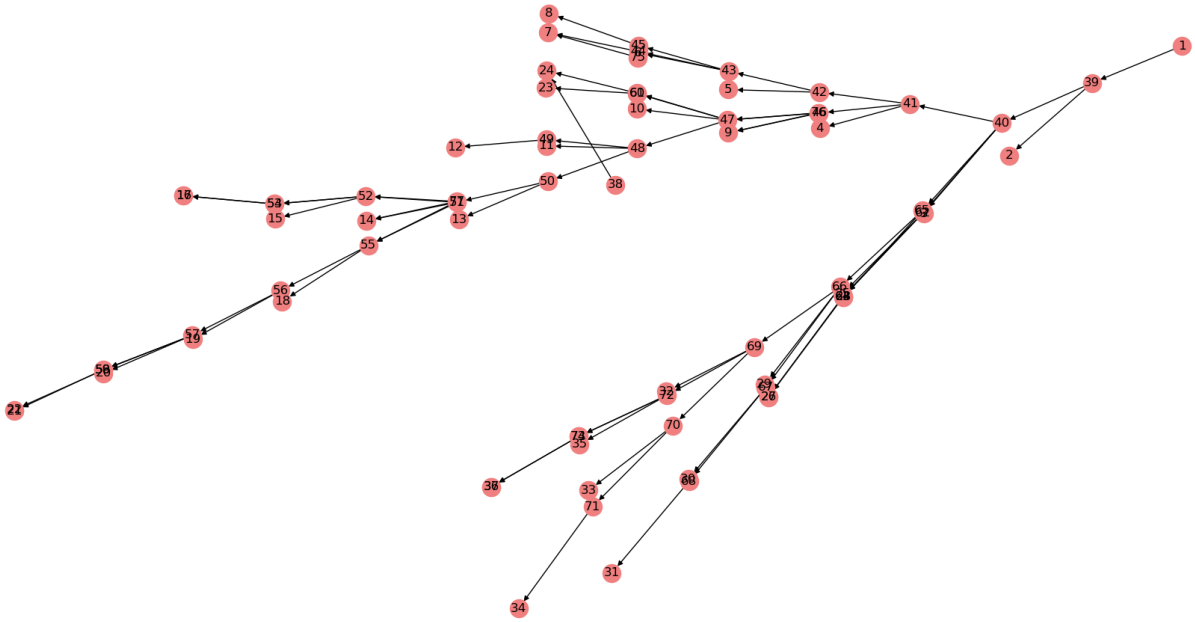


Figure 9.3: Hetero functional graph of IEEE-37 node system

Figure 9.3 illustrates a Hetero functional graph (HGF) of the IEEE 37-node system, and it is different than that of regular graph (RG) shown in Fig. 3.2 of Chapter 3. HFG is generated based on HFAM, where nodes represent processes and links denote their dependencies. The regular graph shows the IEEE-37 node power network with a source node, load nodes, transformers, DERs, and BESS. These are connected by links representing power lines. Rather than 37 nodes that only represent physical facilities in RG, HFG has 77 nodes that represent all system functionalities. The details of the facility locations and their functionalities are provided in Table 9.1 and Table 9.2, respectively. In fact, the RG does not incorporate changes even when end-users add PVs/BESS to their houses/community, while HFG can incorporate those changes explicitly as additional nodes (e.g., nodes 75 to 77). These kinds of fundamental differences reinforce our deduction that HFG is a more rigorous framework compared to regular models. Additionally, HFG captures the logical flow of system functionalities, making it easy to track the services from their start to their end process. Figures 9.1 and 9.2 denote the system knowledge matrix and hetero functional adjacency

matrix for the IEEE 37-node system. To further compare HFG with RG, we simulated several attacks on these graphs and analyzed robustness via failure trajectories in the next subsection.

Table 9.1: Locations of Resources in IEEE 37-node network

Resource	Description	Resource	Description
M1	Source node 1	H14	Power line between node 14 to 15
M2	Load node 2	H15	Power line between node 15 to 16
M3	Load node 3	H16	Power line between node 15 to 17
M4	Load node 4	H17	Power line between node 14 to 18
-	-	H18	Power line between node 18 to 19
M37	Load node 37	H19	Power line between node 19 to2 20
M38	Transfomer for load node 24	H20	Power line between node 20 to 21
M39	DER at load node 7	H21	Power line between node 20 to 22
M40	DER at load node 9	H22	Power line between node 10 to 23
M41	DER at load node 14	H23	Power line between node 10 to 24
H1	Power line between node 1 to 2	H24	Power line between node 3 to 25
H2	Power line between node 2 to 3	H25	Power line between node 25 to 26
H3	Power line between node 3 to 4	H26	Power line between node 25 to 27
H4	Power line between node 4 to 5	H27	Power line between node 3 to 28
H5	Power line between node 5 to 6	H28	Power line between node 28 to 29
H6	Power line between node 6 to 7	H29	Power line between node 29 to 30
H7	Power line between node 7 to 8	H30	Power line between node 30 to 31
H8	Power line between node 4 to 9	H31	Power line between node 29 to 32
H9	Power line between node 9 to 10	H32	Power line between node 32 to 33
H10	Power line between node 10 to 11	H33	Power line between node 33 to 34
H11	Power line between node 11 to 12	H34	Power line between node 32 to 35
H12	Power line between node 11 to 13	H35	Power line between node 35 to 36
H13	Power line between node 13 to 14	H36	Power line between node 36 to 37

9.3 Contingency Analysis in HFGT framework

In this section, we have conducted a percolation-based assessment, where nodes would be removed one at a time, and then the robustness of the residual network is measured at each step. The nodes are removed with two different strategies, i.e., random (randomly selecting nodes) and targeted (nodes are selected based on some criterion). The robustness at each step of the percolation is determined by the size of the largest cluster component (LCC) and the number of connected components (NCC). The larger the LCC, the more connected

Table 9.2: Defined Processes in IEEE 37-node network

Name	Functionality	Name	Functionality
P1	Maintain voltage at source node 1	**P40**	Transport power from M2 to M3
P2	Consume power at load node 2	**P41**	Transport power from M3 to M4
P3	Consume power at load node 3	**P42**	Transport power from M4 to M5
P4	Consume power at load node 4	**P43**	Transport power from M5 to M6
P5	Consume power at load node 5	**P44**	Transport power from M6 to M7
P6	Consume power at load node 6	**P45**	Transport power from M6 to M8
P7	Consume power at load node 7	**P46**	Transport power from M4 to M9
P8	Consume power at load node 8	**P47**	Transport power from M9 to M10
P9	Consume power at load node 9	**P48**	Transport power from M10 to M11
P10	Consume power at load node 10	**P49**	Transport power from M11 to M12
P11	Consume power at load node 11	**P50**	Transport power from M11 to M13
P12	Consume power at load node 12	**P51**	Transport power from M13 to M14
P13	Consume power at load node 13	**P52**	Transport power from M14 to M15
P14	Consume power at load node 14	**P53**	Transport power from M15 to M16
P15	Consume power at load node 15	**P54**	Transport power from M15 to M17
P16	Consume power at load node 16	**P55**	Transport power from M14 to M18
P17	Consume power at load node 17	**P56**	Transport power from M18 to M19
P18	Consume power at load node 18	**P57**	Transport power from M19 to M20
P19	Consume power at load node 19	**P58**	Transport power from M20 to M21
P20	Consume power at load node 20	**P59**	Transport power from M20 to M22
P21	Consume power at load node 21	**P60**	Transport power from M10 to M23
P22	Consume power at load node 22	**P61**	Transport power from M10 to M24
P23	Consume power at load node 23	**P62**	Transport power from M3 to M25
P24	Consume power at load node 24	**P63**	Transport power from M25 to M26
P25	Consume power at load node 25	**P64**	Transport power from M25 to M27
P26	Consume power at load node 26	**P65**	Transport power from M3 to M28
P27	Consume power at load node 27	**P66**	Transport power from M28 to M29
P28	Consume power at load node 28	**P67**	Transport power from M29 to M30
P29	Consume power at load node 29	**P68**	Transport power from M30 to M31
P30	Consume power at load node 30	**P69**	Transport power from M29 to M32
P31	Consume power at load node 31	**P70**	Transport power from M32 to M33
P32	Consume power at load node 32	**P71**	Transport power from M33 to M34
P33	Consume power at load node 33	**P72**	Transport power from M32 to M35
P34	Consume power at load node 34	**P73**	Transport power from M35 to M36
P35	Consume power at load node 35	**P74**	Transport power from M35 to M37
P36	Consume power at load node 36	**P75**	Generate power at node 7
P37	Consume power at load node 37	**P76**	Generate power at node 9
P38	Step down voltage to load node 24	**P77**	Generate power at node 14
P39	Transport power from M1 to M2		

the network is and thus more robust. In contrast, the larger value of NCC corresponds to lower connectivity and lower robustness. Furthermore, in the targeted case, nodes are removed based on the out-degree of a node since the out-degree of a node signifies the importance of that node in terms of providing functionalities to other nodes of the network. The experiments are repeated for both HFG and RG. Figures 9.4 and 9.5 illustrate the LCC and NCC for both random and targeted attacks in HFG and RG, respectively. LCC decreases with the progress of percolation since node removals will disintegrate the graph into various components with a smaller number of nodes. However, LCC decays sharply in the case of RG because it always has fewer nodes and edges compared to HFG. Similar behavior is demonstrated by the NCC. While in the case of a targeted attack, degradation is very fast with a very few attacks compare to that of random case. particularly, in RG representation, removing node 3 prevents the power flow in the entire network. Since DERs located in nodes 7, 9 and 14, they can serve the local nodes. Therefore, a separate analysis is required for the isolated network to evaluate system robustness. However, their contribution to maintain system functionality (in other words enhance system robustness) is clearly ignored in RG. On the other hand, HFG maintains the functionality of transporting power active as long as any of the generation sources (i.e., source node 1, DERs at node $7, 11, 14$) is active in the graph. Thus, HFG enables seamless computation in a single step. This demonstrates the benefit of using HFG while studying system robustness.

9.4 Interdependent system configuration

To exploit the full potential of HFGT for assessing robustness, we extended the test system from a single power distribution network to an interdependent network. Fig. 9.6 shows the integrated urban network of interest. This network comprises of an electric power system [173], a gas network [174], a water system [175], a road transportation network [176], and a district heating network [177]. Further, Table 9.3 tabulates the locations of various in-

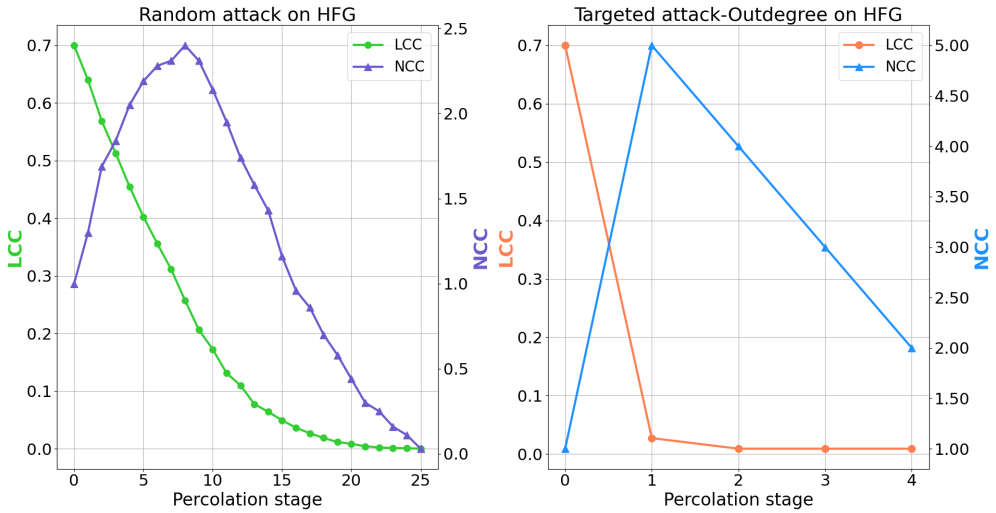


Figure 9.4: Trajectory of robustness metrics in HFG across various stages of percolation

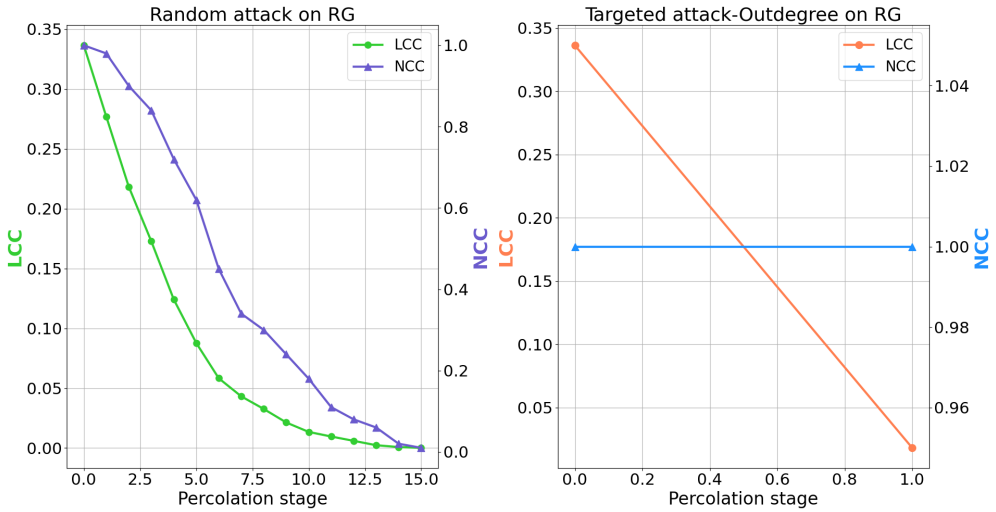


Figure 9.5: Trajectory of robustness metrics in regular graph (RG) across various stages of percolation

frastructure facilities within the networks, including 1 gas-fired power plant (GPP), 1 solar power plant (SPP), 2 water treatment plants (WTP), 3 heating plants (HPL), 2 gas stations (GS), 1 gas-driven CHP, 2 industry parks (IP1-IP2), 3 commercial zones (CZ1-CZ3), 6 residential zones (RZ1-RZ6), 1 water infrastructure repair facility (WIRF), 1 power infrastructure repair facility (PIRF) and 1 natural gas infrastructure repair facility (GIRF). To ensure the proper operation of these facilities, 121 functionalities are required in total. In particular, electricity supplies of industry parks, commercial zones and one residential zone

(RZ1) depend on three electric power sources, including the gas-driven power plant, CHP and the solar plant. The complete process relation matrix is presented in [178], with the same structure as defined earlier for the conceptual diagram with 43 functionalities.

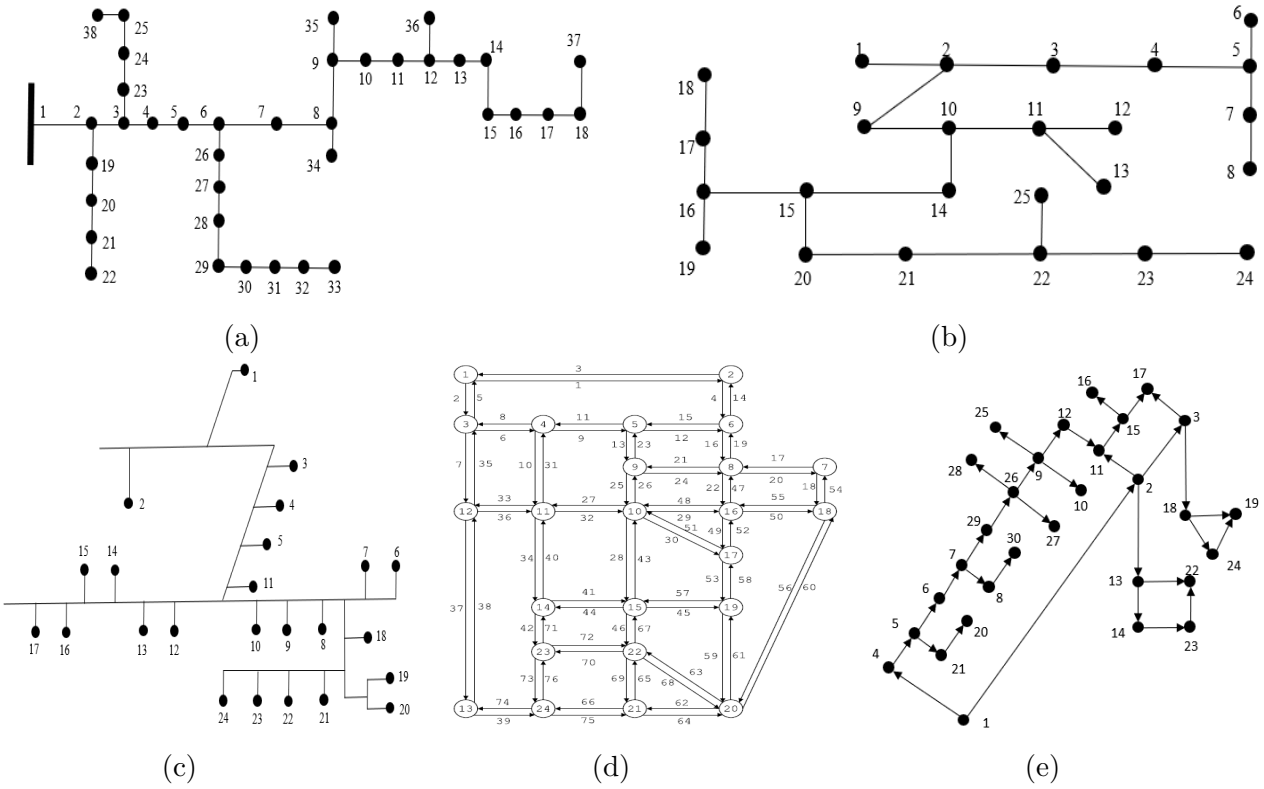


Figure 9.6: Integrated infrastructure with (a) electricity, (b) gas, (c) district heating, (d) transportation, (e) water networks

For this system network configuration, the relation matrix needs to be extended to incorporate the additional dependencies among processes due to the shared path of energy/service deliveries. For instance, the successful execution of the process that transmits power from GPP (located at node 5 in electric power network) to WIRF (located at node 1 in electric power network) depends on the successful execution of process that transmits power from GPP (located at node 5 in electric power network) to WTP1 (located at node 3 in electric power network). This is because, the electricity delivery path between GPP (node 5) and WTP1 (node 3) is a part of the only path for electricity transmission from GPP (node 5) to WIRF (node 1). Similarly, the effects of other network constraints on process dependencies

are integrated into the process relation matrix.

Table 9.3: Locations of Entities in Various Networks

UtilityDescription	Entity	Location	Entity	Location	Entity	Location
Power	WIRF	Node 1	CZ1	Node 9	RZ2	Node 17
	WTP1	Node 3	WTP2	Node 11	RZ3	Node 19
	IP1	Node 4	CZ2	Node 12	RZ4	Node 20
	GPP	Node 5	RZ1	Node 13	GS2	Node 21
	IP2	Node 6	GIRF	Node 14	RZ5	Node 22
	GS1	Node 7	SPP	Node 15	CHP1	Node 23
	PIRF	Node 8	CZ3	Node 16	RZ6	Node 24
Water	WIRF	Node 1	CZ1	Node 9	RZ2	Node 17
	HPL1	Node 2	HPL2	Node 10	HPL3	Node 18
	WTP1	Node 3	WTP2	Node 11	RZ3	Node 19
	IP1	Node 4	CZ2	Node 12	RZ4	Node 20
	GPP	Node 5	RZ1	Node 13	GS2	Node 21
	IP2	Node 6	GIRF	Node 14	RZ5	Node 22
	GS1	Node 7	SPP	Node 15	RZ6	Node 24
	PIRF	Node 8	CZ3	Node 16		
Gas	GS1	Node 1	HPL3	Node 8	RZ3	Node 14
	HPL1	Node 2	CHP1	Node 10	RZ4	Node 15
	GPP	Node 5	RZ1	Node 12	RZ5	Node 17
	HPL2	Node 6	RZ2	Node 13	RZ6	Node 19
Heat	WIRF	Node 1	CZ2	Node 12	RZ3	Node 19
	IP1	Node 4	RZ1	Node 13	RZ4	Node 20
	IP2	Node 6	GIRF	Node 14	RZ5	Node 22
	PIRF	Node 8	CZ3	Node 16	CHP1	Node 23
	CZ1	Node 9	RZ2	Node 17	RZ6	Node 24
Transport	WIRF	Node 1	PIRF	Node 8	HPL3	Node 18
	HPL1	Node 2	HPL2	Node 10	GS2	Node 21
	WTP1	Node 3	WTP2	Node 11	CHP1	Node 23
	GPP	Node 5	GIRF	Node 14		
	GS1	Node 7	SPP	Node 15		

9.5 Weighted hetero functional graph theory (WHFGT)

Using the system knowledge matrix of the HFGT framework, system resources and system processes can be mapped. To more realistically capture functionality interactions, we propose a process relation matrix (P_R), which is developed on top of the conventional dependency

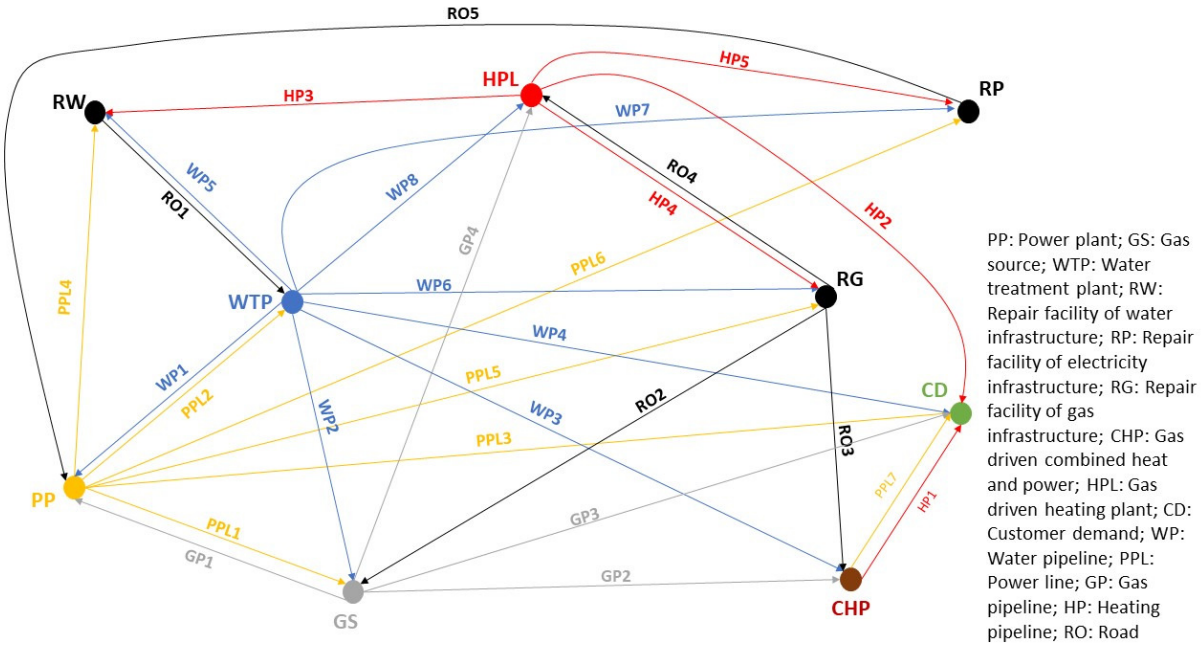


Figure 9.7: Conventional Graph Representation of a Conceptual Interdependent Urban Infrastructure Network (IUN)

matrix of HFGTs. For the conventional dependency matrix, the relationships between the functionalities are expressed in binary terms, i.e., it only captures whether the dependency is present or not. However, it is inadequate to quantify the degree of dependency between the two functionalities. For instance, an electric power consumer can be supplied by multiple power plants. The conventional HFGT framework cannot present the degree of dependency of the electricity consumer on each power plant precisely. Therefore, to address this shortcoming, we propose a WHFGT framework and the corresponding relation matrix (P_R), which quantifies the degree of dependency using weights of real numbers. As performing a particular functionality could depend on multiple other functionalities, the proportion contributed from each functionality signifies the weight. Thus, for a particular node in the WHFGT, the weights of its associated edges in matrix P_R denote the degree of dependencies on other functionalities. For instance, in Fig. 9.7, electricity demand of the community customers can be supplied by two sources, i.e., the gas-driven power plant and CHP, and the dependencies on both of them can be reflected by assigning weights. With the assumption that gas-driven

plant contributes 20% of the total demand and CHP supplies remaining 80%, we can allocate 0.2 to the edge connecting the functionality of "consuming electricity in community" and the functionality of "delivering electricity from gas-driven power plant to community consumer", and 0.8 to the other edge. For the illustration purpose, we have arbitrarily allocated weights in this work. Note that the proposed method can be adapted to accommodate other parameters in practice based on the collected relevant information. The process relation matrix (P_R) for the conceptual diagram depicted in Fig. 9.7 is a 43×43 matrix as 43 functionalities are incorporated. For brevity, the process relation matrix (P_R) is provided in an online repository [178]. The rows and columns of the matrix represent functionalities, while the entries represent the dependencies between two corresponding functionalities. Correspondingly, Fig. 9.8 depicts the process relation graph for a synthetic network shown in Fig. 9.7. The direction of edges are from the source to target nodes, where target nodes (functionalities) are dependent on the source nodes (functionalities) with dependency proportional to the edge weights.

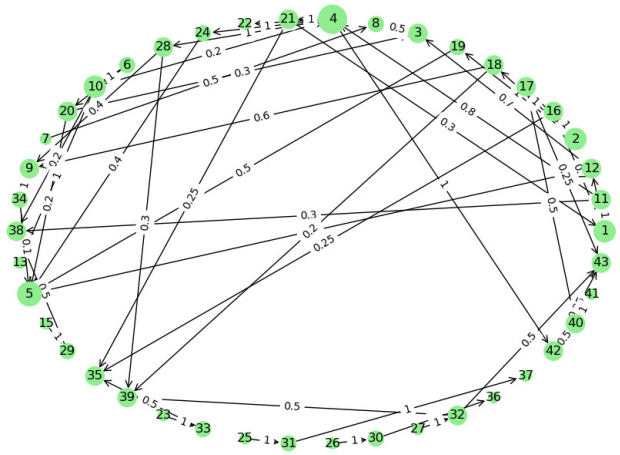


Figure 9.8: Process relation graph corresponding to the network shown in Fig. 1

9.6 Contingency Analysis in WHFGT Framework

The tight interconnections among various infrastructures may exacerbate the impact of potential cascading failures, as contingencies in one part of the system might propagate to other networks. Therefore, robustness evaluation of IUNs is critical for vulnerability analysis and robustness enhancement.

To simulate contingencies on IUNs, the nodes are defined as the targets for attacks. In other words, disabling certain functionalities is equivalent to removing the associated nodes in the WHFGT graphs. Note that the edges merely denote the dependencies among functionalities, thus, they are not considered as attack targets, as disabling dependency relations has no interpretative meaning. Besides, based on graph percolation procedure [179], nodes are removed in stages and the robustness metrics of the residual graph are computed at each stage to study the transition properties of the graph. Along with the removal of a selected node, the child nodes of the selected node are also removed under specific conditions. Here, child nodes are defined as nodes that can be reached (directed path exist) from the selected node and are computed using depth first search algorithm [180]. We would like to reiterate that, for the sake of simplicity, all the contingencies in this paper are referred to as "attacks".

According to the available system information and attacker capabilities/strength, the attacks can be further divided into four categories, as tabulated in Table 9.4. Random/targeted attacks are categorized based on the knowledge level of attackers about the system. Specifically, in a random attack, nodes are removed randomly at each stage of the percolation whereas, in a targeted attack, nodes are removed based on certain importance scores of nodes (assuming the system information is available to the attacker). On the other hand, complete/partial attacks are categorized based on attackers capability. Specifically, in a complete attack, the attacker disables a process completely, while in a partial attack, the attack can only degrade the process to a certain degree. Detailed description of these attack categories will be provided later.

Table 9.4: Types of attack strategies

StrengthInformation	**Random**	**Target**
Complete	complete random	complete target
Partial	partial random	partial target

As mentioned earlier, targeted attacks are launched based on the importance scores assigned to nodes, assuming the attacker has acquired the entire system information. Here, the concept of centrality is applied to quantify importance of nodes in a network. Four types of centrality are widely applied in prior literature, including degree centrality, eigenvector centrality, pagerank centrality and betweenness centrality [181]. In IUN, we select betweenness and weighted out-degree centrality as the basis to conduct targeted attacks. Specifically, removal of nodes with high betweenness centrality and high weighted out-degree centrality in the earlier stages is considered as efficient approaches to impair the system connectivity, thereby simulating extreme conditions.

9.6.1 Complete attack

The most trivial type of attack scenario would be the situation where the attacker is capable of disabling a process completely, referred to as a complete attack. This can be caused by extreme natural events or powerful malicious attackers. In WHFGT, complete attacks can be represented by removal of corresponding nodes entirely, within a single stage. Based on the knowledge level of the attacker, complete attacks can be further categorized into complete random and complete targeted attacks. Specifically, for a complete random attack, it is assumed that the attacker is not aware of the internal configuration of the IUN and hence the targeted nodes (processes) will be attacked randomly. Under this situation, nodes are selected randomly for removal at each stage of the percolation and the percolation is continued until the graph is left with two nodes. Obviously, if a child node (functionality) is dependent on its only parent node (functionality), disabling the parent node (functionality) will lead to obliteration of the child node. For instance, the node representing the functionality of

"transport water from WTP to the residential community" will be removed once the node corresponding to the functionality of "treat water in WTP" is out of the graph. On the other hand, for child nodes with multiple parent nodes, removing some (not all) of its parent nodes will lead to a certain degree of degradation, which is quantified by the out-degree weight reduction. Out-degree weights of a node indicate the amount of dependency the neighbors have on the node, whereas the in-degree weights of a node signify the extent of dependency the node has on its neighbor. Specifically, the decrease of out-degree weights is assumed to be in proportion to the decrease of in-degree weights. In other words, for a particular node (process), with less inputs from parent nodes (functionalities), its performance of supporting other nodes (functionalities) will be degraded accordingly. When the out-degree weight of a node decreases to a pre-defined threshold, namely critical quality of functionality (QoF), it will be removed from the graph. The practical interpretation of this rule is that when a process is unable to support certain follow-up services, it will be considered as dysfunctional. For instance, when the process of "deliver electricity to customers 1-5" is degraded to "deliver electricity to only customer 1" due to attacks, the relevant infrastructure might be considered as dysfunctional and the temporary shutdown or maintenance actions may be needed. Note that the critical QoF value can be assigned flexibly, based on the specific processes and situations.

Furthermore, considering the frequent occurrences of malicious cyber attacks around the globe, it is increasingly vital to address the situation wherein the attacker could obtain insights on the entire system and launch targeted attacks. Targeted attacks aim to remove nodes with higher weighted out-degree/betweenness centrality in the earlier stages of percolation. In other words, functionalities with higher importance are targeted in the earlier stages of percolation. The complete targeted attack is similar to its random counterpart except that the nodes are selected on the basis of predefined importance score, determined by weighted out-degree/betweenness centrality. During targeted attacks, nodes are removed in descending order of importance. The detailed procedure of complete attack is shown in

Algorithm 6 Complete attack

Input: Graph G with V nodes.
Output: List of robustness metrics corresponding to various percolation stages
LOOP Process
while $|V| > 2$ **do**
  nodeselected: Select a node randomly or based on the order of important scores.
  childnodes: Find all child nodes of *nodeselected*.
  remove *nodeselected* from G
  for *child* in *childnodes* **do**
    indeg: Find weighted in degree of *child* node
    outdeg: Find weighted out degree of *child* node
    $Degradationratio = \frac{\text{current indeg of child}}{\text{indeg of child before percolation}}$
    outdegweights: Decrease the out edges weight of *child* by *Degradationratio*
    if (*indeg*$= 0$) or (outdegweights $<$ Quality of Function) **then**
      remove *child* from G
    end if
  end for
  compute robustness metrics
end while
return list of robustness metrics corresponding to various percolation stages

Algorithm 1. The robustness metrics indicated in the output of Algorithm 1 will be elaborated in section 4. Again, it should be noted that the word "attack" used in this paper does not necessarily denote malicious cyber/physical destruction, but also represents natural disasters that may lead to serious damage.

9.6.2 Partial attack

The complete attack is designed for simulating the scenarios where the attack is sufficiently powerful to destroy a functionality completely, which results in loss of the entire process. To examine the robustness of the graph against less severe contingencies, we propose a new type of attack strategy referred to as partial attack. The key difference between partial and complete attack strategies is that during percolation of a partial attack, the attacked nodes will only experience out-degree depletion until the critical QoF value is violated. In contrast to complete attacks where the attacked process is disabled unconditionally, partial attacks will only lead to a certain level of degradation, namely degradation level, which signifies the

extent of performance degradation of functionalities. In other words, functionalities operating at the level above QoF value is considered to be at an operative stage. Once degradation level of a functionality reaches QoF threshold, it becomes no longer operative that results in the removal of corresponding node from the graph. On the other hand, similar to complete attacks, partial attacks can be further subdivided into two types depending upon the knowledge level of the attacker, namely partial random attacks and partial targeted attacks. For partial random attacks, the attacked nodes are selected randomly, while nodes with higher weighted out-degree/betweenness centrality will be targeted in the early stages during partial targeted attacks. Algorithm 2 describes the complete procedure of a partial attack. It should be noted that partial attack analysis only applies to our proposed weighted hetero functional graphs, since partial degradation of weights can be incorporated. Traditional HFGT framework cannot accommodate these realistic attack scenarios.

Algorithm 7 Partial attack

Input: Graph G with V nodes.
Output: List of robustness metrics corresponding to various percolation stages
LOOP Process
while $|V| > 2$ **do**
  nodeselected: Select a node randomly or based on the order of important scores.
  childnodes: Find all child nodes of *nodeselected*.
  outdegweights: Decrease the out edges weight of *nodeselected* by random quantity.
  if (*outdegweights* < Acceptable service) **then**
    remove *nodeselected* from G
  end if
  for *child* in *childnodes* **do**
    indeg: Find weighted in degree of *child* node
    outdeg: Find weighted out degree of *child* node
    Degradationratio= $\frac{\text{current indeg of child}}{\text{indeg of child before percolation}}$
    outdegweights: Decrease the out edges weight of *child* by *Degradationratio*
    if (*indeg*= 0) or (outdegweights < Quality of Function) **then**
      remove *child* from G
    end if
  end for
  compute robustness metrics
end while
return list of robustness metrics corresponding to various percolation stages

9.7 Numerical Study

In this section, we implement WHFGT based modeling of IUN and evaluate its robustness. To begin with, the system network configuration is described, along with its effects on the process relation matrix (P_R). After introducing several robustness metrics, the IUN robustness is quantified by analyzing the impacts of four types of attack scenarios.

9.7.1 Metrics for robustness evaluation

Robustness of IUN in the present work is studied through the percolation process, where nodes of the network are removed in stages and the connectivity of the residual network is assessed at each stage, until the graph is completely disconnected. In a high level definition, the connectivity of the residual graph after node removal represents the robustness of the graph towards that node attack. In general, higher connectivity of the residual graph indicates enhanced robustness. The conventional indices for quantification of graph connectivity include Largest Connected Component (LCC) and the Number of Connected Components (NCC) [182]. Here, a component denotes a subgraph, in which any two vertices are connected to each other by paths and is connected to no additional vertices in the graph. Specifically, LCC denotes the size of the largest component where every node is at least connected to one other node, while NCC represents the number of connected components in the entire graph. Furthermore, we use flow robustness (FR) to quantify robustness, from the components standpoint [96]. It captures the ability of the nodes to communicate with each other in all the clusters and hence characterizes the overall reachability of the graph. Unlike LCC that only accounts for the largest connected component, FR incorporates the number of nodes in all components of the residual graph. The FR metric corresponds to:

$$FR = \frac{\sum_i |C_i|(|C_i| - 1)}{N(N-1)} \tag{9.1}$$

where C_i is the number of nodes in component i and N denotes the total number of nodes in the original graph before attacks. As seen, FR represents the degradation at a global level by monitoring the connectivity situation of all components and hence reveals global robustness.

However, the three indices described earlier are inherently incapable of incorporating weights of edges. Therefore, we design a new robustness metric, namely service robustness (SR). It can be expressed as:

$$SR = \frac{\sum_i Q_i^a}{\sum_i Q_i} \tag{9.2}$$

where Q_i^a denotes the weighted out-degree of node i after each stage of attack and Q_i represents the total weighted out-degree of all nodes in the original graph without attacks. The SR index not only infers connectivity of the graph, but also incorporates the weights of edges. Since the weights of edges quantify the dependencies between source and target nodes, integrating them into the robustness analysis would aid in providing a more precise evaluation of impacts of attacks. In the next sub-section, we examine the robustness of the graph described earlier by observing the trajectories of the four robustness metrics along the percolation stages with different types of attacks.

9.7.2 IUN Robustness Analysis

The robustness of IUN is examined by evaluating the robustness metrics, i.e., LCC, NCC, SR and FR, after each stage of percolation. The impacts of four attack strategies, i.e., complete random attack, complete targeted attack, partial random attack and partial targeted attack, are examined and compared in the following. The attacked node selections in targeted attack are carried out based on betweenness and weighted out-degree. Additionally, Monte-Carlo simulation approach is used here to evaluate the effects of random attack strategies by repeating the simulation for 10,000 times.

Complete attack

As mentioned earlier, in a complete attack, the attacked nodes are removed completely once the attack is imposed. Under this category, we first simulate the random attack situation. The trajectories of the four robustness metrics are depicted in Fig. 9.9. The horizontal axis denotes the numbers of sequential attacks imposed on the system. After each attack, the robustness metrics of the residual graph are evaluated and plotted in the vertical axis. The robustness analysis is performed in steps, namely in percolation stages, which is in consistent with the definition of percolation in [183]. In this way, real-world situations such as sequential cyber-attack events or natural disasters can be simulated. As expected, LCC decreases along with the progress of percolation, as node removals will decompose the graph into various components with a smaller number of nodes. On the other hand, the NCC increases in the initial stages and starts declining after a certain stage. This is because, initially, the entire graph is fragmented into several sub-graphs that lead to an increasing number of connected components. After a certain stage (stage 14 in this case), those sub-graphs will be further decomposed into individual nodes, decreasing the value of NCC. The global metric FR also drops exponentially but with a lower decay rate compared to LCC, which implies the slower degradation of connectivity (weighted or unweighted) among various components across the entire graph (compared to the local connectivity in the largest component). The trajectory of SR shows a gradual decline as well, which captures the diminishing mutual dependencies among nodes.

To demonstrate the effects of attackers' knowledge about the system, we further illustrate the trajectories of robustness metrics with imposed betweenness- and weighted outdegree-based targeted attacks along the percolation stages, as depicted by blue and green curves in Fig. 9.9, respectively. Like the results associated with complete random attacks, the general patterns of trajectories of robustness metrics correspond to the trend of gradual decline except for NCC. The NCC metric initially increases abruptly followed by fluctuations and eventually settles in the final stages. Additionally, the value of FR has a lower decay rate

compared to LCC value, which indicates a certain level of connectivity among components even with few nodes in the largest component. Additionally, compared to a random attack, the degradation of the system robustness is substantially higher with targeted attacks. Specifically, it takes 12 and 10 stages to disconnect the graph entirely if betweenness and weighted out-degree are selected as the basis to launch targeted attacks, compared to an average of 40 stages required to completely disconnect the graph with imposed random attacks. To further quantify the distinctions between random and targeted attacks, the numbers of sequential attacks required for degrading the robustness metrics to certain levels are tabulated in Table 9.5. The results illustrate that targeted attacks require fewer steps to reach the same level of degradation as random attacks. For instance, achieving 50% degradation in FR only takes 3 and 1 stages for betweenness- and weighted outdegree-based targeted attacks, respectively, while 10 steps are needed to attain the same level of degradation with random attacks. In other words, this test demonstrates the value of securing system information, as attackers with system information are capable of causing severe destruction rapidly. Fig. 9.10 depicts the histograms of numbers of nodes in each component at various stages with a complete targeted attack. The rapid decrease of node number in each component at the early stages can be witnessed, while the component decomposition process slows down after stage 6. Furthermore, comparing two types of targeted attacks, we can observe that the weighted outdegree-based targeted attacks outperform betweenness-based targeted attacks for all robustness metrics. This implies that weighted outdegree centrality could provide a more precise indication of nodes importance in this test.

Partial attack

In partial random attacks, attacked nodes are selected randomly with a certain level of degradation at each stage of percolation. The orange curves in Fig. 9.11 depict the trajectories of robustness metrics. The general trends in Fig. 9.11 are similar to the results of a complete random attack. Specifically, the metrics of LCC, FR and SR decrease abruptly

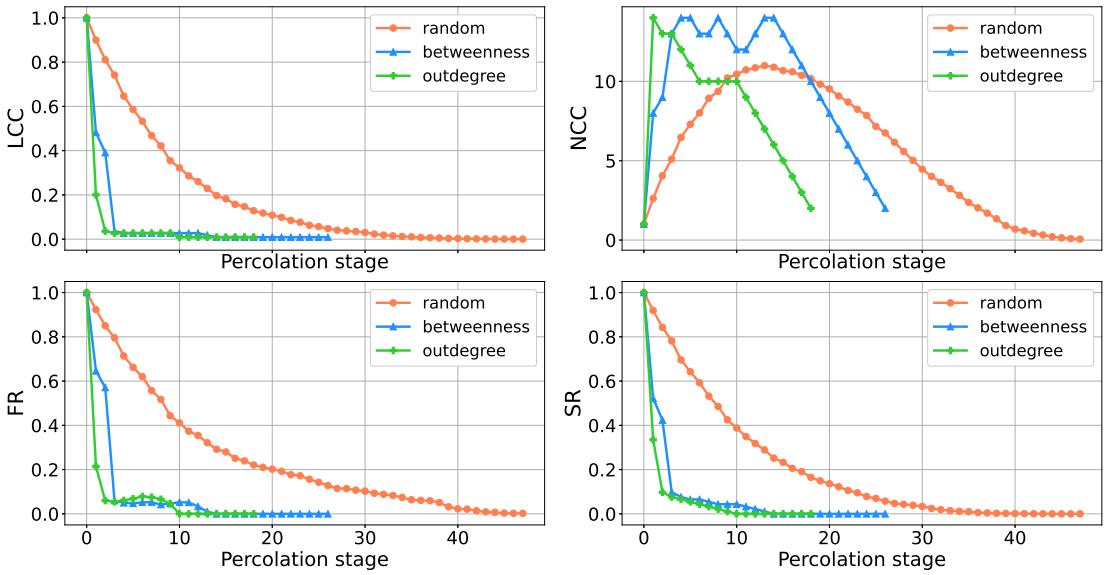


Figure 9.9: Trajectory of robustness metrics across various stages of percolation with complete attack

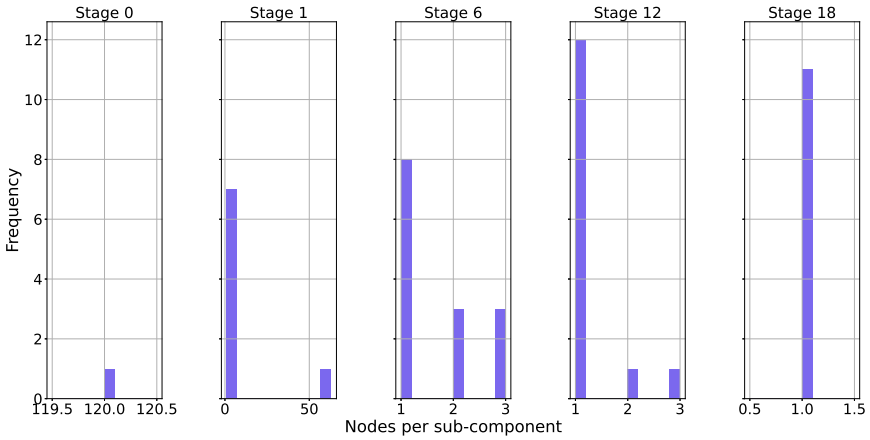


Figure 9.10: Histogram of nodes per sub-component/cluster across different stages of percolation in complete targeted attack

Table 9.5: Number of attacks for various degradation levels corresponding to random and targeted attacks with complete attack strategy RN: Random attack BW: Betweenness based targeted attack OD: Out-degree based targeted attack

Degradation Level	20%			50%			80%		
Metric Attack	RN	BW	OD	RN	BW	OD	RN	BW	OD
LCC	3	1	1	7	1	1	15	3	2
FR	4	1	1	10	3	1	22	3	2
SR	3	1	1	8	2	1	17	3	2

at the beginning followed by a moderate decline, while the index of NCC increases initially and decreases after a certain stage. These observations are consistent with the results from complete attack case described earlier.

We also illustrate the trajectories of robustness metrics along the percolation stages with partial targeted attacks, as depicted in blue and green curves in Fig. 9.11. As seen in the figure, targeted attacks lead to a more abrupt decrease in LCC, FR and SR. To further illustrate the graph fragmentation process, Fig. 9.12 depicts the histograms of numbers of nodes in each component at representative stages of partial targeted attack. The rapid decrease of node number in each component at the beginning can be explicitly observed, while the component decomposition process significantly slows down after stage 10. The results correspond to the conclusion drawn from Fig. 9.11, which again signifies the remarkable ability of targeted attacks in decimating the network rapidly. Additionally, the notable gap between partial random and targeted attack strategies can be observed in Table 9.6, which tabulates the required numbers of attack stages to reach certain degradation levels. As expected, it takes fewer steps for targeted attacks to fragment the graph into a certain level. In a nutshell, the general trend of robustness metrics are similar in both complete and partial attacks. The distinction between complete and partial attacks is compared with the distinction between targeted and random attacks in the following subsection.

Table 9.6: Number of attacks for various degradation levels corresponding to random and targeted attacks with partial attack strategy

Degradation Level	20%			50%			80%		
Metric Attack	RN	BW	OD	RN	BW	OD	RN	BW	OD
LCC	4	1	1	11	4	1	23	6	4
FR	4	1	1	13	6	1	30	6	4
SR	5	1	1	13	4	1	26	6	4

Complete random attack versus partial targeted attack

Evidently, complete, targeted attacks cause more severe damage than partial, random attacks, as demonstrated in Tables 9.5 and 9.6. In this section, the comparison between the

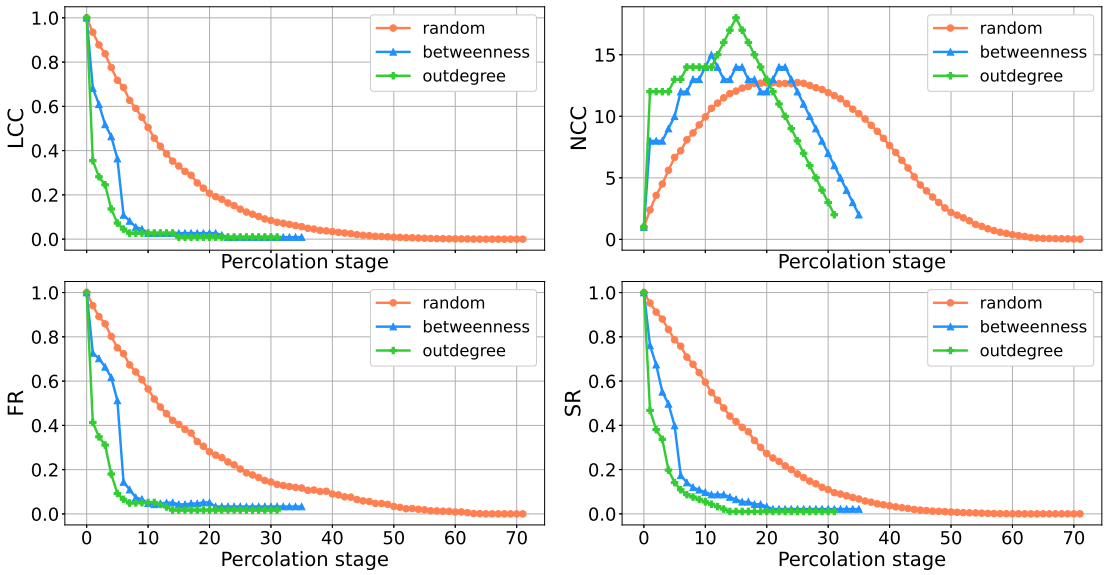


Figure 9.11: Trajectory of robustness metrics across various stages of percolation with partial attack.

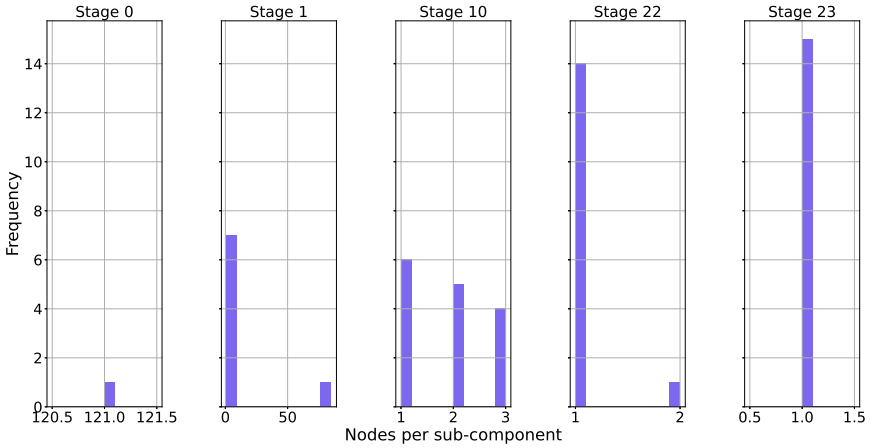


Figure 9.12: Histogram of nodes per sub-component/cluster across different stages of percolation in partial targeted attack

impacts of a complete random attack and a partial targeted attack is conducted to evaluate the merits of securing system information and infrastructure hardening.

From tables 9.5 and 9.6, we can witness that, in general, the distinction between random and targeted attacks is more significant, compared to the distinction between complete and partial attacks. For instance, assuming the targeted attacks are based on out-degree weight, to reach the degradation level of 80% in FR, the gap in the required percolation stages

between a partial random attack and a complete random attack is 8, whereas this gap between a complete targeted attack and a complete random attack is 20. Similar conclusions can be draw from other comparisons from tables 9.5 and 9.6. To further compare the effects of attack strength (complete attack *vs* partial attack) and attack intelligence (targeted attack *vs* random attack), we compare the outcomes of a complete random attack and a partial targeted attack. The results show that the required percolation stages to reach a same degradation level with a complete random attack is more than a partial targeted attack. For instance, it takes 22 and 4 stages for a complete random attack and a partial targeted attack (based on out-degree weight) to reach 80% of degradation in FR, respectively. These results reveal that intelligent attacks are more likely to result in severe consequences than powerful attacks, which call for the demanding needs for securing system information. To further illustrate and demonstrate which factor dominates severity of attacks, strength or information, Fig. 9.13 depicts snapshots of the residual graph at three representative stages with a complete random attack and a partial targeted attack. It can be observed that the partial targeted attack outperforms the complete random attack in rate of node eliminations. This study reinforces the merits of securing system information for robustness enhancement.

9.8 Summary

This chapter leverages a hetero-functional graph theory-based framework for modeling power distribution networks. It highlights the several advantages of HFGT over regular graphs, including their thoroughness in capturing minor system details and their ability to provide a reliable assessment of robustness. For exhibiting the real potential, the test system is extended to an interdependent system comprising of electricity, water, district heating, natural gas, road transportation networks and relevant services. In addition, this chapter also introduces a weighted hetero functional graph theory (WHFGT) based framework that can capture dependencies in real-numbers. We conduct various types of attack simulations on

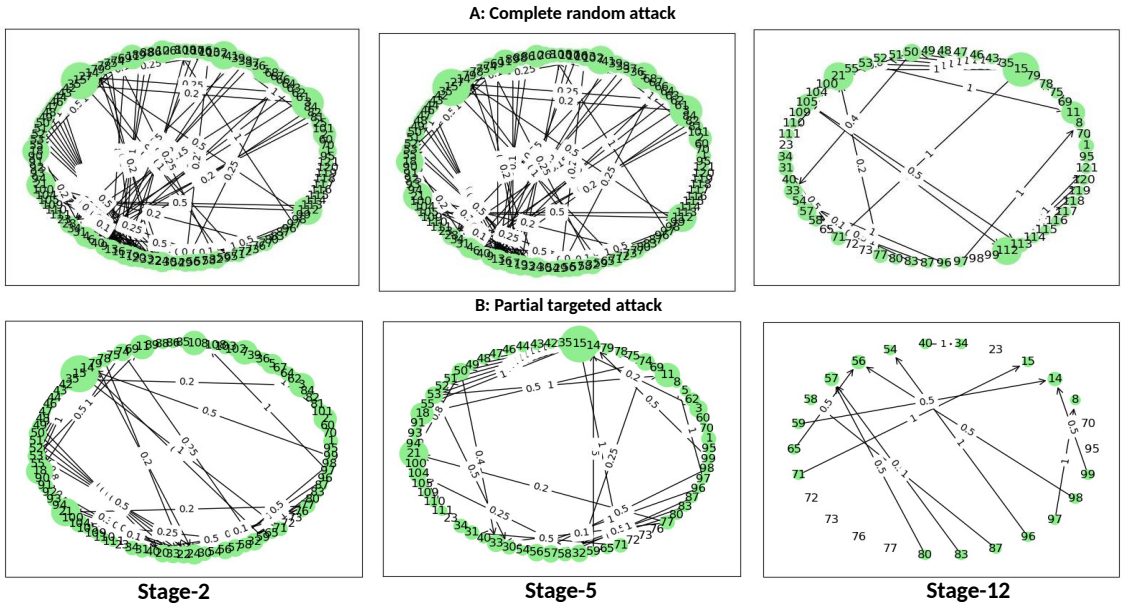


Figure 9.13: Snapshots of graph across representative stages of percolation with complete random attack and partial targeted attack.

the weighted hetero functional graphs and provide in-depth information about the trajectories of graph fragments. Results demonstrate that the proposed WHFGT-based framework and robustness metrics can provide precise insights into the robustness of interdependent infrastructure networks, which highlights the great importance of infrastructure hardening and securing network information. The next chapter concludes the thesis and highlights the potential future research directions.

Chapter 10

Conclusions and Future work

10.1 Conclusions

This concluding chapter summarizes the contributions of this dissertation and discusses future research directions. This dissertation addresses the fundamental questions related to improving complex system resilience against disruptive events. Primarily, it focuses on developing frameworks for accurate assessment of system robustness so that effective actions can be undertaken to mitigate the impact of disruptive events on system operations. Broadly, there are two different ways to access system robustness depending upon the factors involved, i.e., performance and network-based. In this regard, an analytical framework has been proposed for performance-based voltage violations metrics, and a graph machine learning-based predictive model for network-based robustness metrics. The central theme across all the proposed methods is related to "reducing complexity, enhancing scalability, and inducing generalizability". The developed frameworks can be applied to complex systems such as the smart grid, transportation, etc., to identify key elements of robustness, and undertake effective actions to improve system resilience towards extreme events. The accomplishments of this dissertation can be summarized as follows:

Chapter 3 presents a foundational work for performance-based robustness analysis of a

power distribution network. Specifically, it develops a probabilistic formulation of voltage change due to random change in power at multiple locations of the network. The proposed method can be used for various operation and planning-related tasks, including the monitoring of voltage violations in the network. This analytical framework is then combined with information theoretic metrics to derive a novel voltage influencing score in Chapter 4. VIS is a node level metric that measures the voltage influencing capacity of actor nodes (nodes where power varies) in a power distribution grid. VIS has several use cases, including the identification of dominant voltage influencer (DVI) nodes. Results in standard test networks demonstrate that the proposed metric effectively predicts the DVI nodes while substantially reducing the execution time. The DVI nodes can serve as optimal control locations for improving system resilience against detrimental voltage fluctuations.

Chapter 5 develops a graph neural network-based predictive model for identifying critical nodes/links in complex networks. Essentially, the framework consists of two parts, where in the first part, a GNN based embedding and regression model are trained end-to-end on synthetic graphs with a small subset of nodes/links. The second part deals with the prediction of scores for unseen nodes and links in the graph. Results on real-world networks demonstrate the computational efficiency and scalability of the proposed approach over conventional methods. These critical nodes can be prioritized to improve the system's resilience. Moreover, the predictions from GNN could be unreliable due to uncertain input graph or GNN parameters. To address this issue, a Bayesian framework for incorporating aleatoric and epistemic uncertainty into GNN is introduced in Chapter 6. Basically, the aleatoric uncertainty arising from imprecise information about graph structure (probabilistic links) and node features is propagated via the Assumed Density Filter. Furthermore, this method is agnostic to network architecture, algorithms, and the learning tasks. Experimental results show that the proposed method achieves superior performance in quantifying uncertainties for different levels of input noise across several types of graphs. The confidence interval around the mean prediction improves the usability of predictions related to critical nodes.

Chapter 7 further extends the graph neural-based framework of Chapter 5 to graph combinatorial problems. This kind of problem is related to identifying a set of nodes for a desired objective where the sequence of nodes plays a crucial role in determining the optimal node set. Therefore, a Deep Reinforcement learning-based engine is leverage for learning to identify the desired node sequence, and GNN is used for encoding the underlying network information. Experiments on real-world networks show the computational efficiency and higher scalability of the proposed framework.

In Chapter 8, a systematic study of different network node failure-based robustness metrics is conducted by comparing their similarity and dissimilarity in ranking critical nodes of a power distribution network. Then, the efficacy of these metrics in characterizing voltage fluctuations is accessed by comparing their rankings with that of voltage influencing scores. Results show that the hybrid failure-based metrics can express voltage fluctuations to a reasonable extent. Several key challenges related to the assessment of robustness in present-day complex networks are also highlighted. Finally, Chapter 9 leverages a hetero functional graph theory-based framework for modeling power distribution networks. It highlights the several advantages of HFGT over regular graphs, including their thoroughness in capturing even the smallest system details and their ability to provide a reliable assessment of robustness. In addition, this chapter also introduces a weighted hetero functional graph theory (WHFGT) based framework that can capture dependencies in real-numbers. We conduct various types of attack simulations on the weighted hetero functional graphs and provide in-depth information about the trajectories of graph fragments. Results demonstrate that the proposed WHFGT-based framework and robustness metrics can provide precise insights into the robustness of interdependent infrastructure networks, which highlights the great importance of infrastructure hardening and securing network information.

10.2 Future work

This section presents possible future directions in the areas of novel modeling frameworks for large-scale complex networks and effective methodologies for computing their performance and network-based robustness metrics. The following are the potential extensions to the work presented in this dissertation:

- Chapter 4 provides dominant voltage influencer nodes for a given system in a static manner, i.e., DVI nodes do not change with time. However, in practice, DVI nodes could change with the power change profile. Developing a dynamic framework for DVI nodes can be pursued in the future.

- Chapter 5 presents a graph machine learning-based framework for identifying critical nodes and links in a complex network. It will be interesting to investigate the efficacy of this framework, particularly the link identification module, for determining critical links in power transmission lines. Furthermore, the criticality score in the current methodology is determined by obliterating a single node/link at a time. Extension of this framework from single node/link analysis to concurrent cases where multiple nodes/links would be affected at a time could be another area of further inquiry.

- Chapter 6 generates a confidence interval along with the mean predictions in a graph neural network. However, only the mean predictions are used for minimizing the loss while training the model. The future extension of this work could be on utilizing both the mean and variance values in training the model, thereby, improving the model performance and robustness. Additionally, Bayesian posteriors can be used to quantify the epistemic uncertainty arising due to the distribution of model parameters. However, its computation is not feasible for complex deep learning models, including graph neural networks. Developing an analytical approximation of posterior similar to our ADF approach for aleatoric uncertainty could be a potential future work.

- The learning-based models in Chapters 5 and 6 do not include predefined auto-corrective actions while assessing system robustness at different stages of degradation. It will be interesting to incorporate those factors into our predictive models, thereby improving the applicability of the proposed methodologies for real-world deployments.

- Chapter 7 develops a generic framework for graph combinatorial problem taking Influence maximization as a case study. Trimming this framework for robustness related concurrent node attack study could be a potential area for future research.

- Chapter 8 introduces a weighted hetero functional graph theory for modeling interdependent networks. However, it only considers system concepts, which solely involves structural models. Extension to other hetero-functional models involving control and service graphs could be a potential venue to explore. Moreover, the hetero-functional graphs could be significantly large and sparse for a real-world complex system. Therefore, partitioning and analyzing the graphs on the basis of clusters can ease their robustness analyses. This could also be explored as a potential future task.

- Chapters 3 to 9 focus on developing a computationally efficient and scalable framework for determining system robustness. It will also be interesting to explore deep reinforcement learning and the graph neural network-based framework for developing efficient system restoration models after the onset of extreme events.

Bibliography

[1] W. Bank. Total population. [Online]. Available: https://data.worldbank.org/indicator/SP.POP.TOTL

[2] W. B. urban. Urban population (% of total population). [Online]. Available: https://data.worldbank.org/indicator/SP.URB.TOTL.IN.ZS

[3] M. Talaat, S. Alsayyari, A, A. Alblawi, and Y. Hatata, A, "Hybrid-cloud-based data processing for power system monitoring in smart grids," *Sustainable Cities and Society*, vol. 55, p. 102049, 2020.

[4] S. Oikonomou and M. Parvania, "Optimal coordination of water distribution energy flexibility with power systems operation," *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 1101–1110, 2019.

[5] A. Zajacs and A. Borodinecs, "Assessment of development scenarios of district heating systems," *Sustainable Cities and Society*, vol. 48, p. 101540, 2019.

[6] B. Zhao, J. Conejo, A, and R. Sioshansi, "Unit commitment under gas-supply uncertainty and gas-price variability," *IEEE Transactions on Power System*, vol. 32, no. 3, p. 2394–2405, 2017.

[7] R. Mahmoudi, S. Shetab-Boushehri, R. Hejazi, S, and A. Emrouznejad, "Determining the relative importance of sustainability evaluation criteria of urban transportation network," *Sustainable Cities and Society*, vol. 47, p. 101493, 2019.

[8] S. Hosseini, K. Barker, and J. E. Ramirez-Marquez, "A review of definitions and

measures of system resilience," *Reliability Engineering & System Safety*, vol. 145, pp. 47–61, 2016.

[9] T. M. Yates and A. S. Masten, "Fostering the future: Resilience theory and the practice of positive psychology." 2004.

[10] I. De Terte and C. Stephens, "Psychological resilience of workers in high-risk occupations," *Stress and Health*, vol. 30, no. 5, pp. 353–355, 2014.

[11] C. S. Holling, "Resilience and stability of ecological systems," *Annual review of ecology and systematics*, vol. 4, no. 1, pp. 1–23, 1973.

[12] H. C. Stanley, "Engineering resilience versus ecological resilience," *Engineering within ecological constraints*, vol. 31, no. 1996, p. 32, 1996.

[13] F. Berkes, J. Colding, and C. Folke, *Navigating social-ecological systems: building resilience for complexity and change*. Cambridge University Press, 2008.

[14] W. N. Adger, "Social and ecological resilience: are they related?" *Progress in human geography*, vol. 24, no. 3, pp. 347–364, 2000.

[15] S. Mitra, N. Seifert, M. Zhang, Q. Shi, and K. S. Kim, "Robust system design with built-in soft-error resilience," *Computer*, vol. 38, no. 2, pp. 43–52, 2005.

[16] E. Hollnagel, D. D. Woods, and N. Leveson, *Resilience engineering: Concepts and precepts*. Ashgate Publishing, Ltd., 2006.

[17] A. M. Madni and S. Jackson, "Towards a conceptual framework for resilience engineering," *IEEE Systems Journal*, vol. 3, no. 2, pp. 181–191, 2009.

[18] S. Wang, L. Hong, M. Ouyang, J. Zhang, and X. Chen, "Vulnerability analysis of interdependent infrastructure systems under edge attack strategies," *Safety science*, vol. 51, no. 1, pp. 328–337, 2013.

[19] N. Pandey, A. Pal *et al.*, "Impact of digital surge during covid-19 pandemic: A viewpoint on research and practice," *International Journal of Information Management*, p. 102171, 2020.

[20] M. Kashif, M. K. Javed, D. Pandey *et al.*, "A surge in cyber-crime during covid-19," *Indonesian Journal of Social and Environmental Issues (IJSEI)*, vol. 1, no. 2, pp. 48–52, 2020.

[21] Y. Xiang, L. Wang, and N. Liu, "A robustness-oriented power grid operation strategy considering attacks," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 4248–4261, 2018.

[22] K. Jhala, B. Natarajan, A. Pahwa, and L. Erickson, "Real-time differential pricing scheme for active consumers with electric vehicles," *Electric Power Components and Systems*, vol. 45, no. 14, pp. 1487–1497, 2017.

[23] K. Jhala, B. Natarajan, and A. Pahwa, "The dominant influencer of voltage fluctuation (divf) for power distribution system," *IEEE Transactions on Power Systems*, 2019.

[24] R. Yan and T. K. Saha, "Voltage variation sensitivity analysis for unbalanced distribution networks due to photovoltaic power fluctuations," *IEEE Transactions on Power Systems*, vol. 27, no. 2, pp. 1078–1089, 2012.

[25] T. Jamal and S. Salehin, "Hybrid renewable energy sources power systems," in *Hybrid Renewable Energy Systems and Microgrids*. Elsevier, 2021, pp. 179–214.

[26] S. Munikoti, K. Jhala, K. Lai, and B. Natarajan, "Analytical voltage sensitivity analysis for unbalanced power distribution system," 2020.

[27] S. Munikoti, B. Natarajan, K. Jhala, and K. Lai, "Probabilistic voltage sensitivity analysis to quantify impact of high pv penetration on unbalanced distribution system," *IEEE Transactions on Power Systems*, pp. 1–1, 2021.

[28] S. Munikoti, M. Abujubbeh, K. Jhala, and B. Natarajan, "Spatio-temporal probabilistic voltage sensitivity analysis (st-pvsa)-a novel framework for hosting capacity analysis," *International Journal of Electrical Power and Energy Systems*, vol. 134, 2022.

[29] ——, "An information theoretic approach to identify dominant voltage influencers for unbalanced distribution systems," *IEEE Transactions on Power Systems*, 2022.

[30] M. Abujubbeh, S. Munikoti, and B. Natarajan, "Probabilistic voltage sensitivity based preemptive voltage monitoring in unbalanced distribution networks," in *2020 52nd North American Power Symposium (NAPS)*. IEEE, 2021, pp. 1–6.

[31] L. Das, S. Munikoti, B. Natarajan, and B. Srinivasan, "Measuring smart grid resilience: Methods, challenges and opportunities," *Renewable and Sustainable Energy Reviews*, vol. 130, p. 109918, 2020.

[32] S. Munikoti, L. Das, and B. Natarajan, "Scalable graph neural network-based framework for identifying critical nodes and links in complex networks," *Neurocomputing*, vol. 468, pp. 211–221, 2022.

[33] ——, "Bayesian graph neural network for fast identification of critical nodes in uncertain complex networks," in *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2021, pp. 3245–3251.

[34] S. Munikoti, B. Natarajan, and M. Halappanavar, "Gramer: Graph meta reinforcement learning for multi-objective influence maximization," *arXiv preprint arXiv:2205.14834*, 2022.

[35] S. Munikoti, D. Agarwal, L. Das, M. Halappanavar, and B. Natarajan, "Challenges and opportunities in deep reinforcement learning with graph neural networks: A comprehensive review of algorithms and applications," *arXiv preprint arXiv:2206.07922*, 2022.

[36] W. Schoonenberg and A. Farid, "Modeling smart cities with hetero-functional graph theory," in *Proc. 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Banff Center, Banff, Canada, 2017.

[37] R. Aghatehrani and A. Golnas, "Reactive power control of photovoltaic systems based on the voltage sensitivity analysis," in *2012 IEEE Power and Energy Society General Meeting*. IEEE, 2012, pp. 1–5.

[38] G. Valverde and T. Van Cutsem, "Model predictive control of voltages in active distribution networks," *IEEE Transactions on Smart Grid*, vol. 4, no. 4, pp. 2152–2161, 2013.

[39] A. Newaz, J. Ospina, and M. O. Faruque, "Coordinated voltage control in distribution systems with distributed generations," in *2019 IEEE Power & Energy Society General Meeting (PESGM)*. IEEE, 2019, pp. 1–5.

[40] S. Kang, J. Kim, J.-W. Park, and S.-M. Baek, "Reactive power management based on voltage sensitivity analysis of distribution system with high penetration of renewable energies," *Energies*, vol. 12, no. 8, p. 1493, 2019.

[41] Y. Huang, "Day-ahead optimal control of pev battery storage devices taking into account the voltage regulation of the residential power grid," *IEEE Transactions on Power Systems*, vol. 34, no. 6, pp. 4154–4167, 2019.

[42] F. Ding and B. Mather, "On distributed pv hosting capacity estimation, sensitivity study, and improvement," *IEEE Transactions on Sustainable Energy*, vol. 8, no. 3, pp. 1010–1020, 2016.

[43] H. Pezeshki, A. Arefi, G. Ledwich, and P. Wolfs, "Probabilistic voltage management using oltc and dstatcom in distribution networks," *IEEE transactions on power delivery*, vol. 33, no. 2, pp. 570–580, 2017.

[44] O. A. Afolabi, W. H. Ali, P. Cofie, J. Fuller, P. Obiomon, E. S. Kolawole *et al.*, "Analysis of the load flow problem in power system planning studies," *Energy and Power Engineering*, vol. 7, no. 10, p. 509, 2015.

[45] M. Brenna, E. De Berardinis, F. Foiadelli, G. Sapienza, and D. Zaninelli, "Voltage control in smart grids: An approach based on sensitivity theory," *Journal of Electromagnetic Analysis and Applications*, vol. 2, no. 08, p. 467, 2010.

[46] B. B. Zad, J. Lobry, and F. Vallée, "A centralized approach for voltage control of mv distribution systems using dgs power control and a direct sensitivity analysis method," in *2016 IEEE International Energy Conference (ENERGYCON)*. IEEE, 2016, pp. 1–6.

[47] V. Klonari, B. B. Zad, J. Lobry, and F. Vallée, "Application of voltage sensitivity analysis in a probabilistic context for characterizing low voltage network operation," in *2016 International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*. IEEE, 2016, pp. 1–7.

[48] C. Mugnier, K. Christakou, J. Jaton, M. De Vivo, M. Carpita, and M. Paolone, "Modelless/measurement-based computation of voltage sensitivities in unbalanced electrical distribution networks," in *2016 Power Systems Computation Conference (PSCC)*. IEEE, 2016, pp. 1–7.

[49] G. Valverde, T. Zufferey, S. Karagiannopoulos, and G. Hug, "Estimation of voltage sensitivities to power injections using smart meter data," in *2018 IEEE International Energy Conference (ENERGYCON)*. IEEE, 2018, pp. 1–6.

[50] S. Weckx, R. D'Hulst, and J. Driesen, "Voltage sensitivity analysis of a laboratory distribution grid with incomplete data," *IEEE Transactions on Smart Grid*, vol. 6, no. 3, pp. 1271–1280, 2014.

[51] K. Jhala, B. Natarajan, and A. Pahwa, "Probabilistic voltage sensitivity analysis (pvsa)—a novel approach to quantify impact of active consumers," *IEEE Transactions on Power Systems*, vol. 33, no. 3, pp. 2518–2527, 2017.

[52] V. Boginski and C. W. Commander, "Identifying critical nodes in protein-protein interaction networks," in *Clustering challenges in biological networks*. World Scientific, 2009, pp. 153–167.

[53] X. Wang, E. Pournaras, R. E. Kooij, and P. Van Mieghem, "Improving robustness of complex networks via the effective graph resistance," *The European Physical Journal B*, vol. 87, no. 9, p. 221, 2014.

[54] X. Wang, Y. Koç, R. E. Kooij, and P. Van Mieghem, "A network approach for power grid robustness against cascading failures," in *2015 7th international workshop on reliable networks design and modeling (RNDM)*. IEEE, 2015, pp. 208–214.

[55] H. Wang and P. Van Mieghem, "Algebraic connectivity optimization via link addition," in *Proceedings of the 3rd International Conference on Bio-Inspired Models of Network, Information and Computing Sytems*, 2008, pp. 1–8.

[56] P. Van Mieghem, D. Stevanović, F. Kuipers, C. Li, R. Van De Bovenkamp, D. Liu, and H. Wang, "Decreasing the spectral radius of a graph by link removals," *Physical Review E*, vol. 84, no. 1, p. 016101, 2011.

[57] C. Pizzuti and A. Socievole, "A genetic algorithm for enhancing the robustness of complex networks through link protection," in *International Conference on Complex Networks and their Applications*. Springer, 2018, pp. 807–819.

[58] E.-Y. Yu, Y.-P. Wang, Y. Fu, D.-B. Chen, and M. Xie, "Identifying critical nodes in complex networks via graph convolutional networks," *Knowledge-Based Systems*, vol. 198, p. 105893, 2020.

[59] M. Sun, Y. Jiang, Y. Wang, H. Xie, and Z. Wang, "Identification of critical nodes in dynamic systems based on graph convolutional networks," in *2020 3rd International Conference on Unmanned Systems (ICUS)*. IEEE, 2020, pp. 558–563.

[60] A. Graves, "Practical variational inference for neural networks," *Advances in neural information processing systems*, vol. 24, 2011.

[61] C. Blundell, J. Cornebise, and K. Kavukcuoglu, "D., wierstra. weight uncertainty in neural network," in *Proceedings, of the 32nd International Conference on Machine Learning,(ICML-15)*, 2015, pp. 1613–1622.

[62] J. M. Hernández-Lobato and R. Adams, "Probabilistic backpropagation for scalable learning of bayesian neural networks," in *International conference on machine learning*. PMLR, 2015, pp. 1861–1869.

[63] Y. Mae, W. Kumagai, and T. Kanamori, "Uncertainty propagation for dropout-based bayesian neural networks," *Neural Networks*, vol. 144, pp. 394–406, 2021.

[64] Y. Zhang, S. Pal, M. Coates, and D. Ustebay, "Bayesian graph convolutional neural networks for semi-supervised classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 5829–5836.

[65] S. Pal, F. Regol, and M. Coates, "Bayesian graph convolutional neural networks using non-parametric graph learning," *arXiv preprint arXiv:1910.12132*, 2019.

[66] A. Hasanzadeh, E. Hajiramezanali, S. Boluki, M. Zhou, N. Duffield, K. Narayanan, and X. Qian, "Bayesian graph neural networks with adaptive connection sampling," in *International conference on machine learning*. PMLR, 2020, pp. 4094–4104.

[67] S. Zuloaga, P. Khatavkar, L. Mays, and V. Vittal, "Resilience of cyber-enabled electrical energy and water distribution systems considering infrastructural robustness under

conditions of limited water and/or energy availability," *IEEE Transactions on Engineering Management*, 2019.

[68] J. Najafi, A. Peiravi, A. Anvari-Moghaddam, and M. Guerrero, J, "An efficient interactive framework for improving resilience of power-water distribution systems with multiple privately-owned microgrids," *International Journal of Electrical Power and Energy Systems*, vol. 116, p. 105550, 2020.

[69] G. Wang, Z. Xu, F. Wen, and K. P. Wong, "Traffic-constrained multiobjective planning of electric-vehicle charging stations," *IEEE Transactions on Power Delivery*, vol. 28, no. 4, pp. 2363–2372, 2013.

[70] K. Lai, T. Chen, and B. Natarajan, "Optimal scheduling of electric vehicles car-sharing service with multi-temporal and multi-task operation," *Energy*, vol. 204, p. 117929, 2020.

[71] B. Zhao, J. Conejo, A, and R. Sioshansi, "Coordinated expansion planning of natural gas and electric power systems," *IEEE Transactions on Power System*, vol. 33, no. 3, pp. 3064–3075, 2018.

[72] ——, "Using electrical energy storage to mitigate natural gas-supply shortages," *IEEE Transactions on Power System*, vol. 33, no. 6, pp. 7076–7086, 2018.

[73] Y. Li, Y. Zou, Y. Tan, Y. Cao, X. Liu, S. M, S. Tian, and F. Bu, "Optimal stochastic operation of integrated low-carbon electric power, natural gas, and heat delivery system," *IEEE Transactions on Sustainable Energy*, vol. 9, no. 1, p. 273–283, 208.

[74] M. Ouyang, "Review on modeling and simulation of interdependent critical infrastructure systems," *Reliability engineering & System safety*, vol. 121, pp. 43–60, 2014.

[75] M. Ouyang, L. Hong, Z.-J. Mao, M.-H. Yu, and F. Qi, "A methodological approach to

analyze vulnerability of interdependent infrastructures," *Simulation Modelling Practice and Theory*, vol. 17, no. 5, pp. 817–828, 2009.

[76] E. Wong, A. Tai, and S. So, "Container drayage modelling with graph theory-based road connectivity assessment for sustainable freight transportation in new development area," *Computers & Industrial Engineering*, vol. 149, p. 106810, 2020.

[77] K. Lai and M. Illindala, "Graph theory based shipboard power system expansion strategy for enhanced resilience," *IEEE Transactions on Industry Applications*, vol. 54, no. 6, pp. 5691–5699, 2018.

[78] "Identifying the high-level flow model of water distribution networks using graph theory," in *16th Conference on Water Distribution System Analysis, WDSA 2014*. Elsevier, 2014, pp. 1192–1199.

[79] S. Wagner and N. Neshat, "Assessing the vulnerability of supply chains using graph theory," *International Journal of Production Economics*, vol. 126, no. 1, pp. 121–129, 2010.

[80] B. Murgante and G. Borruso, "Smart cities in a smart world," in *Future city architecture for optimal living*. Springer, 2015, pp. 13–35.

[81] L. Dueñas-Osorio, J. I. Craig, B. J. Goodno, and A. Bostrom, "Interdependent response of networked systems," *Journal of Infrastructure Systems*, vol. 13, no. 3, pp. 185–194, 2007.

[82] J.-Z. Yu and H. Baroud, "Modeling uncertain and dynamic interdependencies of infrastructure systems using stochastic block models," *ASCE-ASME J Risk and Uncert in Engrg Sys Part B Mech Engrg*, vol. 6, no. 2, 2020.

[83] D. Bristow and A. Hay, "Graph model for probabilistic resilience and recovery planning of multi-infrastructure systems," *Journal of Infrastructure Systems*, vol. 23, no. 3, 2017.

[84] A. Ahmed and D. Bristow, "Capturing the spatial and operational interdependencies among building systems using building information modelling," *Structure and Infrastructure Engineering*, vol. 15, no. 12, 2019.

[85] R. Johnson and S. Guikema, "Characterising the robustness of coupled power-law networks," *Reliability Engineering & System Safety*, vol. 191, p. 106560, 2019.

[86] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter, "Multilayer networks," *Journal of complex networks*, vol. 2, no. 3, pp. 203–271, 2014.

[87] A. M. Farid, "Static resilience of large flexible engineering systems: Axiomatic design model and measures," *IEEE Systems Journal*, vol. 11, no. 4, pp. 2006–2017, 2015.

[88] A. R. Malekpour and A. Pahwa, "Radial test feeder including primary and secondary distribution network," in *2015 North American Power Symposium (NAPS)*. IEEE, 2015, pp. 1–9.

[89] H. O. Lancaster and E. Seneta, "Chi-square distribution," *Encyclopedia of biostatistics*, vol. 2, 2005.

[90] L.-L. Chuang and Y.-S. Shih, "Approximated distributions of the weighted sum of correlated chi-squared random variables," *Journal of Statistical Planning and Inference*, vol. 142, no. 2, pp. 457–472, 2012.

[91] M. Nakagami, "The m-distribution—a general formula of intensity distribution of rapid fading," in *Statistical methods in radio wave propagation*. Elsevier, 1960, pp. 3–36.

[92] K. Jhala, V. K. Krishnan, B. Natarajan, and Y. Zhang, "Data-driven preemptive voltage monitoring and control using probabilistic voltage sensitivities," National Renewable Energy Lab.(NREL), Golden, CO (United States), Tech. Rep., 2019.

[93] J. Vasilj, P. Sarajcev, and D. Jakus, "Pv power forecast error simulation model," in

2015 12th International Conference on the European Energy Market (EEM). IEEE, 2015, pp. 1–5.

[94] D. M. Endres and J. E. Schindelin, "A new metric for probability distributions," *IEEE Transactions on Information theory*, vol. 49, no. 7, pp. 1858–1860, 2003.

[95] B. Zhao, Z. Xu, C. Xu, C. Wang, and F. Lin, "Network partition-based zonal voltage control for distribution networks with distributed pv systems," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 4087–4098, 2017.

[96] M. J. Alenazi and J. P. Sterbenz, "Comprehensive comparison and accuracy of graph metrics in predicting network resilience," in *2015 11th International Conference on the Design of Reliable Communication Networks (DRCN)*. IEEE, 2015, pp. 157–164.

[97] W. Ellens and R. E. Kooij, "Graph measures and network robustness," *arXiv preprint arXiv:1311.5064*, 2013.

[98] W. Ellens, F. Spieksma, P. Van Mieghem, A. Jamakovic, and R. Kooij, "Effective graph resistance," *Linear algebra and its applications*, vol. 435, no. 10, pp. 2491–2506, 2011.

[99] D. Fay, H. Haddadi, A. Thomason, A. W. Moore, R. Mortier, A. Jamakovic, S. Uhlig, and M. Rio, "Weighted spectral distribution for internet topology analysis: theory and applications," *IEEE/ACM Transactions on networking*, vol. 18, no. 1, pp. 164–176, 2009.

[100] X. Long, D. Tipper, and T. Gomes, "Measuring the survivability of networks to geographic correlated failures," *Optical Switching and Networking*, vol. 14, pp. 117–133, 2014.

[101] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[102] X. Liu, Y. Li, and R. Xia, "Adaptive multi-view graph convolutional networks for skeleton-based action recognition," *Neurocomputing*, 2020.

[103] X. Yu, S. Lu, L. Guo, S.-H. Wang, and Y.-D. Zhang, "Resgnet-c: A graph convolutional neural network for detection of covid-19," *Neurocomputing*, 2020.

[104] Z. Liang, J. Du, Y. Shao, and H. Ji, "Gated graph neural attention networks for abstractive summarization," *Neurocomputing*, vol. 431, pp. 128–136, 2021.

[105] P. Yan, L. Li, M. Jin, and D. Zeng, "Quantum probability-inspired graph neural network for document representation and classification," *Neurocomputing*, vol. 445, pp. 276–286, 2021.

[106] P. Bongini, M. Bianchini, and F. Scarselli, "Molecular generative graph neural networks for drug discovery," *Neurocomputing*, 2021.

[107] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in neural information processing systems*, 2017, pp. 1024–1034.

[108] A. Arulselvan, C. W. Commander, L. Elefteriadou, and P. M. Pardalos, "Detecting critical nodes in sparse graphs," *Computers & Operations Research*, vol. 36, no. 7, pp. 2193–2200, 2009.

[109] C. Fan, L. Zeng, Y. Ding, M. Chen, Y. Sun, and Z. Liu, "Learning to identify high betweenness centrality nodes from scratch: A novel graph neural network approach," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 559–568.

[110] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[111] J. Yamanaka, S. Kuwashima, and T. Kurita, "Fast and accurate image super resolution

by deep cnn with skip connection and network in network," in *International Conference on Neural Information Processing*. Springer, 2017, pp. 217–225.

[112] M. Tu and X. Zhang, "Speech enhancement based on deep neural networks with skip connections," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5565–5569.

[113] W. Chen, T.-Y. Liu, Y. Lan, Z.-M. Ma, and H. Li, "Ranking measures and loss functions in learning to rank," *Advances in Neural Information Processing Systems*, vol. 22, pp. 315–323, 2009.

[114] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection," http://snap.stanford.edu/data, Jun. 2014.

[115] R. A. Rossi and N. K. Ahmed, "The network data repository with interactive graph analytics and visualization," in *AAAI*, 2015. [Online]. Available: http://networkrepository.com

[116] A.-L. Barabási, R. Albert, and H. Jeong, "Scale-free characteristics of random networks: the topology of the world-wide web," *Physica A: statistical mechanics and its applications*, vol. 281, no. 1-4, pp. 69–77, 2000.

[117] M. E. Newman, "Models of the small world," *Journal of Statistical Physics*, vol. 101, no. 3-4, pp. 819–841, 2000.

[118] P. Holme and B. J. Kim, "Growing scale-free networks with tunable clustering," *Physical review E*, vol. 65, no. 2, p. 026107, 2002.

[119] A. Hagberg, P. Swart, and D. S Chult, "Exploring network structure, dynamics, and function using networkx," Los Alamos National Lab.(LANL), Los Alamos, NM (United States), Tech. Rep., 2008.

[120] L. V. Jospin, W. Buntine, F. Boussaid, H. Laga, and M. Bennamoun, "Hands-on bayesian neural networks–a tutorial for deep learning users," *arXiv preprint arXiv:2007.06823*, 2020.

[121] S. Munikoti, L. Das, and B. Natarajan, "Bayesian graph neural network for fast identification of critical nodes in uncertain complex networks," *arXiv preprint arXiv:2012.15733*, 2020.

[122] W. J. Maddox, P. Izmailov, T. Garipov, D. P. Vetrov, and A. G. Wilson, "A simple baseline for bayesian uncertainty in deep learning," *Advances in Neural Information Processing Systems*, vol. 32, pp. 13 153–13 164, 2019.

[123] F. Liu, P. Zhou, S. J. Baccei, M. J. Masciocchi, N. Amornsiripanitch, C. I. Kiefe, and M. P. Rosen, "Qualifying certainty in radiology reports through deep learning–based natural language processing," *American Journal of Neuroradiology*, vol. 42, no. 10, pp. 1755–1761, 2021.

[124] A. Loquercio, M. Segu, and D. Scaramuzza, "A general framework for uncertainty estimation in deep learning," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3153–3160, 2020.

[125] S. Asthana, O. D. King, F. D. Gibbons, and F. P. Roth, "Predicting protein complex membership using probabilistic network reliability," *Genome research*, vol. 14, no. 6, pp. 1170–1175, 2004.

[126] D. P. Kingma, T. Salimans, and M. Welling, "Variational dropout and the local reparameterization trick," *Advances in neural information processing systems*, vol. 28, pp. 2575–2583, 2015.

[127] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*. PMLR, 2016, pp. 1050–1059.

[128] K. Lee, Z. Wang, B. Vlahov, H. Brar, and E. A. Theodorou, "Ensemble bayesian decision making with redundant deep perceptual control policies," in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*. IEEE, 2019, pp. 831–837.

[129] X. Boyen and D. Koller, "Tractable inference for complex stochastic processes," *arXiv preprint arXiv:1301.7362*, 2013.

[130] J. Gast and S. Roth, "Lightweight probabilistic deep networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3369–3378.

[131] T. P. Minka, "A family of algorithms for approximate bayesian inference," Ph.D. dissertation, Massachusetts Institute of Technology, 2001.

[132] B. J. Frey and G. E. Hinton, "Variational learning in nonlinear gaussian belief networks," *Neural Computation*, vol. 11, no. 1, pp. 193–213, 1999.

[133] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[134] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" *arXiv preprint arXiv:1703.04977*, 2017.

[135] M. Zhang and Y. Chen, "Link prediction based on graph neural networks," *Advances in Neural Information Processing Systems*, vol. 31, pp. 5165–5175, 2018.

[136] K. Kaynar, "A taxonomy for attack graph generation and usage in network security," *Journal of Information Security and Applications*, vol. 29, pp. 27–56, 2016.

[137] A. V. Sathanur, M. Halappanavar, Y. Shi, and Y. Sagduyu, "Exploring the role of intrinsic nodal activation on the spread of influence in complex networks," in *Social Network Based Big Data Analysis and Applications*. Springer, 2018, pp. 123–142.

[138] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2003, pp. 137–146.

[139] M. Farajtabar, N. Du, M. G. Rodriguez, I. Valera, H. Zha, and L. Song, "Shaping social activity by incentivizing users," *Advances in neural information processing systems*, vol. 27, 2014.

[140] N. Friendkin and E. C. Johnsen, "Social influence networks and opinion change," *Adv Group Proc*, vol. 16, pp. 1–29, 1999.

[141] K. Saito, R. Nakano, and M. Kimura, "Prediction of information diffusion probabilities for independent cascade model," in *International conference on knowledge-based and intelligent information and engineering systems*. Springer, 2008, pp. 67–75.

[142] S. S. Singh, A. Kumar, S. Mishra, K. Singh, and B. Biswas, "A centrality measure for influence maximization across multiple social networks," in *International Conference on Advanced Informatics for Computing Research*. Springer, 2019, pp. 195–207.

[143] O. A. Hussain and F. Zaidi, "Influence maximization in complex networks through supervised machine learning," in *International Conference on Complex Networks and Their Applications*. Springer, 2021, pp. 217–228.

[144] M. Minutoli, M. Halappanavar, A. Kalyanaraman, A. Sathanur, R. Mcclure, and J. McDermott, "Fast and scalable implementations of influence maximization algorithms," in *2019 IEEE International Conference on Cluster Computing (CLUSTER)*, 2019, pp. 1–12.

[145] S. Munikoti, D. Agarwal, L. Das, and B. Natarajan, "A general framework for quantifying aleatoric and epistemic uncertainty in graph neural networks," *arXiv preprint arXiv:2205.09968*, 2022.

[146] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[147] J. E. Smith and R. L. Winkler, "The optimizer's curse: Skepticism and postdecision surprise in decision analysis," *Management Science*, vol. 52, no. 3, pp. 311–322, 2006.

[148] H. Hasselt, "Double q-learning," *Advances in neural information processing systems*, vol. 23, pp. 2613–2621, 2010.

[149] M. Botvinick, S. Ritter, J. X. Wang, Z. Kurth-Nelson, C. Blundell, and D. Hassabis, "Reinforcement learning, fast and slow," *Trends in cognitive sciences*, vol. 23, no. 5, pp. 408–422, 2019.

[150] M. Zhang, Z. Cui, M. Neumann, and Y. Chen, "An end-to-end deep learning architecture for graph classification," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[151] T. T. Nguyen, N. D. Nguyen, P. Vamplew, S. Nahavandi, R. Dazeley, and C. P. Lim, "A multi-objective deep reinforcement learning framework," *Engineering Applications of Artificial Intelligence*, vol. 96, p. 103915, 2020.

[152] K. Van Moffaert, M. M. Drugan, and A. Nowé, "Scalarized multi-objective reinforcement learning: Novel design techniques," in *2013 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*. IEEE, 2013, pp. 191–199.

[153] H. Dai, E. B. Khalil, Y. Zhang, B. Dilkina, and L. Song, "Learning combinatorial optimization algorithms over graphs," *arXiv preprint arXiv:1704.01665*, 2017.

[154] S. Manchanda, A. Mittal, A. Dhawan, S. Medya, S. Ranu, and A. Singh, "Gcomb: Learning budget-constrained combinatorial algorithms over billion-sized graphs," 2020.

[155] M. Minutoli, M. Halappanavar, A. Kalyanaraman, A. Sathanur, R. Mcclure, and J. McDermott, "Fast and scalable implementations of influence maximization algorithms,"

in *2019 IEEE International Conference on Cluster Computing (CLUSTER)*. IEEE, 2019, pp. 1–12.

[156] M. J. Alenazi and J. P. Sterbenz, "Evaluation and comparison of several graph robustness metrics to improve network resilience," in *2015 7th International Workshop on Reliable Networks Design and Modeling (RNDM)*. IEEE, 2015, pp. 7–13.

[157] S. Munikoti, K. Lai, and B. Natarajan, "Robustness assessment of hetero-functional graph theory based model of interdependent urban utility networks," *Reliability Engineering & System Safety*, p. 107627, 2021.

[158] Y. Koç, M. Warnier, R. E. Kooij, and F. M. Brazier, "A robustness metric for cascading failures by targeted attacks in power networks," in *2013 10th IEEE International Conference on Networking, Sensing and Control (ICNSC)*. IEEE, 2013, pp. 48–53.

[159] L. Cuadra, S. Salcedo-Sanz, J. Del Ser, S. Jiménez-Fernández, and Z. W. Geem, "A critical review of robustness in power grids using complex networks concepts," *Energies*, vol. 8, no. 9, pp. 9211–9265, 2015.

[160] D. Zhou, F. Hu, S. Wang, and J. Chen, "Power network robustness analysis based on electrical engineering and complex network theory," *Physica A: Statistical Mechanics and its Applications*, vol. 564, p. 125540, 2021.

[161] K. Yamashita and H. Ohsaki, "Effective node resistance and its implication on network robustness," in *2021 International Conference on Information Networking (ICOIN)*. IEEE, 2021, pp. 149–153.

[162] S. Wang, W. Lv, J. Zhang, S. Luan, C. Chen, and X. Gu, "Method of power network critical nodes identification and robustness enhancement based on a cooperative framework," *Reliability Engineering & System Safety*, vol. 207, p. 107313, 2021.

[163] Y. Koç, M. Warnier, R. E. Kooij, and F. M. Brazier, "An entropy-based metric to quantify the robustness of power grids against cascading failures," *Safety science*, vol. 59, pp. 126–134, 2013.

[164] A. Nasiruzzaman, H. Pota, and M. Mahmud, "Application of centrality measures of complex network framework in power grid," in *IECON 2011-37th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2011, pp. 4660–4665.

[165] M. Hassanzadeh, M. Etezadi-Amoli, and M. Fadali, "Practical approach for sub-hourly and hourly prediction of pv power output," in *North American Power Symposium 2010*. IEEE, 2010, pp. 1–5.

[166] ""IEEE PES AMPS DSAS Test Feeder Working Group"," https://cmte.ieee.org/pes-testfeeders/resources/, accessed: 2022-06-02.

[167] ""REDS: REpository of Distribution Systems"," https://www.dejazzer.com/reds/, accessed: 2022-06-02.

[168] H. Esmaeilian, M. Mahmoodabadi, and A. Mahdavinia, "Comprehensive large-scale distribution test networks," 2021.

[169] W. C. Schoonenberg, I. S. Khayal, and A. M. Farid, *A hetero-functional graph theory for modeling interdependent Smart City infrastructure*. Springer, 2019.

[170] T. Wardt and A. M. Farid, "A hybrid dynamic system assessment methodology for multi-modal transportation-electrification," *Energies*, vol. 10, no. 5, p. 653, 2017.

[171] W. Schoonenberg and M. Farid, A, "A dynamic model for the energy management of microgrid-enabled production systems," *Journal of Cleaner Production*, vol. 164, pp. 816–830, 2017.

[172] M. Farid, A, "Multi-agent system design principles for resilient coordination & control

of future power systems," *Intelligent Industrial Systems*, vol. 1, no. 3, pp. 255–269, 2015.

[173] D. Singh, K. Misra, R, and D. Singh, "Effect of load models in distributed generation planning," *IEEE Transactions on Power Systems*, vol. 22, no. 4, pp. 2204–2212, 2007.

[174] K. Sundar and A. Zlotnik, "State and parameter estimation for natural gas pipeline networks using transient state data," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 5, pp. 2110–2124, 2019.

[175] D. Sherali, H, S. Subramanian, and V. Loganathan, G, "Effective relaxations and partitioning schemes for solving water distribution network design problems to global optimality," *Journal of Global Optimization*, vol. 19, pp. 1–26, 2001.

[176] T. Chen, B. Zhang, H. Pourbabak, A. Kavousi-Fard, and W. Su, "Optimal routing and charging of an electric vehicle fleet for high-efficiency dynamic transit systems," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 3563–3572, 2018.

[177] J. Zheng, Z. Zhou, J. Zhao, and J. Wang, "Function method for dynamic temperature simulation of district heating network," *Applied Thermal Engineering*, vol. 123, pp. 682–688, 2017.

[178] S. Munikoti. Interdependent urban networks. [Online]. Available: https://github.com/saimunikoti/Interdependent_Urban_Networks

[179] D. S. Callaway, M. E. Newman, S. H. Strogatz, and D. J. Watts, "Network robustness and fragility: Percolation on random graphs," *Physical review letters*, vol. 85, no. 25, p. 5468, 2000.

[180] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT press, 2009.

[181] C. Gao, D. Wei, Y. Hu, S. Mahadevan, and Y. Deng, "A modified evidential methodology of identifying influential nodes in weighted networks," *Physica A: Statistical Mechanics and its Applications*, vol. 392, no. 21, pp. 5490–5500, 2013.

[182] A. Ahmed, M, A. Abubakar, and S. Alghamdi, J, "A study on the uncertainty inherent in class cohesion measurements," *Journal of Systems Architecture*, vol. 57, pp. 474–484, 2011.

[183] S. Yang, "Networks: An introduction by mej newman: Oxford, uk: Oxford university press. 720 pp., 85.00$," 2013.

Appendix A

Nomenclature

A.1 Acronyms

VSA	Voltage sensitivity analysis
PVSA	Probabilistic voltage sensitivity analysis
DERs	Distributed energy resources
PV	Photovoltaic
BESS	Battery energy storage system
DVI	Dominant voltage influencer nodes
VIS	Voltage influencing score
KL	KL divergence
BC	Bhattacharyya distance
MCS	Monte-Carlo simulation
GNN	Graph Neural Network
GCN	Graph convolutional Network
ILGR	Inductive Learner for Graph Resilience
BILGR	Bayesian Inductive Learner for Graph Resilience
PL	Power law

PLC Power law cluster

SBM Stochastic block model

ADF Assumed density filtering

BGCN Bayesian Graph convolutional Network

MAP Maximum aposteriori estimate

NLL Negative log likelihood

DRL Deep Reinforcement learning

GraMeR Graph Meta reinforcement learning

IM Influence Maximization

AIM Activation informed Influence Maximization

MDP Markov decision process

GHC Greedy hill climbing

NEF Network efficiency

ENR Effective node resistance

ECD Electrical coupling connection degree

APL Active power flow loss

ENS Electrical node significance

EBW Electrical betweenness

EDG Electrical degree

ENT Electrical node robustness

HFGT Hetero-functional graph theory

WHFGT Weighted Hetero-functional graph theory

IUN Interdependent urban network

QoF Quality of functionality

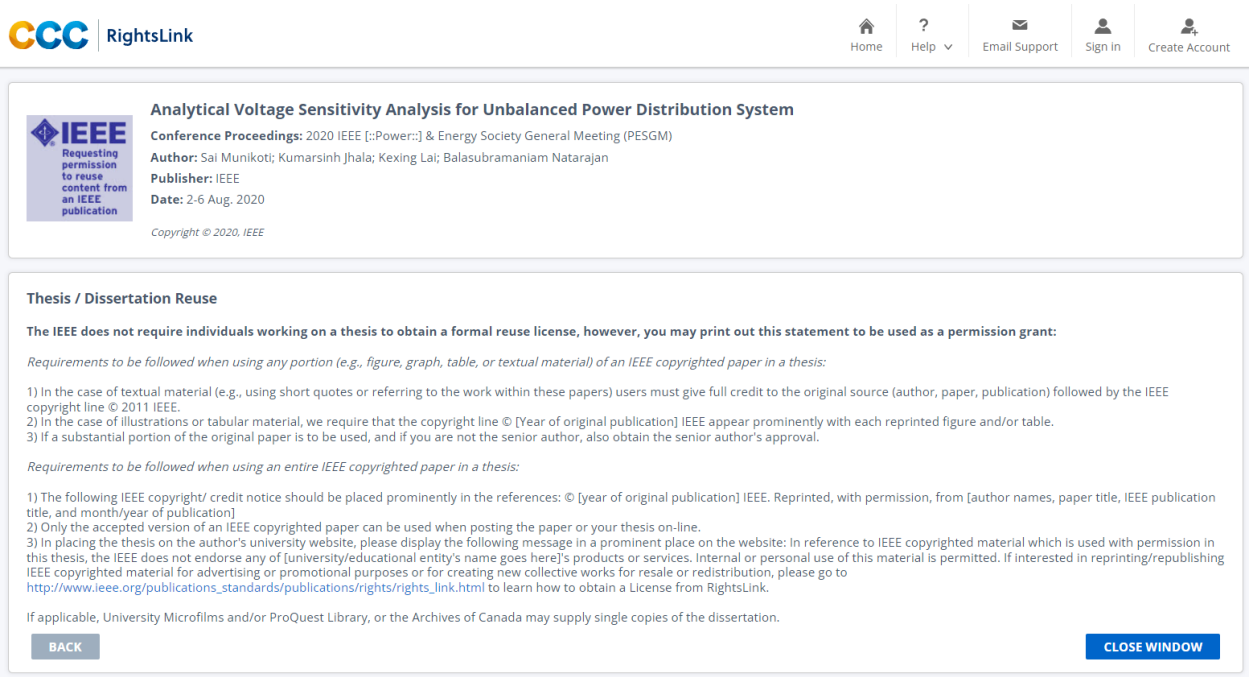
LCC Largest connected component

NCC Number of connected components

FR Flow robustness

Appendix B

Reuse permissions from publishers



CCC RightsLink

Home | Help | Email Support | Sign in | Create Account

IEEE Requesting permission to reuse content from an IEEE publication

Probabilistic Voltage Sensitivity based Preemptive Voltage Monitoring in Unbalanced Distribution Networks

Conference Proceedings: 2020 52nd North American Power Symposium (NAPS)
Author: Mohammad Abujubbeh; Sai Munikoti; Balasubramaniam Natarajan
Publisher: IEEE
Date: 11-13 April 2021

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK CLOSE WINDOW

CCC RightsLink

Home | Help | Email Support | Sign in | Create Account

Robustness assessment of Hetero-functional graph theory based model of interdependent urban utility networks

Author: Sai Munikoti,Kexing Lai,Balasubramaniam Natarajan
Publication: Reliability Engineering & System Safety
Publisher: Elsevier
Date: August 2021

Journal Author Rights

Please note that, as the author of this Elsevier article, you retain the right to include it in a thesis or dissertation, provided it is not published commercially. Permission is not required, but please ensure that you reference the journal as the original source. For more information on this and on your other retained rights, please visit: https://www.elsevier.com/about/our-business/policies/copyright#Author-rights

BACK CLOSE WINDOW

CCC RightsLink

Home | Help | Email Support | Sign in | Create Account

Measuring smart grid resilience: Methods, challenges and opportunities

Author: Laya Das,Sai Munikoti,Balasubramaniam Natarajan,Babji Srinivasan
Publication: Renewable and Sustainable Energy Reviews
Publisher: Elsevier
Date: September 2020

Journal Author Rights

Please note that, as the author of this Elsevier article, you retain the right to include it in a thesis or dissertation, provided it is not published commercially. Permission is not required, but please ensure that you reference the journal as the original source. For more information on this and on your other retained rights, please visit: https://www.elsevier.com/about/our-business/policies/copyright#Author-rights

BACK CLOSE WINDOW