Explainable and Network-Based Approaches for Decision-making in Emergency Management

Anika Tabassum

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computer Science and Applications

B. Aditya Prakash, Chair
Ryan Gerdes, Co-chair
Naren Ramakrishnan
Anuj Karpatne
Supriya Chinthavali

September 10, 2021
Blacksburg, Virginia

Keywords: Critical Infrastructure, Urban analytics, Multivariate Time-series, Urban-Net, Explanations

Explainable and Network-Based Approaches for Decision-making in Emergency Management

Anika Tabassum

(ABSTRACT)

Critical Infrastructures (CIs), such as power, transportation, healthcare, etc., refer to systems, facilities, technologies, and networks vital to national security, public health, and socio-economic well-being of people. CIs play a crucial role in emergency management. For example, the recent Hurricane Ida, Texas Winter storm, colonial cyber-attack that occurred during 2021 in the US, shows the CIs are highly inter-dependent with complex interactions. Hence power system failures and shutdown of natural gas pipelines, in turn, led to debilitating impacts on communication, waste systems, public health, etc. Consider power failures during a disaster, such as a hurricane. Subject Matter Experts (SMEs) such as emergency management authorities may be interested in several decision-making tasks. Can we identify disaster phases in terms of the severity of damage from analyzing changes in power failures? Can we tell the SMEs which power grids or regions are the most affected during each disaster phase and need immediate action to recover? Answering these questions can help SMEs to respond quickly and send resources for fast recovery from damage. Can we systematically provide how the failure of different power grids may impact the whole CIs due to inter-dependencies? This can help SMEs to better prepare and mitigate the risks by improving system resiliency.

In this thesis, we explore problems to efficiently operate decision-making tasks during a disaster for emergency management authorities. Our research has two primary directions, guide decision-making in resource allocation and plans to improve system resiliency. Our work is done in collaboration with the Oak Ridge National Laboratory to contribute impactful research in real-life CIs and disaster power failure data.

1. **Explainable resource allocation:** In contrast to the current interpretable or explainable model that provides answers to understand a model output, we view explanations as answers to guide resource allocation decision-making. In this thesis, we focus on developing a novel model and algorithm to identify disaster phases from changes in power failures. Also, pinpoint the regions which can get most affected at each disaster phase so the SMEs can send resources for fast recovery.

2. **Networks for improving system resiliency:** We view CIs as a large heterogeneous network with nodes as infrastructure components and dependencies as edges. Our goal is to construct a visual analytic tool and develop a domain-inspired model to identify the important components and connections to which the SMEs need to focus and better prepare to mitigate the risk of a disaster.

Explainable and Network-Based Approaches for Decision-making in Emergency Management

Anika Tabassum

(GENERAL AUDIENCE ABSTRACT)

Critical Infrastructure Systems (CIs) entitle multiple infrastructures valuable for maintaining public life and national security. E.g., power, water, transportation. US Federal Emergency Management Agency (FEMA) aims to protect the nation and citizens by mitigating all hazards during natural or man-made disasters. For this, they aim to adopt different decision-making strategies efficiently. E.g., During an ongoing disaster, when to quickly send resources, which regions to send resources first, etc. Besides, they also need to plan how to prepare for a future disaster and which CIs need maintenance to improve system resiliency.

We explore several data-mining problems which can guide FEMA towards developing efficient decision-making strategies. Our thesis emphasizes explainable and network-based models and algorithms to help decision-making operations for emergency management experts by leveraging critical infrastructures data.

All praises to the ALMIGHTY ALLAH, the Most Beneficent, the Most Merciful

Acknowledgments

Like everyone, my PhD journey consists of many unforgettable memories and ups and downs. All these memories have taught me different perspectives and made me more mature and resilient. I wish to express my sincerest gratitude to all those who were by my side and have assisted me in coming so far in my journey.

First, I would like to extend my deepest gratitude with my advisor Dr. B. Aditya Prakash, whose guidance and support throughout this journey was invaluable to me. Aditya's mentoring was not just restricted to research, his vision, ambition, pointing my weakness, and how to improve from there taught me greatly as to how I should walk the path I have chosen. I am particularly thankful for his constant persistence, which motivated me to learn at a fast pace and overcome the fear of accepting any challenges. To summarize, this degree may not have been possible without his constant advice and support, and I am forever thankful for that.

I sincerely thank all other members of my dissertation committee for their valuable comments and critics during my prelim and final defense- Prof. Ryan Gerdes, Prof. Naren Ramakrishnan, Prof. Anuj Karpatne, and Supriya Chinthavali. Their assistantship and guidance were very important towards substantially improve this dissertation.

I want to thank Supriya Chinthavali for always being patient yet providing the mentorship I needed to collaborate with an interdisciplinary team. Without her initiation, her quick response to providing me with timely datasets, and constant feedback during research discussions, my network-based works would never have been completed. My interdisciplinary work would have been impossible without my collaborators at Oak Ridge National Laboratory (ORNL), Dr. Sangkeun Lee, Nils Stenvig, and the entire Urban-Net team. Their

enthusiasm, domain knowledge, and constant feedback were great blessings for me.

I sincerely thank Prof. Naren Ramakrishnan for his practical ideas and feedback during our collaboration, which helped immensely improve my works. I am indebted to him for the financial support I received through NSF UrbComp in my last two years. Thanks to my Virginia Tech collaborators Nikhil Muralidhar, Dr. Liangzhe Chen, and Dr. Sorour Amiri. I still remember moments of our submission deadlines, working on experiments, and reviewing papers. In retrospect, perhaps, these are the moments I enjoyed the most in this entire journey.

This entire journey would be hard without my lab members at Sanghani Center of AI & Data Analytics, Sorour, Bijaya, Alex, Nikhil, Sathappan, and Debanjan, too name a few. I owe Nikhil and Sathappan for all their thoughtful suggestions in my papers, providing the materials, contacts, and helpful feedbacks in my presentation to strive during my job search. I will never forget our coffee breaks and afternoon walks.

Thanks to Roxanne Paul, Wanawsha Hawrami, Sharon Kinder-Potter, Corinne Julien, and Juanita Victoria for their administrative support over the years. Specific thanks to Roxanne and Wanawsha for their help and immense patience.

I am deeply thankful to my parents for tolerating my best and worst, for their never-ending prayers and fasts for my success. Thanks to my brother Saif who has always been my inspiration, mentor, and counselor in distress.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Critical infrastructures (CIs), such as power, transportation, healthcare, etc., play a crucial role in emergency management. Failure of even a small part in CIs, caused by any natural or human-made disaster, can trigger widespread cascading failures impacting many other interdependent facilities and disrupt the functionality of the entire CIs. Consider multivariate time-series power failure sequences during a hurricane. Each sequence represents the number of outages that occurred in a county throughout the time of the hurricane. To manage such disasters, Subject Matter Experts (SMEs) always look for multiple decision-making tasks. Can we identify different phases of the hurricane in terms of the severity of the damage? Can we pinpoint the counties that may be most affected at each phase where SMEs can send resources for fast recovery? For monitoring such scenarios, SMEs need human-friendly explanations that can guide them towards efficient resource allocation for fast recovery. Again, consider nations' critical infrastructures (CIs). Can we analyze the real-time consequences of CIs as to which components are the most vulnerable and need immediate maintenance to reduce risks during a disaster? Analyzing such scenarios can guide SMEs to plan improving system resiliency.

This thesis explores the decision-making tasks mentioned above that emerge from the power failures and critical infrastructure inter-dependencies. We focus on solving such problems from the data-science viewpoint in two directions: explainable models for resource allocation and network-based frameworks for improving system resiliency.

1.1 Motivation

The US Federal Emergency Management Authority (FEMA) states four operations for an efficient decision-making: response, recovery, preparedness, and mitigation [7, 84]. Fig. 1.1 shows four phases of decision-making for a disaster.

Decision-making Operations:

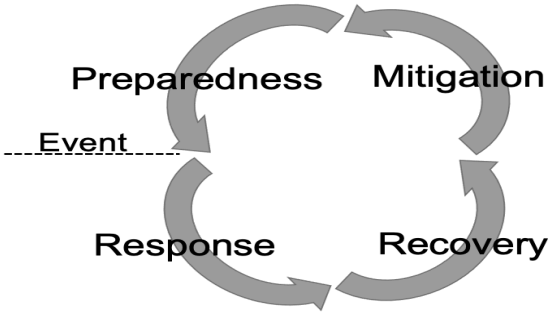


Figure 1.1: **Decision-making phases for emergency management (Adapted from FEMA [7]).**

- *Response:* "Conducting emergency operations to save lives and property by taking action to reduce the hazard to acceptable levels."
- *Recovery:* "Rebuilding communities so that individuals, businesses, and governments can function on their own, return to normal life, and protect against future hazards."
- *Preparedness:* "Building the emergency management function to respond effectively to, and recover from, any hazard."
- *Mitigation:* "Taking sustained actions to reduce or eliminate long-term risk to people and property from hazards and their effects."

Applications and datasets. We emphasize that all our projects are directly motivated by the real-life problems faced by emergency management or public health domain experts.

Almost all our works are in collaboration, scrutinized, and acknowledged by the domain experts at Oak Ridge National Laboratory (ORNL). We study mainly three categories of datasets- (i) Emergency management scenarios include multivariate time- series of power failures data during recent hurricanes. The data is collected from the ORNL Power outage situation awareness tool Eagle-I. (ii) CIs: The original data is collected from two sources, ORNL North American Energy Resiliency Model (NAERM) project and the public HiFld dataset from HomeLand Infrastructures. (iii) Public Health: Our work includes public health data from different public sources of historical and recent infection cases, such as recent Covid-19, Tuberculosis, Diphtheria.

1.2 Organization of Thesis

In this section, we first give an overview of our thesis, introduce and explain our thesis statement, and briefly discuss all our problems.

1.2.1 Thesis Statement

Explainable resource allocation and Networks for improving system resiliency can guide towards efficient decision-making operations in emergency management.

Explainable resource allocation: The recent interpretable and explainable ML are developed for providing answers to which humans can understand the cause of a model decision [78]. For emergency management, we view explanations as human-friendly answers which can guide SMEs for efficient resource allocation during a disaster. For the thesis, we focus on two types of answers, identify disaster phases where significant damage has occurred and pinpoint regions that are significantly affected at each disaster phase. These 'important'

phases and regions can help SMEs send resources and recover quickly from damage.

Networks for system resiliency: We emphasize developing network-based approaches on national level CIs data for planning and improving resiliency through situation awareness. By leveraging networks, we can develop models to be applied to different systems using less domain knowledge. We consider a heterogeneous network for the critical infrastructure system, where nodes represent different CIs components and edges represent the inter-dependent connections. A significant part of the network-based framework consists of graph construction, verification of inter-dependencies, and building a visual analytic tool. We next develop a domain-inspired model to identify trigger components that can cause significant failure in the power system to guide SMEs with power contingency analysis.

Limitations of existing methods. For identifying disaster phases from power failure data, a natural way is to observe how power failure in one region impacts another region. The existing methods are inadequate to model such observation and identify disaster phases systematically for the SMEs. Besides, there is no relevant model to pinpoint significantly affected regions at each disaster phase. First, the state-of-the-art multivariate time-series change point (event) detection algorithms [55, 72, 76], which were proposed in different domains, do not incorporate the spatial information of a disaster track. Second, the existing explainable ML concepts have been so far proposed for classification [66, 91], non-sequential data [45, 103], or to interpret model behavior [47, 75]. We want novel explainable techniques to pinpoint regions for disaster phases for multivariate sequential data. Third, constructing a national-level network systematically from large-scale heterogeneous CIs data (whole US national) is crucial. There is no direct domain rule to define the inter-dependencies among CIs. Existing state-of-the-arts do not adapt to capture national-level CIs inter-dependencies and their failures during different disasters. Finally, the simulation models to identify important CIs and analyze their failures' impact are also expensive.

To summarize, we focus on bringing two different data science perspectives for handling decision-making efficiently. Explainable resource allocation can guide decision-making in response and recovery, and network-based frameworks for improving system resiliency can help decision-making in preparedness and mitigation. To identify disaster phases from multivariate power failure sequences during a hurricane, we develop an automatic segmentation model to capture non-trivial failure changes occurring with hurricane track. We introduce an explainable model to pinpoint the regions that may get affected significantly at each phase. We assert that these models can provide human-friendly explanations for SMEs and help decision-making for response and recovery. We show that our proposed algorithms can extract promising insights that match with historical scenarios. In collaboration with the ORNL, we develop a visual analytic tool for real-time situational awareness before a disaster. We also develop a domain inspired model to identify critical power components and guide traditional power contingency analysis tool.

1.2.2 Thesis Structure

We organize our thesis outline into two directions: explainable resource allocation for response and recovery and network-based approaches for preparedness and mitigation. The thesis structure is in the following list.

1. **Explainable Resource allocation for Response and Recovery**
    (a) Problem 1: Identifying Disaster Phases (**Chapter 3**)
    (b) Problem 2: Pinpointing Regions for Disaster Phases (**Chapter 4**)
2. **Network-based System resiliency for Preparedness and Mitigation**

(a) Problem 3: Improving situation awareness through a Visual Analytic Tool **Chapter 5**

(b) Problem 4: Efficient Power Contingency Analysis through Network Trigger Nodes (**Chapter 6**)

1.3 Part I: Explainable Resource Allocation for Response and Recovery

In this section, we discuss the problems and models we focus for explainable resource allocation. We first identify disaster phases in terms of sudden changes in multivariate power failure sequences during hurricane. Next, we discuss our novel explanation problem of pinpointing the regions for each disaster phase.

1.3.1 Problem 1: Identifying Disaster Phases

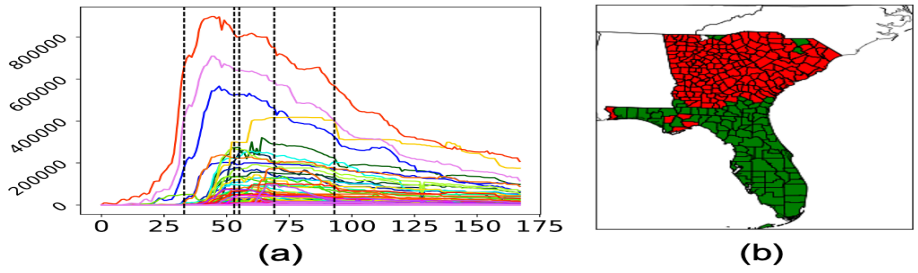


Figure 1.2: **An example of the holistic spatial and temporal analysis results from our novel CnR model to analyze the damage to the power grid during Hurricane Irma.**

We characterize the severity of power failure by measuring the number of outages in a hurricane-affected region (a county in our case) over the entire period of the hurricane. Three critical questions need to be answered for the characterization of this process: Q1.

How can we identify different phases of a hurricane as a function of the severity of the damage (sudden changes) to critical infrastructure (CIs) like the power grid, using sparse power outage data? Q2. Which counties are the culprit for characterizing each phase? Q3. How can counties be grouped based on their overall failure dynamics during the hurricane? Our main goal is to help emergency management SMEs answer the questions, as mentioned above. The limitations of existing works in temporal segmentation are that not many of them easily incorporate spatial information in temporal segmentation. None of the existing models provide any explanation framework wherein 'culprit' counties (at each segment of a power failure) can be identified in space and time. Most segmentation algorithms focus only on identifying temporal cut points [25], with a few applications in computer vision and video analysis modeling spatial relationships [32, 100]. However, no work has identified culprit time-series (i.e., counties), thereby explaining each identified temporal segment. Our algorithm, Cut-n-Reveal (CnR) contains two parts: detecting a good segmentation of the power loss data to capture the main changes, and finding the corresponding explanations (a subset of culprit counties) per segment. For segmentation, we assume a known underlying graph structure captures the relationship among these power failure time-series. For example, in CIs, the number of electric outages in all disaster-affected counties form a set of time series. The relationship among these counties can be based on their geographical proximity. With this knowledge of the segmentation and the explanations for each segment, the expert has a holistic picture of the different phases of the failure process and the specific time series that contributed significantly to each phase change. Fig. 1.2 shows temporal and spatial segmentation found by CnR in Hurricane Irma. Fig 1.2(a) depicts the overall temporal segmentation over a dataset where each time-series indicates the total number of households that lost power in a single county throughout Hurricane Irma (household count per county is recorded every hour). Fig. 1.2(b) represents the spatial clustering of all counties that experienced significant damage (in green) to power infrastructure through the course of

Hurricane Irma, essentially representing the spatial span of damage. We will explain our segmentation and corresponding explanation model of CnR in Chapter 3.

1.3.2 Problem 2: Pinpointing Regions for Disaster Phases

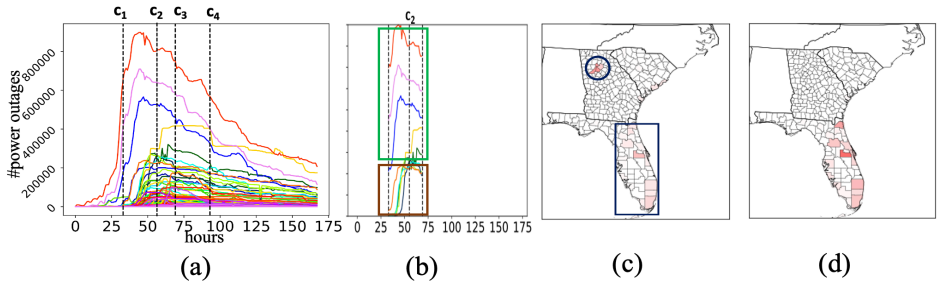


Figure 1.3: **Disaster example (a): 2017 Hurricane Irma times-series (representing power failures of a county during the hurricane). The first and last change point are based on the hurricane landfall (Sept. 10) and end time (Sept. 12). Change points c_2 and c_3 is around the time when hurricane is changing trajectory. (b) shows a snippet and (c) a heatmap of top 10 most important culprit counties found by our algorithm across c_2. (d) shows a simple magnitude-based culprits heatmap across c2. Brighter colors indicate larger importance weights ((c),(d)).**

In Sec. 1.3.1 we design our CnR explanation model to only capture explanations for the corresponding CnR segmentation. However, urban domain experts may use multiple segmentation algorithms [40, 52, 55, 76] based on their data for identifying meaningful events. They want to understand which signals are the 'culprit' or more susceptible across each particular event irrespective of the segmentation algorithm they used. However, most time-series segmentation algorithms are usually complex. They do not give ready actionable explanations to identify the culprits across events. Also, tracking the model behavior for each of these segmentation algorithms (as we did in Sec. 1.3.1) will give different time-series as culprits. Our research question is, how can we then identify 'actionable time-series' as an explanation for events found by any segmentation algorithm?

What is 'actionable time-series'?: For example, consider Fig. 1.3 for the 2017 Hurricane

Irma. Fig. 1.3(a) shows the hurricane time-series data with the segmentation, which roughly corresponds to different phases. Fig. 1.3(b) shows the failure time-series snippet of top 10 'culprit' counties (i.e. time-series) across the second cutpoint (c_2) found by our algorithm. We see some counties undergo a significant change of power failures (green box), which are useful to guide resource allocation as they denote widespread problems. At the same time, there are some less obvious counties near Atlanta, too (brown box in Fig 1.3(b)). These counties have relatively lower but a sudden change of failures compared to the other counties. Further, see the geographic heatmap plot of our culprits (based on their importance weights as learned by our algorithm) in Fig. 1.3(c). Clearly, these counties (denoted by the blue circle) are not obvious: they are located in the north-west corner, far away from where the hurricane is present (the blue rectangle, where the rest of the important culprits are). However, they are still important and useful for situational awareness and resource allocation. Indeed, reports [22] suggest that this is due to a separate tropical storm happening at this time. Note that figuring out these counties by just visualizing the time-series (Fig. 1.3(a)) across c_2 is hard (as they get buried). Further, if we plot a heatmap of all the counties based on their change of magnitude across c_2 (Fig. 1.3(d)), we *cannot* recover these counties located at the middle (as their magnitudes are relatively lower).

Hence, to automatically capture human-friendly and actionable culprits as the explanation for the associated phases or events given by any black-box segmentation, we propose a novel problem Rationalization for time-series segmentation (RaTSS) in terms of constituent time-series.

1.3.3 Lessons Learned

- Leveraging spatial relations among the disaster-affected regions is necessary to identify disaster phases. This spatial relation can be viewed as domain knowledge to be

applicable for general multivariate time-series data.

- Providing explanations for a black-box time-series segmentation is crucial and needs multiple assumptions. First, cutpoints in the segmentation represent meaningful events. Also, explanations can be actionable in terms of constituent time-series for the corresponding event.

1.3.4 Contributions

- **Novel segmentation model for power failure sequences.** We propose a novel segmentation model to capture both temporal and spatial relations among the regions experiencing power failures to identify disaster phases. Our proposed model view power failure process as a time-series segmentation problem. We also propose a novel explanation model to identify the *important* regions for each cutpoint found by our segmentation model.

- **Novel explainable problem for black-box segmentation in multivariate time-series.** We propose a novel problem and an algorithm for the urban experts, which finds human-friendly and actionable time-series as explanation for the events provided by a black-box segmentation.

- **Incorporate domain knowledge for multivariate time-series segmentation and explanation.** For our segmentation model, to incorporate domain knowledge we leverage a network structure to represent spatial relation among the regions. For our explainable problem, we represent multivariate time-series in terms of statistical features which can be viewed as domain knowledge representation for multivariate sequences.

- **Application for ongoing disaster scenarios.** Our segmentation and explanation model can be applied to identify phases for an ongoing hurricane using real-time power failure data and subsume the projected hurricane track as a network structure.

1.4 Part II: Network-based System Resiliency for Preparedness and Mitigation

In collaboration with the ORNL, we develop network-based frameworks for improving system resiliency to support decision-making. We focus on developing a visual analytic tool for situation awareness. Next, we develop a domain-inspired model to guide power system contingency analysis for identifying CIs which may trigger significant failures.

1.4.1 Problem 3: Improving Situation awareness through Visual Analytic Tool

We construct a heterogeneous CIs network from a publicly available geographic shapefile data [4]. This project is built around the idea of viewing CIs as large heterogeneous networks. We build the inter-dependencies among different layers of CIs using multiple suitable techniques ranging from the nearest neighbor, SMEs recommended connections, and exploiting real disaster scenarios. We develop the network-based interactive visualization tool *Urban-net* to view inter-dependencies and failure dynamics over CIs. The visual interface utilizes a network topology-based analysis and simulation model to understand temporal consequences along with inter-dependencies.

1.4.2 Problem 4: Efficient Power Contingency Analysis through Network Trigger Nodes

Modeling failure dynamics within a power system is a complex and challenging process due to multiple inter-dependencies and convoluted inter-domain relationships. Subject matter experts (SMEs) are interested in understanding these failure dynamics for planning future disasters and contingencies associated with them (i.e., losses or failures of power system components, such as transmission lines). Contingency analysis (CA) tools enable such 'what-if' scenario analysis to evaluate the impacts on the power system. As it is impractical to analyze all possible contingencies among N system components, an important step for performing CA is identifying a set k of 'trigger' components, which when failed initially can significantly impact the overall system by causing multiple failures. Currently SMEs focus on identifying these trigger components by running expensive simulations on all possible subsets, which quickly becomes infeasible. Hence finding a relevant set of trigger components (contingencies) rapidly to enable efficient and useful CA is crucial. We aim to make CA more efficient by tackling it from a heterogeneous power system network viewpoint. First, we construct a network with multiple electric grid infrastructure components and dependencies as connections among them. We reformulate the problem of finding a set of trigger components as a problem of identifying critical nodes in the network, which can cascade power failures through connected nodes and cause significant damage to the network. To guide the practical CA tools, we develop a probabilistic model for setting up networks' edge-weights using intricate domain rules.

1.4.3 Lessons Learned

- Visual Analytic tool is important to understand inter-dependencies for national-level CIs.

- Analyzing CIs failures using network topology at least helps understand the worst-case consequences of a disaster.
- State-of-the-art network cascade models are not enough to capture trigger components in power contingency analysis (CA). It is essential to consider domain guided model to map CA to networks.

1.4.4 Contributions

- **Real large-scale heterogeneous CIs network.** Our first contribution is to build a network using US national-level CIs data. We improve ORNL's Urban-Net tool for modeling failures, capture domain-based failure dynamics, and identifying vulnerabilities.
- **Situation awareness tool.** We combine the what-if analysis techniques (i.e., network converter, modeling, and failure dynamic simulation) in an interactive tool Urban-Net, using different open source technologies. From an application viewpoint, Urban-Net can significantly enhance situational awareness for emergency management and draw significant stakeholders' interest.
- **Domain-inspired model for expensive contingency analysis in power system.** We bring a novel technique to identify critical nodes by developing a domain-inspired probabilistic model in the network to guide the practical contingency analysis in power simulation. Our domain-inspired model is an idea to incorporate domain knowledge in power system simulations.

We organize the thesis as follows. First, we survey lines of research related to our works in Chapter 2. Next, Chapter 3- 4 discuss our problems and models on explainable resource

allocation for decision-making regarding response and recovery (Part I). Chapter 5- 6 discuss our work in network-based models for decision-making in preparedness and mitigation (Part II). Finally, we summarize our contributions, limitations, and possible future direction to overcome the limitations. All the codes are publicly available for research purposes [15].

Chapter 2

Literature Survey

In this Section, we survey about the related literature on retrospective and what-if mining in urban computing for emergency management and public health, Mining in multivariate time-series, ML interpretable models, and Analyzing CIs, such as identifying vulnerable facilities, developing situation-awareness models, and what-if mining.

2.1 Urban Computing Applications

Several urban computing applications in energy and public health domain have been stated for past years [108, 110]. Oak Ridge National Laboratory (ORNL) has developed a real-time situational awareness tool Eagle-I [46] to monitor and analyze the nation's energy infrastructure. The ORNL EARSS team [30] has an automated model to take wind speed and location estimates provided by hurricane experts a geospatial assessment on the impact to the electric grid in terms of projected duration of the outage. Hafyani et al. [48] designed a change point detection process to detect the transition in multidimensional environmental crowdsensing data. Moss et al.[79] shows how GPS mobility data can imply a spatial spread of influenza-like infectious diseases. Wah et al. [106] analyzed on time-series Tuberculosis (TB) data to characterize the demographic and temporal trends of the impact of the disease. Various works have been discussed recently on understanding and quantifying the impact of COVID-19 from different types of Non-pharmaceutical (NPI) interventions and exit strate-

gies taken in Europe and China. [68, 94]. [111] developed an interactive visualization tool to observe mobility and sociability trends in different regions in the US due to COVID-19.

2.2 Critical Infrastructures (CIs) Application

In this section, we provide a few examples of work on CIs dealing with power and energy As representative work, we focus on methods that identify vulnerable facilities to protect against unknown natural disasters with known patterns and strategies from network modeling perspective, developing situation-awareness models, and tools for what-if analysis.

2.2.1 Vulnerablility Identification

Identifying vulnerable critical facilities in a power system is necessary to protect and enhance them against unknown natural disasters. The state-of-the-art can be divided into mainly two different techniques: using only the network structure and/or incorporating failure cascade dynamics as well. Several algorithms have been proposed in order to identify critical nodes using only the network structure [26, 34, 37]. Arianos et al. [26] introduced the concept of using geodesic distance for power flow to calculate resiliency of power grids. Chen et al. [37] proposed an algorithm *OPERA* for the connectivity control problem. It aims to find a set of optimal nodes that maximizes the impact on an interdependent CIs network consisting of physical, control and communication layers. In addition, they developed *SUBLINE*, to unify a family of prevalent subgraph connectivity measures to quantify network dependency of the graph and subsequently use it for identifying the optimal nodes.

Additionally, incorporating failure dynamics can help prevent catastrophic failures of the whole system. The 2003 blackout shows an instance where a transmission line failure cas-

caded to failures of water, waste-treatment, and communication systems. Buldyrev et al. [34] developed a framework based on mutually connected clusters to study cascading failures in interdependent networks. They analyzed the presence of a giant connected cluster under simple random failures on Erdos-Renyi networks and showed a phase transition and a critical threshold. In contrast, Chen et al. [39] developed an algorithm *HOTSPOTS* to model more complicated failure cascades and identify critical nodes that may lead to substantial failures. Their heterogeneous network consists of power plants, substations, transmissions and gas compressors. They propose a 'path-based' failure cascade model on this complex system representing how every component of a CIs network interacts with each other. Also, they formulate an optimization algorithm to identify a set of critical transmission nodes whose failure will maximize the number of failed substations.

2.2.2 Situation-awareness Tools

Liang et al. [73] built a classifier to distinguish flooded areas from non-flooded areas from satellite images. The authors developed a semi-supervised learning algorithm which divides the satellite image into several patches based on the proximity and intensity of the pixels. A user is asked to label a few patches and based on that the classifier automatically classifies all other patches. They used crowd-sourced knowledge for getting user feedback for their classifier instead of using domain expert to label the image patches. Muralidhar et al. [81] also developed an online monitoring system *ILLIAD* for anomaly detection and state estimation in cyber physical systems (CPS), e.g., wireless and wired networks. They combined model-based and data driven approaches to learn invariant functional relationships between components of the CPS (shown to represent the underlying network structure of the CPS) and checked for the violation of any of these invariant relationships over time which was treated as an anomaly in the CPS.

Huang et al. [59] designed a crowd-sourcing based anomaly prediction system which allows

a user to report urban anomalies they encountered. Based on these historical anomaly data they developed a Bayesian inference model to understand dependency among regions regarding the anomaly distribution. Next, they built a Markov model to learn state transitions between normal and anomaly data and predict the state of the next time slot.

What-if analysis is necessary to predict emergency and differentiate casual events from non-casual events, and many approaches use crowd-sourcing in some way for this along with other techniques. Several CIs tools have been developed to support the decision-making process. Argonne National Lab has developed a risk-based decision support system and a simulation model named Critical Infrastructure Protection and Decision Support System (CIPDSS) [3] to protect CI systems against vulnerabilities, natural or human-made disasters, etc. For studying energy development and impacts of climate Los Alamos National Lab (LANL) designed Climate-Energy Assessment for Resiliency (*CLEAR*) model which enables to assess the interdependency of CIs regarding their relationship with climate.

There are several tools developed by Oak Ridge National Lab (ORNL) to facilitate decision making for urban infrastructures. *URBAN-CAT* [16] has been developed to understand the impacts of climate change, *URBAN-MET* [17] has been designed to study interactions between urban and environmental systems, *LANDSCAN* [9] has been developed to model the distribution of population and settlement based on demographic and remote sensing imagery data. Besides, several CIs datasets are also available to aid research and decision making: a geospatial and US domestic infrastructure HSIP gold data [8], NHDPlus and USGS hydrology data [1, 2], and EIA energy data [6].

2.2.3 Network Mining in CIs

Several literature have been proposed to quantify importance of nodes in CI network using well-known network-based centrality measures [44, 105, 107]. Tabassum et al. [99] conducted a study on the state-of-the-art research and tools to address computational challenges in CIs using machine learning techniques. Dzangier et al. [43] developed a framework to address the recovery coupling in a multilayered CI collecting recovery time of millions of power grid failures. Lee et al. [69] first proposed how to construct a CIs network from geographic shape-files and discussed how to compute importance of nodes using CI network reachability score. Chen et al. [39] proposed a new path-based cascade model in CI to find critical transmission lines using path-based cascade model. However, their path-based model not efficient to use in large-scale power network. Recently Oliva et al. [85] proposed an approach to measure criticality indices of CI nodes by aggregating different centrality metrics through *Logarithmic Least Squares* method. However, their model finds critical nodes through structural graph analysis in contrast to our dynamic model.

2.3 Time-series Mining

For explaining time-series classification models. Senin et al. [97] proposed how to discover the significant characteristic pattern of time-series using TF-IDF approach in vector space. Karlsson et al. [64] developed an algorithm to find out the minimum number of tweaks in time-series data that can change a classifier decision like random shapelet forest classifier.. Jain et al. [61] proposed a problem to identify repeated sequences of pattern or motifs in time-series segmentation. However, all these explanation models mainly focus on interpretation for non-domain experts on non-sequential problems and are specific to the underlying machine learning models. We find that temporal segmentation is inherently unsupervised

and the intuition behind the segments might not be readily apparent or explainable in certain applications.

Time-series Segmentation There has been an abundance of work on time series segmentation based on monitoring changing temporal patterns, such as modeling co-evolving time series and segmenting them using multi-level HMMs [76] on motion capture data, discovering patterns in data streams [92] using distributed video data, developing online algorithms for frequent sequence mining [80] on different application domain, i.e., robotics, wild life, and health monitoring, and time series segmentation using temporal mixture model and Bayesian information criterion on railway data [95] and Kalman filters on motion capture sequences and chlorine measurement data [72]. GOALIE [88], is another algorithm applied in the context of biological process data that produces segmentations of multivariate time series but its focus is on finding cutpoints where significant shifts of clusters (of time series) occur. Change point detection has also been a popular topic in the climate sciences [89, 96]. Characterizing the dynamics of natural disasters like hurricanes lends itself naturally to a change point detection approach but there has been little work conducted in this regard. Zhao and Chu [109] propose a hierarchical Bayesian framework for detecting shifts in annual hurricane counts while Ruggieri [93] introduces a Bayesian change point detection algorithm to detect changes in temperature using climate data records.

Despite the extensive research conducted in time series segmentation and change point detection, we have found that there exists little prior work in leveraging them to characterize the dynamic effect of natural disasters for emergency management.

2.4 Explainable Models

Recently there has been a push towards quantifying model uncertainty [51] and making machine learning model outputs quantifiable, explainable and simple [91]. Burkart et al. conducted a survey on multiple explainable ML models, where they categorized the explainable models with respect to different domains and type of outputs model generates for explanations [35]. Mohseni et al. distinct between the interpretable and explainable models and states about multiple metric to evaluate the explainable models [77]. Sangdeh et al. presented literature where they designed several quantitative and qualitative experiments to investigate the impact of features and model transparency on model prediction. Poursabzi et al. [87] proposed a measure of trust and explainability. Ehsan et al. proposed an AI rationalization approach to explain the behavior of autonomous agents in in terms of natural language [47]. They evaluated their results on a synthetic game environment and showed how generating rationalizations to describe agent behavior are more satisfying than other explanation models.

Part I

Explainable Resource Allocation for Response and Recovery

Chapter 3

Identifying Disaster Phases

In this Chapter, we describe about our work in identifying disaster phases for decision-making regarding resource allocation in emergency management.

3.1 Introduction

Power outages during several recent hurricanes have had a severe impact on our national security, economy, and public safety. The 2017 hurricane season was the most expensive in U.S. history resulting in huge economic losses (greater than $250 billion). Hurricane Irma caused one of the largest power outages which reportedly knocked out power to 4.5 million of the 4.9 million Florida Power & Light customers. Hence, better understanding of power outages and how they evolve during hurricanes is a very important task for damage prevention and control.

Subject matter experts in critical infrastructure systems (CIS) constantly seek solutions and ideas on how to reduce power outages during hurricanes. For example, Oak Ridge National Laboratory's (ORNL) Energy Awareness and Resiliency Standardized Services (EARSS) project developed a fully automated procedure to take wind speed and location estimates provided by hurricane monitoring experts and provide a geo-spatial estimate on the impact to the electric grid in terms of outage areas and projected duration of outages [30]. Retrospectively identifying 'cut-points' with a sudden change in the number of outages in historical

data can help in many aspects like identifying phases and causes through inter-dependency analysis. This helps disaster management personnel learn from past events and be better prepared for future contingencies. For example, a retrospective analysis of hurricane Sandy highlighted the underlying causes due to inter-dependencies with communication, oil and natural gas infrastructures [24]. Further, pinpointing 'culprit' counties responsible for each such cut-point helps domain experts localize points of failure and analyze restorative periods [50]. Hence such analysis can be used to shorten restoration periods of vulnerable points in the grid thereby improving grid resiliency to future disasters.

The above analysis goals may be addressed using the time-series mining task of 'segmentation'. However computing interpretable 'culprits' for each cut-point is a task which has not been studied before.

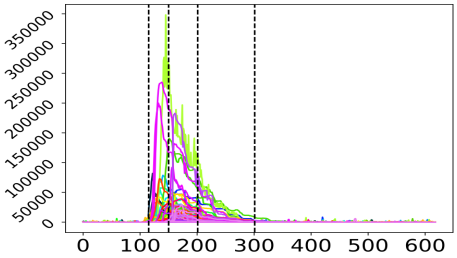
Our Contributions:

- We propose a novel problem and algorithm CnR for computing segments of power outage data. CnR captures temporal and spatial relationships between counties experiencing power outages, modeling the power failure process as a segmentation problem. We also propose a novel explanation algorithm that identifies the *culprit* counties for each segmentation cut-point.

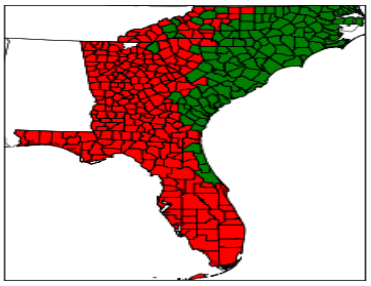
- Our proposed formulation uses low dimensional latent factor models and achieves significant speed up.

- We perform experiments with CnR and other popular segmentation algorithms on synthetic and real datasets including historical hurricane power outage data. The other segmentation procedures perform significantly worse relative to CnR on the real hurricane data, due to their inability to model complex spatial dynamics of the failure process. Although CnR is developed for power outage data, it can be applied to any

multivariate time series.



(a) Temporal Segmentation to identify different failure phases of the power grid across all counties affected by Hurricane Matthew.



(b) Spatial Clustering revealed by CnR of counties in Florida that experienced extensive damage (in green) to power infrastructure due to Hurricane Matthew.

Figure 3.1: **An example of the holistic spatial and temporal analysis results from our novel CnR model to analyze the damage to the power grid during Hurricane Matthew. Fig. 3.1a depicts the overall temporal segmentation over a dataset where each time series indicates the total number of households that lost power in a single county over the course of Hurricane Matthew (household count per county is recorded every hour). Fig. 3.1b represents the spatial clustering of all counties that experienced significant damage to power infrastructure through the course of Hurricane Matthew, essentially representing the spatial span of damage.**

The rest of the Chapter is organized as follows. In Section 3.2 we formally state the segmentation and explanation problems. Section 3.3 introduces our novel spatio-temporal CnR-UV segmentation and explanation model which is designed to incorporate extensive spatial information for segmenting and explaining failure process dynamics in a multivariate time series. Section 3.4 showcases the performance of CnR-UV with respect to other state-of-the-art algorithms, and on real-world problems and finally Section 3.5 discusses some possible limitations on current analysis and explores potential avenues for future work.

3.2 Preliminaries

Motivation: Large power grids usually contain thousands of generators, hundreds of thousands of transmission lines and millions of consumers. Grid components have strong interdependencies like in the transmission grid where multiple paths exist between generators and consumers and these paths typically are arranged in a mesh grid manner. Hence, if one path or line fails, the electricity instantaneously follows an alternate path governed by Kirchhoff's voltage and current laws. If the alternate path however cannot handle the overload in flow, it in-turn fails and this failure cascades to neighboring components. Due to the well studied property of cascading failures and small-world properties in the power grid [56, 58], a few initial points of failure due to a hurricane quickly cause network instability in a region potentially causing millions of people to suffer the effects of brownouts or blackouts. Natural disasters like hurricanes exhibit multiple phases of varied intensity along their path causing failures with different levels of severity at different regions. We model the progression of this grid failure process as a temporal segmentation problem. Modeling this failure process over time, across different regions (e.g. counties) affected by a hurricane, is essential for improving the resilience of critical infrastructure to disasters.

Focus: We characterize the severity of this grid failure process by measuring the number of people in a hurricane affected region (a county in our case) without power over the entire time period of the hurricane. Three critical questions need to be answered for characterization of this process:

- How can we identify different phases of a hurricane as a function of severity of the damage to critical infrastructure like the power grid, using sparse customer power loss data?
- Which counties are most important for characterizing each phase?

- How can counties be grouped together based on their overall failure dynamics during the hurricane?

Our main goal is to help domain experts answer the aforementioned questions.

Notation: We assume we are given a set of time series $X = \{x_1, x_2, ..., x_n\}$, where each time series $x_i = [x_i(t_1), x_i(t_2), ..., x_i(t_m)]$, and $x_i(t_j)$ represents the value at time stamp t_j for the i^{th} time series. We also assume there is a known underlying graph structure G that captures the relationship among these time series $\{x_i\}$. For example, in critical infrastructure systems, the number of electric outages in all disaster affected counties form a set of time series, and the relationship among these counties can be based on their geographical proximity.

Definitions: Our algorithm CnR (Cut-n-Reveal) contains two parts: detecting a good *segmentation* of the outage data to capture the main changes; and finding the corresponding *explanations* (subset of important counties) per segment. With this knowledge of the segmentation and the explanations for each segment, the expert has a holistic picture of the different phases of the failure process as well as the specific time series that contributed significantly to each phase change. We now formally state our definition of a segmentation and an explanation.

Definition 3.1 (Segmentation S)**.** A segmentation of X contains a set of distinct temporal cut-points $S = \{c_1, c_2, ..., c_k\}$, where $c_i \in \{t_1, t_2, ..., t_m\}$.

Definition 3.2 (Explanations E)**.** $E = \{e_1, e_2, ..., e_k\}$, where e_i is an n by 1 non-negative explanation vector. $||e_i||_1 = 1$ and e_{ij} represents the importance of time series j for explaining the cut-point c_i.

Set-up: The cut-points of S naturally divide the entire time period in the dataset into a set of disjoint time segments. The i^{th} time segment is denoted as a set of contiguous time steps

$s_i = [c_{i-1}, c_i)$ with $i \in \{1, 2, .., k+1\}$, $c_0 = t_1$, and $c_{k+1} = t_m$. Two sets / segments s_i, s_j are said to be neighboring segments if $s_i = \{t_l, .., t_{l+\Delta l}\}$ and s_j contains t_{l-1} or $t_{l+\Delta l+1}$.

Assuming we are given a segmentation S of X, containing a set of cut-points $\{c_i\}$ and corresponding segments, $\{s_i\}$, a desired explanation of the segmentation should be simple yet effective enough to guide efforts to prevent or curtail the effects of critical infrastructure failure in future disasters.

To this end, we introduce an explanation vector e_i for each cut-point c_i in S. Each e_i is an $n \times 1$ vector where n represents the number of counties and e_{ij} represents the importance of the j^{th} time series/county in explaining the cut-point. Intuitively, if time series x_j shows very different patterns before and after the cut-point c_i, we consider it important in explaining why c_i is a good cut-point. On the other hand, if x_j remains constant/unchanged across c_i, it does not provide useful information in terms of the cut-point c_i and should have low values in e_i. In the hurricane outage data where there are hundreds of time series/counties, such explanation vectors are able to highlight the "culprit" time series/counties

3.3 Proposed Problem

In this section we propose our Cut-n-Reveal Spatio-temporal (CnR-UV) model which performs segmentation on a multivariate time series jointly using temporal and information and also yields explanations for each segmentation cut-point.

3.3.1 CnR-UV Segmentation

Modeling the power outage failure process using only temporal changes presents a few drawbacks. Firstly, the segmentation process does not account for or attempt to model spatial

Table 3.1: **Definitions**

$X \in \mathbb{R}^{n \times m}$	The data matrix consisting of n time series each with m time steps.
$D \in \mathbb{R}^{n \times n}$	Depicts a degree Matrix
$A \in \mathbb{R}^{n \times n}$	Depicts an adjacency Matrix
$L = D - A$	Represents the Laplacian Matrix of A.
$U \in \mathbb{R}^{n \times l}$	Spatial feature matrix with l latent features.
$V \in \mathbb{R}^{l \times m}$	Temporal feature matrix with l latent features. in CnR-UV, $l << m$ (Eq. 3.1).
$R \in \mathbb{R}^{m \times m-1}$	A lower triangular matrix with -1's on the primary diagonal and 1's on the second diagonal
$E \in \mathbb{R}^{n \times k}$	Represents the explanation matrix to quantify importance of each time series in explaining each cut-point.
$\mathbf{1} \in \mathbb{R}^{n \times 1}$	Denotes a vector of ones
$\lambda, \alpha, \lambda_i, \gamma_i$	Scalar hyper-parameters used in the segmentation and explanation formulations.

relationships between entities (counties in our case) over which the failure process (power outage) occurs. However, phenomena like cascading failures indicate the existence of strong spatial interactions between components in the power grid and incorporating spatial relationships can aid in more effective modeling of the power outage process. We state our problem as follows:

Problem 3.3.1. Given a set of time series X, the Laplacian matrix L of the underlying spatial network, and a number k, **find** a spatial weight matrix U, a temporal weight matrix V, the k-segmentation of S derived from V that captures the main temporal pattern changes in X.

Overview of our approach

For effective spatio-temporal modeling, we allow temporal latent matrix V to consider the underlying spatial relationships between counties. To this end, we introduce a spatial weight

matrix U, jointly learnt with V. U, V are latent weight matrices, and the latent factor modeling approach is used for our segmentation model because of its success in similar sparse settings like recommendation systems [67, 83].

Formulation

We develop a temporal segmentation formulation influenced by spatial constraints, where the failure process at each time step is represented in a rich low-dimensional latent space l such that $V \in \mathbb{R}^{l \times m}$, $U \in \mathbb{R}^{n \times l}$ and $l << m$, $l << n$. Let $v_i \in \mathbb{R}^{l \times 1}$ and $v_j \in \mathbb{R}^{l \times 1}$ represent the i^{th} and j^{th} column vectors of V respectively where $i, j \in \{1, .., m\}$ and $i \neq j$. We can also consider v_i and v_j to be the latent representation for the i^{th} and j^{th} time steps respectively. The goal of this formulation is that the similarity of v_i, v_j is not solely influenced by the temporal proximity of v_i to v_j but also by the underlying spatial behavior of the counties at time steps i and j. To achieve this goal, we formulate a novel temporal segmentation model in Eq. 3.1 to jointly model spatial and temporal characteristics of power outage during natural disasters.

$$\begin{aligned} \min_{U,V} \frac{1}{2}||X - UV||_F^2 + \lambda_1||U||_1 + \frac{\beta_1}{2}Tr(U^T L U) \\ + \lambda_2||V||_1 + \lambda_3||VR||_{1,2} \\ \text{subject to} \quad U \geq 0, V \geq 0 \end{aligned} \tag{3.1}$$

In Eq. 3.1, the matrix U learns the latent representation for each of the n counties. The first term calculates the re-construction error wherein the original failure behavior observed in X, is re-created as a combination of U (latent county outage characteristics), and V (latent temporal outage characteristics). The l_1 norm terms on U and V ensure sparsity, in line with our goal of designing simple interpretable explanations of our segmentation. The term VR (R is defined in Table 3.1) calculates the difference of each time step with

its previous time step in the latent V space. This term essentially serves as a smoothness constraint penalizing the dissimilarity of neighboring time steps. The $l_{1,2}$ norm term forces whole column similarity between two columns of V, i.e. between neighboring time steps in V as opposed to just element-wise similarity in the case of a simpler l_1 norm on VR. The term $Tr(U^T LU)$ represents the Laplacian regularization constraining the matrix U to be influenced by the underlying geographic layout of the counties in X. Here, the matrix L represents the Laplacian of the county-county adjacency matrix A.

We employ the alternating direction method of multipliers (ADMM) [33] method to solve Eq. 3.1 using the Lagrangian formulation as represented in Eq. 3.2. To separate each term in Eq. 3.1, we assign $J = U$, $K = V$, $P = KR$.

$$\begin{aligned}\mathcal{L}(U,V,J,K,P) &= \frac{1}{2}||X - JK||_F^2 + \lambda_1||U||_1 + \frac{\beta_1}{2}\mathbf{Tr}(J^T LJ) \\ &+ \lambda_2||V||_1 + \lambda_3||P||_{1,2} + \langle G, V - K\rangle + \frac{\gamma_1}{2}||V - K||_F^2 \\ &+ \langle H, U - J\rangle + \frac{\gamma_2}{2}||U - J||_F^2 + \langle F, P - KR\rangle \\ &+ \frac{\gamma_3}{2}||P - KR||_F^2\end{aligned} \tag{3.2}$$

We solve for U, V, J, K, P in an alternating manner. The update steps for each term in Eq. 3.2 are discussed below

1. **Update V:**

   (a) Fixing U, J, K, P we solve for V.

$$\min_V \quad \lambda_2||V||_1 + \langle G, V - K\rangle + \frac{\gamma_1}{2}||V - K||_F^2$$

   This can be restated as follows:

$$\min_V \quad \lambda_2||V||_1 + \frac{\gamma_1}{2}||V - \big(K - \frac{G}{\gamma_1}\big)||_F^2 \tag{3.3}$$

Equation 3.3 has element-level closed form solutions that can be obtained using the soft thresholding operator [27, 74, 102]. The element-level closed form solution is as defined in Eq. 3.4

$$V = sign\left(K - \frac{G}{\gamma_1}\right) max\left(\left|K - \frac{G}{\gamma_1}\right| - \frac{\lambda_2}{\gamma_1}\right) \tag{3.4}$$

(b) Fixing U, V, J, P we solve for K.

$$\begin{aligned}\min_K \frac{1}{2}||X - JK||_F^2 + \langle G, V - K\rangle + \frac{\gamma_1}{2}||V - K||_F^2 \\ + \langle F, P - KR\rangle + \frac{\gamma_3}{2}||P - KR||_F^2\end{aligned} \tag{3.5}$$

Differentiating Eq. 3.5 w.r.t K and setting the derivative to zero yields:

$$\begin{aligned}(X^T J^T JX + \gamma_1 I)K + \gamma_3 KRR^T = \\ X^T JX + G + \gamma_1 V + FR^T + \gamma_3 PR^T\end{aligned} \tag{3.6}$$

If we set,

i. $A = (X^T J^T JX + \gamma_1 I)$

ii. $B = \gamma_3 RR^T$

iii. $C = X^T JX + G + \gamma_1 V + FR^T + \gamma_3 PR^T$

then Eq. 3.6 takes the form of a sylvester equation.

$$AK + KB = C \tag{3.7}$$

The solution to Eq. 3.7 is a well studied problem. [31, 53].

(c) Fixing U, V, K, J we solve for P.

$$\min_P \lambda_3 ||P||_{1,2} + \langle F, P - KR \rangle + \frac{\gamma_3}{2} ||P - KR||_F^2 \tag{3.8}$$

this is equivalent to

$$\min_P \lambda_3 ||P||_{1,2} + \frac{\gamma_3}{2} ||P - \left(KR - \frac{F}{\gamma_3} \right)||_F^2 \tag{3.9}$$

Let $M = KR - \frac{F}{\gamma_3}$ then Eq. 3.9 has the following closed form solution [102]

$$P(:,i) = \begin{cases} \frac{||M(:,i)|| - \frac{\lambda_3}{\gamma_3}}{||M(:,i)||} M(:,i) & if\ ||M(:,i)|| > \frac{\lambda_3}{\gamma_3} \\ 0 & otherwise \end{cases} \tag{3.10}$$

2. **Update U:**

(a) Fixing V, J, K, P we solve for U. We can follow a similar procedure to the V update step above and obtain the following element-wise closed form solution for U.

$$U = sign\left(J - \frac{H}{\gamma_2} \right) max\left(\left| J - \frac{H}{\gamma_2} \right| - \frac{\lambda_1}{\gamma_2} \right) \tag{3.11}$$

(b) Fixing U, V, K, P we solve for J.

$$\begin{aligned} \min_J \frac{1}{2} ||X - JK||_F^2 + \frac{\beta_1}{2} Tr(J^T L J) + \langle H, U - J \rangle \\ + \frac{\gamma_2}{2} ||U - J||_F^2 \end{aligned} \tag{3.12}$$

Differentiating Eq. 3.12 w.r.t J and setting the derivative to 0 similar to the K update procedure, the expression in Eq. 3.13 is obtained.

$$\beta_1 LJ + J(XKK^T X + \gamma_2 I) = XKX^T + H + \gamma_2 U \tag{3.13}$$

We once again solve Eq. 3.13 by reduction to a Sylvester equation,

i. $A = \beta_1 L$

ii. $B = XKK^T X + \gamma_2 I$

iii. $C = XKX^T + H + \gamma_2 U$

3. **Update G:**

$$G = G^{old} + \gamma_1(V - K)$$

4. **Update H:**

$$H = H^{old} + \gamma_2(U - J)$$

5. **Update F:**

$$F = F^{old} + \gamma_3(P - KR)$$

6. **Update $\gamma_1, \gamma_2, \gamma_3$:**

$$\gamma_1 = \rho\gamma_1^{old}; \gamma_2 = \rho\gamma_2^{old}; \gamma_3 = \rho\gamma_3^{old}$$

Solving Eq. 3.2 yields an $l \times m$ temporal weight matrix V and an $n \times l$ spatial weight matrix U. We construct an affinity matrix $W = V^T V$ which is passed to the normalized cuts algorithm to obtain the temporal segmentation S from V. As an additional step, we also

construct a separate spatial affinity matrix $W_U = UU^T$ which represents county similarity, and obtain a spatial clustering of W_U using the normalized cuts procedure. In addition to the temporal segmentation and the explanation of each temporal segment, we believe the spatial clustering of counties provides an additional level of insight about aggregate county behavior during natural disasters in a given region.

3.3.2 CnR-UV Explanation

It need not always be the case that the effects of a power outage in a particular county are felt only in the neighboring counties or that only counties directly affected by a hurricane experience outages. As outlined in [57], the influence graph for county power outage need not necessarily be exactly similar to the grid topology or geographic county layout. Intuitively this means that outage in a county in one part of a state can have far-reaching effects leading to outages or in counties located in a different part of the state in an instantaneous or delayed manner. Accommodating for such effects in our explanation methodology requires a smoother, less stringent spatial constraint. This leads to better explanations for domain experts as well.

Overview of our approach

Our segmentation model proposed in Eq. 3.1 learns a temporal weight matrix V. In addition to the V matrix, the segmentation formulation of the CnR-UV model also learns a rich latent factor representation of each disaster-affected county U. Due to the richness of the latent factor representation and flexibility of design, the U matrix in addition to local spatial effects, is also able to capture far-reaching effects of counties on each other. Hence, the affinity matrix UU^T would capture far-reaching county-county similarities extending significantly beyond

the immediate neighborhood of a county. We utilize this property and construct $W_U = UU^T$ which can be considered the adjacency matrix of a weighted undirected graph of counties whose degree matrix D is a diagonal matrix where each D_{jj} represents the weight of county j and is calculated as the sum of row j of W_U. Matrix L_U is the Laplacian calculated as $L_U = D - W_U$.

Problem 3.3.2. Given a set of time series X, a number k, the k-segmentations of S, and the Laplacian L_U derived from the spatial latent factor matrix U, **find** the associated explanations E, that capture the main pattern changes in X.

Formulation

The explanation formulation employed by CnR-UV is defined in Eq. 3.14. For a smoother Laplacian matrix L_U in the regularization term which allows e_i to consider counties in a larger spatial radius as opposed to in the explanation step of CnR-V wherein a strict spatial constraint based on the county graph is imposed through the Laplacian matrix L.

$$\begin{aligned} &\arg\max_{E} \sum_{i=1}^{k} [{e_i}^T d(S,i) + \alpha e_i^T L_U e_i] - \lambda \sum_{i=1}^{k} ||e_i||_1 \\ &\text{subject to} \quad 0 \leq e_{ij} \leq 1, ||e_i||_1 = 1 \end{aligned} \tag{3.14}$$

The function $d(S,i)$ returns an explanation vector $e_i \in \mathbb{R}^{n \times 1}$. Equation 3.14 can be solved by optimizing each e_i as a separate QP problem convex in e_i. The complete pseudo-code for CnR-UV is given in Algorithm 1.

Remark 3.3. CnR-UV takes worst-case time $O(\#iter l^2(X + m^2 + n^2)))$, i.e, quadratic in the number of time steps and time series. In practice we found QP to be very fast, and total time to be sub-quadratic on the dataset size. The space complexity of CnR-UV is near-linear $O(X + nl + ml)$.

Algorithm 1 CnR-UV Segmentation with Explanation

Input: X: Hurricane Power Outage Data, G: Spatial Graph, l: Num. Latent Features
Result: $S = \{c_1, ..., c_k\}$, Temporal Segmentation
$E = \{e_{c_1}, .., e_{c_k}\}$, Temporal Explanation
Init: U = V = 0
while *not converged* **do**
  Estimate, U, V using Eq. 3.2
  Retrieve S using Normalized Cuts
  Estimate explanation vectors $e_{c_1}, .., e_{c_k}$ using Eq. 3.14
end

3.4 Experiments

We implement CnR in Python and Matlab. Our experiments were conducted on a 4 Xeon E7-4850 CPU with 512 GB of 1066Mhz main memory.

3.4.1 Setup

Datasets: We collect datasets from different domains with the ground truth segmentations to quantitatively evaluate our performance. For efficiency purposes, we perform a standard rolling average as a pre-processing step to all the data. The final statistics are in Table 3.2.

1. *ChickenDance*: *ChickenDance1* and *ChickenDance2* [40] are recorded as motion capture sequences of 4-dimensional data points with ground-truth segmentation [76] and is originally from CMU motion capture database [18]. The ground-truth segmentation is based on different motions in the chicken dance.

2. *WalkJog*: We used two variants of the *WalkJog* datasets, *WalkJog1* where we uniformly sampled 1000 data points from the *WalkJog* dataset used in [52] and *WalkJog2* used in [39]. These datasets adapted from the REALDISP Activity recognition Dataset [29] have recordings of walking and jogging motions with segments between different mo-

tions.

3. *GrandMal Seizures*: [52] has 3-min recordings of neural activity (pre-seizure, seizure and post-seizure) of a subject, recorded using a scalp electrode.

4. *Synthetic Data*: We also generated synthetic data consisting of 4 time-series sampled from normal distributions with different means and standard deviations. Time-series were perturbed at different times to cause segments and the goal is to identify these segments.

5. *NILM*: Non Intrusive Load Monitoring dataset. This dataset consists of real power measurements for various household appliances like lamps, laptops, and refrigerators, recorded through the use of MAU (Measurement and Actuation Units) connected between the device and the wall-socket (more details are in [90]). We use a 24-hr hour snapshot of the NILM data from 2012-01-17 00:00:00 to 2012-01-17 23:59:59 sampled at two minute intervals, and use the time when a device switches states as the ground truth cut-points.

6. *Hurricane Outage data*: ORNL has developed several grid situational awareness products over the last decade such as VERDE, EARSS and EAGLE-I [46] for different stakeholders like DOE and FEMA, primarily for emergency management. For example, the National Outage Map within EAGLE-I collects distribution outage data of all the customers from utility websites every 15 minutes. Due to the recent coverage expansion (with more utilities exposing data from their Outage Management Systems), in this paper, we consider the more recent hurricane outage data namely for Matthew, Harvey and Irma since it covers nearly 90% of the population in the hurricane affected areas.

Baselines We wish to evaluate the *segmentation* and *explanation* parts of our CnR-UV

algorithm. We first start with evaluating the performance of the CnR-UV segmentation procedure and later detail the CnR-UV explanation evaluation.

Segmentation Baselines First, we compare the segmentation of CnR-UV with several state-of-the-art multivariate time series segmentation algorithms.

- CnR-V [82] provides segmentation based on an optimization model to learn latent temporal changes separately and not considering latent spatial changes.

- *Autoplait* [76] is a hidden markov model (HMM) based algorithm which discovers different regimes in co-evolving time series. Each regime can be thought of as the segments for our problem.

- *TICC* [55] is a recent algorithm for multivariate time series to discover repeated patterns. It clusters time stamps into segments using their model.

- *Dynammo* [72] learns a dynamical system (Kalman Filter) and segments the time-series wherever the reconstruction error becomes high.

- *Floss* [52] is an unsupervised semantic segmentation algorithm which learns the segmentation from the local minimas obtained in the matrix profile.

Explanation Baselines To the best of our knowledge, there is no method that retrieves explanations for each segment the way CnR-UV does. Hence, we are unable to compare CnR-UV explanations with those of other state-of-the-art algorithms. We do however evaluate explanation performance for the aforementioned datasets with ground-truth segments. The evaluation procedures for segmentation and explanation are detailed in section 3.4.2.

Table 3.2: **Datasets Used.**

Dataset	#Time stamps	#Time series	Ground Truth
Synthetic	1000	4	✓
NILM	721	17	✓
ChickenDance 1	1590	4	✓
ChickenDance 2	322	4	✓
WalkJog 1	1000	2	✓
WalkJog 2	303	2	✓
GrandMal	1000	2	✓
Harvey	264	250	
Irma	169	271	
Matthew	252	369	

Table 3.3: **Evaluation of segmentation (seg) and explanation (exp) on Ground Truth Datasets based on F1-score.**

Method / Dataset	CnR-UV		CnR-V	Auto Plait	TICC	Dyn.	Floss
	seg	exp					
Synthetic	**1.0**	1.0	0.58	0.5	**1.0**	0.52	0.85
NILM	**0.83**	1.0	0.56	0.4	0.82	0.71	0.73
Chicken1	**0.93**	1.0	0.63	0.85	0.92	0.54	0.53
Chicken2	**0.85**	1.0	0.73	0.73	0.5	0.75	0.71
WalkJog1	0.22	1.0	0.57	0	0.86	0.54	**1.0**
WalkJog2	0.75	1.0	**1.0**	0	0.33	0	0
GrandMal	0.86	1.0	0.58	0.5	**1.0**	0.36	0.5

3.4.2 Quantitative Evaluation

Segmentation: We compare CnR-UV performance with several segmentation baselines on datasets with ground truth segmentations: *NILM, ChickenDance, Synthetic, GrandMal* and *WalkJog*. We evaluate the detected cut-points by calculating the F1 score based on the ground truth cut-points (as in [76]). Higher F1 scores indicate better segmentation. For all our experiments with CnR-UV, we set $l = 2$ (our algorithm was robust to varying latent factor dimensions) and chose the hyperparameters using *gridsearch*. We show the results in Table 3.3 where we observe that CnR-UV outperforms all methods on most datasets except *GrandMal, WalkJog1*.

Explanation: CnR-UV is also able to retrieve reasonable explanations for proposed segmentations in each case. Since there is no existing literature performing explanation in an automatic and principled way, we were unable to compare our explanation algorithm with other baselines. Also, since we did not have any ground truth for explanations, we created a ground truth dataset by manually generating explanations for each cut-point. We did this by identifying a subset k_i of the n time series in a dataset that experienced perturbation across a cut-point c_i. This subset k_i of time series can be considered the ground truth explanation for cut-point c_i. We then compare the top $|k_i|$ (cardinality of set k_i) values in the explanation vector e_i against the ground truth explanations using the F1 score. This is repeated for all cut-points and an average F1 score is calculated for explanations on the dataset. Results of this procedure have been outlined for each dataset in Table. 3.3 (CnR-UV exp). Explanations were evaluated on *ChickenDance*, *WalkJog*, *NILM, Synthetic* and *GrandMal* datasets. We only consider true positive segments identified by CnR-UV while calculating explanation F1 scores. For *WalkJog1* dataset, even though CnR-UV segmentation is low (F1 score = 0.22) our explanation performs well (F1 score = 1.0). This is because we only use the true positive segments and calculate explanations for those segments because of the availability of ground

truth explanation data only for true positive segments.

Discussion of CnR-UV compared to other baselines: For the best performance, CnR-UV models should be provided with the spatial graph relating the time series being modeled. CnR-UV was so designed, bearing in mind the goal of modeling power system failure processes during hurricanes using real-world data. However, in order to holistically evaluate temporal segmentation performance, we compared CnR-UV to many state-of-the-art baselines on several datasets. In this context, all time series were considered spatially independent (as we did not have prior knowledge of spatial inter-dependencies). Due to this lack of spatial information, CnR-UV underperformed as the 'U' matrix was unable to learn the best possible representation. Despite this, our model matches or outperforms strong baselines like TICC, Dynammo, Autoplait and Floss in 5 out of 7 datasets. It must be noted that despite the lack of spatial information, CnR-UV also outperforms the CnR-V model on 5 out of 7 datasets (Table 3.3) indicating that the low-dimensional latent factor U and V matrices in CnR-UV are indeed able to learn rich representations of the failure process.In the datasets where spatial information is missing, we treat all time-series as independent. The reason for the underperformance of CnR-UV in Walkjog and GrandMal datasets w.r.t TICC may have to do with this time series independence assumption being sub-optimal.

Dynammo performs segmentation based on re-construction error w.r.t a tolerance threshold specified by the user. We found that the segmentation was sensitive to this tolerance threshold parameter which directly governs the number of segments allowed. In most cases, Dynammo was found to over-segment or under-segment depending on the error tolerance.

In the case of the *Auto-plait model*, it is found to perform better on datasets with less or non-spiky sudden changes (*ChickenDance* dataset) in time series in contrast to the time series that have a much more spiky and sudden changing behavior (*WalkJog* and *Synthetic*).

Floss performs segmentation in multivariate timeseries by finding local minima on the average CAC curve [52]. Thus, for *GrandMal* and *ChickenDance1* data where multiple groups of time series exhibit large and sometimes spiky changes at different time steps, CAC curve of all time series do not exhibit local minima at the same or close timesteps. This results in the average CAC curve not yielding local minima at all the groundtruth cut points which causes the low F1 scores for Floss in Table 3.3. Also, we must note that the *ChickenDance2* dataset is smoother than *ChickenDance1* (i.e changes are smaller / more gradual than *ChickenDance1*), and we immediately see a significant performance improvement in this scenario in the F1 score of Floss.

3.4.3 Case Studies: Hurricanes

In Section 3.2, we set out to design a model to address the following goals

1. Identify phases of a hurricane as a function of severity of damage to critical infrastructure like the power grid.

2. Identify the most important counties that characterize each phase (i.e "explain" each phase).

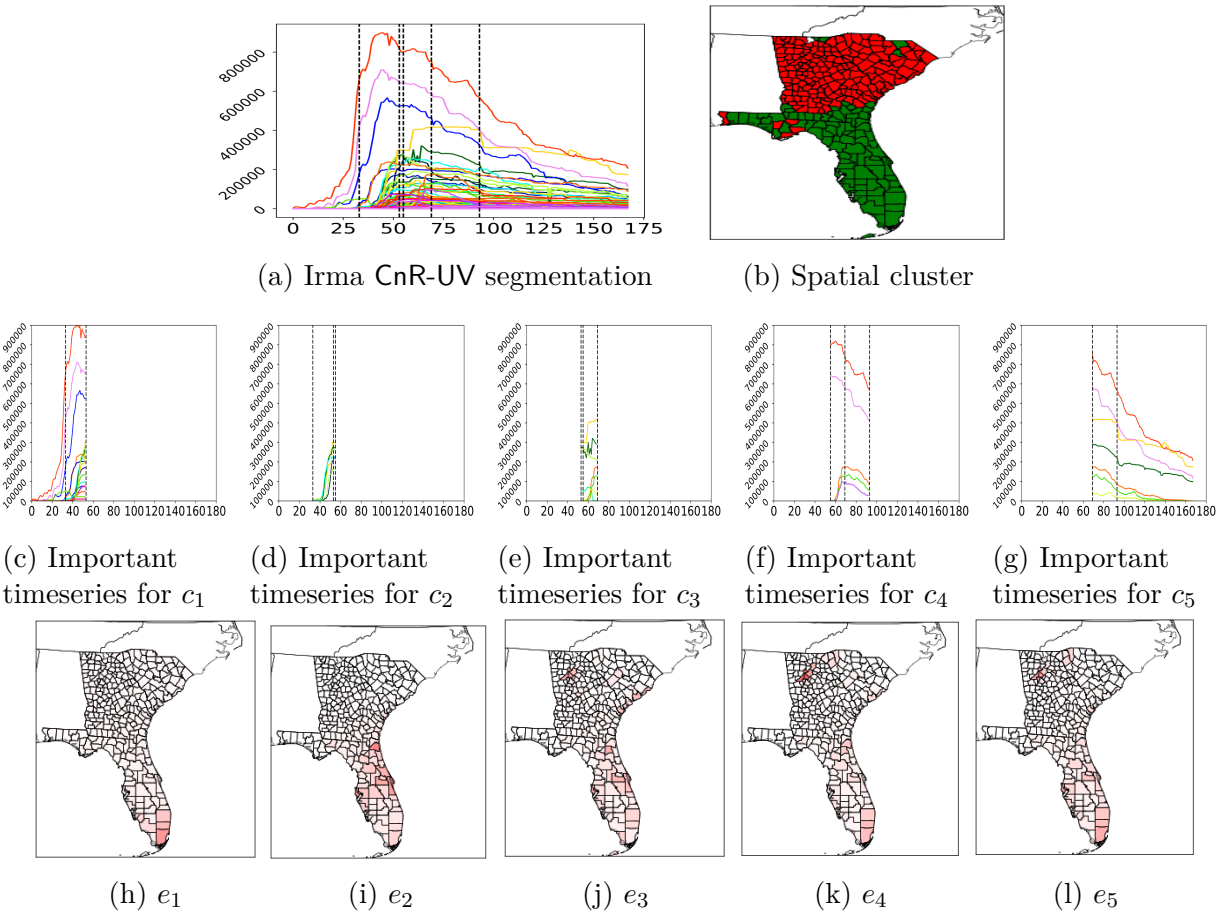
3. Group counties together based on their overall failure dynamics through the hurricane, to allow for overall assessment of spatial span of the damage.

With the aforementioned goals in mind, we ran CnR-UV for power outage failure data from three recent hurricanes. We show that CnR-UV can find meaningful pattern changes and insightful associated explanations. Specifically, the current culprit definition can be used to distinguish between regimes while also applicable to the hurricane failure setting where the failure process of each county follows a typical increasing, peak, decreasing trend pattern.

This is because multivariate hurricane failure time series are highly complex. Although the failure pattern (increasing failure rate, peak, decreasing failure rate) is consistent across counties, the time of failure and rate of failure differs widely across counties requiring a formulation as defined by us in Eq. 3.1 to capture the complexities (and local changes around cut-points) of this spatio-temporal multivariate failure process. The effectiveness of our model in capturing complex patterns in multiple hurricanes will be demonstrated in the following section. For the segmentation model we set the number of latent dimensions (l) to 5 and for each cut-point, we consider counties with explanation weight > 0.1 as important.

Hurricane Irma We show the results in Fig. 3.2. Fig. 3.2a represents the overall segmentation that CnR-UV yields for hurricane Irma, while Fig. 3.2c to Fig. 3.2g show the explanations yielded by CnR-UV across each cut-point in the segmentation. All explanation figures (except those for the first and last cut-point) consist of three cut-points i.e the cut-point being explained, along with the previous and the next cut-point. Each explanation figure is accompanied with a spatial visualization of the important counties highlighted by the explanation of the cut-point. Fig. 3.2h to Fig. 3.2l are spatial depictions of the explanations in Fig. 3.2c to Fig. 3.2g respectively. Finally, Fig. 3.2b represents clustering results of the spatial matrix U. All the following hurricane result visualizations are organized in the same manner.

The first cut-point (showcased in Fig. 3.2c) at time step 35 (around September 10^{th}) shows hurricane landfall when the outages of a few counties seem to rise sharply. Indeed, Fig. 3.2h shows these counties at the southern tip of Florida indicating the location of landfall of hurricane Irma. The second and the third cut-points in Fig. 3.2a might seem redundant owing to their close proximity. However, the second cut-point (showcased in Fig. 3.2d) is capturing a small rising trend of county power outages, for the counties highlighted in Fig. 3.2i. On the other hand, the third cut-point captures fluctuations and plateaus in a different set of counties. The fourth cut-point c_4 (and the corresponding explanation e_4)



(a) Irma CnR-UV segmentation (b) Spatial cluster

(c) Important timeseries for c_1 (d) Important timeseries for c_2 (e) Important timeseries for c_3 (f) Important timeseries for c_4 (g) Important timeseries for c_5

(h) e_1 (i) e_2 (j) e_3 (k) e_4 (l) e_5

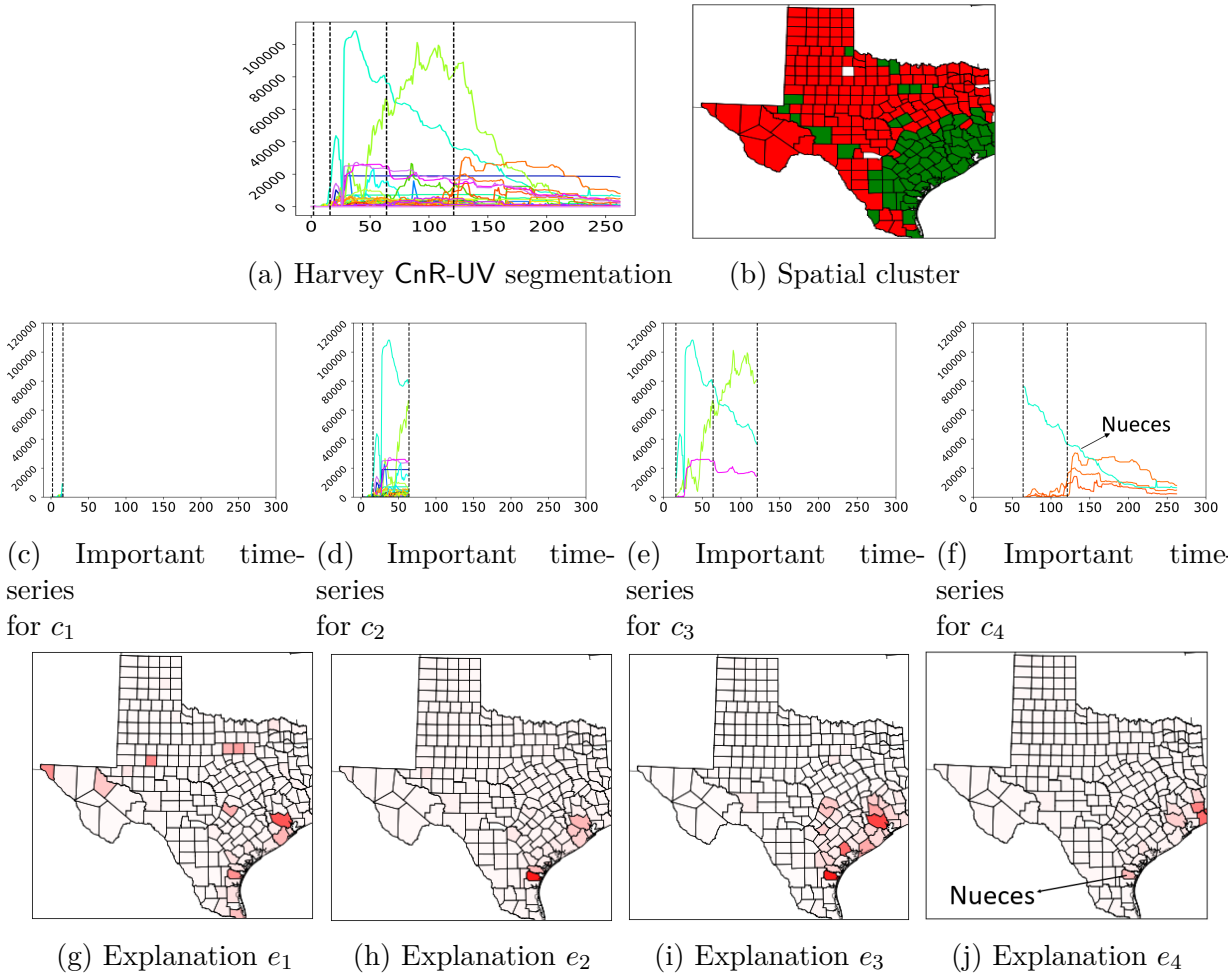
Figure 3.2: **Segmentation and the corresponding explanations for Irma. (a) shows all the counties having grid failures during Hurricane Irma. Each county is represented by a timeseries with an individual color in solid line. The vertical dashed lines are the cutpoints obtained by CnR-UV. (b) Spatial clustering result showing the spatial span of grid failure, based on spatial proximity of counties and similarity in failure patterns of their time series. (h)-(l) e_i visualizations in geographic space for each cut-point. Counties with higher e_i values (higher values represented by darker red) are more important for the cut-point, and are marked with a color closer to red. (c)-(g) The most important time series for each cut-point in the segmentation obtained from e_i whose explanation weight > 0.1.**

(Figs. 3.2f and 3.2k) is interesting: first, it captures a short rising outage trend (of smaller magnitude) at Dekalb, Fulton and Gwinnett counties in the North-West. Reports [21] suggest this is due to a separate tropical storm. At the same time, it also captures the start of the decrease in outages at Miami and Boward counties, both of which rise at the beginning in the first cut-point. Thus, CnR-UV can correctly capture the power restoration period of these counties (Miami, Boward) automatically. The last cut-point c_5 (and corresponding explanation e_5) at time step 93 (around September 12^{th}) captures the date when hurricane Irma was downgraded to a category 2 storm and the outages of the counties started to decrease. Note that these cut-points and explanations are non-trivial, and are successfully modeled since CnR-UV is able to capture the diverse trends in power outages of different magnitudes including in far away counties which do not follow the hurricane trajectory. As mentioned in Section 3.1, retrospective analysis of hurricanes through CnR-UV helps capture failure and restorative phases (ex: Miami, Boward counties) through *segmentation*, that can help experts understand grid resilience and restoration patterns. At the same time, CnR-UV *explanations* in addition to pin-pointing hurricane affected regions that incurred major power outages, can also uncover subtle trends in regions where consequential events occur, ex: like the tropical storm at Dekalb, Fulton and Gwinnett which was caught by CnR-UV explanations. Such insights can alert grid maintainers about the potential for such situations in future.

Spatial Clusters: Fig. 3.2b shows the spatial clustering of all counties affected by hurricane Irma (i.e. the clustering based on W_U where $W_U = UU^T$ is the spatial affinity matrix as explained in Section 3.3.2). It turns out that the green cluster contains counties most affected by power outages, whereas the red cluster shows the counties whose power outages were comparatively lower. This extent of the green cluster (towards the Western/North-Western part) is challenging to estimate by hand or through physical surveys, but has been

uncovered by CnR-UV solely based on time-series dynamics and spatial constraints. This ultimately can aid disaster management experts and power companies to plan recovery for future hurricanes [36, 86].

Hurricane Harvey: The CnR-UV results for hurricane Harvey are depicted in Fig. 3.3.



(a) Harvey CnR-UV segmentation (b) Spatial cluster

(c) Important time-series for c_1 (d) Important time-series for c_2 (e) Important time-series for c_3 (f) Important time-series for c_4

(g) Explanation e_1 (h) Explanation e_2 (i) Explanation e_3 (j) Explanation e_4

Figure 3.3: **Segmentation and the corresponding explanations and spatial clustering for Hurricane Harvey obtained by CnR-UV analogous to Fig 3.2. See detailed discussions in Section 3.4.3.**

The spatial depiction of explanations in Fig. 3.3g - 3.3j broadly trace the trajectory of the hurricane along the eastern coast of Texas, with a few additional non-coastal counties also being highlighted as important in the Northern and North-Western parts of Texas.

In Fig. 3.3a, the first and the second cut-point might seem redundant owing to their close proximity, and their both capturing increasing outage trends. However, upon closer investigation, we find that the first cut-point is detected when there is a sharp spike in El Paso and Howard counties at the very beginning of the hurricane. As no other counties have begun to experience outages at this point, even the relatively low absolute values of power outages (around 1600 homes) are captured by our segmentation model, leading to the first cut-point.

The spatial explanation Fig. 3.3g, depicts as important a few disconnected counties in the Northern and North-Western part of Texas which might seem counter-intuitive at first. However, reports [5, 20] suggest that these counties were hit by floods as a result of hurricane Harvey causing major damage.

Similarly, the explanation of the second cut-point (Fig. 3.3d) highlights the spike in outages at Nueces and Aransas (around $100,000$ homes), but also captures Fort Bend, Brazoria and Harris (Fig. 3.3h group of three counties highlighted in the Northern part of the East coast of Texas) as important. Although their outages are low (around 2000) compared to Nueces, they have a very sharp peak at this cut-point. Report [19] states that the sudden rise of this peak is due to an EF1 level tornado on August 26, which caused major damage at Fort Bend also potentially affecting the surrounding counties.

The explanation for the third cut-point (Fig. 3.3e) captures two different patterns, the high outage spike of Harris county (green line) and the declining trend at Victoria and Nueces counties. Finally, for the last cut-point (Fig. 3.3f) while the outages in many counties are decreasing, our algorithm correctly highlights a sudden rise of outages in counties Orange, Jefferson, Hardin. The main reason for this increase is due to the rising water of the Neches river, which causes the city to lose service from its major pump stations. As in case of Irma, spatial clustering results for hurricane Harvey (Fig. 3.3b) also help us glean the overall

picture of the spread of damage due to the hurricane. This explanation is beneficial for doing an inter-dependency study. Note that CnR-UV captures long range county dependencies even if the counties are not geographically close to each other; such information of subtle county relationships is often buried deep in the original set of hundreds of time series and cannot be uncovered through simple models or through rudimentary visual or statistical analysis of the original data.

3.5 Discussion

In this Chapter, we have developed a novel effective and scalable combined framework CnR for providing segmentations and simple interpretable explanations for multivariate time-series like outage data. We evaluated the performance of our methodology against state of the art segmentation and time-series clustering procedures on open ground truth datasets. We have also conducted retrospective analysis on the failure of the power grids during three hurricane events. CnR-UV is very fast and provides segmentation within 3mins for each hurricane data.

There are several avenues our model can be adapted and beneficial for analyzing a future disaster. For analyzing power failures, our segmentation model CnR-UV assumes the underlying spatial network L is based on the adjacency matrix of all the counties in the hurricane-affected state. For analyzing an ongoing or future hurricane, we can represent L in terms of a projected hurricane trajectory. For the adjacency matrix, we can define an edge as the neighboring counties only if both fall in the hurricane trajectory. Besides, for our analysis, we consider the presence of power failures sequences for the entire hurricane period. We can run the power failure sequence data based on partial hurricane time for real-time hurricane events. As future work, methodologically, we can explore real-time learning of the

segmentations and explanations.

Chapter 4

Pinpointing Regions for Disaster Phases

In the previous chapter, the explanation model of CnR-UV can only capture important counties to explain changes (cutpoint) for the CnR-UV segmentation. However, we may want such actionable guidance by explaining changes for any given segmentation. Next, we briefly discuss our work on pinpointing vulnerabilities to explain changes when the segmentation is black-box.

4.1 Introduction

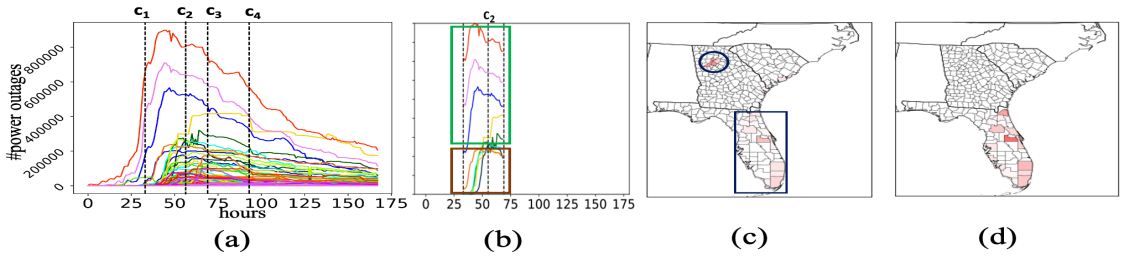


Figure 4.1: **Disaster example (a): 2017 Hurricane Irma times-series (representing power failures of a county during the hurricane). The first and last cut point are based on the hurricane landfall (Sept. 10) and end time (Sept. 12). Cupoints c_2 and c_3 is around the time when hurricane is changing trajectory. (b) shows a snippet and (c) a heatmap of top 10 most important counties found by our algorithm across cutpoint c_2. (d) shows a simple magnitude-based *TOIs* heatmap across c_2. Brighter colors indicate larger importance weights ((c),(d)).**

Multivariate time-series data, where each timestamp has readings from the observations of multiple entities or sensors, are prevalent in various urban scenarios, ranging from critical infrastructures, public health, and so on.

Urban subject matter experts (SMEs) such as emergency management and health care authorities segment such data for identifying meaningful events. They want to understand which signals are more susceptible across each particular event. For example, in a hurricane disaster scenario, a set of multivariate time-series can represent power failures across a set of counties. Further a subject matter expert can identify the events which correspond to different phases of the hurricanes (e.g., based on severity of damage), by using multiple segmentation algorithms [40, 55, 76]. However, most time-series segmentation algorithms [52, 55, 76] are usually complex, and while they give high quality segmentations, they do not give ready actionable insights to identify the time-series of interests (*TOIs*) across events.

Finding time-series of interests (*TOIs*): We next discuss how finding *TOIs* gives actionable insights which are useful for domain experts in context of urban analytics. For the hurricane power failure data (mentioned above) domain experts (SMEs) such as emergency management authorities seek solutions to reduce power outages in different counties. If SMEs know which time-series/counties are the most important w.r.t. an event, they can send personnel to fix damage and alert local authorities to reduce further loss [30]. For example, consider Fig. 4.1 for the 2017 Hurricane Irma (where Fig. 4.1(a) shows the hurricane time-series data with the segmentation, which roughly correspond to different phases). Fig. 4.1(b) shows the failure time-series snippet of top 10 counties (i.e. time-series) across the second cutpoint (c_2) found by our algorithm. We see some of the counties have a very high change of power failures (green box) which are obviously useful to guide resource allocation as they denote widespread problems. At the same time there are also some less obvious

counties near Atlanta too (brown box in Fig 4.1(b)). These counties have relatively lower but a sudden change of failures compared to the other counties. Further see the geographic heatmap plot of our *TOIs* (based on their importance weights as learnt by our algorithm) in Fig. 4.1(c). Clearly these counties (denoted by blue circle) are not obvious: they are located at the north-west corner, far away from where the hurricane is present (the blue rectangle, where the rest of the important *TOIs* are). However they are still important and useful for situational awareness and resource allocation. Indeed, reports [22] suggest that this is due to a separate tropical storm happening at this time. Note that figuring out these counties by just visualizing the time-series (Fig. 4.1(a)) across c_2 is hard (as they get buried). In fact, further if we plot a heatmap of all the counties based on their rate of change of magnitude across c_2 (Fig. 4.1(d)) we *cannot* recover these counties located at the middle (as their magnitudes are relatively lower).

How can we then find such *TOIs* for events? One plausible way to solve such problems might be to somehow interpret the internal change of state of the segmentation algorithm across any cut-point. For instance, one might use a recent segmentation algorithm Autoplait [76] to segment, which learns a HMM internally. Hence one way to get the *TOIs* will be to see which time-series cause the internal HMM to change across the cut-point. However it is easy to see these will only point to time-series which help explain *model behavior*, but not necessarily give actionable insights to the DE (as explained above such actionable insights are based more on the change in the time-series itself, and how similar/dissimilar these changes are to others). Additionally, DEs may end up using multiple segmentation algorithms (such as TICC [55] which uses multilayer MRF in contrast to Autoplait) for different datasets based on what constitute meaningful events for the data. Tracking the model behavior for each of these segmentation algorithms will give different *TOIs*. Further this may become too complex to the DE, who will need additional technical help to understand the technical

intricacies of the segmentation algorithm to get any actionable insights.

Our Contributions: Hence we need to find a different way to get actionable insights for any segmentation algorithm. Our contributions are:

- We introduce and formalize a novel problem Rationalization for Time-series segmentations (RaTSS) to provide actionable insights for the DE.
- We propose an algorithm Find-RaTSS to solve RaTSS which can rationalize any black-box segmentation in terms of constituent time-series efficiently.
- Finally, we evaluate performance of Find-RaTSS with baselines on both synthetic and real general and urban data. Also, we showcase how rationalizations (*TOIs*) by Find-RaTSS are actionable directive for the DEs in urban domains like disasters and disease spread.

The rest of the chapter is organized as follows. We first describe some terminologies in Sec. 4.2. We propose and formalize our novel problem RaTSS. Next, we design an algorithm to solve RaTSS and showcase performances of our algorithm in the synthetic and real-life urban domain. Additionally, in the experiments (Sec. 4.5) we show how our *TOIs* can help a variety of SMEs, including public health for understanding impact of interventions in the Covid-19 pandemic. We then explore the related literature on the closest works in urban analytics and time-series mining and finally conclude with the avenues for future work.

4.2 Preliminaries

In this Section, we give a brief description of the terminologies we use for our problem and model.

1. **Time-series of interests (*TOIs*)**: The signals which are more susceptible across

each particular event or cutpoint.

2. **Actionable Insights**: In urban analytics context, actionable insights are the time-series of interests which can guide SMEs for decision-making regarding resource allocation, situation awareness, etc.

3. **Rationalization**: Time-series of interests which can provide actionable insights to the SMEs for the associated events in a given segmentation. Rationalization and actionable insights are two inter-changeable terms.

4.3 Proposed Problem

Notations. Suppose, we have a multivariate time-series data matrix of m sequences $\mathbf{X} = \{x_1, x_2, ...x_m\}$, where each $x_u = x_u(t_1), ...x_u(t_t)$ has t observations. We are given a set of cutpoints $\mathcal{C} = \{c_1, c_2, \ldots, c_k\}$, each $1 \leq c_j \leq t$. For the rest of the paper we use the term rationalizations and *TOIs* as interchangeable. An overview of all the notations are in Table 4.1. The main challenges for formulating such rationalization problem across a segmentation are: **P1.** We require a general framework which works for any $\mathbf{X}$ and any segmentation $\mathcal{C}$ given by a black-box segmentation algorithm B. For generalization, we do not know anything about B. **P2.** We need to associate constituent time-series across each cutpoint to actionable directives. Hence we propose an intermediary weighting scheme to measure importance of each time-series which can map them towards actionable directives.

Problem 4.3.1 (**Informal RaTSS**). **Given** a multivariate time-series $\mathbf{X}$ and a segmentation $\mathcal{C}$ for $\mathbf{X}$. **Find** the rationalization weight vector of size $m \times 1$, $\mathbf{r}_j$ across each cutpoint c_j in $\mathcal{C}$, where each value r_j^u in $\mathbf{r}_j$ is a scalar and represents the importance weight for time-series x_u across c_j.

Table 4.1: **Notations.**

Symbol	Description
$\mathbf{X} \in \mathbb{R}^{m \times T}$	Data matrix consisting of m time-series each having t timestamps
$\mathcal{C}$	Segmentation with a set of cutpoints for $\mathbf{X}$
B	Any segmentation algorithm which outputs a set of cutpoints on $\mathbf{X}$.
$\mathbf{r}_j$	$m \times 1$ rationalization weight vector
$G(\mathcal{S}, \mathcal{E})$	A segment graph of $\mathcal{S}$ nodes (s_{ij}) and $\mathcal{E}$ edges (e_{ijk}) connected by weight $\mathbf{w}_{ijk}$
s_{ij}	A node (segment) of G consisting of the sub-sequence of m time-series within $i - j$ timestamps
e_{ijk}	An edge of G representing edge weight of the connected nodes s_{ij} & s_{jk}
$\mathbf{w}_{ijk}$	A $m \times 1$ weight vector for e_{ijk}, each cell is weight of x_i in e_{ijk}
K	Total number of paths in G
$F(s_{ij})$	A function to return a $m \times f$ matrix for each time-series in s_{ij} having f features
P_B	The path in G which maps to $\mathcal{C}$
P_{rest}	All possible paths in G except the path π_B
$\boldsymbol{\pi}_B$, $\boldsymbol{\pi}_{\text{rest}}$	Cost of path P_B, P_{rest}
$\boldsymbol{\alpha}$	Global latent weight vector
λ_1, λ_2	Scalar hyper-parameters used for rationalization Eq. 4.2

What is r_j^u? One possible approach to measure importance is to assign a numerical weight r_j^u to the time-series x_u in terms of its change across a cutpoint c_j. However, we do not know anything about the segmentation algorithm B (**P1**), besides the way we should measure change (e.g., using time-series features) may be different for each cut-point. Hence feature-engineering to work for all B correctly is not possible.

Hence we present another idea: since $\mathcal{C}$ is selected by B among all possible segmentations, we may assume $\mathcal{C}$ by B is the best in some sense (as after all the algorithm B output $\mathcal{C}$ as the segmentation, choosing it over *all the other possible segmentations*). It is natural to assume that the rationalization weight vectors $\mathbf{r}_j$ should be set in a way that explains *why* $\mathcal{C}$ becomes better compared to all other possible segmentations. Hence our main idea is to consider $\mathcal{C}$ as the best compared to other possible segmentations, and choose $\mathbf{r}_j$s to make it so.

Segment Graph. We want to efficiently represent all possible segmentations (as they will be exponential in number to the length of the sequence). Recently [40] proposed a 'segment graph' data structure G to represent $\mathbf{X}$ and its segmentations efficiently. Hence we leverage this data structure and convert our rationalization problem in terms of a graph optimization over G. The segment graph $G(\mathcal{S}, \mathcal{E})$ is a *Directed Acyclic Weighted Graph* (DAWG) consisting of a set of nodes $\mathcal{S}$ and set $\mathcal{E}$ of edges. Next, we define its components.

Definition 4.1 (**Node set of** G)**.** The node set $\mathcal{S}$ consists of all possible segments in X. That is, node s_{ij} in node set $\mathcal{S}$ of G consists of the subsequence of $\mathbf{X}$ of consecutive timestamps i to j. Additionally $\mathcal{S}$ consists of s_o and t_a, two dummy nodes to represent the start and end of the time-series.

Definition 4.2 (**Edge set of** G)**.** An edge e_{ijk} in edge set $\mathcal{E}$ connects two adjacent segments s_{ij} and s_{jk}, where $i < j < k$ and $1 \leq i < T-1$, $1 < k \leq T$. Each e_{ijk} is mapped to a possible cutpoint c_j a timestamp t_j in $\mathbf{X}$.

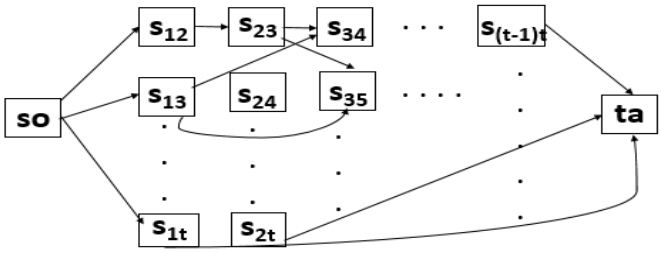


Figure 4.2: **A segment graph.**

The edge weight vector $\mathbf{w}_{ijk}$ for the corresponding e_{ijk} represents the 'change' of every x_u between the adjacent segments of any cutpoint c_j. Note that all segmentations of $\mathbf{X}$ (including the segmentation $\mathcal{C}$) naturally get mapped to paths in G. We further define K (the total number of segmentations) as the total number of possible paths from s_o to t_a paths in G.

Definition 4.3 (**Segmentation $\mathcal{C}$ in G**)**.** The path P_B in G is the path from s_o to t_a s.t. each edge $e_{ijk} \in P_B$ is mapped to the corresponding $c_j \in \mathcal{C}$.

Definition 4.4 (**Other possible segmentations**)**.** P_{rest} is the set of all other possible paths in G from s_o to t_a other than P_B. Every path $p_v \in P_{\text{rest}}$ represents a possible segmentation of X.

Thus, in terms of G our idea can be stated as that P_B should be the best path from s_o to t_a in G (compared to all paths in P_{rest}). Further, $\mathbf{r}_j$ should be set in a way that helps make P_B the best path. Clearly the weight vectors $\mathbf{w}_{ijk}$ (which represent how different adjoining segments are across each edge) also should play a role in quantifying the quality of each path.

Hence, we need a 'quality' metric *Quality* to compare the paths which should depend on the rationalization weights, the paths and the weights over the edges in the paths. Then our required condition can be stated as: *Quality*(P_B, $\mathbf{w}_{ijk}$ for each $e_{ijk} \in P_B, \mathbf{r}$) > *Quality*($p_v$,$\mathbf{w}_{ijk}$

for each $e_{ijk} \in p_v, \mathbf{r}), \forall p_v \in P_{\text{rest}}$. It is not clear how $\mathbf{r}$ which has been defined only over P_B can affect the quality of other paths. One way to handle this issue is to create 'rationalizations' over all edges - which will lead to severe over-parameterization as the number of edges in G is $O(T^3)$. Hence, we propose to instead have a *global latent* weight vector $\boldsymbol{\alpha} \in \mathbb{R}^{m \times 1}$ whose magnitude captures the global latent importance of each time-series, and then use $\boldsymbol{\alpha}$ over the edges in P_B to get $\mathbf{r}_j$ for each cutpoint $c_j \in \mathcal{C}$. Our condition can be re-written as follows:

$$Quality(P_B, \alpha) > Quality(p_v, \alpha), \forall p_v \in P_{\text{rest}} \tag{4.1}$$

with $\mathbf{r}_j = \text{someFunctionOf}(\alpha, \mathbf{w}_{ijk})$.

We now informally state our problem in terms of a graph-based framework.

Problem 4.3.2 (**Informal Graph based RaTSS.**)**. Given, X**, $\mathcal{C}$, and a segment graph G on **X**, so that its path P_B corresponds to $\mathcal{C}$. **Find** $\boldsymbol{\alpha}$ to satisfy Eq. 4.1 and generate the rationalization weight vector $\mathbf{r}_j$ across each edge e_{ijk} in P_B.

Formalizing Problem 4.3.2. To formalize this problem, several questions arise: **Q1.** *What is Quality?* Due to **P1**, we cannot consider any prior knowledge about the segmentation algorithm while designing the quality metric. A possible approach is to assume *Quality* as *cost* of the path. Hence, we simply consider the *weighted length* of the path as its cost. Note that, the length of the path will intuitively measure how different its adjacent segments are. Intuitively, segmentation algorithms try to find cutpoints so that each segment corresponds to a distinct phase in some sense, and hence adjacent segments are expected to be very 'different' from one another. In other words, in our framework, we want to learn $\boldsymbol{\alpha}$ which makes P_B the *longest* weighted path in G from s_o to t_a.

Definition 4.5 (**Length of a path**)**.** The length $\boldsymbol{\pi}_v$ of a path p_v is an $m \times 1$ vector, each

component represents the sum of all the edge weights $\mathbf{w}_{ijk}$ in p_v for time-series x_u. Hence, $\boldsymbol{\pi}_v = \sum_{e_{ijk} \in p_v} \mathbf{w}_{ijk}$.

Hence we can now write the *Quality* function for any path p_v (i.e. the cost of p_v) as follows:

$$Quality(p_v, \boldsymbol{\alpha}) = \sum_{e_{ijk} \in p_v} \boldsymbol{\alpha}^T \mathbf{w}_{ijk} = \sum_{e_{ijk} \in p_v} \sum_{u=1}^{m} \boldsymbol{\alpha}_u \mathbf{w}_{ijk}^u$$

Q2. *How to generate $\boldsymbol{r}_j$ for each c_j using $\boldsymbol{\alpha}$?* As we discussed before (in Eq. 4.1), $\mathbf{r}_j$ = someFunctionOf(α,$\mathbf{w}_{ijk}$). $\mathbf{w}_{ijk}$ represents the change across an edge, while $\boldsymbol{\alpha}$ gives us the global latent weight for each time-series, Additionally, recall that $\mathbf{r}_j$ is intuitively the importance of each time-series over every cut-point $c_j \in \mathcal{C}$. Hence a simple way to get $\mathbf{r}_j$ is $\mathbf{r}_j = |\alpha \odot \mathbf{w}_{ijk}|$, where $\odot$ is the standard element-wise vector dot product (and the modulus is because $\boldsymbol{\alpha}$ can be negative). **Q3.** *How to set $\boldsymbol{w}_{ijk}$?* Note that we still haven't precisely defined $\mathbf{w}_{ijk}$. As discussed before $\mathbf{w}_{ijk}$ just should reflect some change in each time-series across the edge. Keeping in mind **P2**, we just represent each node s_{ij} as a $m \times f$ feature matrix where $F(s_{ij}) = [\mathbf{f}_{ij}^1, ...\mathbf{f}_{ij}^m]$, each $\mathbf{f}^u$ is a $f \times 1$ feature vector of time-series x_u in segment s_{ij}. We can set $\mathbf{w}_{ijk} = ||F(s_{ij}) - F(s_{jk})||_{1,2}$.

This is not feature-engineering. We choose only basic statistical features $\mathbf{f}$ (mean, variance, minimum and maximum in this paper), as we only need to capture basic changes across the segments - not how the segmentation algorithm B regards as the change (a feature ablation test in the experiment shows all these features are necessary and useful for any segmentation). This is different than directly choosing features to set $\mathbf{r}_j$, as now the rationalization weights will need to best explain *why* the cut-point is present, not *how* the time-series change.

Putting everything together. We next formalize our task as an optimization problem. Note that Eq. 4.1 will result in one inequality for each path in G, which are exponential in

number (to T). Hence directly using Eq. 4.1 is infeasible. Instead, we tackle a simplified version, by just adding all the inequalities to get a consolidated objective. Let $\boldsymbol{\pi}_B$ be the length of path P_B. We also define $\boldsymbol{\pi}_{\text{rest}}$ as follows.

Definition 4.6 (**Total length of** P_{rest})**.** $\boldsymbol{\pi}_{\text{rest}}$ is an $m \times 1$ vector, each component represents the length of all other paths in P_{rest} other than P_B over each time-series. Hence, $\boldsymbol{\pi}_{\text{rest}} = \sum_{p_v \in P_{\text{rest}}} \boldsymbol{\pi}_v = \sum_{p_v \in P_{\text{rest}}} \sum_{\substack{e_{ijk} \in p_v \\ e_{ijk} \notin P_B}} \mathbf{w}_{ijk}$.

We can also rewrite $\boldsymbol{\pi}_{\text{rest}}$ as $\sum_{e_{ijk} \in \mathcal{E} - P_B} p_{ijk} \mathbf{w}_{ijk}$ where p_{ijk} is the number of paths in G passing through e_{ijk}. Suppose, $\boldsymbol{\Delta\pi} = \sum_{p_v \in P_{\text{rest}}} \boldsymbol{\pi}_B - \boldsymbol{\pi}_v = (K-1)\boldsymbol{\pi}_B - \boldsymbol{\pi}_{\text{rest}}$, where K is the total number of paths in G. Hence the set of inequalities in Eq. 4.1 will imply maximizing $\boldsymbol{\alpha}^T \boldsymbol{\Delta\pi}$. Therefore, we formalize RaTSS as follows.

Problem 4.3.3 (**Formal Graph based RaTSS.**)**. Given,** $\mathbf{X}$, $\mathcal{C}$, a segment graph G on $\mathbf{X}$, so that its path P_B corresponds to $\mathcal{C}$, and a Function $F(.)$ to represent a node s_{ij} on G in $m \times f$ feature matrix. **Find** $\boldsymbol{\alpha}$ that satisfy $\mathit{Quality}(P_B, \alpha) > \mathit{Quality}(p_v, \alpha), \forall p_v \in P_{\text{rest}}$ and generate rationalization $\mathbf{r}_j$, for each edge e_{ijk} in P_B such that

$$\begin{aligned} &\arg\max_{\boldsymbol{\alpha}} \boldsymbol{\alpha}^T (\boldsymbol{\Delta\pi}) - \lambda_1 ||\boldsymbol{\alpha}||_1 \\ &\text{subject to } \boldsymbol{\alpha} \neq 0, ||\boldsymbol{\alpha}||_2^2 = 1 \end{aligned} \tag{4.2}$$

$$\mathbf{r}_j = |\alpha \odot \mathbf{w}_{ijk}| \tag{4.3}$$

Details: We want $\boldsymbol{\alpha}$ to assign a higher weight for the time-series with high value in $\boldsymbol{\Delta\pi}$ (each $\boldsymbol{\alpha}_u$ represents a weight for time-series x_u). The term $\lambda_1 ||\boldsymbol{\alpha}||_1$ is to encourage sparsity for simple explanations. The constraint $||\boldsymbol{\alpha}||_2^2 = 1$ is to ensure that $\boldsymbol{\alpha}_u$s are bounded and comparable. An overview of RaTSS is shown in Fig. 4.3

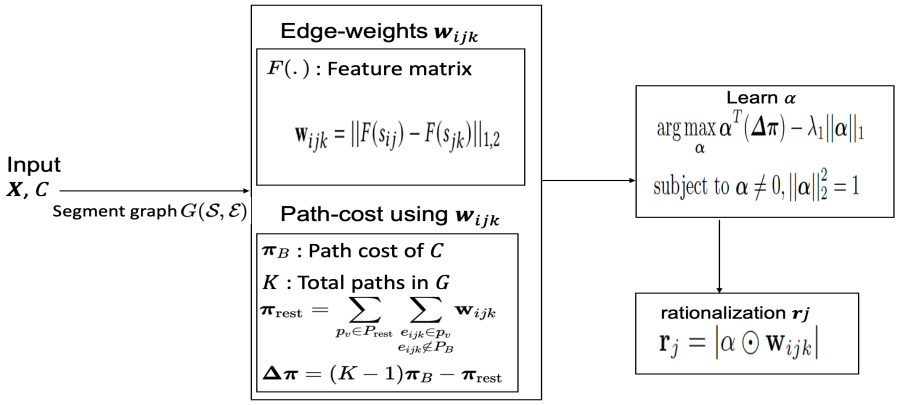


Figure 4.3: **Overview of RaTSS**

4.4 Approach

In this section we present an approach to solve RaTSS. We aim to design an efficient approach for Problem 4.3.3 to identify the *TOIs* for RaTSS. Our main steps are- (A) Computing $\boldsymbol{\Delta\pi}$ and (B) Optimizing $\mathbf{r}_j$. Next, we discuss in detail about these steps, present an overall algorithm Find-RaTSS and its complexities and implementation.

(A) Computing $\boldsymbol{\Delta\pi}$. We construct the segment graph $G(\mathcal{S}, \mathcal{E})$ to compute $\boldsymbol{\Delta\pi}$ from G. The computation is of three steps: (i) Computing the length of P_B, i.e. $\boldsymbol{\pi}_B$, which is trivial, (ii) Calculating the total length of all the possible paths, except $\boldsymbol{\pi}_B$ in G, i.e., $\boldsymbol{\pi}_{\text{rest}}$, and (iii) Calculating the total number of all possible paths in G, i.e., K. Our main challenge is to efficiently compute (ii), and (iii) since the number of paths in G is exponential. Next, we discuss these steps (ii) and (iii).

For computing $\boldsymbol{\pi}_{\text{rest}}$, we design an efficient technique to calculate the number of paths p_{ijk} passing through an edge e_{ijk} in constant time, by exploiting the properties of G. Next, we show how we find p_{ijk} with the help of Lemma 4.7.

Lemma 4.7 (Number of paths through an edge e_{ijk} is p_{ijk})**.** *Given, i and k be the begin and end timestamp of the adjacent segments s_{ij} and s_{jk} of e_{ijk}. For total timestamp T, the*

number of paths through e_{ijk} is,

$$p_{ijk} = \begin{cases} 1 & i = 1, k = T \\ 2^{(i-2)} & k = T \\ 2^{(T-k-1)} & i = 1 \\ 2^{(i-2)+(T-k-1)} & otherwise \end{cases} \tag{4.4}$$

Proof. Following is the proof for different i, j, and k.

- **Case** $i = 1, k = T$: Trivial. Since, there is only on segment from 1 to T and only one path.

- **Case** $i = 1, k < T$: (By induction). Suppose, $T = k + 1$, then the number of paths is $2^{k+1-k-1} = 2^0 = 1$. This is trivial, because from s_{jk} there is only one possible edge $s_{jk} - s_{k(k+1)}$ and thus there is only one possible path.

  Suppose, with $T = k + q$ where $q > k$, the number of paths is 2^{q-1}. We need to show, with $T = k + q + 1$, the number of paths become 2^q. Since $q + 1 > q$, all the paths that can be possible to reach from s_{jk} to $s_{j(k+q)}$ can also be possible to reach from s_{jk} to $s_{j(k+q+1)}$ using the edge $s_{j(k+q)} - s_{j(k+q+1)}$. That is, 2^{q-1} paths possible from s_{jk} to $s_{j(k+q+1)}$ using the edge $s_{j(k+q)} - s_{j(k+q+1)}$.

  Again, by the edge construction property of G, a path ends when end timestamp of a node is $T = k + q + 1$. If we remove the timestamp $k + q$ and consider only $k, k + 1, \ldots, k + q - 1, k + q + 1$, then the number of possible paths become 2^{q-1} to reach $k + q + 1$ (by induction). Because, all edges that used nodes ended with $k + q$ now can use nodes ended with $k + q + 1$.

  Thus, the total number of possible paths with and without using the edge $s_{j(k+q)} -$

$s_{j(k+q+1)}$ becomes $2^{q-1} + 2^{q-1} = 2^{q-1+1} = 2^q$ (proved).

- **Case** $i > 1, k = T$**:** Following the above if we consider $T = i$ and the $k = 1$. The number of paths is $2^{T-k-1} = 2^{i-1-1} = 2^{i-2}$.

- **Case** *otherwise***:** Hence, if there are 2^{i-2} paths possible to reach a node s_{ij} and 2^{T-k-1} possible paths to reach timestamp T from a node s_{jk}. The total number of paths possible through the edge $s_{ij} - s_{jk}$ or e_{ijk} is $p_{ijk} = 2^{(i-2)+(T-k-1)}$.

□

Next, using Lemma 4.7, we can further compute the following corollary on the total number of paths K, intuitively, by setting $i = 1$, and $k = 2 \dots T - 1$.

Corollary 4.8 (**Total number of possible paths in segment graph** G **is** K)**.** *Given total timestamp is* T*, and number of paths through an edge* e_{ijk} *is* p_{ijk}*. If we consider all the edges in edge-set* G *as* $\mathcal{E}$*, the total number of possible paths in* G *is-* $K = 1 + \sum_{\substack{k=2 \\ i=1}}^{T-1} p_{ijk} = 2^{T-2}$

Proof. According to the path property of G, every path starts from $i = 1$ and ends at timestamp T. For $i = 1, k = T$ there is only one path possible (the whole segmentation). For $1 < j < T-1$, the first node of all other paths in G has to start from some s_{1j}. Consider $j = k$, using Lemma 4.7 we find p_{1kk}. Thus, $K = 1 + \sum_{k=2}^{T-1} p_{1kk} = 1 + 2^{T-2} - 1 = 2^{T-2}$. □

(B) Optimizing $\mathbf{r}_j$**.** To add the constraint to the objective of Eq. 4.2, we use Lagrange multiplier λ_2. We solve Eq. 4.2 using gradient descent learning, considering the gradient $-\boldsymbol{\Delta\pi} + \lambda_1 sign(\boldsymbol{\alpha}) + \lambda_2 \boldsymbol{\alpha}$. To find hyper-parameters λ_1, λ_2 we use gridsearch technique and principally choose $\boldsymbol{\alpha}$ for which L2 norm (second term in Eq. 4.2) is closer to 1. We calculate the importance weight $\mathbf{r}_j$ for an edge e_{ijk} in $\boldsymbol{\pi}_B$ using Eq. 4.3. To select the most

important time-series at every e_{ijk}, we sort $\mathbf{r}_j$ in descending order and principally select the top k time-series, when cumulative fraction of their total rationalization weight >= 0.95.

Algorithm Find-RaTSS. We next give the pseudo-code for our Algorithm 1.

Input: $\mathbf{X}$: a set of time-series, $\mathcal{C}$: a segmentation
Result: $\mathbf{r}_j = \{r^1, ..., r^m\}$, rationalization weight of $\mathbf{X}$ for every c_j in C
Consider F: a function for characterizing time-series. Given a segment, it returns a feature matrix $m \times f$
1. Construct nodes in segment Graph G
2. Construct edges in G
3. **foreach** $e_{ijk} \in \mathcal{E}$ **do**
   Calculate total number of paths p_{ijk} in e_{ijk} using Lemma 4.7
   Calculate edge weight $\mathbf{w}_{ijk} = ||F(s_{ij}) - F(s_{jk})||_{1,2}$
   Edge cost: $\boldsymbol{\pi}^{ijk}_{\text{rest}} = p_{ijk}\mathbf{w}_{ijk}$
end
4. $\boldsymbol{\pi}_{\text{rest}} = \sum_{e_{ijk} \in \mathcal{E}} \boldsymbol{\pi}^{ijk}_{\text{rest}}$
5. Compute $\boldsymbol{\pi}_B$
6. $K = 2^{T-2}$
7. Solve $\boldsymbol{\alpha}$ using Eq. 4.2, hyper-parameters λ_1, λ_2
8. **foreach** $c_j \in \mathcal{C}$ **do**
   Compute $\mathbf{r}_j$ using Eq. 4.3
end

Algorithm 1: Algorithm Find-RaTSS.

Implementation Details. For faster computation and handling large data for Algorithm 1, we adopt several techniques, such as (i) parallelization, (ii) floating point precision, and (iii) normalization.

(i) *Parallelization:* For efficient and fast computation of $\boldsymbol{\pi}_{\text{rest}}$, we parallelize Algorithm 1. We divide the task of calculating number of paths for edges among a set of processors n. Also we use a shared memory from Python Multiprocessing library to compute $\boldsymbol{\pi}_{\text{rest}}$.

(ii) *Floating point precision:* To efficiently store large value of p_{ijk} and K (for $T > 900$), we rearrange Eq. 4.2 as $\frac{\boldsymbol{\Delta\pi}}{K} = \boldsymbol{\pi}_B - \frac{\boldsymbol{\pi}_{\text{rest}}}{K}$. And we ignore $\frac{\boldsymbol{\pi}_{\text{rest}}}{K}$ for a very small value.

(iii) *Normalization of* $\boldsymbol{\Delta\pi}$*:* For efficient optimization and better $\boldsymbol{\alpha}$ convergence, we normalize on $\frac{\boldsymbol{\Delta\pi}}{K}$ by their max value and rearrange as $\boldsymbol{\Delta\pi} = \frac{\boldsymbol{\Delta\pi}}{KM}$, $M = \max(\frac{\boldsymbol{\Delta\pi}}{K})$.

Lemma 4.9 (**Time and Space Complexity**)**.** *The worst-case time complexity of our algorithm is as follows:*

- *Case serial:* $O(T^3mf) + O(I) + O(\mathcal{C}mf)$*,* I*= number of iterations in the gradient descent phase.*
- *Case parallel:* $O(\frac{T^3}{n}mf) + O(I) + O(\mathcal{C}mf)$*,* n *= number of processors.*

The space complexity of Find-RaTSS *is* $O(m + mn)$*.*

Proof. We discuss separate proof for time and space complexity.

Time complexity: (i) *Case serial:* The first term is to calculate π_{rest} for each e_{ijk}. Total e_{ijk} in G is $O(T^3)$. The second term is to solve α using Gradient Descent learning, and the third term is to solve $\mathbf{r}_j$ for every cutpoint c_j.

(ii) *Case Parallel:* Parallelization can be applied in Find-RaTSS for π_{rest} computation, then the total time complexity will be distributed among n processors and hence this is $O(\frac{T^3}{n}mf)$. However, time complexity of gradient descent and $\mathbf{r}_j$ computation is similar as in serial cases.

Space complexity: (i) Case serial: The first term $O(m)$ is to store π_{rest} and π_B. Since we need $O(m)$ π_{rest} and $O(m)$ π_B for m time-series. Hence $O(m) + O(m) = O(2m) \approx O(m)$.

(ii) Case parallel: To store final computation of π_{rest} and π_B it takes $O(m)$. Whereas the second term $O(mn)$ is to store temporary $\boldsymbol{\pi}_{\text{rest}}$ computation for each processor among n processors, $O(mn)$ is the number of possible edges in a segment graph G. Hence temporary space for n processor and final computation of $\boldsymbol{\pi}_B$ and $\boldsymbol{\pi}_{\text{rest}}$ is $O(mn) + O(m) = O(m + mn)$. □

Find-RaTSS is clearly linear in the number of time-series m. Although the serial version is also cubic in T, in practice we found that we were able to run the parallel version easily for all our datasets and it was quadratic in terms of T.

4.5 Experiments

We implement RaTSS in Python and Matlab. Our experiments were conducted on a 4 Xeon E7-4850 CPU with 512 GB of 1066Mhz main memory. We collect both general data and

Table 4.2: **Datasets Used.**

SI #	**Dataset**	**Time stamps**	**Time series**	**Cut points**	**Black box (B)**	**GT**
1)	*Gaussian*	350	8	3	[55]	✓
2)	*Insect*	5000	4	2	[52]	✓
3)	*Chicken Dance*	322	4	7	[76]	✓
4)	*Great Barbet*	2200	2	2	[52]	✓
5)	*Sudden Cardiac*	7000	2	3	[52]	✓
6)	*Hurricane Harvey/Irma*	264/169	250/271	3/4	[82]	
7)	*COVID-19*	53	104	11	state emergency date[1]	
9)	*Diptheria/TB*	52/41	90/60	4/4	[55]	

domain specific urban data, to quantitatively and qualitatively evaluate the performance of Find-RaTSS. The detail description of the datasets with ground-truth (GT) rationalizations are discussed here. An overview of the datasets and the segmentation algorithm (B) used on each dataset are shown in Table 4.2. We use B which gives the most meaningful cutpoints (based on GT or historical events).

General datasets: We use a variety of datasets (both synthetic and real) where we infer

the ground-truth *TOIs* (Table 4.2 SI 1-6).

1. *Gaussian*: We generate a synthetic data, where each time-series is a univariate Gaussian. We select three cutpoints and change the parameter of the Gaussian for specific $2-3$ time-series at those cutpoints. Ground-truth (GT) rationalizations are the time-series whose Gaussian parameters change at a cutpoint.

2. *Insect* [52]: EPG recording of insect vector feeding, where each time-series represents an insect. Each cutpoint is an event when feeding state of the insect changes and GT is the insect whose feeding state changes at that event.

3. **ChickenDance** [76]: Motion capture sequences of a set of sensors (left-right hands/legs) in a chicken dance originally collected by CMU[2]. An event occurs, when is a change of dance state, e.g., wings, tail feather, etc. GT are the set of sensors which have high change of motion while changing a dance state.

4. *Great Barbet* [52]: Voice recordings of same species Barbet birds in MFCC format. Each recording is a mixture of two different birds with approximately half-a-minute snippet of their song. The cutpoints are set of events when snippet of one bird ends and other starts. GT are the set of birds, whose call ends or starts at an event.

5. *Sudden Cardiac* [52]: ECG channel reading of hearts of patients. Events occur when heart state changes, e.g., normal heart to sudden heart failure or contraction. GT are the patient whose heart activity changes at an event.

Domain specific urban datasets: Next, we describe our domain specific datasets from urban analytics. Here as there is no ground truth, we discuss the qualitative performance our algorithm.

[2]http://mocap.cs.cmu.edu

6. *Hurricane Outage:* Oak Ridge National Laboratory (ORNL) has developed situational awareness tool *EAGLE-I* to collect power outage distribution of all the customers from utility websites every 15 minutes. We collect this power outage distribution of different counties for Hurricanes **Irma** and **Harvey** in the hurricane-affected areas used by [82]. Rationalizations on such data can help DE with retrospective analysis for emergency management planning.

7. COVID-19 : COVID cases of different states are collected daily from Jan-May by New York Times[3]

8. *Diptheria* and *TB*: Diptheria and Tuberculosis (TB) cases collected bi-weekly from the year 1900-2014 by Project Tycho[4]. Time-series are different cities of US representing Diptheria (TB) cases over time period (years). We run Dynammo [72] to replace the missing values of original data.

Baselines: We compare Find-RaTSS against plausible approaches. Our intuition is to find out the effect, by considering each cutpoint independent of the overall segmentation and without principally considering any optimization. Following we describe each baseline model.

(i) *Feature-based*: We calculate the change across every c_j using basic features, i.e., w_{ijk} and select x_u which have high w^u_{ijk}.

(ii) *Magnitute-based*: We calculate rate of change of time-series values across c_j for a window (5% of the timestamp). Our intuition is to pick the time-series easily figured out by visualization.

(iii) *Forecast error-based*: For every time-series x_u, we train an LSTM forecast model based on the segment before c_j and test the model for the segment after c_j. We select the

[3]https://github.com/nytimes/covid-19-data
[4]https://www.tycho.pitt.edu/

time-series whose forecasting error is high. Since high forecasting error denotes high measure of unpredictability on the time-series magnitude after the cutpoint. In other words change of time-series is high across the cutpoint.

We do not compare against a baseline method of interpreting the internal change of state of segmentation algorithm (as described in Sec. 4.1) since our algorithm is general for a black-box segmentation, whereas each dataset is a practitioner of a different segmentation algorithm.

4.5.1 Quantitative evaluation

We compare Find-RaTSS with the baselines on the ground-truth data. For rationalizations we select top k as the maximum number of GT in the segmentation. Table 4.3 shows F1-measures of all the baselines mentioned above. From the table, we observe that Find-RaTSS consistently performs better than all the baselines (upto 41%).

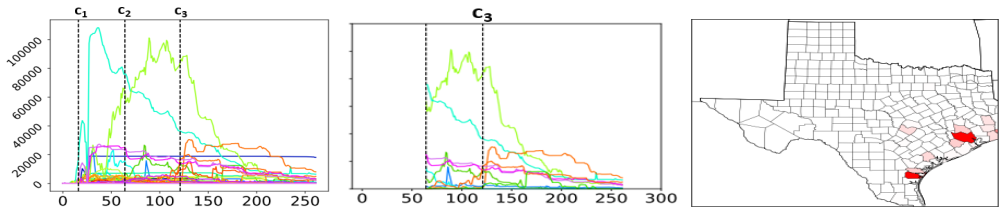
Table 4.3: **F1-scores of Find-RaTSS and baselines on the datasets with ground-truth**

Dataset	Find-RaTSS	Feature	Magnitude	Forecast
Gaussian	**1.0**	0.27	0.42	0.17
Insect [52]	**0.83**	0.33	**0.83**	0.5
Chicken Dance [76]	**0.86**	0.81	0.71	0.5
Great Barbet [52]	**1.0**	**1.0**	0.5	0.5
Sudden Cardiac [52]	**1.0**	0.67	0.67	0

4.5.2 Case-Studies in domain-specific urban data

Next, we show *TOIs* by Find-RaTSS in hurricane power failures, COVID-19 interventions, and epidemiology. Additionally to show that our algorithm correctly captures rationalizations

even for a large number of time-series (3000), we provide results on Wikipedia articles data.



(a) (b) (c)

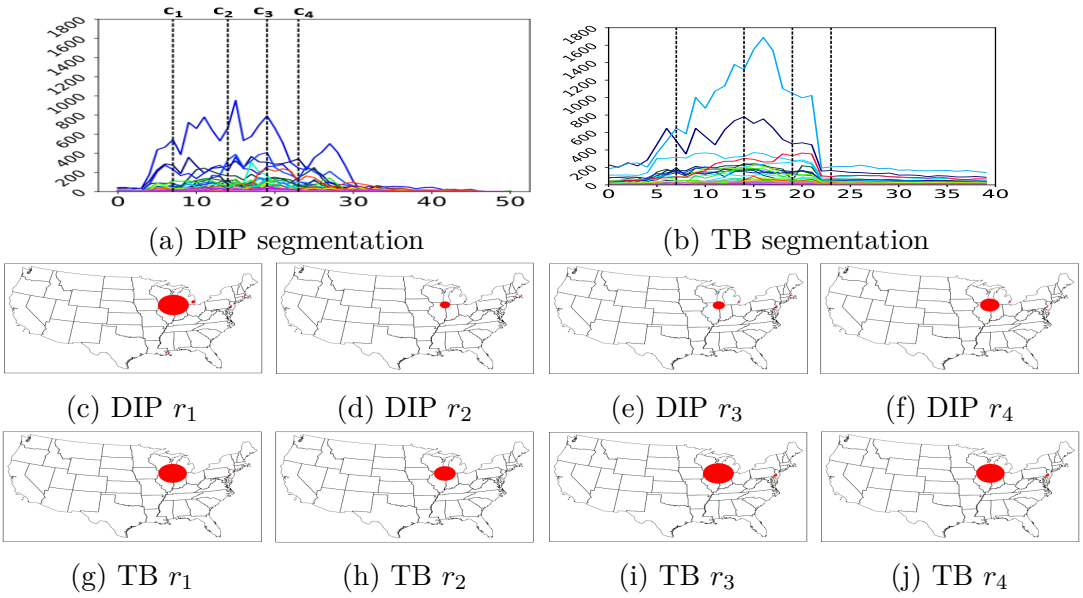
Figure 4.4: **(a) 2017 Hurricane Harvey time-series. (b) A snippet of our rationalizations r_j for cutpoint c_3 and (c) Heatmap plot of r_j for c_3. Find-RaTSS finds non-obvious rationalizations separate from Hurricane trajectory**

Hurricane Harvey: We already explained how our rationalizations are actionable for cutpoint c_2 (See Fig. 4.1 in Sec. 4.1 for detail). The segmentation of Harvey is given during hurricane landfall (around Aug. 26), change of hurricane trajectory (around Aug. 28), and end time of hurricane (around Aug. 30). Fig. 4.4b shows our rationalizations for cutpoint c_3. Note that, this is the cutpoint when hurricane is ending. Along with the decrease of the power failures of other counties, Find-RaTSS correctly highlights sudden increase of power failures of Orange, Jefferson, Hardin (Fig. 4.4c South-east corner). However, the main reason of this increase is due to rising water of Neches river due to which city lose service from major pump stations [23]. Note that, finding these non-obvious counties by visualizing the time-series (Fig. 4.4a) across the cutpoint when power failures of all other counties are decreasing is hard. These *TOIs* can help DEs prioritize resource allocation for quick recovery.

Diptheria (DIP): For Diphtheria and TB segmentation, we consider the time period from 1900 to 1950. This segmentation includes some historical significant events, i.e., campaign (c_1), outbreak (c_3), and invention of vaccination (c_4)[5].

Our rationalizations across c_1 gives actionable insights on the cities where Diphtheria cases

[5]https://timelines.issarice.com/wiki/Timeline_of_diphtheria



(a) DIP segmentation (b) TB segmentation

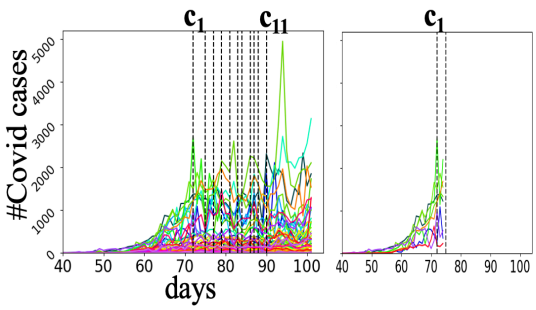
(c) DIP r_1 (d) DIP r_2 (e) DIP r_3 (f) DIP r_4

(g) TB r_1 (h) TB r_2 (i) TB r_3 (j) TB r_4

Figure 4.5: **Find-RaTSS finds rationalizations in Diphtheria and TB. Fig.(c)-(f) are our rationalizations (larger size by higher weight in r_j) in US map found by Find-RaTSS for the second and third cutpoint. Similar results for TB are also shown in Fig.(g)-(j)**

fluctuate the most after the campaign started.If we compare rationalization weights from Fig 4.5c-4.5f with Fig 4.5g-4.5j we observe, Find-RaTSS finds an interesting association between TB and Diphtheria. r_j^u weights of the affected cities are positively correlated for both Diphtheria and TB across the cutpoints. The same report (mentioned above) suggests this association is mainly due to presence of an iron-repressor gene. Clearly, by providing these insights, rationalizations can help DE understand the correlation between the diseases and design vaccination policies [101, 104].

COVID-19 Interventions: For the current COVID-19 pandemic, our goal is to extract which states had interventions (like school closures, etc.) using Find-RaTSS and the disease trajectories. We collect (https://covidvis.github.io/) daily COVID incidences from Jan-early May and consider cutpoints as the state emergencies after 2 weeks (mean incubation period of COVID is 2-14 days). For c_1 (Fig. 4.6(b)), Find-RaTSS infers all the states which



Rationalizations for c_1	Intervention on Jan 19
CA, OK	State emergency
MA, LA	School closure
MI	Gathering>500 ban
CT	Restaurant closed

(c)

Figure 4.6: **Covid-19 pandemic: (a) Time-series and segmentation (interventions). (b) Majority of the states for the first cutpoint c_1 inferred by Find-RaTSS had interventions in the past 2 weeks. (c) Types of interventions that happened before 2 weeks.**

had some intervention around 2 weeks back (Jan 19). Fig. 4.6(c) shows an example of different interventions happened for the rationalizations across c_1 before 2 weeks. Overall, across all cutpoints, Find-RaTSS infers 87% states with interventions >= 2 weeks. For most rationalized states (60%), the interventions happened exactly 2 weeks back. This is useful as in real-time it was very hard to have a complete knowledge of interventions, e.g., indeed, there was no centralized database of such acts. Hence epidemiologists needed to use indirect methods for knowing which interventions are in place and is crucial for modeling the disease spread. Find-RaTSS can potentially give such a method to the epidemiologists to direct attention to such states.

4.5.3 Feature ablation test

Table 4.4: **Feature Ablation test in terms of F1-score for Ground-truth datasets. f_1: mean, f_2: variance, f_3: min, f_4: max**

Dataset	Features removed			
	f_1	f_2	f_3	f_4
Gaussian	0.52	0.44	0.49	0.64
ChickenDance	0.73	0.86	0.401	0.643
Great Barbet	0.5	0.5	0	1.0

For feature robustness by we use feature ablation test. Tbl. 4.4 shows how rationalization affected on the Synthetic, ChickenDance and GrandMal dataset by removing each feature used in RaTSS.

4.6 Discussion

In this paper, we introduce a novel problem Rationalizing time-series segmentation in terms of constituent time-series (RaTSS), to identify actionable time-series of interests for urban domain experts in a set of events found by time-series segmentation algorithms. We propose an algorithm Find-RaTSS to solve RaTSS in terms of a novel graph optimization problem using a segment graph data structure. Find-RaTSS successfully finds *TOIs* in several domains such as emergency management and epidemiology datasets, such as historical Diphtheria cases and the recent COVID-19. In addition, we compare its performance with non-trivial baselines using synthetic and real-life general datasets with inferred ground-truth.

A silent assumption in RaTSS is that the black-box segmentation (events) is meaningful to the SME. For example, when the time-series is constant or segmentation is user-provided where cutpoints donot resemble to any event, then rationalizations (TOIs) found by RaTSS may not be meaningful. Indeed, arguably, it is not clear in this case if there are any meaningful TOIs in the first place. E.g., intuitively, for homogeneous data, cut points are not useful. Besides, we define our rationalizations in terms of constituent time-series only. As we discussed in Sec. 4.1 and show in our experiments later, such a definition is meaningful in an urban analytics context (e.g., actionable counties in the hurricane example). However, there can be situations where a group of time-series can also be actionable. As future work, we can extend RaTSS for more complex *TOIs* (like scoring *groups* of time-series instead of individual ones as we do) which may be more suitable for some other applications.

Part II

Network-based System Resiliency for Preparedness and Mitigation

Chapter 5

Improving Situation Awareness through Visual Analytic Tool

In this Chapter we will talk about our visual situation awareness tool to help decision-making to prepare and plan for a future disaster improving system resiliency.

5.1 Introduction

Critical infrastructure systems (CIs) such as power, cyber, water, and transportation are vital to sustaining national and economic life. Recent natural and human-made disasters like hurricanes, earthquakes and cyber terrorist events show how inter-dependencies across these CIs components can have catastrophic effects, by enabling hazards to potentially propagate and disrupt the functionality of the entire system. As a vivid example, the massive power outage during the 2003 blackout in the north-east US cascaded to impact water-waste treatments, transportation, communication and food industries [38]. To better manage the effects of such disruptive events, domain experts need to understand the complex inter-dependencies and failure dynamics over these systems. This is challenging due to the several layers of multidimensional data (such as types of inter-dependencies, types of failures and coupling) at various temporal and spatial scales.

We present Urban-Net, a network-based interactive visualization tool to model inter-dependency

and failure dynamics over CIs. This project is built around the idea of viewing CIs as large *heterogeneous networks*. The interface can also utilize topology-based analysis and create simulations to understand consequences due to inter-dependencies by intentionally created perturbations.

This broad project is a collaboration between Oak Ridge National Laboratory (ORNL) and Virginia Tech (VT). The tool is a copyright by ORNL and VT, and has been successfully used by many domain experts at ORNL since the last year. We also plan to integrate Urban-Net with the US Department of Energy (DOE) situational awareness platform EAGLE-I and the Federal Emergency Management Agency (FEMA) analytical framework for emergency planning and response purposes. To the best of our knowledge, no existing system provides a software stack that performs vulnerability analysis for *national scale* CIs.

Our project adopts a large scale network-based data analysis on a US national level and also has novel formulation, modeling and algorithmic contributions for domain-based failure dynamics and identification of vulnerabilities. In addition, as mentioned before, we bring together these ideas in an interactive tool Urban-Net, using different open source technologies. From an application viewpoint, Urban-Net can significantly enhance situational awareness for emergency management and we believe will be of significant interest to the stakeholders.

Next we will briefly describe our framework in Sec. 5.3 and our demonstration plan using a use-case scenarios in Sec. 5.4.

5.2 Preliminaries

In this Section, we define some terminologies for ease of understanding the analytic tool.

1. **Criticality**: Nodes or components whose initial failure can cascade to multiple other

nodes and cause large number of failures.

2. **Vulnerability**: Nodes or components which fail or get heavily affected due to the initial disruption of some other nodes.

3. **Inter-dependency**: Any physical or logical connections among two nodes of different infrastructure layer. Physical connections can be denoted as connections based on geographical distance. Logical connections can be denoted as service provided among different layers, e.g, power distribution, water supply, etc.

5.3 Framework

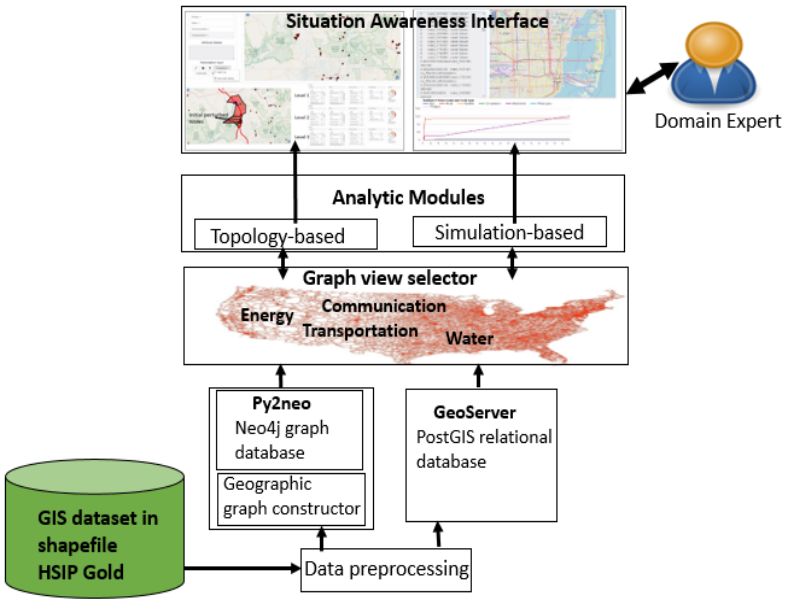


Figure 5.1: **Urban-Net overview. It first converts the geographic data into graph data for analysis (bottom). A domain expert interacts through the interface and selects different CIs components for vulnerability analysis (top) and send query to the server to fetch data for analysis through its modules (in the middle).**

Urban-Net consists of an interactive interface which also provides an option for two types of analysis which necessitates developing two back-end analytic modules: namely a topology

based analytic module and a simulation based analytic module. We present the workflow in the figure above. It first converts the geographic data into graph data for analysis (bottom). A domain expert interacts and selects different CIs components for vulnerability analysis (top) and send query to the server to fetch data for analysis through its modules (in the middle). Next, we discuss the three components of Urban-Net, datasets, and our key contributions.

5.3.1 CIs Network construction

We import raw shapefiles of geographic CIs data into a PostGIS/PostGreSQL[1] relational database. We then convert the relational data into a graph structure (e.g., vertices and edges) and import into a Neo4j graph database using Py2neo. We use the GeoServer to host and show geographical objects stored in the PostGIS data on a map [69, 70].

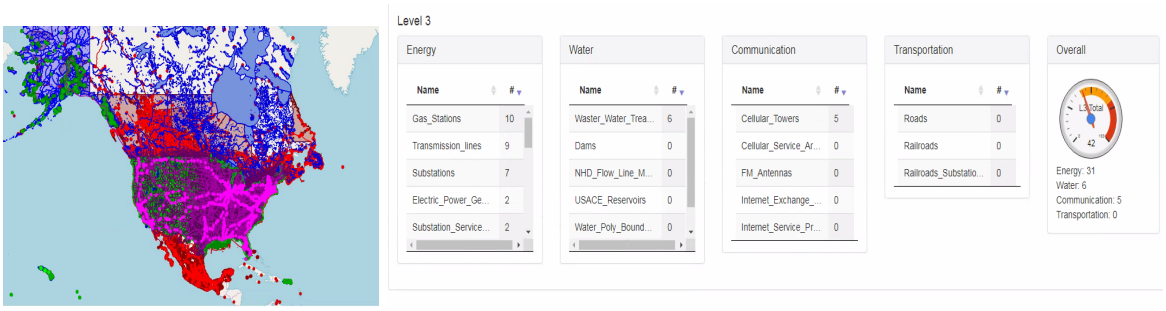
5.3.2 Urban-Net Interface

We also create a map interface, to allow both an overview and detailed access of the whole or part of the CIs networks based on the user's selection to analyze the nodes. Using the map interface, the user can easily visualize an overview of the network and identify critical nodes. For a more detailed analysis of the map, we allow navigation (zoom and pan) as well as auxiliary information of the selected nodes and networks to show in the left panel. Also, to have a clear understanding about vulnerability analysis along with the network visualization, we also integrate tabular and chart interface for each type of CIs such as energy, water, communication, and transportation system.

[1]https://postgis.net/,https://neo4j.com/,https://py2neo.org/v4/,http://geoserver.org/

5.3.3 Analytic Modules

Next we give more details on the interface design for each module.



(a) US CIs network (b) Affected networks

Figure 5.2: **Visualization of Topology-based analytic module. (a) shows the whole CIs network. Red, blue, green, purple indicates energy, water, communication and transportation network respectively. Note, it is difficult to identify the physical inter-dependency between them from the overall visualization. (b) shows a summary of the affected CIs network within** 15 **hops in tabular form.**

Topology-based Analytic module: The topology based analytic module aims to identify potentially affected entities across different CIs networks, based on different *perturbations* (say a failure). When converting shapefile data into a graph data to store into Neo4j database, we link nodes originating from different CIs based on a set of rules. For instance, we link a node that represents a power plant with a node that represents a substation that is geographically nearest to the power plant. When a user selects a node from the shown map, the module generates a graph query to find all entities that can be reached within k hops from the perturbation nodes. This can give users an idea about what could be the potentially affected entities and help them understand the influence of selected failures.

The whole CIs network has been categorized into four layers: energy, water, transportation, and communication. With the CIs layer selection tab the users can select any node types they want to view (and each type of node has a different color). With an interactive zoom, a user can select some random input nodes on the real-time map and see the topological effect

of interdependent nodes and edges. The demo provides the inter-dependency analysis of CIs components in three levels. The first level contains the initial perturbation nodes selected by the user, second and third level identifies the nodes within $\leq k$ and $\leq 2k$ edge hops from the initially perturbed nodes. For visualization, we consider $k = 7$ under the assumption that these would suffice the inter-dependency analysis study for the existing network. Fig. 5.2a shows the initial visualization of the module considering whole CIs network. By perturbing some particular CIs nodes it shows the affected networks within $2k = 15$ hops with the help of geographic map(Fig. 5.3a) as well as tabular form (Fig 5.2b).

In short, topology-based analytic module is graph theory based approach to identify the potentially affected entities based on geographic closeness. Our main contribution in this module is to provide an efficient dynamic computational model to identify the nearest CIs using the procedure mentioned above from any set of perturbed nodes selected by a user. The dynamic model provides user to choose the types of CIs for understanding inter-dependencies and analyzing vulnerabilities within (2-3 sec.)

Simulation-based Analytic module: With topology-based analysis it is hard to estimate the temporal delay of a failure node in a cascade which is very important in order to understand the restoration period of a node. Simulation-based analytic module on an energy based heterogeneous network allows the users to analyze real-time consequences of CIs grid layers of a region by varying seed nodes for perturbation, i.e., random perturbation (randomly generated seed nodes), regional perturbation (nodes generated centered around the selected region). This module also considers temporal aspects along with physical inter-dependencies. When considering temporal aspects, we allow users to control four parameters for every CI node types: i) average time a CIs node can support before it loses control (β) ii) average time (hrs) a node takes to recover from failure (α) iii) average load for a node and iv) average capacity for a node. To create the heterogeneous network, we consider the

current physical graph stored in the database.

- **Step 1:** In this step, we will explain how we generate perturbed nodes to provide an environment for real-time simulation — creating perturbed nodes based on user choice in necessary to understand inter-dependencies and vulnerability by simulating different consequences. We generate a list of seed nodes initially which are considered to be failed nodes. We adopt two different approaches to generate seed nodes based on user's choice. For the first, we randomly sample 10 nodes of the selected state and consider them as initially failed nodes. For the second, we randomly choose one node as the center node of the disaster affected state and list all the nodes as initially failed nodes which are within a $10km$ geographic radius of the center node.

- **Step 2:** We developed state-of-the-art tractable cascade models, which can be utilized using various user parameters (like load capacity etc.) and which are based on novel *path-based* failures between the various CIs components [38]. Tractable cascade models can help approximate the failure nodes fast and give the notion of simulation with more real-time consequence. We construct a heterogeneous network for energy based on two types of electric power plants (with and without using natural gas to generate power), substations, transmission or bus nodes, natural gas compressors, natural gas pipelines, road network. To build the cascade model the nodes in the network itself are connected if there is an edge and to other networks based on the idea of failure cascade proposed by [69]. Each type of node in the network has a certain power load it needs to transfer for operating conditions and the maximum power capacity it can transfer. If a node's load condition goes beyond its capacity it will be inactive for a time based on β. Once a node fails in a cascade, it can recover after a time period based on α. All these four parameter values (load, capacity, α, β) can be set by user's choice. Nodes under transmission bus, gas compressor, and power plants have the ability to distribute their

load to its neighbor by probability P if they fail. The amount of load it can distribute is also based on the user's choice.

- **Step 3:** In this step, we identify the nodes that will fail next in the cascade based on the current list of seed nodes and schedule recovery time of the new failed nodes exponentially distributed by their corresponding α. A node is considered a 'failed' and added to the list of seed nodes if it fulfills any of the following conditions. 1) For the nodes which are considered 'failed' in the list of seed nodes, it can redistribute a load of their neighbors by P. 2) For an active node, if load $>$ capacity then it is considered a new failed node. 3) If all the neighbors of an active node fail, then the node will be considered as 'failed'. 4) If all the nodes of its connected networks according to the cascade fail, it will be added as a failed node.

- **Step 4:** Next, we check if any nodes recover from its failed state, i.e., their scheduled recovery time starts, they will be added in the list of active nodes.

- **Step 5:** We implement step 2 and step 3 as long the simulation continues. We consider running the cascade unless all the nodes become active or if the recovery schedule of a failed node reaches its maximum constraint.

In the interface, for each timestamp t, we list the failed nodes and nodes recovered from a failed state in the current cascade and run the simulation based on the failed nodes for the next cascade. In order to understand the impact of nodes with time, we show a dynamic graph of time vs a number of failed nodes for each node type at the bottom. The nodes which fail can be shown both on the map with their detail information to the right.

The main contribution in this module is to develop a real-life tractable cascade model and integrate path-based failures among the various CIs components for the simulation. To the best of our knowledge, no one used this approach before for analyzing failures in CIs.

5.3.4 Implementation

Here we briefly describe the different technologies we used to implement Urban-Net. We pursue to use the open-source, easily accessible, and easy-to-integrate technologies for Urban-Net.

Interface: We design the user interaction of the interface using JavaScript, and the OpenLayers API Map library[2]. We also use a Python Tornado web server[3] to connect the user interface with the analytic tool and the geographic database (GeoServer and Neo4j).

Analyzing data: We perform graph operations on the constructed network, and it is implemented using the python library Py2neo (a python library for the Neo4j graph database) and a separate docker.

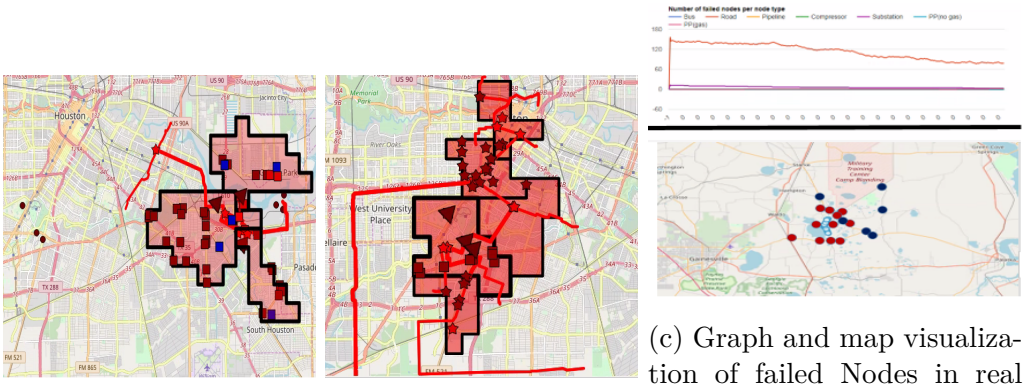
Dynamic map and concurrent hash table: We use the OpenLayers map library to maintain a dynamic map view based on CI nodes selected by the user. OpenLayers has fast thread-safe hash table and vector structure; WMS and WFS to store features or selected layers on map and query them on the client side.

5.4 Scenarios

For the demonstration, we store the existing CIs layers over US and their interdependent relations in the server. We also store analysis output of the simulation-based analytic module for one US state to present a faster visualization to the audience.

In the interface, users can interact with and navigate the map for each module. In the topology-based analytic module users can select perturbation nodes they want to consider

[2]https://openlayers.org/
[3]https://www.tornadoweb.org/en/stable/index.html



(a) Scenario 1 (b) Scenario 2 (c) Graph and map visualization of failed Nodes in real time

Figure 5.3: **(a)-(b)Analyzing physical inter-dependency of CIS in Houston. We perturb two different sets of power plants (red triangle) in (a),(b) which (a) affect some water treatment plants (blue rectangle). (b) affect highest number of energy network. (c) Simulation-based analytic. Bottom Fig. shows the initial seed/failed nodes (red) while others are the effect of failure cascade . (Top Fig.) is a graph showing the no. of failed nodes of different CIS type where the final cascade heavily damaged the road network.**

using 'command' and dragging over the nodes. They can also select the CIs layers they want to consider for analysis for each of the modules. Our plan is to invite the audience, let them interact with Urban-Net, turn them into analysts and develop use-case scenarios related to understanding the CI network using our tool.

Next, for each module we will demonstrate scenarios of Urban-Net illustrating its capability, performance and our plan for the attendees.

Next, we first talk about the machine configuration for Urban-Net. Then we present the important scenarios for each module of the system.

5.4.1 Datasets

We chose the four categories of CIS: energy, water, communication, and transportation based on the HSIP Gold data [8]. Urban-Net system has been deployed to 2.7 GHz 12-Core Intel

Xeon E5 system with 64GB DDR3 Memory. The constructed graph contains $81M$ CI nodes and $83M$ interdependent CI edges, and its total size of the data is approximately 60GB. To simplify the future deployment, we provide Docker containers for our backend modules including databases and geoserver.

5.4.2 Understanding inter-dependencies

We already explained the use-case scenario for Fig. 5.3a in our paper. In Fig. 5.3b, we selected another three power generation plants in the middle of Houston which effected a large portion CI network consisting of 68 energy nodes in total. However, these do not affect any other type of interdependent CI nodes except electric power plants. Thus we showcase two use cases for identifying vulnerabilities as well as a realistic effect on interdependent networks.

5.4.3 Understanding vulnerabilities

Here we explain in detail about the use-case scenario given in the paper. To showcase a use case for a simulation-based analytic module, we considered the state Florida, the most affected state during hurricane Irma. We selected some random nodes centered around the $10km$ of hurricane area using regional perturbation method used in Step 1 of Simulation-based Analytic module in Sec. 5.3 and ran the simulation for three mins. (considered maximum schedule recovery time as 10 mins.) to understand which nodes have been affected the most. We chose all types of CIs of energy network for the simulation. Bottom Fig. 5.3c shows the initial seed nodes where majority of them are gas power plant generators. The last cascade of the simulation after 3 minutes shows that the failure cascade heavily damaged the road networks, when failure of all other nodes reduced (Top Fig. 5.3c). This implies the

end time of the hurricane.

5.5 Discussion

Urban-Net has contributions in several dimensions. First, building CIs networks from geographic data is non-trivial as it requires handling millions of nodes and edges, coming from different data sources. Urban-Net automatically constructs a large-scale graph database from raw *shapefile* format of geographic data sets. Second, to identify vulnerabilities, Urban-Net brings both structural and more failure dynamic-based simulation based analysis. In contrast, most related work looks into only static analysis. In the topology based module, Urban-Net provides a fast, updatable computational model to determine the nearest CIs by granting the user to choose the types of CIs for understanding inter-dependencies and analyzing vulnerabilities. In simulation based analytic module, we developed state-of-the-art tractable heterogeneous failure cascade models in collaboration with domain experts at ORNL (e.g. taking into account novel *path-based* failures between the various CIs components [38]). We have also developed new efficient algorithms to quick pinpoint vulnerable entities (the 'hotspots'). Finally, CIs consists of many complex inter-dependencies distributed over multiple components, which makes a simple visualization of the entire network not very useful. To handle this, Urban-Net provides an easy-to-handle interactive visualization tool which can give both overview and detailed access of CIs networks based on user preferences (See Sec. 5.3).

Chapter 6

Efficient Power Contingency Analysis through Network Trigger Nodes

6.1 Introduction

Power systems within the US are highly interconnected and hence vulnerable to multiple failures triggered by extreme events (such as a hurricane or a man-made disaster). Modeling these failure dynamics for domain experts is very challenging, due to the complex relationships and the possibility of multiple contingencies. Contingencies (or unplanned outages) refer to a loss or failure of part of the power system, such as a transmission line, a generator substation etc. Usually domain experts approach contingency analysis (CA) for measuring grid reliability via the so-called 'N-k' analysis. This entails trying out *all possible* combinations of k failures and then subsequently evaluating the resulting impact using multiple metrics (load loss, line loss etc). Clearly, this standard analysis is very expensive, due to the combinatorial nature of the possibilities as well as due to the expensive power simulations which need to be run to evaluate impact. For instance, even for a very modest size of a power system with $N \approx 1000$ components and $k \leq 4$ leads to over billions of contingency scenarios [41]. Hence, even power experts have proposed several approaches to form an efficient list of contingencies using power-flow simulation tools [41] and scale them using high performance computing approaches [60], the issue is still challenging.

Hence, we aim to develop a complementary pre-analysis tool for the CA by considering the power system from a heterogeneous network perspective. In this network, the system components can be considered as nodes, the interconnections among the components as edges and failure dynamics as cascading failures on the network. Secondly, instead of trying out all possible failures of k components in the network (similar to the $N-k$ analysis), we propose to look for the k 'trigger' or crucial components whose failures can cause the highest impact (largest number of failures) on the network. Once found, these k trigger components allow domain experts to focus on the most critical components, and can then be used as an initial list of contingencies for further analysis with expensive simulation tools [14].

In this chapter, we develop a network-based method of finding trigger components to help power system CA, collaborating between computer scientists and power experts. In contrast to most network approaches which are only structure-based [28], we propose a new domain-inspired method that extracts the trigger nodes by capturing the various failure dynamics of a power system efficiently. Our main contributions are:

- We propose a **D**omain **I**nspired Method in **He**terogeneous **N**etwork (**DIHeN**) for efficient CA through a heterogeneous network-based model considering dynamics over graphs instead of purely structural analysis.

- We develop a network failure model with a probabilistic edge-weight setup inspired from complex domain-based mechanisms.

- We conduct a detailed empirical analysis over nation-scale rich real electric grid infrastructure datasets. We find that our approach can beat the state-of-the-art and also has practical benefits to power experts. Further leveraging an existing power simulation analysis, we also show that our model can find trigger nodes and is beneficial for power-flow simulation tools.

6.2 Preliminaries

Before starting our methodology, we give a brief definition of all necessary terminology used for the method and experiments.

1. **Contingency Analysis (CA)**: Contingency analysis (CA) in power systems are the tool to enable 'what-if' scenario analysis to evaluate the impacts on the power system based on the initial disruption of k components.

2. **Critical Nodes**: The nodes in the network whose initial failure causes a large number of failures in other nodes due to network cascading effects.

3. **Trigger Components**: Trigger components are the contingencies (losses or failure of components) which are initially disrupted to check the impacts quantified by line loss, voltage loss, etc. on the power system CA tools.

4. **Independent Cascade (IC) Model:** Given a directed weighted network G, consisting of set of N nodes and E edges. F represents set of edge-weights for corresponding edges in E. IC model is a popular information diffusion model [65] where once a node fails, in the next time step it gets one chance to fail its connected child with the probability equal to the connecting edge-weight. The cascade starts with a given set of seed nodes that 'initially' fails and ends when there is no new failed node.

5. **User defined critical nodes**: A critical infrastructure facility that are highly important for US national security or property, e.g., a military base.

6.3 Methodology

To analyze and prevent unplanned outages, the standard approach of $N-k$ analysis identifies trigger components that cause a maximum impact on the system through multiple failures quantified by metrics, such as line loss, load loss, etc. This procedure of identifying such trigger components becomes exponentially more difficult with the high value of k due to the combinatorial nature. Hence, we aim to find a set of trigger components for the SMEs without the help of a simulation tool. We can define our problem for finding trigger components in power system as:

Problem 6.3.1 (**Trigger Components in CA**)**.** Given a power system of N components and a value k. *Find* a set S of k contingencies or trigger components that can cause a maximum impact on the system through failures.

To address this problem, considering the power system as a heterogeneous network can be advantageous to capture the interconnections among the components, model various dynamics of failures and quantify the failure impact on the component. Hence, we propose a complementary network-based method **DIHeN** instead of purely using CA simulation tools, which are accurate but computationally expensive. For **DIHeN**, we first construct a heterogeneous network modelling the power system inter-dependency, develop a network-based failure cascade model capturing different failure dynamics inspired from the power domain, and finally find trigger components in the heterogeneous network using the failure model.

6.3.1 Network Construction

Constructing heterogeneous network from a power system is a crucial task, as every type of system component is available in geospatial format (shapefile). Each component is rep-

resented as a geographic object, e.g., point, polygon, line, etc. Since no inter-dependencies or relationships among the components (geographic objects) are addressed explicitly in the data, the researchers need to understand the geographic object and hence define relationships among them. Electric grid infrastructure consists of four types of components (Electric power plants, Transmission Substations, Transmission lines, and Distribution substations) as they have the ability to generate/distribute power [63] and hence are very important for CA. We specifically select these four types of power system components and collect from the publicly available critical infrastructure (CI) data HIFLD[1] for the US [13]. We consider each component as a node in the network. To capture the inter-dependency (connection) across different types of components, we use the nearest-neighbor based on geographical distance between the components using a distance threshold (25 kilometers) to avoid creating dependencies between far distant components. To build the network from the geospatial format we use an existing DoE tool by Lee et al. [69]. Table 6.1 shows the four components (node) types in the network and their functionalities. For the rest of the paper, we use the terms component and node as interchangeable.

The connections among the different components are described in the following. We call a node as parent if it supplies power to its connected node, and child if it consumes power from its connected node.

Electric Power-plants (EP): An EP generates and distributes generated power to multiple Transmission Substations (TS). We model an EP node to have one or multiple children (TS nodes).

Transmission Substations (TS): TS receives power from one or multiple EPs, and the received power is transmitted to another TS or a distribution substation (DS). Thus, we model a TS to have one or multiple EP nodes as parents and one or multiple TL nodes as children.

[1]Homeland Infrastructure Foundation Level Data

Component type	Functionality
Electric Power-plant (EP)	Generates power which transmits to transmission substations for power distribution.
Transmission Substations (TS)	Connects two or more transmission lines.
Distribution Substations (DS)	Transform voltage and distributes power to consumers.
Transmission Lines (TL)	Transmits power from TS to TS or TS to DS.

Table 6.1: **Description of components of power systems**

Transmission Lines (TL): TL connects TS and TS or TS and DS. Thus, A TL node may have TS nodes as its parents and have DS or TS nodes as children.

Distribution Substations (DS): DS is the terminal component, where it receives power from a TL and distributes received power to multiple consumers within its service area. Note that, we did not include consumer components in our model. Each DS node has one or multiple parent TL nodes.

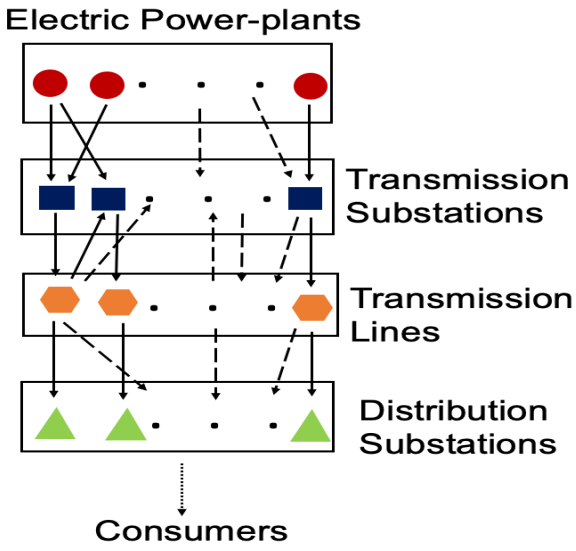


Figure 6.1: **Structure of a power system network.**

6.3.2 Failure Cascade Model

Modeling failure dynamics through trigger components is difficult due to its complex mechanisms, and no single model can capture all the dynamics [98]. E.g., failure in one component may induce stress to its neighbors due to multiple conditions, such as miscoordination of relays, line tripping, overloading, etc. All these might result in cascading failures to other

components [54].

To model such failure, we leverage a network-based independent cascade (IC) model that serves as complementary products to the traditional power flow simulation tools to evaluate the impacts of extreme events on power systems. The main challenge in the IC model is to model the edge-weights in the network for the cascade. Each edge-weight represents an impact weight which denotes the probability that a node may fail if its parent node fails. Our goal is to model such probabilities in a realistic way, inspired from real power systems.

6.3.3 Impact Weights

Impact Weights F is a set of edge-weights for the edges, where each f_{ij} for the edge e_{ij} denotes the probability of failure of a node n_j if its parent node n_i (node that supplies power to n_j) fails. In the power system, the failure of a parent cannot guarantee failure of its child due to the following properties: **P1.** A node will not fail if it can have a consistent power supply through its other parents due to power redistribution [54]. **P2.** The components in the power system is associated with a protection system (e.g., circuit breakers) which tries to isolate the failed component from the entire network to prevent further damage due to a cascade. These protection systems may break due to an increase of stress induced by some conditions and only then retain the ability to cascade [49, 62]. Considering these properties, we propose two probability models for F.

Failure Propagation Probability.

To satisfy **P1**, we make the following assumption: when a node of type t fails, its children (nodes that consume power from it) can gain power from other similar type parents. E.g., if a TS node fails, and its child transmission line (TL) component has another parent which

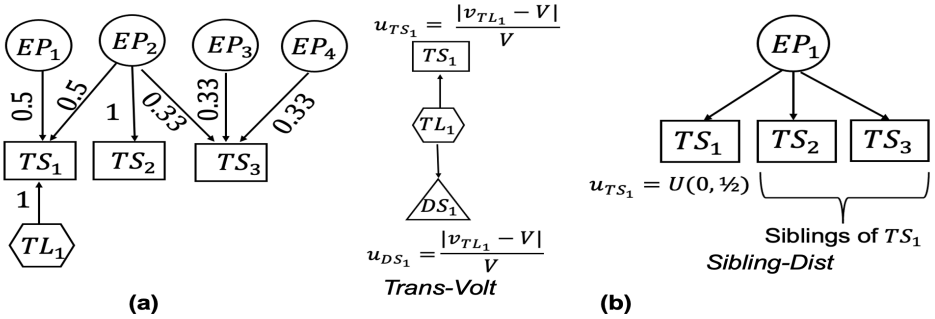


Figure 6.2: **Example of (a) Failure Propagation Probability (k_{ij}) (b) Cascade-blocking probability (u_j).**

is TS, it still may get power through redistribution and hence this decreases the chance of failure of the TL. Thus, for failure propagation probability we simply assume, a node has less chance to fail if it has high number of parents. In other words, a node with higher number of parent are less vulnerable (see example Fig. 6.2(a)). Using this assumption, we first identify the type of each node and the number of parents each node has. For each edge e_{ij}, the probability that a node n_i can propagate its failure to node n_j is:

$$k_{ij} = \frac{1}{\sum_{z \in par(v_j)} I(t_i, t_z)} \tag{6.1}$$

where, $par(v_j)$ are the parents of v_j, t_x is the type of node x and $I(.,.)$ denotes if two nodes are of same type, i.e.,

$$I(t_i, t_j) = \begin{cases} 1 & \text{both nodes are same type, i.e., } t_i == t_j \\ 0 & \text{otherwise} \end{cases} \tag{6.2}$$

Cascade-blocking Probability.

For satisfying **P2**, we model each node to have a probability to prevent cascading failure to its child. In other words, the node cannot cause any further damage and cascade stops.

To capture such blocking probability, we come up with two approaches, one where domain knowledge is involved and provided by the SMEs, the other is based on a realistic assumption but inspired from the scenarios of power simulations. The latter is applicable to the cases where we cannot apply the SME rule.

Trans-Volt: Recommended by the SMEs, for modeling u_{ij} we use the following domain knowledge : Transmission Lines $> 330KV$ are typically deemed critical since they supply more power[2]. Failure of these nodes can highly impact its connected nodes[3]. Based on this knowledge, we come up with the following assumption: The high voltage TL nodes if failed induce more stress on its child to cascade which cause less chance of preventing (blocking) the cascade [10]. Following this assumption, we first collect the associated voltage of every TL node. We set a threshold value V as maximum voltage of all the TL nodes in a network. For edge e_{ij}, with the connecting nodes n_i and n_j, if n_i is a TL node then $u_{ij} = \frac{|v_i - V|}{V}$, where v_i is the associated voltage of node n_i and V is the threshold voltage value in the network. High voltage TL (v_i) leads to lower $|v_i - V|$, hence decrease the chance of blocking a cascade (u_{ij}).

Sibling-Dist: For the edges where above domain knowledge is not applicable, i.e., no TL nodes in the connecting edge, we make the following assumption: the cascade blocking probability of a node n_j is higher if the node has fewer number of siblings (nodes other than n_j which consumes power from the same parent as n_i) [62]. If n_i fails, it impacts on all its children. This induces stress on the protection system on all the siblings of n_j. As a result, few siblings cause less stress on the protection system to deal with, hence more chance to block the cascade. On the other hand, more siblings cause more stress on the protection system to deal with, hence less chance to block the cascade. Based on this assumption, we design u_{ij} of node n_j in edge e_{ij} as the following,

[2]https://www.nerc.com/pa/Stand/Reliability%20Standards/CIP-002-5.1a.pdf
[3]https://www.nerc.com/pa/Stand/Reliability%20Standards/CIP-014-1.pdf

$$u_{ij} = \mathcal{U}(0, b) \tag{6.3}$$

$$b = \frac{1}{\sum_{z \in Ch(par(n_j))\ n_j} I(t_z, t_j)} \tag{6.4}$$

where t_x is the type of node n_x, $\mathcal{U}(.)$ is the randomly sample from uniform distribution, $par(n_j)$ gives all the parents of n_j, and $Ch(par(.))$ represents the union of all the child node of all the parents in $par(.)$, i.e., the siblings of the node in (.). Since, there is no specific domain knowledge involved about impact of protection system for a particular number of siblings, we randomly select u_j from a uniform distribution based on the number of siblings the node has. The fewer sibling yields higher values of b, hence the higher expectation of u_j (see example in Fig. 6.2) (b).

Finalizing F.

k_{ij} and u_{ij} are based on two independent properties **P1** and **P2** and f_{ij} is based on the probability of both the events occuring together. Suppose, n_i and n_j are the connecting nodes of e_{ij}. If n_i fails, then probability that it will fail n_j is $f_{ij} = k_{ij}(1 - u_{ij})$, where $1 - u_{ij}$ is the probability that n_j can cascade.

6.3.4 Problem Formulation

We formulate Problem 6.3.1 as identifying a set of critical nodes which can cause maximum downstream damage to the network due to cascading failures.

Problem 6.3.2 (**Critical Nodes in Network**)**. Given**, a heterogeneous power system network G, a set of edge-weights F on G, the IC model M which uses F, and a budget k.

Find the best set S^* of k nodes that can cause maximum downstream damage on G using M, i.e.,

$$\arg\max_{S} E[S] = \sum_{n_i \in N-S} P(n_i|S). \tag{6.5}$$

$P(n_i|S)$ is the probability that a node n_i can fail in the IC, given the initially failed nodes S. N represents all the nodes in G. $E[S]$ represents the expected number of nodes that may fail given S.

6.3.5 Finding Critical Nodes S

To find the set of critical nodes as mentioned in the Problem 6.3.2 is a well-known Influence Maximization (IM) problem using IC, where $E[S]$ is submodular [65]. To iteratively select k nodes from the network and to solve IM faster on a large network we use the Degree-Discount independent cascade heuristic [42] over greedy [65, 71]. Our framework for **DIHeN** is shown in Algorithm 2.

Input: D: Geographic shapefiles of the components of power system,
k: a budget
Result: S: a set of k critical components in D .
1. $S = \{\}$
2. Construct heterogeneous network G from D as mentioned in Sec. 6.3.1
3. Compute Impact weight F in G
foreach $e_{ij} \in E$ **do**
  Calculate Failure Propagation probability k_{ij} mentioned in Sec. 6.3.3
  Calculate Cascading-blocking probability u_{ij} mentioned in Sec. 6.3.3
  Compute $f_{ij} = k_{ij}(1 - u_{ij})$
end
4. Construct the IC model M using F.
5. Select k nodes of S using M solving Problem 6.3.2.
6. **return** S

Algorithm 2: DIHeN Framework.

6.4 Datasets and Experiments

In this section, we design various experiments to evaluate our results showing **DIHeN** using F is able to find critical nodes and has practical benefits to contingency analysis tools. All experiments herein were conducted on a 4 Xeon E7-4850 CPU with 512 GB of 1066Mhz main memory. We implement **DIHeN** using Python.

Datasets. We construct both national and regional level heterogeneous networks based on the US power system. We consider different sectors of energy infrastructures, such as Bulk Electricity System (BES). We collect different categories of power system components pertaining to BES. Every component in a category also contains multiple power attributes (such as voltage, load, generated power, etc.). Following, we describe each network in detail.

- *National-level Distance-centric*: We construct the network using the US national level geospatial critical infrastructure (CI) HIFLD [13]. For the edges, we define the inter-dependencies among the components based on the geographical distance between two components with a distance threshold of 25km as described in Sec. 6.3.1.

- *National-level Domain-centric*: For the network, the components and inter-dependencies (edges) are entirely supplied by the SMEs in contrast to using the geographical distance.

- *Regional-level*: The national power system in the US is comprised of three major interconnections of electric grid infrastructures which operate independently from each other, Eastern Interconnection (EI), Western Interconnection (WI), and Electric Reliability Council of Texas ($ERCOT$) [12]. We construct networks using HIFLD data [13] based on two major interconnections- EI (encompasses the eastern region) and $ERCOT$ (entire Texas state covers most part of $ERCOT$).

We aim to show that **DIHeN** is adaptable for both national and regional networks consisting of different interconnection types. The overview of the networks are shown in Table 6.2.

Network type	**Name of Network**	**Total Nodes**	**Total Edges**	**Components**		**Description**
				Type	**Size**	
National	*Distance centric*	118956	82750	1. EP 2. TS 3. TL 4. DS	10564 1806 41600 64986	Entire power system of the US.
National	*Domain centric*	93579	135951	1. EP 2. TS 3. TL 4. DS	10732 3553 51425 27869	Power system of US based on the interconnections taken from SMEs.
Regional	*EI*	77124	51851	1. EP 2. TS 3. TL 4. DS	4138 6705 25610 40671	A major interconnection encompassing most of the area east of US.
Regional	*ERCOT*	9719	6294	1. EP 2. TS 3. TL 4. DS	375 646 3456 5242	Entire TX covers most of *ERCOT*, a very different infrastructure compared to *EI* and *WI* [12].

Table 6.2: **Overview of the networks utilized for the study.**

Research Questions. Our goal is to demonstrate that S found through **DIHeN** provides a good solution to Problem 6.3.2. Specifically we want to address:

Q1. Does S fail large number of nodes to satisfy the goal of maximum downstream damage?

Q2. Does S contains critical nodes useful for contingency analysis?

Q3. How do different modules of F in **DIHeN** affect the performance on S?

Q4. Does **DIHeN** enable the nodes to use as trigger components in real power simulations?

Q1, **Q2** are aligned with the goal of the output mentioned in Problem 6.3.1 and Problem 6.3.2. We plan to showcase the performance of **DIHeN** on multiple power system datasets and compare with different non-trivial models. For **Q3**, we analyze the performance of dif-

Criteria Name	Criteria	Description	Networks Used
R_1	$V \geq 345$KV	Transmission lines ≥ 345KV are deemed critical since they serve > 1000MVA load [11]	*Distance-centric*, *Domain-centric*, *EI*, *ERCOT*
R_2	Near user-defined critical nodes	A power system component that is one hop away from a user-defined arbitrary critical component	*Distance-centric*, *Domain-centric*, *EI*, *ERCOT*
R_3	$V \geq 345$KV and near user-defined critical node	Components of ≥ 345KV which are near a user-defined arbitrary critical component and located within one or two hop away	*Distance-centric*, *Domain-centric*, *EI*, *ERCOT*
R_4	$L \geq 300$MW &	Components that supply load ≥ 300MW are critical	*Domain-centric*
R_5	Near-NG	A power system component that is one hop away from a Natural gas station	*Domain-centric*

Table 6.3: **Criticality criteria for M2.**

ferent variants of F in **DIHeN** as described in Sec. 6.3.5. With **Q4**, we are interested to retrospect whether **DIHeN** provides S, that experts can use for contingency analysis.

Measures of Success. For evaluating S, we choose the following metrics:

M1. *failure spread*: Motivated from the use of IC (Sec. 6.3.2) for selecting S, we plan to measure in terms of the expected number of nodes failed in the network by S, i.e., *failure spread*$= E[\#n|S]$.

M2. *coverage gain*: To measure that nodes in S are critical, we use the criticality criteria of power components identified by the SMEs[11]. We select these criteria based on availability of the attributes of the HIFLD data. Table 6.3 shows the expert identified criteria, description for the corresponding criteria. For each criteria R_x and for each node $n_i \in N$, let $Z(n_i, R_x)$ be 1 if n_i satisfies R_x and 0 otherwise. We define the *coverage gain* for criteria R_x as the ratio of coverage of S for R_x to the expected

number of randomly selected k nodes that satisfy R_x:

$$M_{Gain} = \frac{\sum_{s_i \in S} Z(s_i, R_x)/k}{\sum_{n_i \in N} Z(n_i, R_x)/N} \tag{6.6}$$

Baselines. For baselines, we consider popular network-based model to demonstrate utility of DIHeN. We check the performance of DIHeN against the popular network analysis techniques (1-3) and models with uniform edge-weights (4).

1. *PageRank (PR)*: Compute PageRank scores of nodes using edge-weights as our F. Pick k nodes with highest PageRank scores.

2. *Degree-Centrality (DC)*: Pick k nodes with highest degree.

3. *Critical Score (CS_{xy})*: Model developed to quantify criticality of nodes in US national level critical infrastructure network by computing a weighted average of the PageRank scores and the PageRank scores in the reverse graph (reverse PageRank) of each node [44]. We pick k nodes with the highest CS_{xy} (x, y denotes the weight used for each score).

4. *IC with edge-weight $x \in [0, 1]$ (IC_x)*: Use same edge-weight x for all the edges and select k nodes running the DIHeN framework.

6.4.1 Effectiveness (Q1-Q2)

We show that DIHeN is effective in finding S for which (i) the expected number of nodes that will fail given failure of S is large (in terms of **M1**) and (ii) contains critical nodes and can be used for CA (in terms of **M2**).

(i) Evaluation using M1. For **Q1**, the S selected by DIHeN and the baselines, we run our

IC model M. Fig. 6.3 shows *failure spread* (vertical axis) with the increase of k (horizontal axis) on the *Distance-centric* network (we do not show the results for other networks, as the scenario is the same for all other networks). Our model outperforms all the baselines and the difference of failure spread grows as k increases. **(ii) Evaluation using M2.**

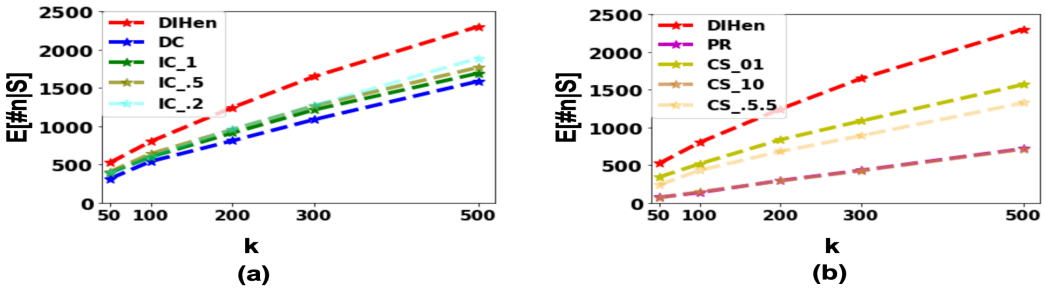
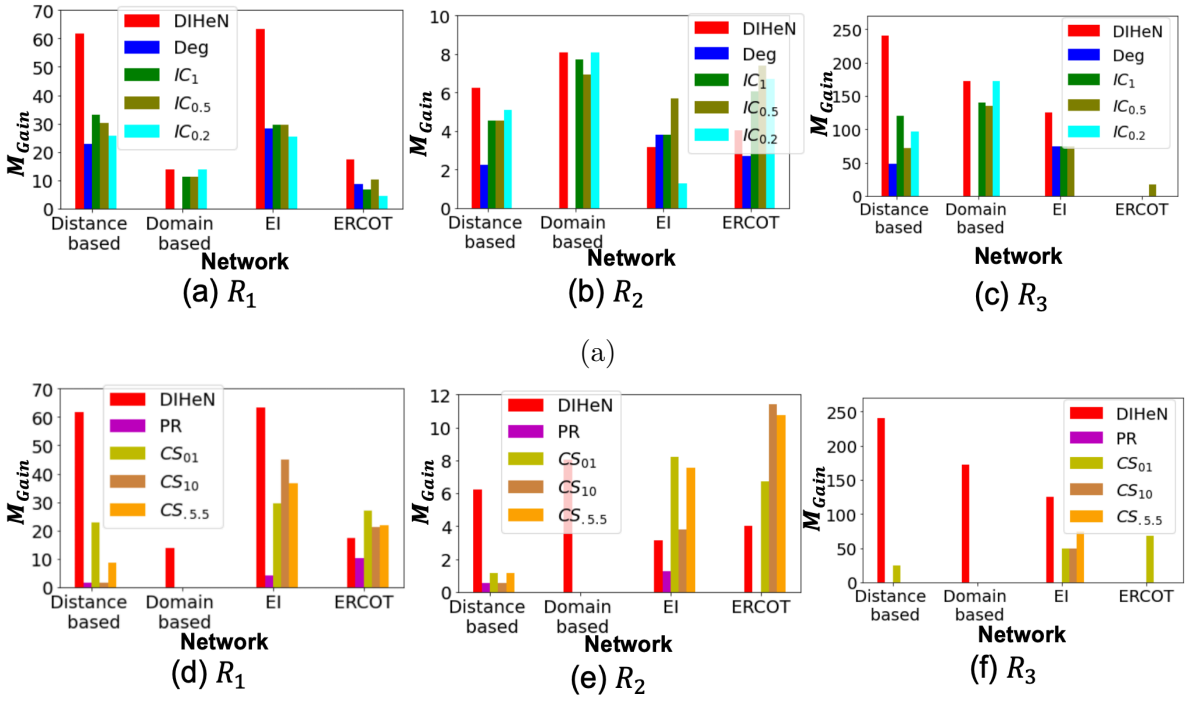


Figure 6.3: **Evaluating S in terms of M1 : *failure spread* vs k. High *failure spread* is the better. DIHEN outperforms all the baselines shown in (a) and (b).**

(ii) Evaluation using M2. For **Q2**, similar to the above experiments, we aim to show performance of **DIHEN** against the baselines as well as varying k on different G. Fig. 6.4 shows the performance of **DIHEN** comparing with the baselines using **M2** on all the datasets. The horizontal axis represents the datasets mentioned in Table 6.2 and vertical axis represents M_{Gain} for each model. Each bar represents a model. For the space constraints, we only show the performance of IC_x for the best values of M_{Gain}, i.e., $x = 1, 0.5, 0.2$.

Our model consistently outperforms all the baselines on the national networks for $R_1 - R_3$. Among all the networks, **DIHEN** best performs 3.5× in R_1, 2.75× in R_2, and $5x$ in R_3[4]. For R_3, **DIHEN** is very high and around 240× *coverage gain* (in the *Distance-centric* network) which no other baselines can perform. For $R_4 - R_5$, **DIHEN** also outperforms the other baselines on *Domain-centric* network (see Fig. 6.5). The *Distance-centric* national network and the regional networks *EI* and *ERCOT* do not contain the necessary attributes to evaluate R_4, R_5. Fig. 6.4 (a)-(c) also indicates that for the baseline IC_1, when failure of initial nodes

[4]The values are found considering the maximum value of ratio of M_{Gain} of **DIHEN** and M_{Gain} of a baseline model, i.e., max $\frac{M_{gain}^{\text{DIHEN}}}{M_{gain}^{\text{baseline}}}$, $M_{gain}^{\text{baseline}} \neq 0$



(a)

(b)

Figure 6.4: **Performance of DIHeN comparing with baselines in terms of the criticality criteria R_1-R_3 on different datasets. Fig.(a)-(c) shows the comparison for the baseline models with uniform edge-weights. Fig.(d)-(f) shows comparison against the popular network analysis approaches. Here, we consider $k = 500$ for *Domain-centric* network, and for all other datasets we consider $k = 50$. High M_{Gain} is the better.**

cause every connected nodes in the cascade to fail, (i.e., a large spread of failure) this can not guarantee critical nodes useful for CA.

For *EI*, we outperform all the baselines in R_1, R_3. For R_2, our *coverage gain* are comparable in 5 out of 8 baselines. On average ratio of our M_{Gain} with respect to other baselines is 1.95× in R_1, 1.18× in R_2, and 1.67× in R_3. For *ERCOT*, DIHeN consistently outperforms the baselines in R_1 (5 out of 8 baselines). M_{Gain} for all the baseline models and DIHeN is 0 in R_3 due to the presence of very sparse data (only 11 nodes in the network satisfy R_3). In R_2 DIHeN provides consistent M_{Gain} on all the network based models (performs best in 2 out 8 baselines). Overall, mean ratio of our model gain to the baseline is 3.45× in R_1 and .83× in R_2. Fig. 6.6a- 6.6c shows the

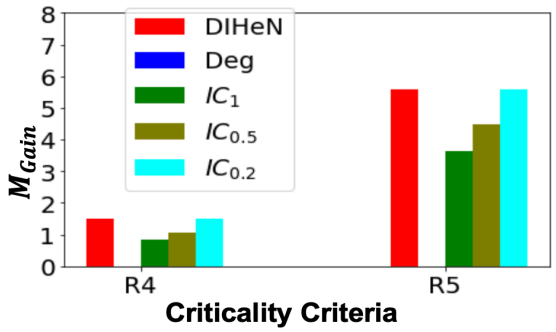
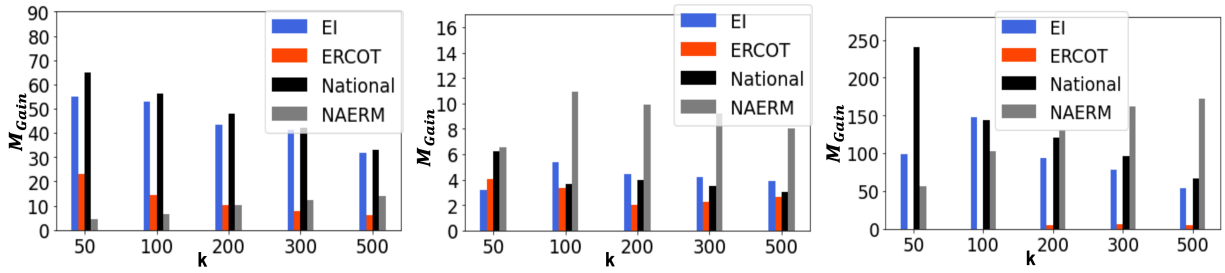


Figure 6.5: ***Domain-centric* network for $R_4 - R_5$.**



(a) R_1 (b) R_2 (c) R_3

Figure 6.6: **Performance of DIHeN for the criticality criteria R_1-R_3 mentioned in Table 6.3 using M2 on different networks in terms of M_{Gain} vs size of k for each R_x. High M_{Gain} is the better.**

performance of our failure model F for different G varying k, for R_1-R_3 (Table 6.3). Each bar represents a power system network. The horizontal axis represents size of k and vertical axis represents M_{Gain} for each R_x. The higher M_{Gain}, the better the performance of DIHeN.

DIHeN consistently provides good set of critical nodes for both national and regional networks and average M_{Gain} across all the networks is 39.3× better than cherry picking the nodes for R_1, 4.87× for R_2, and 108× for R_3. For R_1, the results show decrease of M_{Gain} with the increase of k, although the number of nodes in S satisfy R_1 increases with k. This indicates that we can get a good set of S satisfying R_1, even for small value of k. For R_3, M_{Gain} for DIHeN is low in $ERCOT$ (0 for $k < 200$), since number of nodes satisfy R_3 in $ERCOT$ is very scarce (only 11 nodes out of 9500 satisfy R_3).

6.4.2 Ablation Studies (Q3)

To evaluate the impact of different variants of F, we describe the models based on each of these variants (Sec. 6.3.3). Note that, we only set the edge-weights using each of the following model. To select the top k nodes, we use DIHeN framework.

1. *Failure-propagating (FP)*: Set the edge-weights, $f_{ij} = k_{ij}$ using Failure Propagation probability to check satisfying **P1** is not enough.

2. *Cascade-blocking (CB)*: Set edge-weights, $f_{ij} = (1 - u_{ij})$ using Cascade-blocking probability (Sec. 6.3.3) to check satisfying **P2** is not enough.

3. *No expert rule (NER)*: Assign the edge-weights $f_{ij} = k_{ij} * (1 - u_j)$ using Eq. 6.3 for all the edges to check effect of using only model Sibling-Dist (sec. 6.3.3).

4. *Random Gaussian (RG)*: Select edge-weights randomly from the Gaussian distribution $N(\mu, \sigma)$, where μ is the mean of DIHeN F and σ is the variance obtained from DIHeN F. We intend to compare our domain inspired meaningful method of edge selection (F) over random selection of edge-weights.

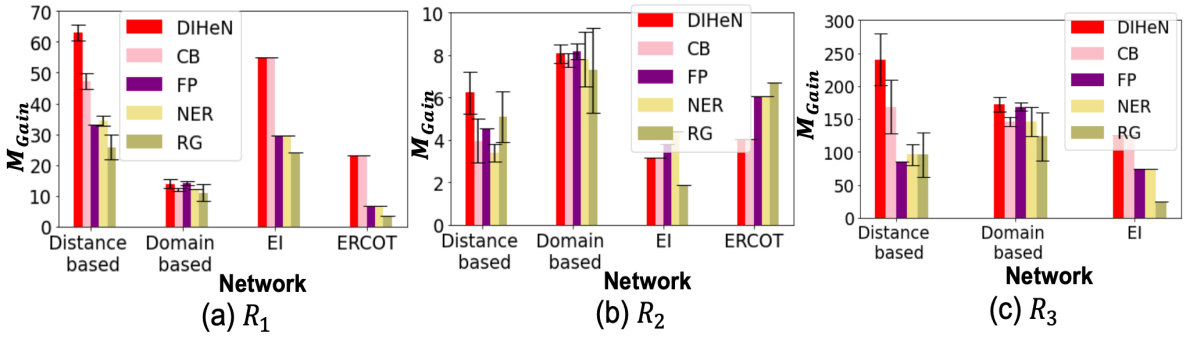


Figure 6.7: **Ablation study on different variants of F on national and regional network for criticality criteria $R_1 - R_3$. For each model, the black vertical bars show standard deviation of M_{Gain} running DIHeN framework 1000 iterations. In *ERCOT*, for R_3,** $M_{gain} = 0$ for all the models, hence not present in Fig. (c)

Using **M2**, in Fig. 6.7, we compare performance of **DIHeN** against the models consists of different variants of F on all the national and regional datasets.

Our model outperforms all other models in the national network and shows competitive performance on the regional network for criticality criteria R_1-R_3. For checking the consistency of our performance, we run the **DIHeN** framework for each model 1000 times. From Fig. 6.7(a)-(c), the low deviation f M_{Gain} (black vertical lines on each model) shows M_{Gain} of all the models are consistent. In the regional networks *EI* and *ERCOT*, **DIHeN** performs almost similar comparing with *CB*. This might be due to sparse graph unlike *Distance-centric* and *Domain-centric* national network, where most components have only one parent, hence $k_{ij} = 1$. Hence, edge-weights F of **DIHeN** is similar weights as model CB (u_{ij}).

Remark: Note that, if we only consider the nodes which apply to criticality criteria $R_1 - R_5$, the total number of components is 7000 in the *Domain-centric* national network. Analyzing combination of 7000 nodes for CA is also exponential. On the other hand, if we choose nodes randomly from 7000 nodes, it can not guarantee the failure of a large number of nodes downstream. Hence, our goal behind using M_{Gain} is to provide the SMEs sufficient reason to filter the nodes that maybe important for CA.

6.4.3 Qualitative Case-study: Power-flow Analysis (Q4)

In this section, we show that the selected nodes using **DIHeN** can be used for real CA simulations by leveraging high fidelity simulation analysis on the power system of *ERCOT* conducted through an in-depth study by the SMEs. Due to privacy concern we cannot share the reported results.

Power-flow analysis. For the analysis, the SMEs choose 81 high voltage transmission lines ($= 345KV$). They first conduct a *base case* analysis, i.e., ideal scenario when no components are removed (i.e., nothing has failed). From the analysis, several parameters are recorded, e.g., the total load due to power flow, number of voltage violation (rise of drop of voltage), the convergence of the simulation, etc.). Next, multiple simulation scenarios are created, each time removing a different subset of transmission lines. The parameters are compared with the *base case* scenario (amount of load loss, number and nature of components which cause voltage violation, the simulation is convergent, etc.). Based on the above comparisons, the experts rank each transmission line. DS components that give the worst number of voltage violations are identified. If a simulation does not converge, the SMEs consider such scenario as highly critical.

Our goal is to check nodes in S with the expert provided components in *ERCOT*. However, due to privacy concerns we are unable to collect the exact *ERCOT* components used by the utility company. Hence, we plan to use our *ERCOT* power system data as mentioned in the Table 6.2. Due to the difference of these two datasets, we cannot compare top k nodes in S using **DIHeN** with the SMEs provided rank of transmission lines. Instead, we aim to qualitatively evaluate S in terms of the available attributes of the nodes as provided by HIFLD [13]. We also record the attributes of the components reported by the SMEs in the scenario analysis. Table 6.4 shows these attributes, its descriptions, and *coverage gain*

(M_{Gain}) for each of the following attributes R.

Name	Attributes	Rationale	M_{Gain}
R_6	TL $= 345KV$	The transmission lines ranked by DEs are of $345KV$.	17.44×
R_7	DS $\geq 138KV$	In most analysis DS $\geq 138KV$ cause worse voltage violations.	37.28×
R_8	DS $\geq 138KV$& TL $= 345KV$	Using combination of $345KV$ TLs results to fail its associated $DS \geq 138KV$ and results simulation to not converge.	21.15×

Table 6.4: **Attributes for case-study and DIHeN performance.**

Analyzing detected nodes. Table. 6.4 shows performance of **DIHeN** using **M2** for $R_6 - R_8$. Note, all the nodes detected by **DIHeN** are transmission lines (TL) which the SMEs selected to use for their analysis (removing different sets of TL). For R_7, we analyze voltage of the DS connected with our selected TL which failed due to S. Among all the DS failed due to S, 35% of them are high voltage $\geq 138KV$. In all the attributes $R_6 - R_8$, **DIHeN** gain on average is 25.3. For R_8, which is a potential attribute for a simulation to not converge, *coverage gain* of **DIHeN** is 21.2×. This indicates that selected nodes through **DIHeN** are suitable as trigger nodes for contingency analysis.

6.5 Lessons Learned

Developing **DIHeN** method by collaborating with the power system experts to aid contingency analysis tools was not an easy task. Following we describe the lessons learned from our experience apart from the observations discussed in Sec. 6.4.

- Contingency analysis are dependent on multiple failure dynamics model that arise

from complex relationships and attributes. No single model can capture all types of dynamics for CA. We choose independent cascade model which are convenient to consider for networks as well as can be mapped as cascading nature of power failure dynamics.

- It is non-trivial to directly use Physics Informed Machine Learning (PGML) concepts for CA. The power flow equation is based on *Kirchoff law* and has no closed-form. Besides, it is hard to incorporate different domain-based constraints as they can only arise during real-time flow analysis. Hence, we leverage a probabilistic model, which is based on some domain-guided properties (**P1**, **P2**) as well as independent of the power flow equation.

- Traditional real-time CA tools consider components' failure quantified by various metrics, e.g., voltage loss, line loss, etc., and can represent multiple disaster scenarios. However, such metrics are only applicable to the power flow equation. Since we cannot consider such metrics, we planned to represent a unit failure for any disaster.

- Properties to capture power failure dynamics are primarily independent of the criteria that power experts consider as trigger components for CA. Criteria $R_1 - R_5$ (Table 6.3) recommended by power experts do not arise from any properties of power failure dynamics. Hence we cannot incorporate such criteria for modeling *Failure Probability* F. Only criteria R_1 is indirectly mapped to *Trans-Volt* model for u_{ij} computation, since they complement both the domain guided property **P2** as well as experts' choice for contingency analysis.

6.6 Discussion

We proposed a complementary network-based framework **DIHeN** for power system CA to identify a set of contingencies ('trigger nodes') using a heterogeneous network. For the first time, we provide a capability within power system inter-dependency analysis that identifies critical nodes within an inter-dependent network not using network connectivity (similar to baseline model IC_1), instead use the knowledge provided by the subject matter experts. We first construct a heterogeneous network using large-scale real power system US data for national and regional levels. **DIHeN** provides a good set of nodes that cause significant failures in the network and are critical for CA. We consistently outperform the baselines in both national and regional level networks. Qualitatively, we find a set of trigger nodes (25.38× coverage gain) that has similar attributes as the experts used for their CA simulations. Our results contain 21% components (number of components among all the selected nodes by **DIHeN**) that have the similar attributes ($R_6 - R_8$ in Table 6.4) which the SMEs consider as a potential contributor to a highly critical scenario. **DIHeN** is based on carefully designed variants inspired from the real CA simulation tools. Our model is very fast, and all the experiments (apart from the experiment to check M_{Gain} deviation for ablation study) finish within 5-15 minutes. The speed and applicability of results are promising towards real-time use of **DIHeN** in evaluating geographically expansive threats, where high numbers of failures are expected but cannot be predicted or assessed in reasonable time frames nor with any expectation of solution convergence using traditional CA tools. For an ongoing disaster, given the network of the corresponding disaster-affected regions (e.g., hurricane track), **DIHeN** can quickly find the potential triggers as wells as the failed components due to cascade.

There are several interesting directions to further study and guide CA through **DIHeN**. First, the network construction may play an important role in **DIHeN** performance. E.g., while our performance was promising in all the datasets, the gain of our model in *ERCOT* was

not as high as the others. This might be due to the sparsity of the network across critical components (e.g., fewer high voltage nodes). Investigating this aspect across regional grids more in detail would be interesting. Second, for modeling Failure probability F, along with the network, the type of every component and its associated voltage play an important role, which restrict analyzing **DIHeN** on synthetic or any publicly available datasets. Third, although we use multiple criticality criteria for evaluating S, there are some more criteria, such as black start resources, control center etc, that are also deemed useful by the SMEs [11]. However, due to privacy concerns, we are unable to collect such data since they are classified information due to national security. Using such new criticality criteria can be more beneficial for CA and hence is an interesting direction of study. Finally, we find power failure dynamics extremely complex with multiple convoluted relationships among them. Our model for failure probability F is based on only two properties (**P1** and **P2**) and hence such domain knowledge are noisy. Investigating uncertainties involved for each domain knowledge for modeling F and analyzing performance of **DIHeN** will be an interesting research direction. As a part of future extension, we plan to analyze **DIHeN** quantitatively with the ground-truth trigger components collected by real CA simulations. We also aim to integrate our framework with existing US DoE tools to provide trigger nodes in real-time for high-fidelity simulations.

Chapter 7

Conclusion

This thesis shows usefulness of inter-disciplinary research and explores data-science research problems to aid decision-making operations for emergency management experts. Our research direction is twofold. Chapter 3- 4 (Part I) present our proposed spatio-temporal time-series segmentation model and a novel explainable problem for black-box time-series segmentation. Our goal is to provide human-friendly explanations for resource allocation decision-making to SMEs so they can quickly respond and recover fast from the damage during a disaster. Besides showcasing our performance based on retrospective hurricane power failures, we explained in Sec. 3.5 how our model can be adapted to real-time and future disasters. Chapter 5- 6 (Part II), describe our network-based visual analytic tool and a domain-inspired failure model for identifying trigger nodes for power contingency analysis. Our visual analytic tool Urban-Net in Chapter 5 has been integrated with ORNL Eagle-I tool to analyze situation-awareness for different hurricane and tornado tracks. We manifest in Sec. 6.5 how we motivate towards a domain-guided model in contrast to using the existing network-based model to aid high-fidelity simulations in power systems. We plan to integrate our framework with the DoE based project NAERM led by ORNL. Our goal is to aid decision-making for preparedness and mitigation for improving system resiliency.

7.1 Lessons Learned and Contributions

For Chapter 3 to identify disaster phases as time-series segmentation, it is necessary to understand the change of power failure process both in space and time. Hence, we develop a joint learning process to capture both temporal and spatial changes. For Chapter 6 no network-based cascade model can directly capture power failure dynamics. Hence, we motivate towards building a domain-inspired model over a network cascade model. Also, it is hard to incorporate domain-based constraints directly as they arise in real-time power flow analysis. Hence, we choose to represent domain-guided properties as a probabilistic failure model.

7.2 Limitations and Future Work

Limitations: First, our segmentation model in Chapter 3 assumes an underlying network structure based on user-defined relations (e.g., spatial structure) among the time-series. It may fail to provide good quality segmentation for datasets that lack such well-defined relations. Second, our proposed problem RaTSS considers multiple assumptions. Such as, black-box segmentation represents meaningful events. Also, explanations are actionable only in terms of constituent time-series. Finally, Our domain-inspired model mentioned in Chapter 6 is based on the assumption that domain knowledge is non-evolving.

Future Work: Following we discuss possible future research directions.

- **Multi-channel time-series segmentation model**: Our current spatio-temporal segmentation model is currently applicable for multivariate sensor data. An exciting direction is to extend our learning model to provide segmentation for multi-channel sensor data. Each sensor is represented across multiple dimensions as a tensor-like

structure. These multi-channel sensors will be leveraged for identifying disaster phases, e.g., cyber-attacks. Besides, such segmentation can also be applied to identify disease phases from ICU patient data. Each patient information is represented across multiple dimensions.

- **Explanations as group of time-series**: Our current problem RaTSS currently finds rationalizations for constituent time-series. We plan to develop a more complex model for scoring groups of time-series which can also hint if the corresponding event is meaningful.

- **Incorporate domain knowledge for explainable segmentation**: Currently our models for learning explanations (importance weight) for the segmentation are only based on the changes of time-series features. We can bring physics-guided machine learning concepts by incorporating some domain-based constraints, such as power grid violation rules for learning such explanations. We can represent such domain constraints as a vector or matrix like structure.

- **Identify critical nodes for uncertain environment**: Our network-based situation-awareness tool *Distance-centric* and DIHeN method to identify trigger components do not consider uncertainties in the data or environment. Our work is based on national level large scale CIs data, and hence it is hard to validate every CIs connection which currently are based on a geographic distance threshold . In practice, uncertainties can be present in the network. Besides, cascading impact due to initial failures can also be different for multiple disasters. E.g., impact of a initial failure during a tornado can be different from a winter-storm. We can extend our model handling all these uncertainties.

- **Uncertainty quantification for domain-inspired model** : Our domain-inspired

probabilistic failure model in Chapter 6 cannot handle noisy and faulty domain knowledge. Develop a model to quantify uncertainty for such domain knowledge and while identifying critical facilities can also be an interesting future research direction.

- **Identify co-located disruptions for situation-awareness**: We define co-located disruptions as two sets of components at distant locations which failed initially around the same time for two different disasters. E.g., Failures of components due to a cyber-attack may happen around the same time as a hurricane. It is possible to identify scenarios analyzing similar or dissimilar changes in outages. We can extend our current models (Part I and Part II) to determine if failure two components occur for the same or different disaster.

Bibliography

[1] Nhdplus. URL http://www.horizon-systems.com/nhdplus/.

[2] Usgs water data. URL https://waterdata.usgs.gov/nwis/rt.

[3] Critical infrastructure protection decision support system (cipdss). URL http://www.bwbush.io/projects/cipdss.html.

[4] Hifld open data. URL https://hifld-geoplatform.opendata.arcgis.com/.

[5] Boston herald hurricane harvey. http://www.bostonherald.com/news/national/2017/08/harvey_lashes_texas_coast_with_high_wind_torrential_rain.

[6] Us energy information administration. URL https://www.eia.gov/.

[7] Fema phases of emergency management. https://www.hsdl.org/?view&did=488295.

[8] Homeland security infrastructure program (HSIP). https://gii.dhs.gov/HIFLD/hsip-guest.

[9] Landscan. URL https://landscan.ornl.gov/.

[10] Addressing security and reliability concerns of large power transformers, . URL https://www.energy.gov/oe/addressing-security-and-reliability-concerns-large-power-transformers.

[11] Bes cyber system categorization, . URL https://www.nerc.com/pa/Stand/Reliability%20Standards/CIP-002-5.1a.pdf.

[12] U.S. electric system is made up of interconnections and balancing authorities - Today in Energy - U.S. Energy Information Administration (EIA), . URL https://www.eia.gov/todayinenergy/detail.php?id=27152.

[13] Homeland Infrastructure Foundation-Level Data (HIFLD), . URL https://hifld-geoplatform.opendata.arcgis.com/.

[14] Psse – transmission planning and analysis psse power system simulation and modeling software, . URL https://new.siemens.com/global/en/products/energy/energy-automation-and-smart-grid/pss-software/pss-e.html.

[15] Code repository. URL https://github.com/anikat1.

[16] Urban climate adaptation tool (urban-cat), . URL https://udi.ornl.gov/content/urban-climate-adaptation-tool-urban-cat.

[17] Urban-met: Modeling urban energy savings scenarios using earth system microclimate and urban morphology, . URL https://studylib.net/doc/11896703/urban-met--modeling-urban-energy-savings-scenarios-using.

[18] Cmu graphics lab motion capture database. http://mocap.cs.cmu.edu, 2014.

[19] Severe weather event review for saturday august 26 2017. http://www.spc.noaa.gov/exper/archive/event.php?date=20170826, 2017.

[20] Major storm system impacting holiday travel through friday. https://www.weather.gov/crp/hurricane_harvey, 2017.

[21] Georgia power working to restore service to 161,000 in dekalb. https://www.dekalbcountyga.gov/news/georgia-power-working-restore-service-161000-dekalb, 2017.

[22] Georgia power working to restore service to 161,000 in dekalb. https://www.dekalbcountyga.gov/news/georgia-power-working-restore-service-161000-dekalb, 2017.

[23] Flooding hits texas towns devastated by harvey. https://www.nytimes.com/2019/09/19/us/houston-beaumont-flooding-imelda.html, 2019.

[24] M Allen, S Fernandez, O Omitaomu, and K Walker. Application of hybrid geo-spatially granular fragility curves to improve power outage predictions. *J Geogr Nat Disast*, 4 (127):2167–0587, 2014.

[25] Samaneh Aminikhanghahi and Diane J Cook. A survey of methods for time series change point detection. *Knowledge and information systems*, 51(2):339–367, 2017.

[26] Sergio Arianos, Ettore Bompard, Anna Carbone, and Fei Xue. Power grid vulnerability: A complex network approach. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 19(1):013119, 2009.

[27] Francis Bach, Rodolphe Jenatton, Julien Mairal, Guillaume Obozinski, et al. Convex optimization with sparsity-inducing norms. *Optimization for Machine Learning*, 5: 19–53, 2011.

[28] Joydeep Banerjee, Arun Das, and Arunabha Sen. A survey of interdependency models for critical infrastructure networks. *arXiv preprint arXiv:1702.05407*, 2017.

[29] Oresti Banos, Mate Attila Toth, Miguel Damas, Hector Pomares, and Ignacio Rojas. Dealing with the effects of sensor displacement in wearable activity recognition. *Sensors*, 14(6):9995–10023, 2014.

[30] Alan M Barker, Eva B Freer, Olufemi A Omitaomu, Steven J Fernandez, Supriya Chinthavali, and Jeffrey B Kodysh. Automating natural disaster impact analysis: An open resource to visually estimate a hurricane's impact on the electric grid. In *2013 Proceedings of IEEE Southeastcon*, pages 1–3. IEEE, 2013.

[31] Richard H. Bartels and George W Stewart. Solution of the matrix equation ax+ xb= c [f4]. *Communications of the ACM*, 15(9):820–826, 1972.

[32] Durell Bouchard. Automated time series segmentation for human motion analysis. *Center for Human Modeling and Simulation,University of Pennsylvania: Pennsylvania, PA, USA*, 2006.

[33] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[34] Sergey V Buldyrev, Roni Parshani, Gerald Paul, H Eugene Stanley, and Shlomo Havlin. Catastrophic cascade of failures in interdependent networks. *Nature*, 464(7291):1025, 2010.

[35] Nadia Burkart and Marco F Huber. A survey on the explainability of supervised machine learning. *Journal of Artificial Intelligence Research*, 70:245–317, 2021.

[36] Liang Chang and Zhigang Wu. Performance and reliability of electrical power grids under cascading failures. *International Journal of Electrical Power & Energy Systems*, 33(8):1410–1419, 2011.

[37] Chen Chen, Jingrui He, Nadya Bliss, and Hanghang Tong. On the connectivity of multi-layered networks: Models, measures and optimal control. In *2015 IEEE International Conference on Data Mining*, pages 715–720. IEEE, 2015.

[38] Liangzhe Chen, Xinfeng Xu, Sangkeun Lee, Sisi Duan, Alfonso G Tarditi, Supriya Chinthavali, and B Aditya Prakash. Hotspots: Failure cascades on heterogeneous critical infrastructure networks. In *Proceedings of the CIKM*, pages 1599–1607, 2017.

[39] Liangzhe Chen, Xinfeng Xu, Sangkeun Lee, Sisi Duan, Alfonso G Tarditi, Supriya Chinthavali, and B Aditya Prakash. Hotspots: Failure cascades on heterogeneous critical infrastructure networks. In *CIKM*, pages 1599–1607. ACM, 2017.

[40] Liangzhe Chen, Sorour E Amiri, and B Aditya Prakash. Automatic segmentation of data sequences. AAAI, 2018.

[41] Qiming Chen and James D McCalley. Identifying high risk n-k contingencies for online security assessment. *IEEE Transactions on Power Systems*, 20(2):823–834, 2005.

[42] Wei Chen, Yajun Wang, and Siyu Yang. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 199–208. ACM, 2009.

[43] Michael M Danziger and Albert-László Barabási. Recovery coupling in multilayer networks. *arXiv preprint arXiv:2011.04623*, 2020.

[44] Pravallika Devineni, Bill Kay, Hao Lu, Anika Tabassum, Supriya Chintavali, and Sangkeun Matt Lee. Toward quantifying vulnerabilities in critical infrastructure systems. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 2884–2890. IEEE, 2020.

[45] Mary T Dzindolet, Scott A Peterson, Regina A Pomranky, Linda G Pierce, and Hall P Beck. The role of trust in automation reliance. *International journal of human-computer studies*, 58(6):697–718, 2003.

[46] Eagle. Eagle-i. https://eagle-i.doe.gov/, 2012.

[47] Upol Ehsan, Brent Harrison, Larry Chan, and Mark O Riedl. Rationalization: A neural machine translation approach to generating natural language explanations. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 81–87. ACM, 2018.

[48] Hafsa El Hafyani, Karine Zeitouni, Yehia Taher, and Mohammad Abboud. Leveraging change point detection for activity transition mining in the context of environmental crowdsensing. *Proceedings of 9th International Workshop on Urban Computing ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020.

[49] David C Elizondo, J de La Ree, Arun G Phadke, and Stan Horowitz. Hidden failures in protection systems and their impact on wide-area disturbances. In *2001 IEEE Power Engineering Society Winter Meeting. Conference Proceedings (Cat. No. 01CH37194)*, volume 2, pages 710–714. IEEE, 2001.

[50] Rozhin Eskandarpour and Amin Khodaei. Machine learning based power grid outage prediction in response to extreme events. *IEEE Transactions on Power Systems*, 32 (4):3315–3316, 2017.

[51] Zoubin Ghahramani. Probabilistic machine learning and artificial intelligence. *Nature*, 521(7553):452, 2015.

[52] Shaghayegh Gharghabi, Yifei Ding, Chin-Chia Michael Yeh, Kaveh Kamgar, Liudmila Ulanova, and Eamonn Keogh. Matrix profile viii: Domain agnostic online semantic segmentation at superhuman performance levels. In *ICDM*, pages 117–126. IEEE, 2017.

[53] Gene Golub, Stephen Nash, and Charles Van Loan. A hessenberg-schur method for the problem ax+ xb= c. *IEEE Transactions on Automatic Control*, 24(6):909–913, 1979.

[54] Hengdao Guo, Ciyan Zheng, Herbert Ho-Ching Iu, and Tyrone Fernando. A critical review of cascading failure analysis and modeling of power system. *Renewable and Sustainable Energy Reviews*, 80:9–22, 2017.

[55] David Hallac, Sagar Vare, Stephen Boyd, and Jure Leskovec. Toeplitz inverse covariance-based clustering of multivariate time series data. In *KDD*, pages 215–223. ACM, 2017.

[56] Paul Hines, Karthikeyan Balasubramaniam, and Eduardo Cotilla Sanchez. Cascading failures in power grids. *IEEE Potentials*, 28(5), 2009.

[57] Paul DH Hines, Ian Dobson, and Pooya Rezaei. Cascading power outages propagate locally in an influence graph that is not the actual grid topology. *IEEE Transactions on Power Systems*, 32(2):958–967, 2017.

[58] Åke J Holmgren. Using graph models to analyze the vulnerability of electric power networks. *Risk analysis*, 26(4):955–969, 2006.

[59] Chao Huang, Xian Wu, and Dong Wang. Crowdsourcing-based urban anomaly prediction system for smart cities. In *Proceedings of the 25th ACM international on conference on information and knowledge management*, pages 1969–1972. ACM, 2016.

[60] Zhenyu Huang, Yousu Chen, and Jarek Nieplocha. Massive contingency analysis with high performance computing. In *2009 IEEE power & energy society general meeting*, pages 1–8. IEEE, 2009.

[61] Saachi Jain, David Hallac, Rok Sosic, and Jure Leskovec. Casc: Context-aware segmentation and clustering for motif discovery in noisy time series data. *arXiv preprint arXiv:1809.01819*, 2018.

[62] Mostafa Jazaeri, Mehdi Farzinfar, and Farzad Razavi. Evaluation of the impacts of relay coordination on power system reliability. *International Transactions on Electrical Energy Systems*, 25(12):3408–3421, 2015.

[63] Stan M Kaplan. Electric power transmission: background and policy issues. Library of Congress, Congressional Research Service, 2009.

[64] Isak Karlsson, Jonathan Rebane, Panagiotis Papapetrou, and Aristides Gionis. Explainable time series tweaking via irreversible and reversible temporal transformations. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 207–216. IEEE, 2018.

[65] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *KDD*, pages 137–146. ACM, 2003.

[66] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. *arXiv preprint arXiv:1703.04730*, 2017.

[67] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8), 2009.

[68] Shengjie Lai, Nick W Ruktanonchai, Liangcai Zhou, Olivia Prosper, Wei Luo, Jessica R Floyd, Amy Wesolowski, Mauricio Santillana, Chi Zhang, Xiangjun Du, et al. Effect of non-pharmaceutical interventions to contain covid-19 in china. 2020.

[69] Sangkeun Lee, Liangzhe Chen, Sisi Duan, Supriya Chinthavali, Mallikarjun Shankar, and B Aditya Prakash. Urban-net: A network-based infrastructure monitoring and analysis system for emergency management and public safety. In *Proceedings of the IEEE BigData*, 2016.

[70] Sangkeun Lee, Supriya Chinthavali, Sisi Duan, and Mallikarjun Shankar. Utilizing semantic big data for realizing a national-scale infrastructure vulnerability analysis system. In *Proceedings of the International Workshop on Semantic Big Data*, 2016.

[71] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 420–429. ACM, 2007.

[72] Lei Li, James McCann, Nancy S Pollard, and Christos Faloutsos. Dynammo: Mining and summarization of coevolving sequences with missing values. In *KDD*, pages 507–516. ACM, 2009.

[73] Jiongqian Liang, Peter Jacobs, and Srinivasan Parthasarathy. Human-guided flood mapping: From experts to the crowd. In *Companion of the The Web Conference 2018 on The Web Conference 2018*, pages 291–298. International World Wide Web Conferences Steering Committee, 2018.

[74] Jun Liu and Jieping Ye. Efficient l1/lq norm regularization. *arXiv preprint arXiv:1009.4766*, 2010.

[75] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4765–4774, 2017.

[76] Yasuko Matsubara, Yasushi Sakurai, and Christos Faloutsos. Autoplait: Automatic mining of co-evolving time sequences. In *SIGMOD*, pages 193–204. ACM, 2014.

[77] Sina Mohseni, Niloofar Zarei, and Eric D Ragan. A multidisciplinary survey and framework for design and evaluation of explainable ai systems. *arXiv preprint arXiv:1811.11839*, 2018.

[78] Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2020.

[79] Robert Moss, Elham Naghizade, Martin Tomko, and Nicholas Geard. What can urban mobility data reveal about the spatial distribution of infection in a single city? *BMC public health*, 19(1):1–16, 2019.

[80] Abdullah Mueen and Eamonn Keogh. Online discovery and maintenance of time series motifs. KDD '10, pages 1089–1098, 2010.

[81] Nikhil Muralidhar, Chen Wang, Nathan Self, Marjan Momtazpour, Kiyoshi Nakayama, Ratnesh Sharma, and Naren Ramakrishnan. illiad: Intelligent invariant and anomaly detection in cyber-physical systems. *ACM TIST*, 9(3):35, 2018.

[82] Nikhil Muralidhar, Anika Tabassum, Liangzhe Chen, Supriya Chinthavali, Naren Ramakrishnan, and B Aditya Prakash. Cut-n-reveal: Time series segmentations with explanations. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11 (5):1–26, 2020.

[83] Xia Ning and George Karypis. Slim: Sparse linear methods for top-n recommender systems. In *ICDM*, pages 497–506. IEEE, 2011.

[84] Penjani Hopkins Nyimbili and Turan Erden. Spatial decision support systems (sdss) and software applications for earthquake disaster management with special reference to turkey. *Natural Hazards*, 90(3):1485–1507, 2018.

[85] Gabriele Oliva, Annunziata Esposito Amideo, Stefano Starita, Roberto Setola, and Maria Paola Scaparra. Aggregating centrality rankings: A novel approach to detect critical infrastructure vulnerabilities. In *International Conference on Critical Information Infrastructures Security*, pages 57–68. Springer, 2019.

[86] Min Ouyang. Review on modeling and simulation of interdependent critical infrastructure systems. *Reliability engineering & System safety*, 121:43–60, 2014.

[87] Forough Poursabzi-Sangdeh, Daniel G Goldstein, Jake M Hofman, Jennifer Wortman Vaughan, and Hanna Wallach. Manipulating and measuring model interpretability. *arXiv preprint arXiv:1802.07810*, 2018.

[88] Naren Ramakrishnan, Satish Tadepalli, Layne T Watson, Richard F Helm, Marco Antoniotti, and Bud Mishra. Reverse engineering dynamic temporal models of biological processes and their relationships. *Proceedings of the National Academy of Sciences*, 107(28):12511–12516, 2010.

[89] Jaxk Reeves, Jien Chen, Xiaolan L Wang, Robert Lund, and Qi Qi Lu. A review and comparison of changepoint detection techniques for climate data. *Journal of Applied Meteorology and Climatology*, 46(6):900–915, 2007.

[90] Andreas Reinhardt, Paul Baumann, Daniel Burgstahler, Matthias Hollick, Hristo Chonov, Marc Werner, and Ralf Steinmetz. On the accuracy of appliance identification based on distributed load metering data. In *SustainIT (2012)*, pages 1–9. IEEE, 2012.

[91] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *KDD*, pages 1135–1144. ACM, 2016.

[92] Guy Rosman, Mikhail Volkov, Dan Feldman, John W Fisher III, and Daniela Rus. Coresets for k-segmentation of streaming data. In *NIPS*, pages 559–567, 2014.

[93] Eric Ruggieri. A bayesian approach to detecting change points in climatic records. *International Journal of Climatology*, 33(2):520–528, 2013.

[94] Nick Warren Ruktanonchai, JR Floyd, Shengjie Lai, Corrine Warren Ruktanonchai, Adam Sadilek, Pedro Rente-Lourenco, Xue Ben, Alessandra Carioli, Joshua Gwinn, JE Steele, et al. Assessing the impact of coordinated covid-19 exit strategies across europe. *Science*, 2020.

[95] Allou Samé and Gérard Govaert. Online Time Series Segmentation Using Temporal Mixture Models and Bayesian Model Selection. *ICMLA*, 1:602–605, 2012.

[96] Rishu Saxena, Layne T Watson, Randolph H Wynne, Evan B Brooks, Valerie A Thomas, Yang Zhiqiang, and Robert E Kennedy. Towards a polyalgorithm for land use change detection. *ISPRS journal of photogrammetry and remote sensing*, 144:217–234, 2018.

[97] Pavel Senin and Sergey Malinchik. Sax-vsm: Interpretable time series classification using sax and vector space model. In *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, pages 1175–1180. IEEE, 2013.

[98] Jiajia Song, Eduardo Cotilla-Sanchez, Goodarz Ghanavati, and Paul DH Hines. Dynamic modeling of cascading failure in power systems. *IEEE Transactions on Power Systems*, 31(3):2085–2095, 2015.

[99] Anika Tabassum, Supriya Chinthavali, Liangzhe Chen, and Aditya Prakash. Data mining critical infrastructure systems: Models and tools. *IEEE Intelligent Informatics Bulletin*, 19(2), 2018.

[100] Yuliya Tarabalka, Guillaume Charpiat, Ludovic Brucker, and Bjoern H Menze. Spatio-temporal video segmentation with shape growth or shrinkage constraint. *IEEE Transactions on Image Processing*, 23(9):3829–3840, 2014.

[101] Swapna Thorve, Mandy L Wilson, Bryan L Lewis, Samarth Swarup, Anil Kumar S

Vullikanti, and Madhav V Marathe. Epiviewer: an epidemiological application for exploring time series data. *BMC bioinformatics*, 19(1):449, 2018.

[102] Stephen Tierney, Junbin Gao, and Yi Guo. Subspace clustering for sequential data. In *IEEE CVPR*, pages 1019–1026, 2014.

[103] Alfredo Vellido, José David Martín-Guerrero, and Paulo JG Lisboa. Making machine learning models interpretable. In *ESANN*, volume 12, pages 163–172. Citeseer, 2012.

[104] Srinivasan Venkatramanan, Jiangzhuo Chen, Sandeep Gupta, Bryan Lewis, Madhav Marathe, Henning Mortveit, and Anil Vullikanti. Spatio-temporal optimization of seasonal vaccination using a metapopulation model of influenza. In *2017 IEEE International Conference on Healthcare Informatics (ICHI)*, pages 134–143. IEEE, 2017.

[105] Trivik Verma, Wendy Ellens, and Robert E Kooij. Context-independent centrality measures underestimate the vulnerability of power grids. *International Journal of Critical Infrastructures 7*, 11(1):62–81, 2015.

[106] Win Wah, Sourav Das, Arul Earnest, Leo Kang Yang Lim, Cynthia Bin Eng Chee, Alex Richard Cook, Yee Tang Wang, Khin Mar Kyi Win, Marcus Eng Hock Ong, and Li Yang Hsu. Time series analysis of demographic and temporal trends of tuberculosis in singapore. *BMC Public Health*, 14(1):1121, 2014.

[107] Haihua Yang and Shi An. Critical nodes identification in complex networks. *Symmetry*, 12(1):123, 2020.

[108] Scott L Zeger, Rafael Irizarry, and Roger D Peng. On time series analysis of public health and biomedical data. *Annu. Rev. Public Health*, 27:57–79, 2006.

[109] Xin Zhao and Pao-Shin Chu. Bayesian multiple changepoint analysis of hurricane

activity in the eastern north pacific: A markov chain monte carlo approach. *Journal of climate*, 19(4):564–578, 2006.

[110] Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. Urban computing: concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(3):1–55, 2014.

[111] Fan Zuo, Jingxing Wang, Jingqin Gao, Kaan Ozbay, Xuegang Jeff Ban, Yubin Shen, Hong Yang, and Shri Iyer. An interactive data visualization and analytics tool to evaluate mobility and sociability trends during covid-19. *arXiv preprint arXiv:2006.14882*, 2020.