# STATISTICAL AND MACHINE LEARNING MODELS FOR CRITICAL INFRASTRUCTURE RESILIENCE

BY

JACOB HEGLUND

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Aerospace Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2020

Urbana, Illinois

Advisor:

Research Assistant Professor Huy T. Tran

# Abstract

This thesis presents a data-driven approach to improving predictions of critical infrastructure behaviors. In our first approach, we explore novel data sources and time series modeling techniques to model disaster impacts on power systems through the case study of Hurricane Sandy as it impacted the state of New York. We find a correlation between Twitter data and load forecast errors, suggesting that Twitter data may provide value towards predicting impacts of disasters on infrastructure systems. Based on these findings, we then develop time series forecasting methods to predict the NYISO power system behaviors at the zonal level, utilizing Twitter and load forecast data as model inputs.

In our second approach, we develop a novel, graph-based formulation of the British rail network to model the nonlinear cascading delays on the rail network. Using this formulation, we then develop machine learning approaches to predict delays in the rail network. Through experiments on real-world rail data, we find that the selected architecture provides more accurate predictions than other models due to its ability to capture both spatial and temporal dimensions of the data.

*For my family, friends, and the scientific community at large.*
*"If I have seen further it is by standing on the shoulders of Giants."*
*- Sir Isaac Newton*

# Acknowledgments

This work would not have been possible without the support of those around me. Thank you to my advisor, Professor Huy T. Tran, for giving me the opportunity to work on research that is both interesting and valuable. It is only with his guidance that I have been able to come this far, and his guidance will continue to help me grow as I take my next steps as a PhD student in the Tran Research Group.

Thank you to the others in the Tran Research Group including Nick Chase, Neale Van Stralen, Ha Kewon, Walker Dimon, and Seung Kim. While we often were working on other problems, it was great to hear feedback on my own work and to discuss the interesting problems you were working on as well.

Thanks to the undergrads I've had the opportunity to mentor. While there are too many to mention by name, it is one of life's greatest joys to mentor such hard-working people.

Thank you to my family, to my parents, my sister, my step-mother, and the rest of my family. You have always provided love and support, and there is no way to repay you for the fascination in science you have always fostered in me.

Thank you to my closest friends, both on and off campus. No matter what direction life takes us, we will always remain a part of each other's lives.

Finally, thank you to my fiancée for being with me for all the ups and down that grad school brings. Every day brings new challenges, and I can't wait to face them head on together!

# Contents

# List of Tables

# List of Figures

# Chapter 1 : Introduction

## 1.1 Motivation

Critical Infrastructures (CIs) form the foundation upon which modern societies and economies are built. Due to their ubiquity, disruption of normal operation of CIs can have severe primary effects, such as loss of life, property damage, and economic losses, as well as secondary effects, such as mass displacement of residents, widespread health consequences, and decreased quality of life for those affected. Throughout history, both human-made incidents and natural disasters have caused disruptions to CIs, and in some instances, CI disruptions could be described as "disasters" in their own right[1]. For example, the Flint water crisis (2014-present) is a disruption of the city's water infrastructure which has directly been linked with increased death rates among children and other vulnerable groups due to brain and nervous system damage from lead in the water supply [2]. The 2001 World Trade Center attacks caused disruptions to the transportation infrastructure and power systems, among other CIs, in New York City, some of which were observed to last several months following the attacks [3]. Other disruptions or events causing disruptions to CIs include the 2003 Northeast Blackout [4], Hurricanes Sandy, Harvey, and Irene, among many others.

Due to threats from state- and non-state actors, as well as the increased severity and frequency of severe weather events, developing CI resilience is a issue of utmost importance for ensuring both national security and the common good [5]. The need for CI resilience has become more widely recognized in recent years, and improving CI resilience is a key strategy toward reducing the overall impact of CI disruptions. It is not a question of "if" CI disrup-

[1] While the term "disruption" is the standard term used in the CI literature, this term does not effectively communicate the severity of CI disruptions. To get a better feel for the severity of these events, please feel free to replace "disruption" with "disaster", "calamity", or "crisis" at any point during the reading of this thesis.

tions will occur in the future, but of "when" they will occur, "how severe" the disruptions will be, and "how many" lives will be forever changed.

One promising approach to improving CI resilience is the development of models to quantify the behaviors of CIs. Models can facilitate understanding and predictions of the future behaviors of CIs, which can increase information available to first responders and CI operators, and reduce the overall impact of CI disruptions. However, increased connectivity and interdependency within and between CIs can lead to complicated interactions between CI elements. Some previous efforts to model CI resilience rely on manual analysis of post-disaster data sources [3], [6], [7]. However, these approaches do not develop models to predict CI behaviors during future disruptions. Other approaches use statistical models [8] or network-based models [9] to gain insight into CI behaviours, but lack validation on real-world data sources. These works suggest that novel approaches need to be developed and validated on real-world data sources in order to effectively provide real-time insights into, and predictions of, the behaviors of CIs.

Data-driven approaches have lead to unprecedented advances in areas such as computer vision and natural language processing. Innovations in machine learning, particularly in deep learning, have facilitated the development of predictive models that outperform traditional statistical approaches by leveraging large datasets. At the same time, increased integration of sensors into infrastructure systems provides an opportunity collect large-scale data on the daily operation of CIs. Combining these, we can develop novel data-driven methods for improving CI resilience. In our approach, we leverage recent advances in the availability of large datasets as well as deep learning models to develop real-time models for prediction of CI performance toward the goal of improving CI resilience.

## 1.2 Contribution

In this thesis, we approach the problem of improving CI resilience by examining methods that improve predictions of CI behavior. We accomplish this via two primary approaches. Firstly, we approach this problem by leveraging new data sources. We explore the feasibility of using social media data to improve predictions of the disruptions of power infrastructure as it is affected by

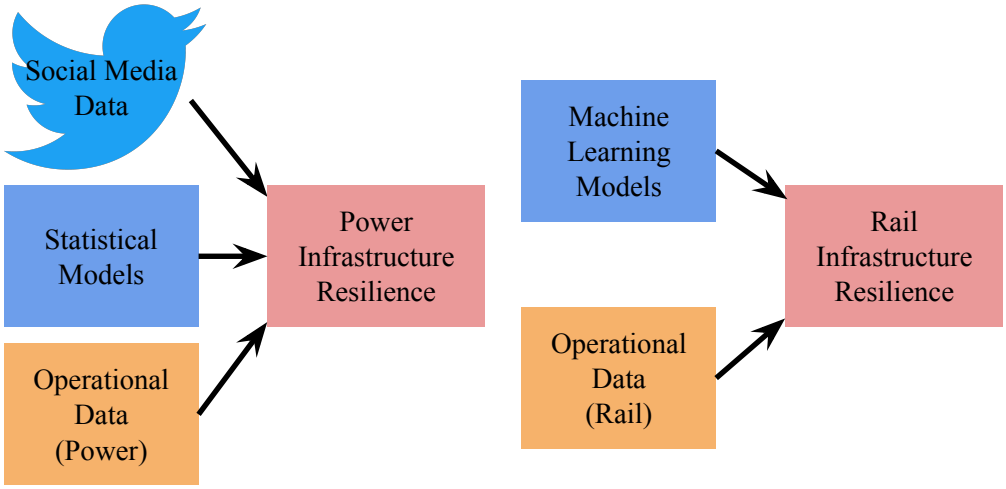


Figure 1.1: An overview of the approaches used in this thesis. (Left) In Chapter 3 we discuss the use of social media, statistical models, and operational power data toward the goal of improving power infrastructure resilience. (Right) In Chapter 4 we discuss the use of machine learning models and operational rail data toward the goal of improving rail infrastructure resilience.

extreme weather. In particular, we develop predictive models that leverage social media data to predict the behavior of the power infrastructure of New York state before and during Hurricane Sandy. We find that the inclusion of social media data improves the prediction accuracy of an optimized time series model, supporting future investigations into the use of social media as a sensor for CI resilience. Secondly, we approach this problem by developing models to predict the behaviors if CIs. We explore a class of predictive models known as graph neural networks (GNNs) for the problem of predicting delays in rail traffic infrastructure. Through novel graph-based formulations, we use data-driven models to capture interconnections and network effects in this CI toward the goal of predicting delays. We find GNN methods give more accurate predictions than classical statistical models for delays in the rail transportation context. We visualize the general approaches of each chapter of this thesis in Figure 1.1.

## 1.3 Outline

In Chapter 2, we begin with a background of the relevant infrastructure as well as methods applied to study these infrastructure. In Chapter 3, we explore and develop social media as a novel source of data for improving predictions in the power infrastructure context. In Chapter 4, we present a novel graph-based formulation of the delay prediction problem in the rail context which we use to develop statistical and machine learning models to improve delay predictions. Finally in Chapter 5 we summarize our findings and detail future extensions of this work.

# Chapter 2 : Background

## 2.1 Critical Infrastructure Resilience

A 1997 report by the President's Commission on Critical Infrastructure Protection identifies the following systems as critical infrastructures: transportation, electric power, water supply systems, information and communications services, banking and finance, government services, and oil production and storage [5]. The partial or complete disruption of any one of these systems can lead to disastrous outcomes, and the threat of disruption has led to research into how to define, measure, and improve resilience in infrastructure systems. Most definitions of resilience share common themes with the definition provided by the National Academy of Sciences [10], which defines resilience as follows:

> *Resilience is the ability to prepare and plan for, absorb, recover from, and more successfully adapt to adverse events.*

Based on this definition and others like it, researchers have developed methods for measuring resilience, many of which are discussed in recent surveys [11, 12, 13, 14, 15]. While these metrics are valuable tools for understanding the high-level resilience of a given system, they often provide little guidance for real-time decision support in the time immediately preceding, during, and after a disaster. This limitation in the field of CI resilience has been captured by a recent push for "resilience analytics" [16] which aims to develop data-driven methods for enabling descriptive, predictive, and prescriptive modeling of infrastructures to enhance their resilience. In this thesis, we focus on power and rail infrastructures to develop and validate new models for predicting CI behaviors toward the goal of improving CI resilience to disruptive events.

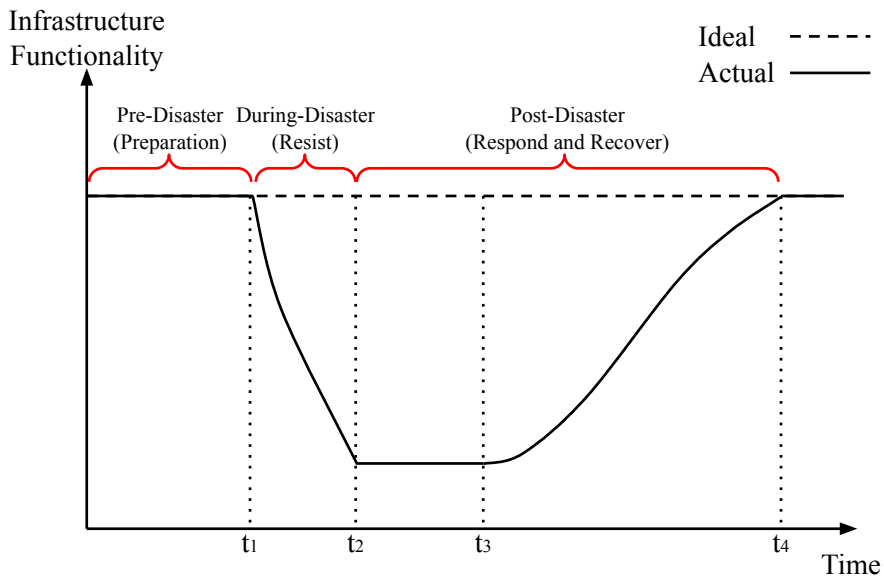


Figure 2.1: The timeline of CI functionality during a disruption event. $t_1$ marks the beginning of an event which causes a CI disruption, $t_2$ marks the end of the disruptive event, $t_3$ marks the beginning of the recovery efforts, and $t_4$ marks the full recovery. A highly resilient CI would not experience disruption in functionality during the event which would reduce the functionality of a less-resilient CI.

### 2.1.1 Power Infrastructure

Several surveys [15, 17, 18] have identified key strategies for improving resilience in power grids. One key strategy identified is the development of models to predict power outages and other abnormal behaviors in the power grid. [19] compares several regression methods to predict outages during Hurricane Ivan, and validates their method on two other hurricanes. [20] develops a set of statistical models to both preallocate resources before hurricanes and manage resources after hurricanes to improve restoration of power infrastructure. [8] utilizes Monte-Carlo simulations to produce models of power infrastructure damage due to tornadoes. Finally, [21] uses SVM to model the state of power grid components to predict outages under extreme weather.

Another key strategy to improving power grid resilience is increasing situational awareness of grid operators [22], which can be accomplished through increasing the capabilities of grid sensors. Working toward this goal, several papers have examined novel sources of data, namely social media as a dis-

tributed sensor for improving operator situational awareness. [23] develops a metric to better-understand the connection between the number of social media posts in an geographic area during a disaster and the disruptions caused by the disaster. [24] proposes the use of Twitter data as a distributed sensor to improve predictions of power use, and compares a topic modeling approach to predicting power outages to a weather-based model of power outage prediction. [25] performs a correlation analysis of Twitter activity around the time of Hurricane Sandy to estimate economic damage caused by the hurricane. Finally, [26] proposes the use of Twitter as a distributed sensor to improve predictions of power outages during Hurricane Sandy.

One drawback of these methods is that the models don't explicitly take into account the time series nature of data, and may have difficulty adapting to quickly-evolving situations as is often the case with disasters. The modeling techniques also only model outages, while modeling more specific aspects of data could provide more specific predictions and better-inform operational decisions. Finally, while the social media-focused papers demonstrate the feasibility of utilizing Twitter data for improved power grid predictions, they fail to provide robustly validated models that could provide real-time insights to grid operators.

### 2.1.2 Transportation Infrastructure

In the context of transportation infrastructure, one measure of resilience is the deviation from the day's schedule. Resilient transportation infrastructure will experience fewer overall deviations from the day's schedule compared to less resilient transportation infrastructure. By determining the overall delay state of a particular hub of transportation, such as a rail station, we may draw conclusions about the overall resilience of the transportation infrastructure.

#### Rail Transportation

Many mathematical and statistical models have been established in the literature to predict delays and understand its propagation throughout a railway network. [27] proposed the modeling of a railway system as a linear system in max-plus algebra with zero-order dynamics that represent delay propaga-

tion. [28] proposed different distributions for eleven delay types ranging from bad weather to fault in tracks and utilized maximum likelihood estimation (MLE) and the Kolmogorov-Smirnov test (K-S) to evaluate each proposed distribution. [29] developed an algorithm that analyzes real-world data to identify delays and cascading delays defined according to various conditions, returning a network of dynamic delay propagation. [30] utilized the closed episode algorithm to mine cascading delays through a Belgian railway network focusing on specific reference points throughout the network. Finally, [31] developed three regression models to predict the delay of trains at stations, each of which introduced different assumptions about the current delay and previous delays.

With recent advancements in the field of machine learning, many have explored the use of machine learning models to predict delays and understand the mechanism of delay propagation. [32] proposed several regression models including random forests and feed-forward neural networks toward the problem of estimating time of arrival in United States rail networks. [33] proposed the adoption of recurrent neural networks (RNNs) alongside Irish Rail System data with labeled delay types to perform a one station step delay forecast. [34] produced a train delay prediction system, forecasting the time taken for the train to reach its next checkpoint considering its scheduled journey up to terminal station. [35] utilized weather records, historical delays, and train schedules to identify delay-inducing factors, and utilized gradient-boosted regression trees to predict delays along the Beijing Guanzhou line. Finally, [36] explored weather data for delay prediction in rail networks through the use of kernel-based methods, extreme learning machines, and ensemble methods.

While these efforts demonstrate the value of machine learning for predicting delays in railway systems, previous approaches have typically focused on small or single-line railways, and have not yet been validated on larger or more complex rail networks. At the same time, previous approaches have not explicitly considered the connections between elements in the rail network, limiting their capabilities in capturing the delay-propagation dynamics in the railway network.

## 2.2 Modeling Methods

There are many approaches to predicting the future, and both statistical and machine learning approaches give mathematically-grounded methods for making predictions. Both approaches seek to develop a generalized predictive model for some dataset. This model is typically denoted $f(x;\theta)$, where $x$ is the input data and $\theta$ is the set of model parameters. In this thesis, we utilize both statistical and machine learning approaches to developing predictive models.

### 2.2.1 Statistical Models

Statistical methods allow us to draw conclusions from data in a principled manner. In our analysis, we use basic but powerful statistical techniques to support the use of social media as a distributed sensor to improve predictions of CI behaviors.

#### Correlation Analysis

In a basic statistical setting, correlation and covariance are two common measures used to describe the relationship between sets of data. Correlation is used to measure the linear relationship between between two variables. Similarly, covariance is used to measure the strength of the correlation between two sets of data. These measures of relationship have analogues in a time series setting, namely cross-correlation and cross-covariance, which are used to measure the similarity between two series of data. Using cross-correlation and cross-covariance, a straightforward correlation analysis can be used to identify the predictive power of one variable as it relates to another variable.

#### Establishing Causality

While correlation analysis can establish a relationship between features of the time series, it does not establish a causal relationship. A causal relationship between two variables can typically only be established in highly controlled settings where individual factors are varied to test a given hypothesis. Since our data were generated and collected under uncontrolled conditions, we

cannot directly establish a strong causal relationship between our variables. However, for the purposes of improving predictions, there is another type of causality we consider, namely Granger Causality. Granger Causality is a concept based on prediction and according to the definition, is useful for establishing whether one variable will improve predictions of future values of another variable in a multi-regression setting.

### Time Series Models

While many statistical methods work well in non-time series settings, time series settings require a different set of methods. For the time series prediction problem, we consider the input to our prediction model as $T$ realizations of a time series random variable, $X$. One naive approach to predicting the future value of this variable is to simply predict the next value to be the present value of the variable, that is $x_{t+1} = x_t$. Another approach involves taking the average over input time steps as the prediction, such that $x_{t+1} = \frac{1}{T}(x_{t-T} + x_{t-T+1} + ... + x_{t-1} + x_t)$. These are just a few of the simplest approaches to time series forecasting, and more sophisticated models exist to capture temporal relationships between linear and nonlinear temporal random variables, as well as including exogenous variables to improve predictions.

## 2.2.2 Machine Learning Models

Machine learning is the study of algorithms that automatically optimize themselves through exposure to data. Due to the generalized approach, machine learning methods have seen success in many different fields including computer vision and natural language processing. There are several paradigms of machine learning, including unsupervised, supervised, and reinforcement learning. In this thesis, we primarily utilize supervised learning, which involves the development of predictive models which are optimized using labelled, input-output pairs of data. We use the supervised learning approach as a contrast to more-typically utilized statistical models.

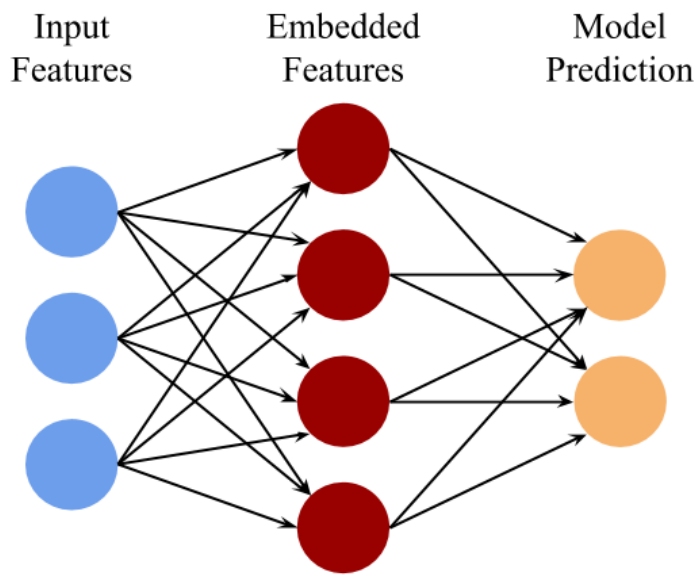


Figure 2.2: Artificial neural network architecture with an input layer, a single fully-connected hidden layer with four nodes, and an output layer. The activation functions between layers of the network are not shown.

## Artificial Neural Networks

Artificial neural networks (ANNs) are a connectionist model of human cognition that are often used in supervised learning contexts. These models and their extensions have been the subject of intensive research in recent years. ANNs are optimized, or trained, through process known as backpropagation. This process calculates some measure of error, or loss, between the model prediction $f(x;\theta) \equiv \hat{y}$ and the ground truth $y$ with respect to some loss function. As an example, the mean-squared error loss function Equation (2.1) is commonly used for regression tasks, or prediction of a continuous value. Please see Figure 2.2 for a visual depiction of an ANN model.

$$L(\hat{y};\theta) = \frac{1}{N}\sum_{i=1}^{N} ||f(x_i;\theta) - y_i||^2 \tag{2.1}$$

One reason for the increase in research interest in this class of models is due to advances in specialized computational capabilities, such as improved graphical processing units (GPUs) and the development of tensor processing units (TPUs), which have facilitated the training of these models. These models are universal function approximators [37, 38], meaning they can theoretically approximate any function. In practice, these models typically re-

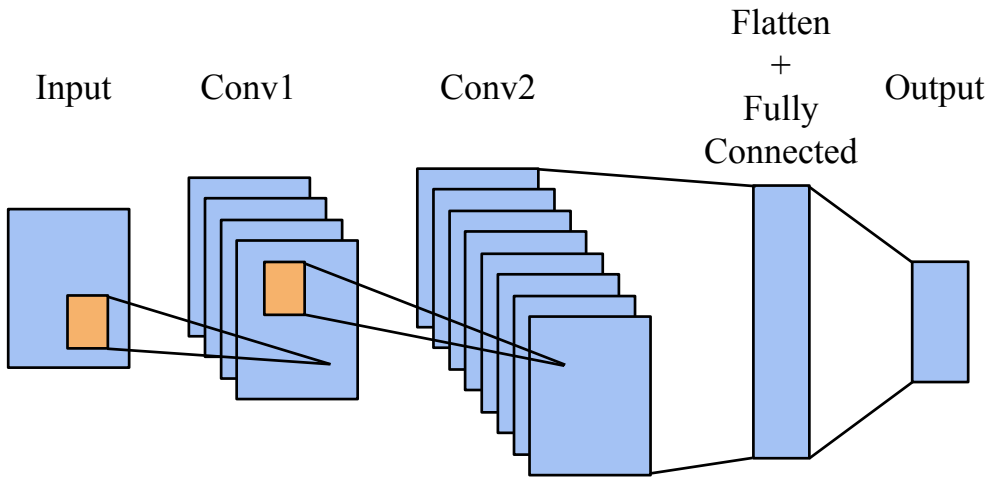


Figure 2.3: A basic convolutional neural network designed to extract features from image data. The convolution operation captures local spatial dependencies between pixels in the image and extracts embedded features. These models also maintain a relatively low number of model parameters.

quire large amounts of data to perform well.

The input and output of an ANN are typically denoted as $X \in R^{F_{\text{in}}}$ and $Y \in R^{F_{\text{out}}}$ respectively. One may notice that the input to an ANN is a vector of features, which limits their use in domains with spatial or temporal dimensions of data. To overcome this limitation, convolutional neural networks (CNNs) [39, 40] have been developed to capture spatial aspects of data, such as the local connections found in images. See Figure 2.3 for a visualization of a basic CNN architecture. Similarly, methods such as recurrent neural networks (RNNs) [41, 42] and temporal convolutions [43] have been developed for time series data. Both CNNs and RNNs have been shown to be universal function approximators [44, 45], which means they retain many desirable theoretical properties of standard ANNs.

## Graph Neural Networks

Graph neural networks (GNNs) are an extension of ANNs which operate on irregularly-structured or non-Euclidean data which may be represented as a graph. A graph $G(V, E)$ is uniquely defined by the set of nodes, or vertices,

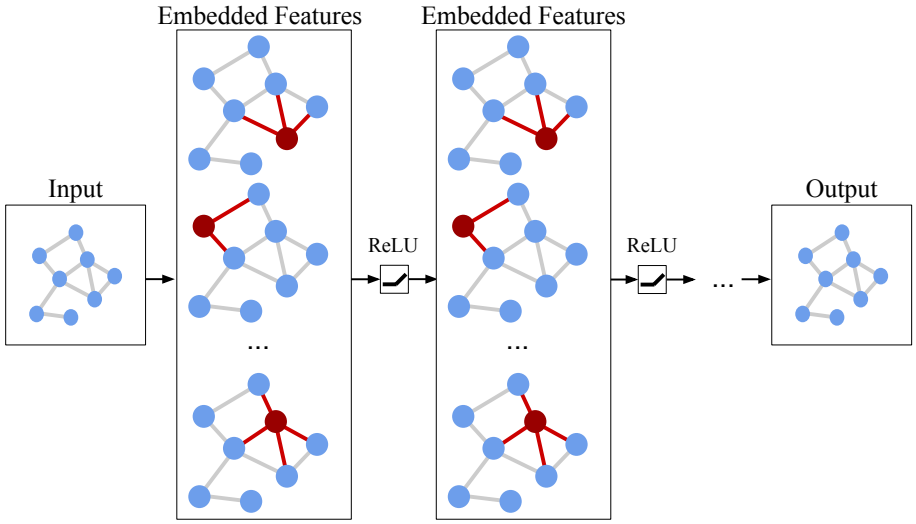


Figure 2.4: A generic GNN architecture implementing repeated layers of graph convolutions and ReLU activations. Red lines are used to denote information propagation between nodes. The graph convolution creates a set of embedded node-wise features used for prediction.

and edges. The edges between nodes are typically defined by an adjacency matrix $A \in R^{|V|\times|V|}$, where $|V|$ is the number of nodes. We may also consider node-wise features to get an attributed graph G(V, E, X) where $X \in R^{|V|\times F}$ is a matrix of node-wise features, where $F$ is the number of features per node. Note that we may also consider edge-wise features, but they are not considered as part of this work.

GNN methods typically leverage convolution, or aggregation, operations to capture spatial relationships within the data. GNNs extend methods developed for CNNs to be applicable for graph-structured data by specifically leveraging graph convolutions to propagate information between neighboring nodes and embed provided graph features into a latent space. This embedding provides a high-level representation of the data, which is then typically combined with a multi-layer perceptron or softmax output layer to provide node-level predictions. A generic GNN model is shown in Figure 2.4. For surveys of specific GNN architectures and their applications, please see [46] and [47].

While the case of proving or disproving the universal function approximator property for ANNs was fairly straightforward, it is not as straightforward with GNNs. Recent papers such as [48, 49] have discussed how the power of a GNN architecture, or the ability of a GNN to distinguish between graphs with

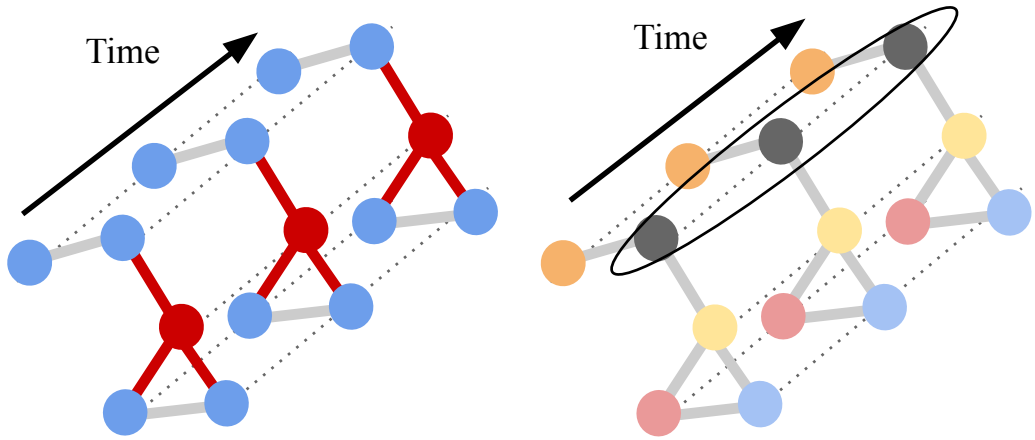


Figure 2.5: (Left) The spatial convolution operation for a single node visualized on a spatio-temporal graph. This operation is used to aggregate information between nodes along a single time step of the graph. (Right) The 1-dimensional temporal convolution operation for a single node is visualized on a spatio-temporal graph. This operation is used aggregate information across multiple time steps of a node's history. In this image, the same node is depicted as the same color across multiple time steps.

different structure, depends on the choice of graph convolution. Additionally, GNN properties such as invariance and equivariance become important for characterizing the properties of GNNs. Proving or disproving the power of GNNs is central to characterizing the properties of specific GNN architectures and determining whether they are an appropriate choice for a given task. This topic is very much a new area of research, and for a survey of work in this area please see [50].

## Spatio-Temporal Graph Neural Networks

Spatio-temporal GNNs (STGNNs) extend standard GNNs to domains where data has both spatial and temporal aspects. Real-world phenomena often exhibit both spatial connections, which are well-modeled by GNNs, and temporal aspects, and STGNNs provide a novel method for modeling such phenomena. In the context of STGNNs, the input values for each node are allowed to vary over time such that the input tensor $X \in R^{|V| \times X \times t}$, where t is the

number of time steps given as input, and $t \leq T$ where $T$ is the total number of time steps in the dataset. As discussed in Section 2.2.2, there are several methods to extend neural network architectures to a temporal domain including RNNs and temporal convolutions. RNN approaches suffer from issues such as vanishing gradient and computationally expensive backpropagation. Meanwhile, convolution-based approaches have advantages such as stable gradients and fast backpropagation facilitated by parallel computations, and are the operation of choice for modern STGNN architectures.

There have been several innovations in STGNNs in recent years. [51] proposed one of the first STGNN architectures with Graph Convolutional Recurrent Network (GCRN), which utilized a ChebNet [52] to capture spatial dependencies in conjunction with LSTM to capture temporal aspects of data. [53] extended this work to a ground-traffic prediction setting with a diffusion convolutions operation as well as a GRU. [1] further extended their architecture this work to include temporal convolutions, and [54] further built off of this by introducing an attention-based spatial convolution as part of their architecture.

Current STGNN architectures yield high performance on certain tasks, but there is still room for improvement. In particular, current architectures have not thoroughly explored the effectiveness of different spatial aggregation operations. An exploration of different spatial aggregation schemes could lead to reduced the number of model parameters and computational requirements, as well as improved theoretical guarantees of performance. Additionally, the set of tasks that STGNNs are evaluated on is fairly limited at this point in time. Current architectures are typically evaluated on ground-traffic prediction problems, but comparing results across architectures is not always simple since different papers tend to use different datasets. At the same time, it is not clear that an architecture which performs well on one tasks will perform well on another task with different parameters, such as a smaller number of nodes. As a result, a set of benchmarks for STGNNS should be developed to foster easier comparisons of models across different valuable data settings.

# Chapter 3 : Power Infrastructure and Social Sensing

## 3.1 Introduction

Toward the goal of improving situational awareness of CI operators, we focus on potential benefits of social media as a distributed sensor of CI behaviors, which we refer to as *social sensing*. When geocoded and timestamped to an appropriate resolution, social media has the potential to supplement existing data sources available to authorities and provide increased levels of situational awareness. Realizing the full potential of this data source may save on costs of physical sensors, provide coverage in areas difficult to reach with existing sensors, provide backup sensing, and provide cross-validation of other data sources. To demonstrate the feasibility social sensing, we develop social media data as a feature for statistical models to provide real-time predictions of CI behaviors.

Given the widespread of Hurricane Sandy on New York's power system, we focus on the time period immediately before, during, and after Hurricane Sandy as a case study for this study. Hurricane Sandy travelled from the Caribbean Sea to the Northeastern US along the Atlantic Ocean from October 24th to October 30th in 2012. Sandy made landfall in New Jersey sometime in the early morning of October 30th. The hurricane is estimated to have caused over 200 fatalities along its path and economic losses of between 78 and 97 billion US dollars in the US [55]. Estimated losses from impacts to power systems alone are on the order of 16.3 billion dollars. Over 20 million people are estimated to have been affected by power outages, including those in highly populated areas like Manhattan in New York City. Recovery of these power services significantly varied from region to region, with only 84% of the system restored one week after landfall.

## 3.2 Methods

### 3.2.1 Power Infrastructure Data

The dataset we use to study power infrastructure is publicly available through The New York Independent System Operator (NYISO), the primary operator of the electrical grid in New York. For our study, we consider the integrated load and load forecast data, which are both provided hourly across the 11 NYISO load zones. These zones are are visualized in Figure 3.6. We consider data for October and November 2012 for this study, focusing on the "day of load forecast" and "integrated load" fields to characterize the behaviors of the New York power system.

We focus on analyzing abnormality in infrastructure behaviors to capture impacts of disasters on CIs. We characterize abnormality in power systems by calculating the forecast load error, $\epsilon_i(t)$, for load zone $i$ during hour $t$ as,

$$\epsilon_i(t) = \frac{\hat{L}_i(t) - L_i(t)}{L_i(t)} \tag{3.1}$$

where $\hat{L}_i(t)$ and $L_i(t)$ are the day of load forecast and actual integrated load for load zone $i$ during hour $t$, respectively.

### 3.2.2 Social Media Data

As part of our exploration of novel data sources, we examine the use of social media, particularly Twitter, to serve as a distributed sensor. We focus on collecting and preprocessing Twitter data to enable $n$-gram analysis of tweets. Three methods for collecting Twitter data include paid services, open source data sets, and use of an API for collecting a live stream of tweets. For our study of power infrastructure, we use an open-source data set of 6.5 million geotagged tweets from Washington DC, Connecticut, Delaware, Maine, Maryland, New Jersey, New York, North Carolina, Ohio, Pennsylvania, Rhode Island, South Carolina, Virginia, and West Virginia, posted between October 22 and November 02 of 2012 [56]. We only consider geotagged tweets due to our desire to enable high-resolution spatiotemporal modeling. Due to data restrictions in Twitter's Terms of Service, the data set only contains tweet ID numbers, not the tweets themselves. Therefore,

we hydrated the data set using Hydrator (a publicly available tweet hydration tool) on March 14, 2018 to extract 4.8 million of the original tweets in full JSON format. We store these tweets in a MongoDB database. Missing tweets were likely deleted between posting of the original tweet ID data set and our date of hydration.

We perform the following steps to preprocess a tweet string for $n$-gram analysis. We first tokenize the "full_text" or "text" field to identify contiguous sequences of $n$ words within the tweet. We then convert all words to be lowercase due to inconsistent capitalization within Twitter data, and remove common words (i.e., stopwords such as "the") and punctuation to reduce noise in the data. The hashtag symbol "#" is removed as punctuation, and thus unigram tokens from a tweet's string and its hashtags are not differentiated from each other. We apply stemming to map words to their word stem. For example, words like "damage", "damaging", and "damaged" that have the same stem but different suffixes are mapped to the same stem "damag." We use these tokens to create filtered data sets that only include tweets containing keywords or hashtags related to the disaster or infrastructures of interest. We geolocate tweets to map to geographic regions of interest, at relevant spatial resolutions. We preprocess the Twitter data used in our case study with Python's Natural Language Toolkit [57], using the "english" stopword corpus and Porter stemming [58].

Tweets are geolocated to identify which infrastructure geographic region they were tweeted from within. We use New York Independent System Operator (NYISO) load zones as regions for this study, with load zone geotagging performed using NYISO geojson files available from ArcGIS Online. Two keyword and hashtag searches are then performed to create tweet data sets focused on Hurricane Sandy or power systems. The sets of keywords and hashtags for each search are shown in Table 3.1, along with the resulting count of related tweets. Keywords and hashtags for Sandy-related tweets are based on those used in [23]. Keywords and hashtags for power-related tweets were selected by the authors.

We focus on analyzing normalized tweet counts to track changes in social media networks during disaster events. We calculate normalized Sandy- and power-related tweet counts as the number of related tweets posted within a NYISO load zone during a given hour, relative to the total number of posted tweets within that same spatiotemporal grouping. For example, the

Table 3.1: Keywords and Hashtags for Twitter Searches.

| | Keywords and Hashtags | Count |
|---|---|---|
| Sandy-related tweets | hurricane, sandy, storm, #sandy, #hurricanesandy, #njsandy, #masandy, #stormde, #sandydc, #rigov | 30,290 |
| Power-related tweets | blackout, electricity, grid, light, nyiso, outage, power, service, #blackout, #electricity, #outage, #power, #poweroutage | 12,129 |

normalized count of Sandy-related tweets, $S_i(t)$, for load zone $i$ during hour $t$ is calculated as,

$$S_i(t) = \frac{s_i(t)}{n_i(t)} \tag{3.2}$$

where $s_i(t)$ is the number of Sandy-related tweets posted within load zone $i$ during hour $t$, and $n_i(t)$ is the total number of tweets posted within load zone $i$ during hour $t$. A similar ratio is calculated for the normalized count of power-related tweets, $P_i(t)$.

### 3.2.3 Proposed Models

We utilize several statistical methods and models to investigate the viability of social media data to act as a feature to augment predictions of CI behaviors. We formulate the problem as a time series prediction problem where features are sampled on an hourly basis. We use these features to first establish a correlative relationship between the CI behavior and social media data. Once this relationship is identified, we establish weak causality using a Granger Causality test. Finally, we develop a predictive model for CI behavior using the ARMA-family of statistical models, which are standard time series modeling methods. We depict this modeling approach in Figure 3.1.

#### Correlation Analysis

We analyze time series representations of the data to understand temporal trends, relationships among features, and the potential predictive power of various features for CI behaviors. We use sample auto-correlations and the

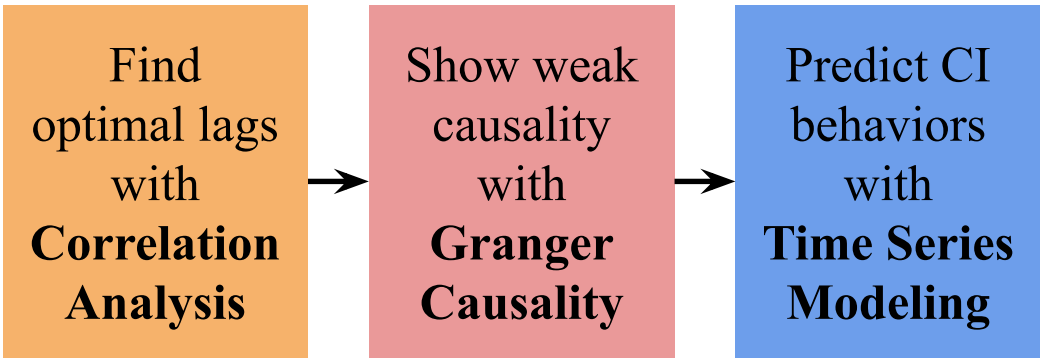


Figure 3.1: We use three primary steps in our analysis to establish social media as a viable distributed sensor. First, we find optimal time lags using correlation analysis, next we show weak causality using Granger Causality, and finally we predict the CI behavior using time series modeling techniques.

Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test to assess stationarity of the processes generating our time series, and apply transformations or differencing as needed to approximate weak-stationarity [59]. Once we have achieved (or nearly achieved) weak-stationarity in our time series, we calculate sample cross-correlations among the series to assess their potential predictive power for one another.

Based on [60], we calculate the sample auto-correlation function, $\hat{\rho}(h)$, and sample auto-covariance function, $\hat{\gamma}(h)$, for time series $x$ as

$$\hat{\rho}(h) = \frac{\hat{\gamma}(h)}{\hat{\gamma}(0)} \tag{3.3}$$

$$\hat{\gamma}(h) = \frac{1}{n}\sum_{t=1}^{n-h}(x_{t+h} - \bar{x})(x_t - \bar{x}) \tag{3.4}$$

where $h = 0, 1, \ldots, n-1$ is the lag, $n$ is the length of the full time series, $x_t$ is the time series value at time $t$, and $\bar{x} = n^{-1}\sum_t x$ is the sample mean of the time series. We calculate the sample cross-correlation function, $\hat{\rho}_{xy}(h)$,

and sample cross-covariance function, $\hat{\gamma}_{xy}(h)$, between series $x$ and $y$ as

$$\hat{\rho}_{xy}(h) = \frac{\hat{\gamma}_{xy}(h)}{\sqrt{\hat{\gamma}_x(0)\hat{\gamma}_y(0)}} \tag{3.5}$$

$$\hat{\gamma}_{xy}(h) = \frac{1}{n}\sum_{t=1}^{n-h}(x_{t+h} - \bar{x})(y_t - \bar{y}) \tag{3.6}$$

We then identify the optimal time lag, $h^*$, on Twitter data that maximizes sample cross-correlation with load forecast error. We use these optimal time tags as potential lags to apply to Twitter data for forecasting impacts to infrastructure systems.

## Granger Causality

While sample cross-correlation is useful for understanding correlative relationships, we ideally aim to understand causative relationships. However, our data were collected through uncontrolled methods, resulting in potentially many external, unobserved variables that affect the data. We therefore cannot show strong causality between our modeled processes. Instead, we aim for Granger causality, a weaker version of causality. Granger causality develops two auto-correlative models on an observed variable $y$, defined as,

$$y_t = \beta_0 + \beta_1 y_{t-1} + \beta_2 y_{t-2} + ... + \beta_T y_{t-T} \tag{3.7}$$

$$y_t = \beta_0 + \beta_1 y_{t-1} + \beta_2 y_{t-2} + ... + \beta_T y_{t-T} + \beta_{T+1} x_{i,t-1} + ... + \beta_{2T} x_{i,t-T} \tag{3.8}$$

where $T$ is the maximum lag included in the auto-correlative model and $x_i$ is the tested exogenous variable. To determine Granger Causality, an $F$-test is then performed on both of the model errors. If Equation (3.8) performs better than Equation (3.7), then the inclusion of lagged values of exogenous variable $x_i$ provides information that is useful in predicting endogenous variable $y$. In our case, the endogenous variable is load forecast error, while the exogenous variable is Twitter data. In our study of the power infrastructure, we analyzed Granger Causality for maximum lags within the set $T \in \{1, 2, 3, \ldots, 30\}$; we chose $T = 30$ as the largest maximum lag because we

did not observe Granger causality above this lag for our data. We note that while Granger causality can help identify the value of one time series towards modeling another, it does not establish strong causality or the direction of causality.

## Time Series Models

There are several classes of models typically used for time series data, ranging from statistical models such as naive, moving average (MA), and autoregressive (AR) models, to deep learning approaches such as recurrent neural networks (RNNs). We focus on the class of autoregressive moving average (ARMA) models for our analysis because, unlike deep learning models, they do not require large amounts of data to give accurate, short-term forecasts. ARMA models are also more sophisticated than naive or average forecast models, and work well in situations where the data may be noisy.

An ARMA model is represented as a sum of an MA($q$) model and an AR($p$) model. An MA model of order $q$ is defined as,

$$y_t = \mu + \delta_t + \theta_1 \delta_{t-1} + ... + \theta_q \delta_q \tag{3.9}$$

where $\mu$ is the expected value of the available time series data points at the time of forecast, $\delta_t$ is the value of a white-noise random variable at time $t$, and $\theta_t$ are model parameters typically chosen using maximum likelihood estimation (MLE). An AR model of order $p$ is defined as,

$$y_t = \mu + \epsilon_t + \phi_1 y_{t-1} + \phi_2 y_{t-2} + ... + \phi_p y_{t-p} \tag{3.10}$$

where $\mu$ is the expected value of the available time series data points at the time of forecast, $\epsilon_t$ is the value of a white-noise random variable at time $t$, $\phi_t$ are model parameters typically chosen using MLE, and $y_{t-n}$ are past values of the time series.

There are also several variations of ARMA models, including autoregressive integrated moving average (ARIMA) and autoregressive integrated moving average with exogenous variable (ARIMAX) models. ARIMA models build on ARMA by forecasting on differenced time series data. The order of differencing is treated as a parameter of the model, and the forecast is integrated in discrete-time to yield forecasts on the non-differenced time series. The

integration is performed as,

$$\hat{y}_t = \hat{y}_t^{(d)} + y_{t-d} \tag{3.11}$$

where $\hat{y}_t^{(d)}$ is the forecast on the differenced time series at time $t$, $d$ is the order of differencing, and $\hat{y}_t$ is the forecast on the time series at time $t$. ARIMAX builds further on ARIMA by performing regression on both endogenous and exogenous variables. We focus on ARIMA and ARIMAX models for our study of power infrastructure and the effects of social media as a distributed sensor for improving predictions of CI behaviors.

## 3.3 Results

### 3.3.1 Data Exploration

Our presented results focus on analyzing and modeling relationships between normalized counts of Sandy-related tweets, normalized counts of power-related tweets, and load forecast errors for NYISO load zones (see Figure 3.6 for a visualization of the zones). We consider data from the beginning of October 28th through the end of November 1st to focus on the time period surrounding landfall of Hurricane Sandy in the US.

Figure 3.2 shows how our Twitter and load forecast error features vary during this time period. Most NYISO zones show a peak in Sandy- and power-related tweets during the day before and day of landfall. The N.Y.C. zone shows the most gradual increase and decrease in Sandy-related tweets, as it extends to two days before and after landfall. The increased Twitter activity in this zone is likely due to a combination of N.Y.C.'s coastal location, its large population, and its active Twitter userbase. We also see that peaks in Sandy-related tweets tend to precede those in load forecast errors by several hours, suggesting these tweets may have value towards forecasting future impacts on power systems. Peaks in power-related tweets appear to be closer (temporally) to those in load forecast errors, which is expected as people are most likely to post power-related tweets after their power supply has actually been impacted.

Regarding load forecast errors, we see that LONGIL, MILLWD, HUD VL,

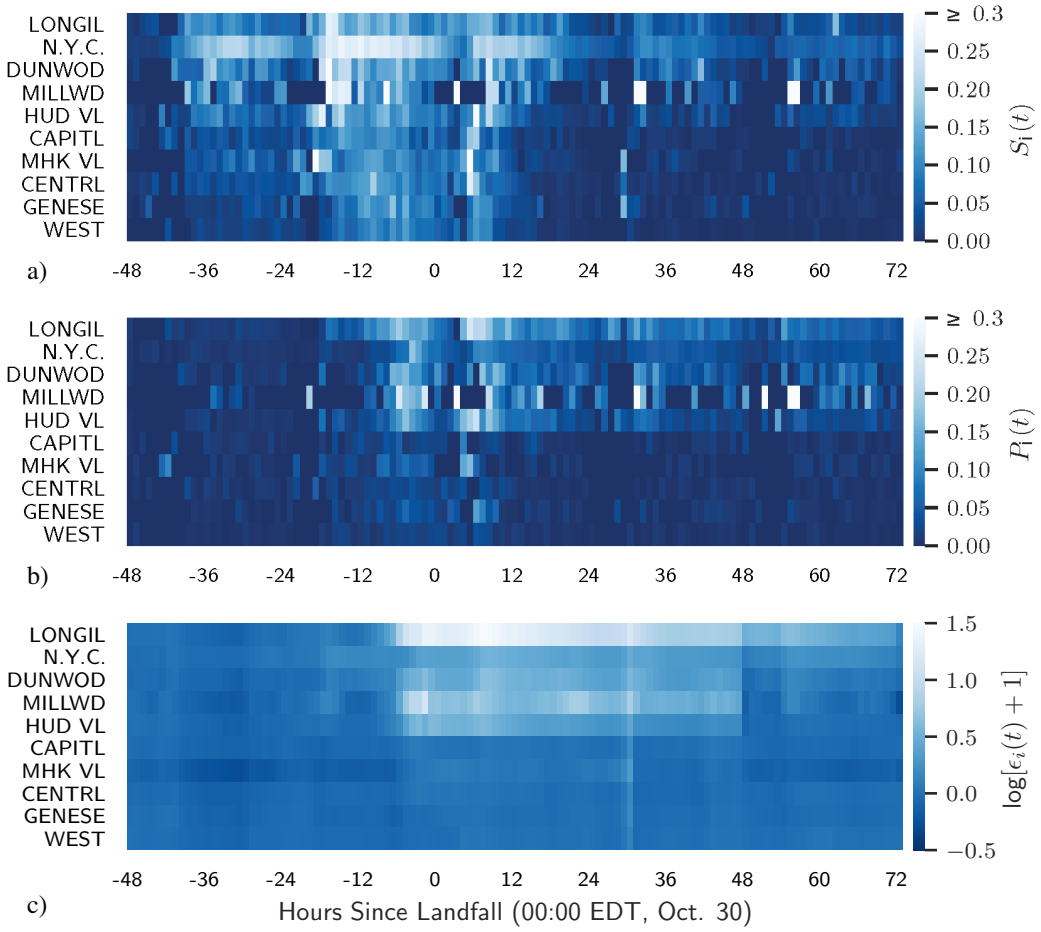


Figure 3.2: Heat maps of Sandy-related tweets ($S_i(t)$), power-related tweets ($P_i(t)$), and load forecast errors ($\log[\epsilon_i(t)+1]$) during Hurricane Sandy for NYISO zones. Load forecast errors are log transformed to better visualize scale differences in the data (with one added to ensure positive values). Data for zone four is not shown due to a lack of relevant tweets.

DUNWOD, and N.Y.C. show the most notable errors, likely due to their proximity to the coast. These results also show that the impact of Sandy on forecast errors for these zones primarily spans the 48-hour period after landfall, suggesting most power systems were recovered within 48 hours of being impacted. Note that forecast errors were defined such that positive errors indicate an over-prediction of load; this situation may occur, for example, when power is unavailable due to damaged infrastructure.

### 3.3.2 Correlation Analysis

Having identified some general trends in our data, we more formally analyze relationships within the data using time series methods. We only analyze time series that have at least 20 data points; this filter removes Twitter data for the NORTH zone from our analysis. Figure 3.3 provides a detailed visualization of Twitter data and load forecast errors for the N.Y.C. and LONGIL zones. We see similar trends among the three series for both zones, again seeing that peaks in Sandy-related tweets appear to precede peaks in power-related tweets and load forecast errors. The data also suggest that, within our time period of interest, the underlying processes generating these series are non-stationary, as the data show a clear trend and some cyclic behaviors. This non-stationarity is problematic for our analysis, as many time series methods assume stationarity in the processes being modeled. Most other zones, particularly those showing the largest increases in load forecast error during Sandy, show similar trends in their time series data and are therefore not shown here.

We address the issue of non-stationarity by applying transformations and differencing to our data. We find that first-differencing with no transformations best approximates weak stationarity in our data, based on visual analysis of sample auto-correlations and KPSS tests. Figure 3.4 shows sample auto-correlations for N.Y.C. after first-differencing. The data are not still not truly stationary, as there are still visually apparent cycles and lags producing auto-correlations of statistical significance. However, application of the KPSS test suggests that all of our considered time series are stationary after first-differencing; i.e., the test finds sufficient evidence to reject the null hypothesis of stationarity (for both trended and constant models). We therefore focus on first-differenced data for our models, being careful to note that the underlying processes are only approximately weakly stationary.

Based on these results, we analyze sample cross-correlations for first-differenced data to further understand relationships between our Twitter features and load forecast errors. Figure 3.5 shows these results for the N.Y.C. zone. Here, we see a large spike in sample cross-correlation between Sandy-related tweets and load forecast error at a lag of $h = -24$ hours (with the lag being applied to Sandy-related tweets). This result provides quantitative evidence for the visual trend of Sandy-related tweets preceding load forecast errors seen in

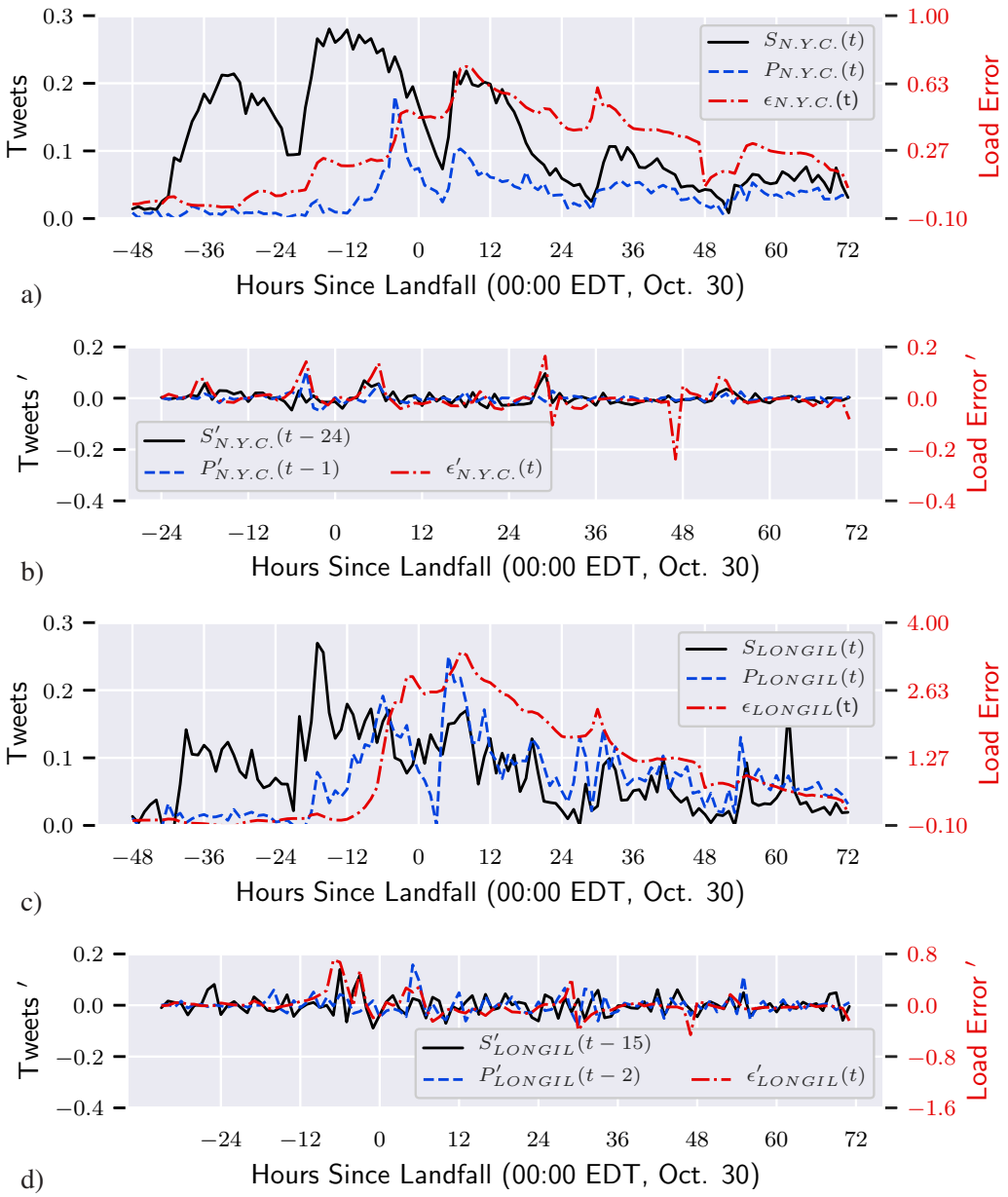


Figure 3.3: Visualization of (a) Sandy-related tweets ($S_{N.Y.C.}(t)$), power-related tweets ($S_{N.Y.C.}(t)$), and load forecast errors ($\epsilon_{N.Y.C.}(t)$) during Hurricane Sandy for N.Y.C.. (b) shows the same data, but with Twitter data lagged to maximize sample cross-correlation with load forecast error. Sandy-related tweets are lagged by $h^* = -24$ hours, power-related tweets are lagged by $h^* = -1$ hour, and all three series are first-differenced. (c) and (d) show analogous data for LONGIL.

Figure 3.2. We also see a large spike in sample cross-correlation between power-related tweets and load forecast errors at a lag of $h = -1$ hours, which again corresponds to the visual trends seen in Figure 3.2. We see similar

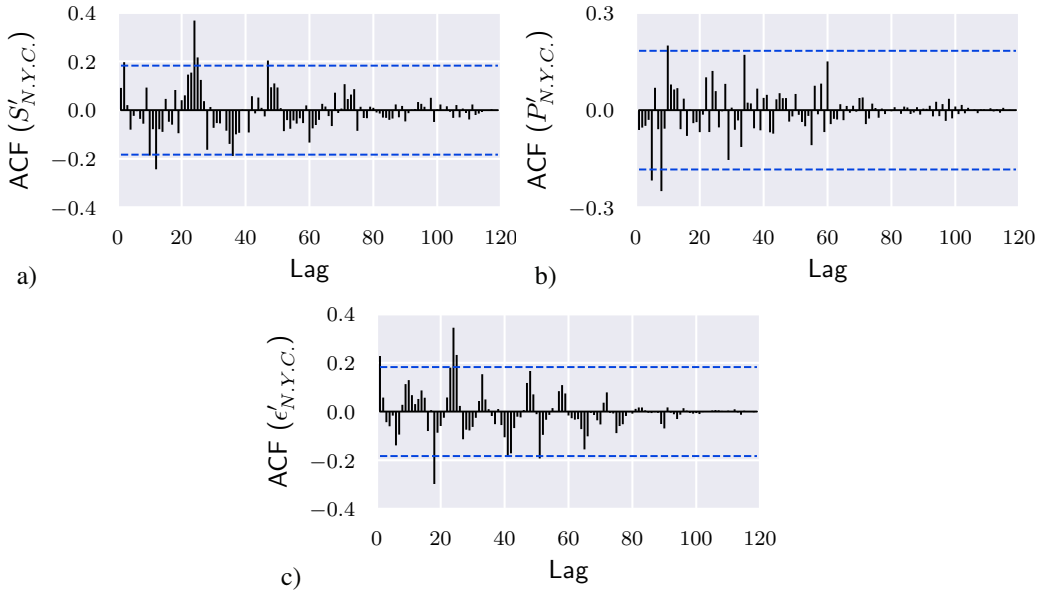


Figure 3.4: Sample auto-correlation function results for first-differenced (a) Sandy-related tweets, (b) power-related tweets, and (c) load forecast error during Hurricane Sandy for N.Y.C.. Dashed lines indicate 95% significance levels, calculated as $\pm 2/\sqrt{(n)}$ where $n$ is the length of the series.

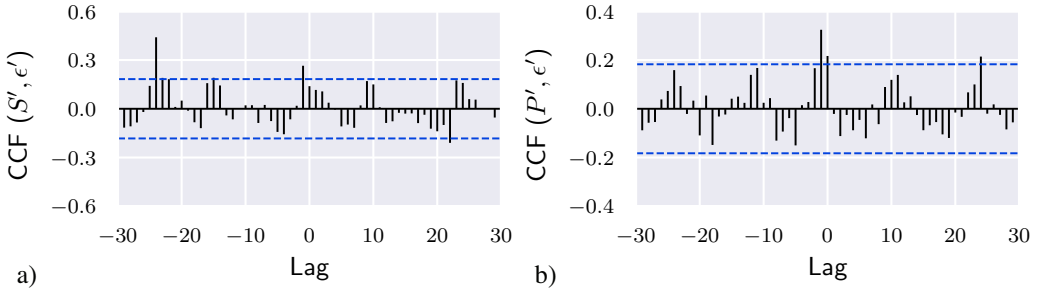


Figure 3.5: Sample cross-correlation function results between (a) Sandy-related tweets and load forecast error and (b) power-related tweets and load forecast error during Hurricane Sandy for N.Y.C.. Sample cross-correlations are calculated using first-differenced data. Dashed lines indicate 95% significance levels, calculated as $\pm 2/\sqrt{(n)}$ where $n$ is the length of the shorter series.

trends for other zones (not pictured here), though the lags at which spikes in cross-correlations occur vary among zones.

Figure 3.6 shows maximum sample cross-correlations for all zones (except for NORTH). We see weak to moderate maximum cross-correlations between Sandy-related tweets and load forecast error for most zones, including the coastal ones that were most strongly impacted by the hurricane. We see

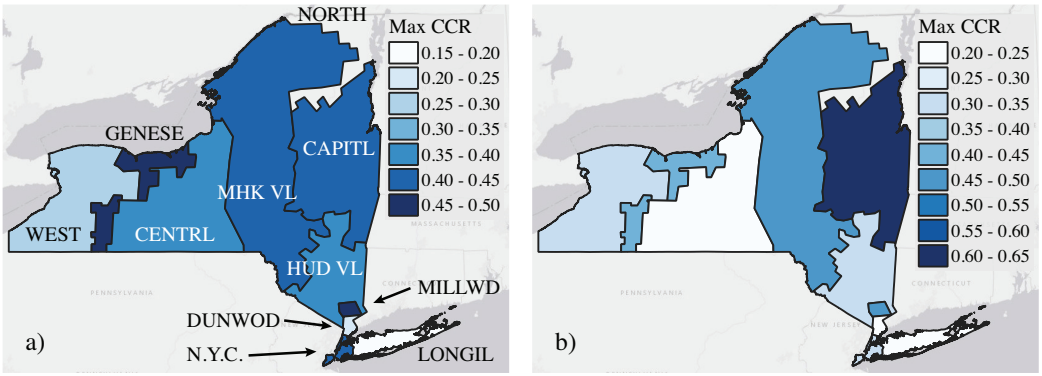


Figure 3.6: Maximum sample cross-correlations between (a) Sandy-related tweets and load forecast error and (b) power-related tweets and load forecast error during Hurricane Sandy for NYISO zones. Sample cross-correlations are calculated using first-differenced data with lags ranging from -30 to 30 hours. Note that data for NORTH is not plotted due to its small sample size.

similar trends in max cross-correlation between power-related tweets and load forecast error. MILLWD in particular shows high max cross-correlations between Twitter data and load forecast errors. Though cross-correlations are weak to moderate, we note that every max cross-correlation value is statistically significant at the 95% level. This result suggests that, given appropriate feature engineering and model development, Twitter data may provide value towards predictive modeling of power systems and other CIs.

### 3.3.3 Granger Causality

To further understand the potential predictive power of Twitter data for load forecast error, we also assess Granger causality for our time series. Applying the method as described in Section 3.3.2 to first-differenced data, we find that data for some zones show Sandy- and power-related tweets to be Granger causal for load forecast errors at a 95% significance level. More specifically, Sandy-related tweets are determined to be Granger causal for all zones except HUD VL, MILLWD, and DUNWOD. Power-related tweets are determined to be Granger causal for all zones except CENTRL, MILLWD, DUNOWD, and LONGIL. Sandy-related tweets tend to show Granger causality around the lags for which the cross-correlation was maximized, but this trend was not observed for the power-related tweets. This result partially supports the

conclusion drawn from general correlation analyses that Twitter data may improve our ability to model infrastructure impacts during disaster events. The inconclusiveness also motivates further research into actual model comparisons with and without Twitter data (as seen in Section 3.3.4), as well research on the topic of feature engineering for social media data in the context of CI forecasting.

### 3.3.4 Modeling

Given the potential value of Twitter data towards modeling load forecast errors, we now describe results implementing such forecast models. We develop models for all zones, but focus our presented results on N.Y.C.. We focus on N.Y.C. because it is densely populated and shows the most Twitter activity among NYISO zones, while also showing impacts to load forecast error during Sandy.

#### Model Comparisons

We first train and compare the accuracies of ARIMA and ARIMAX models for forecasting load errors, to understand effect of including Twitter data as a model feature. We train our ARIMA and ARIMAX models on first-differenced data, using load errors as the endogenous variable with Sandy-related tweets or power-related tweets as the exogenous variable. We lag Sandy-related and power-related tweets by the lag that maximizes sample cross-correlation between those tweets and load errors, based on results described in Section 3.3.2. For example, Sandy-related tweets for N.Y.C. are lagged by $-24$ hours, with power-related tweets for N.Y.C. lagged by $-1$ hour. We use lagged tweets as model features to prevent the need to include a significant amount of previous hourly data. We select $p$ and $q$ parameters to minimize the Akaike information criterion (AIC), and therefore refer to these models as being AIC-optimal. We consider forecasts every three hours during the hurricane, forecasting the load error up to three hours in advance of each of these forecast times. For each forecast time, we retrain ARIMA and ARIMAX models using data from the last 24 hours. We use this continuous retraining approach to utilize the most recent data that would have

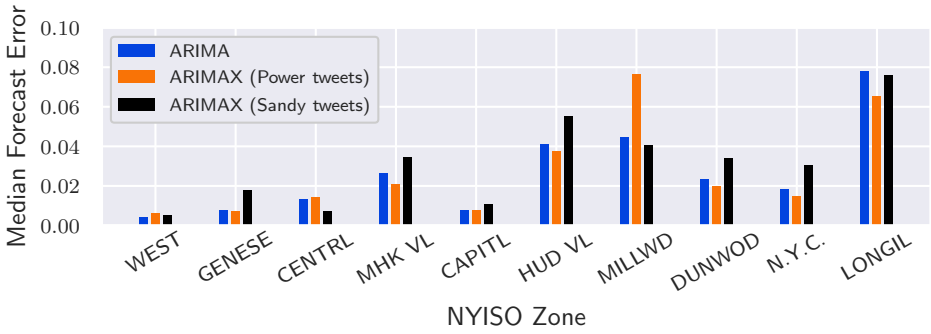


Figure 3.7: Median forecast errors for over all NYISO zones. We omit NYISO zone 4 from the analysis due to insufficient Twitter data.

been available at the time of forecast. We then integrate and back-difference our forecasts to transform them back to their original units, and use these forecasts to calculate model forecast errors relative to the actual observed load errors at our considered forecast times.

Figure 3.7 shows the median forecast errors from this study. We see that errors for ARIMAX models using power-related tweets are lower than ARIMA model errors for several zones, including HUD VL, DUNWOD, N.Y.C., and LONGIL. That is, most of the zones whose load errors were strongly impacted by Sandy, other than MILLWD, showed improved forecasting with power-related tweets used as an exogenous variable. These results demonstrate that including Twitter data as a model feature can, in fact, improve our ability to forecast the impacts of Sandy on the power system within the state of New York. However, we also see that errors for ARIMAX models using Sandy-related tweets are higher than ARIMA models for all zones except CENTRL, MILLWD, and LONGIL. Thus, while Twitter data can improve forecasting errors for power system impacts, one must be careful to ensure relevant features are derived from the data.

## Training Data Optimization

We also examine the effect of varying the amount of training data, $n$, on the performance of ARIMAX models. We use load forecast error as the endogenous variable and lagged power-related tweets as the exogenous variable in these models, with all data being first-differenced. We aim to understand this effect for two primary reasons. One, available data may be sparse for dis-

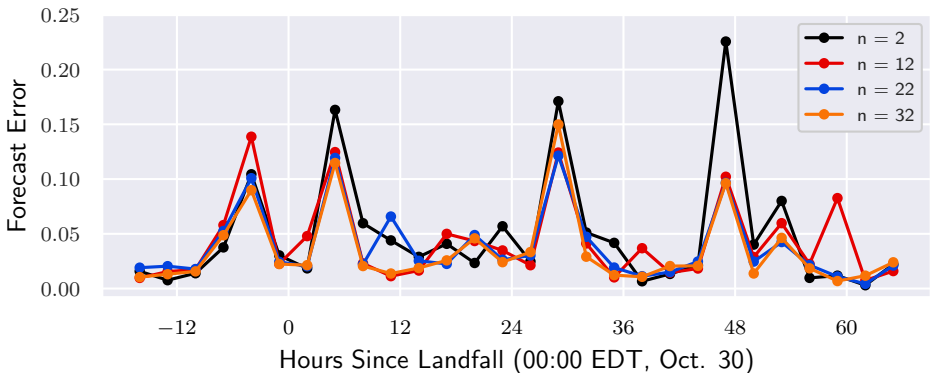


Figure 3.8: The forecast error for each AIC-optimal model generated during the training-data optimization process.

aster events, requiring an understanding of model sensitivity to the amount of available training data. Two, disaster events themselves are highly non-stationary processes, which may result in data becoming less representative of the current process as the disaster unfolds. In this case, only including recent data may be more beneficial than including all available data, even at the cost of training data size. We therefore implement a training scheme that identifies (in hindsight) the optimal training data size $n^*$ with respect to forecast error, and explore how $n^*$ varies over the course of Hurricane Sandy.

Figure 3.8 shows the results of this study. We see that the optimal amount of training data, $n^*$, varies over the course of the hurricane. However, we see no clear relationship between $n^*$ and the point in time at which the forecast is being performed. Furthermore, overall error rates do not significantly change with respect to $n$, suggesting these models are relatively insensitive to the amount of training data provided.

Using the training method described above, we then evaluate ARIMAX models that are continuously retrained at each forecast time. We again allow the model to select the optimal $p$ and $q$ parameters based on AIC; however, we also allow the model to select the optimal amount of training data, $n^*$, to be used at each forecast time. We refer to these models as being AIC and forecast error optimal. This continuous re-training approach is one possible method for deployment of ARIMAX models during a disaster event, as the method incorporates only the most relevant amount of recent observations into model training data as it becomes available. Figure 3.9 shows forecasts at various times during the hurricane using these continuously re-

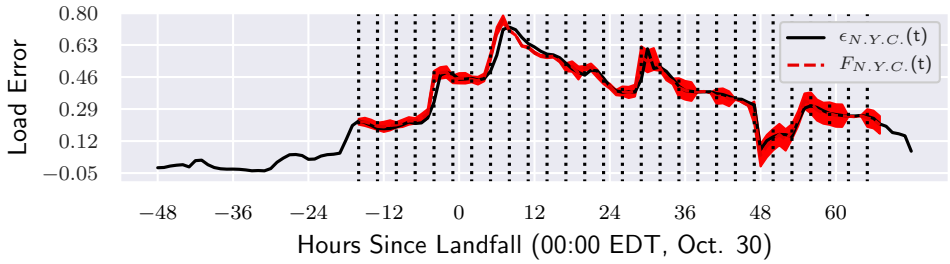


Figure 3.9: Forecasts ($F_{N.Y.C.}$) of a continuously-retrained set of ARIMAX models for N.Y.C. during Hurricane Sandy, compared to actual observed load errors ($\epsilon_{N.Y.C.}$). Each vertical, dotted line represents a new model that is optimized for both AIC and forecast-error at the time of forecast. The prediction intervals shown are calculated at the 95% significance level.

trained ARIMAX models. These models visually perform well in forecasting the load error, with 95% prediction intervals typically containing the actual observed load error.

## 3.4 Discussion

The main goal of social sensing is to provide spatially and temporally localized predictions of the effects of disasters as they relate to CIs. We demonstrate the feasibility of this approach through processing and statistical analysis of real-time social media data, followed by development of time series forecasting models using this data. While physical sensors are often available for measuring electrical grid response, our approach may supplement physical sensors by providing additional coverage for locations that lack funding, or where physical sensors are not as prevalent. Additionally, physical sensors may fail in disaster events, e.g., due to flooding, structural damage, or failures in communication infrastructure. In this case, social sensing would improve the robustness of power systems by providing supplementary information and redundancy to the existing suite of physical sensors. While social sensing may take on a supplementary role for power systems, the lack of real-time physical sensor data for other CIs (e.g., road transportation) means that social sensing could act as the primary sensor for determining the real-time effects of disasters on these CIs. Social sensing may also capture information that physical sensors may not, such as damages to CIs and adjacent areas.

Whether acting in a primary or supplementary role, social sensing methods have the potential to improve our understanding of the effects of disasters as they relate to CIs.

# Chapter 4 : Rail Infrastructure Delay Prediction

## 4.1 Introduction

Toward the goal of improving situational awareness of CI operators, we focus on the potential benefits of a graph-based formulation of the rail infrastructure for predictive models. When applied at a global level, a graph-based formulation allows us to explicitly model the interactions between every element of the rail network simultaneously. Through the use of a machine learning model, which optimizes itself based on real-world operational data, we are able to implicitly model interactions on the rail network which are not easily captured with other modeling techniques. To demonstrate the feasibility of this approach, we consider a subset of the British rail network from 2016-2017 a case study to provide real-time predictions of CI behaviors.

The British rail industry is currently experiencing a stagnation in performance affecting a rapidly growing commuter population. The Rail Research UK Association [61] predicts the increase of cascading delays, delays that are a result of its prior delays or the propagation of delay from any other train, from 600,000 minutes annually to 800,000 minutes annually in the last five years. With the number of passengers travelling on British train networks almost doubling from 1 billion to 1.7 billion in the past two decades [62], this trend will continue unless appropriate measures are put into place. However one major roadblock to reducing overall delays on the rail network is in understanding and modeling the propagation of delays on the rail network. These delays exhibit complex nonlinear spatio-temporal behaviors, and are inherently difficult to predict.

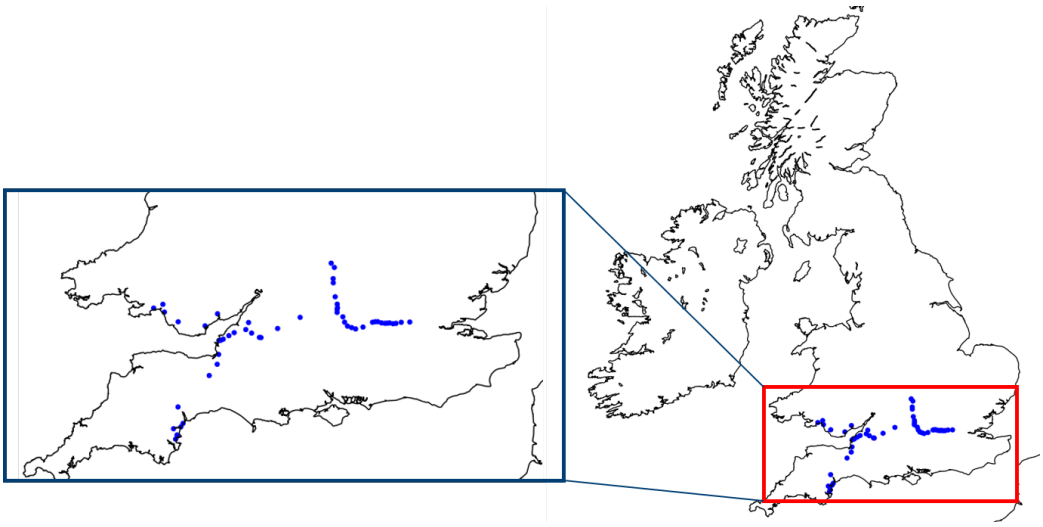

Figure 4.1: The subset of the British rail network with Didcot Parkway and London Paddington as the gateway stations.

## 4.2 Methods

### 4.2.1 Problem Formulation

We formulate the prediction of delays on the rail network as a time series regression problem in which observed delays on links, or connections between stations, at the previous $N_{\text{past}}$ time steps are used to predict the most likely delay at the $t + N_{\text{future}}$ time step. We use the following definition for links of the rail network:

**Definition 1.** Rail Link: A rail link AB exists between Station A and Station B if a train on the rail network does not pass through any other station on the network in between Station A and Station B.

Based on this definition, we formally state the regression problem on the rail network as,

$$\hat{y}_{t+N_{\text{future}}} = \underset{v_{t+N_{\text{future}}}}{\operatorname{argmax}} \log P(v_{t+N_{\text{future}}} | v_{t-N_{\text{past}}}, ..., v_t) \tag{4.1}$$

where $v_t \in R^{N \times F}$ is a tensor of $F$ delay features on $N$ links of the rail network at time $t$, and $\hat{y}_t \in R^{N \times F}$ is a tensor of model predictions at time $t$. Note

that we only consider one feature, link delay, for this study (i.e., $F = 1$).

We then represent the rail network as an undirected and attributed graph $G(V, E, X(t))$, defined by nodes $V$, edges $E$, and time-varying node features $X \in R^{N \times F \times T}$, where $N = |V|$ is the number of nodes in the graph, and $T$ is the total number of time steps in the dataset. This graph may be represented by its adjacency matrix $A_G$, defined as,

$$\begin{aligned} A_G(i,j) &= 1 \quad \text{if stations } i \text{ and } j \text{ share a link,} \\ &= 0 \quad \text{otherwise.} \end{aligned} \tag{4.2}$$

Note that while the node features are time-dependent, the underlying graph structure remains static throughout the data period.

This graph formulation considers the delays of links in the rail network as edge-wise features of $G$. While recent work has explored the use of edge-wise features for graph prediction, these architectures often do so in an effort to simultaneously leverage node- and edge-wise features. Since we only consider one set of features (i.e., link delays) we do not require such an architecture. We therefore invert the nodes and edges of $G$ to produce a line graph of the rail network $\mathcal{L}$ to enable the use of architectures with only node-wise features. This line graph then has an adjacency matrix $A_{\mathcal{L}}$, defined as,

$$\begin{aligned} A_{\mathcal{L}}(i,j) &= 1 \quad \text{if links } i \text{ and } j \text{ are connected by a station,} \\ &= 0 \quad \text{otherwise.} \end{aligned} \tag{4.3}$$

We use this line graph to capture spatial relationships within the data in our proposed model architectures.

Machine learning methods have shown promise for predicting delays in transportation systems. These methods typically leverage convolution operations to capture spatial relationships within the data. Graph neural networks (GNNs) extend these methods to be applicable for graph-structured data by specifically leveraging graph convolutions to propagate information between neighboring nodes and embed provided graph features into a latent space. This embedding provides a high-level representation of the data, which is then typically combined with a multi-layer perceptron or softmax output

Table 4.1: Service metrics provided by Darwin's HSP

| Key | Description |
|---|---|
| Origin Location | Computer Reservation System (CRS) code of origin |
| Destination Location | CRS code of destination |
| gbtt ptd | Public departure time at departure station |
| gbtt pta | Public arrival time at destination station |
| TOC code | Code of train operating company |
| RIDs | Train ID |
| Matched services | List of all train RIDs |
| Tolerance Value | Tolerance for difference between actual and public arrival time |
| Num not tolerance | Number of trains outside the tolerance |
| Num tolerance | Number of trains within the tolerance |

layer to provide node-level predictions. In this work, we compare a GNN model against two other common models to better-understand the benefits and drawbacks of each model. For surveys of GNNs and their applications, see [46] and [47].

### 4.2.2 Data Description

The dataset we use to study rail infrastructure is provided through Darwin, Great Britain's official railway information engine [63]. Specifically, the application programming interface (API) we utilize is the Historical Service Performance (HSP) API [64]. This API provides two datasets through two separate calls in Javascript Object Notation (JSON) Format, which are used in conjunction with each other. The first call, Service Metrics, requires the origin and destination stations, first departure and final arrival times, and start and end dates to be defined as inputs. The second call, Service Details, requires train IDs provided by the Service Metrics API as input. The data received through the Service Metrics call is outlined in Table 4.1 and data received through the Service Details call is outlined in Table 4.2.

All time values provided by the Service Details API are accurate to the nearest minute and include all origin-destination trips that pass through the gateway origin and destination stations. For our study of rail infrastructure,

Table 4.2: Service details provided by Darwin's HSP

| Key | Description |
|---|---|
| Date of service | Date of service of the specified train RID |
| TOC Code | Code of train operating company |
| RID | Inputted RID |
| Location | CRS code of train location |
| gbtt ptd | Public departure time |
| gbtt pta | Public arrival time |
| Actual td | Actual departure time |
| Actual ta | Actual arrival time |
| Late canc reason | Code that specifies late or cancellation reason |

we select Didcot Parkway and London Paddington as the gateway stations; i.e., all train journeys that include both these stations in their schedule in both inbound and outbound directions are included in the dataset. Journeys between these stations were chosen due to their notoriety in providing prevalent delayed services [65]. Figure 4.1 shows the rail network stations included for the gateway stations of Didcot Parkway and London Paddington. In the inbound direction, Darwin provides 10,767 journeys in 2016 and 10,742 journeys in 2017 initiating at various stations, passing through Didcot Parkway, and terminating at London Paddington. In the outbound direction, Darwin provides 9,069 journeys in 2016 and 8,969 journeys in 2017 initiating at London Paddington and passing through Didcot Parkway on the way to their respective destination stations.

### 4.2.3 Data Preprocessing

Given the raw data provided by Darwin, we include train journeys on non-holiday weekdays starting between 5:30 AM and 12:00 PM from 2016 and 2017. This time range was selected to capture the mechanics of delay propagation during peak usage of the rail network. We construct a line graph of the rail network $\mathcal{L}$ by setting links between stations as nodes of the graph and stations connecting links as edges. For consistency of the graph structure, we remove any links that are included in one year but not the other. We also only consider inbound trips for this set of experiments; i.e., we only

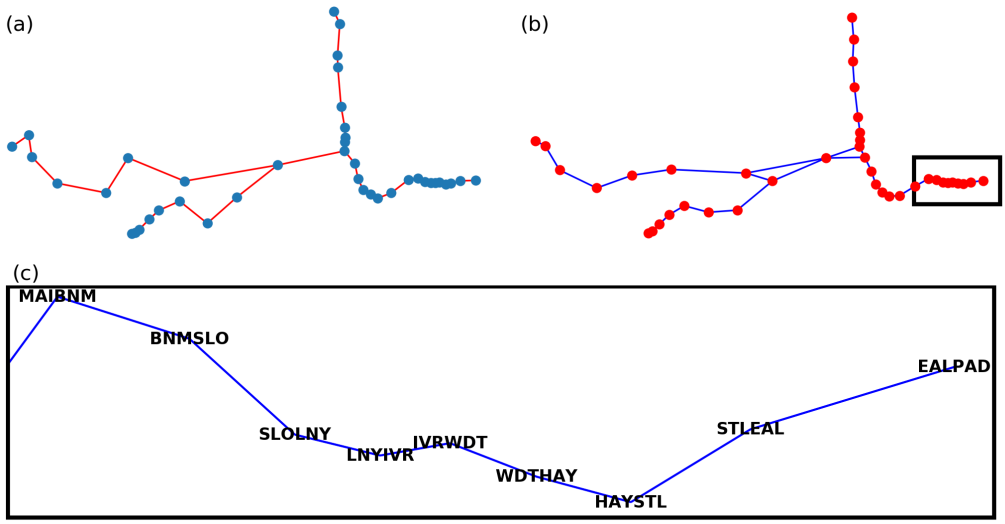


Figure 4.2: (a) The stations of the British rail network which form our rail network graph $G$. (b) The links of the British rail network which form our rail network line graph $\mathcal{L}$, which is used in the STGCN model. (c) A zoomed-in view of the highlighted links in (b). This inbound corridor of the rail network terminates at the London Paddington station, making it one of the busiest rail corridors in Britain.

include the trips beginning at some station, passing through Didcot Parkway, and terminating at London Paddington. Finally, we remove stations that serve an average of fewer than one train per day in order to reduce noise in the graph signal. Our resulting graph $G$ and line graph $\mathcal{L}$ are shown in Figure 4.2. We use the NetworkX library to calculate the line graph from $G$. The rail links included in this graph, and their usage during the considered time period, are shown in Figure 4.4.

Our model uses the arrival delay of trains passing through links on the rail network as its feature. This feature is used in order to measure the congestion experienced on each link of the rail network. Arrival delay is defined as $d_{\text{arr}} = t_{\text{arr, sched}} - t_{\text{arr, actual}}$, where $t_{\text{arr, sched}}$ and $t_{\text{arr, actual}}$ are the scheduled and actual arrival time of the train, respectively. We attribute the experienced arrival delay to each link traversed by the train in between its origin and destination stations. The following definitions explain the delay attribution process:

**Definition 2.** Route: A route is the set of rail links traversed by a train in between stations at which it stops. We denote the number of links in a route

as $n_{\text{links}}$. Note that a train may traverse a link as part of a route without stopping at either of the terminating stations on that link.

For example, consider rail network A $\rightarrow$ B $\rightarrow$ C $\rightarrow$ D and a train which departs from A, does not stop at B or C, and stops at D. The train's route for this section of its trip would be (AB, BC, CD).

**Definition 3.** Links Traversed during Time Period $(t_0, t_1)$: Consider rail network A $\rightarrow$ B $\rightarrow$ C $\rightarrow$ D and a train which departs from A, does not stop at B or C, and stops at D. Links AB, BC, and CD along route (AB, BC, CD) are considered traversed during $(t_0, t_1)$ if any of the following are true:

i) the time at which the train departed from Station A falls within $(t_0, t_1)$,

ii) the time at which the train arrives at Station D falls within $(t_0, t_1)$,

iii) the average time at which the train was traversing the route falls within $(t_0, t_1)$.

**Definition 4.** Link Attributed Arrival Delay: Denote Link Attributed Arrival Delay as $d_L := \dfrac{d_{\text{arr}}}{n_{\text{links}}}$. $d_L$ is a feature of the link AB during time period $(t_0, t_1)$ if and only if the link is part of a route that was traversed during $(t_0, t_1)$.

We consider a sequence of node features $(v_{t-N_{\text{past}}}, ..., v_t)$ as our model input, and a single interval $v_{t+N_{\text{future}}}$ as the model output. Since the majority of the routes traversed in the dataset last fewer than 15 minutes, we choose a sampling time interval of 10 minutes. That is, we sample the delay along each rail link at consecutive 10 minute intervals (e.g., [0900, 0910], [0910, 0920], ...). For numerical stability, all features are normalized and the z-score of link attributed delay is used to train the model. Finally, we implement a uniformly sampled 70 / 20 / 10% split of the data for the training, validation, and testing datasets, respectively. All metrics presented in this work were calculated on the test data which is not observed during model training.

### 4.2.4 Proposed Model

#### Network Architecture

As part of our explorations of GNN methods, we select an architecture that leverages node-wise features for the graph prediction problem. We use the

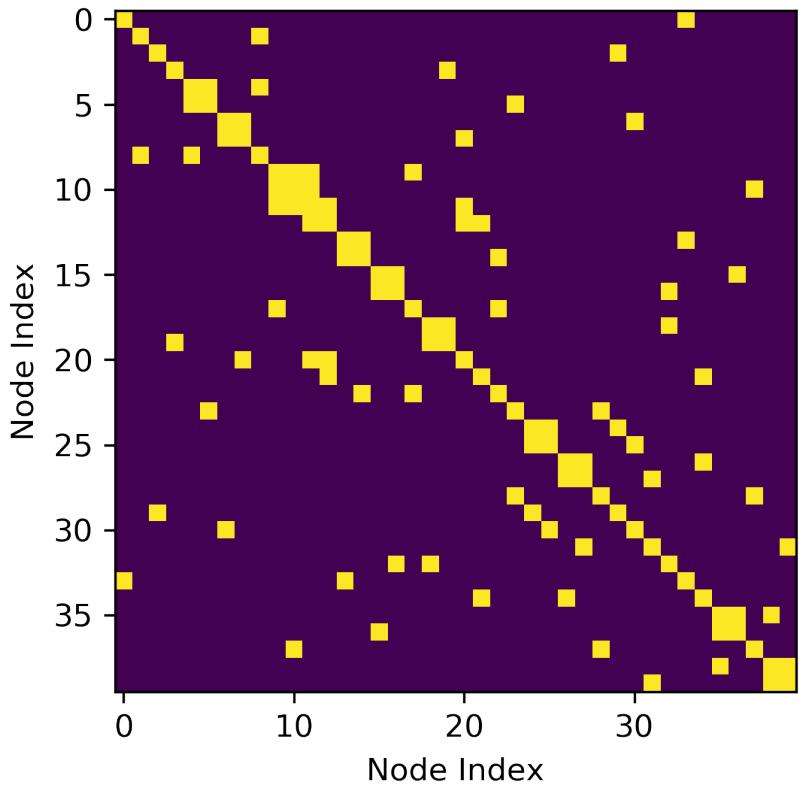


Figure 4.3: The resulting adjacency matrix for line graph $\mathcal{L}$. We add self-loops as part of the standard preprocessing for the STGCN architecture. The matrix is highly sparse due to the relatively small number of connections between links of rail.

STGCN model for this effort because it explicitly considers spatial and temporal dimensions network data [1]. The model architecture is summarized in Figure 4.5. The architecture contains two stacked spatio-temporal convolutional blocks (ST-Conv blocks) followed by an output block, which is itself composed of a temporal convolution followed by a fully-connected layer. We use the L2 loss function to train this architecture, defined as,

$$L(\hat{y};\theta) = \sum_{t} ||f(v_{t-N_{\text{past}}}, ..., v_t; \theta) - v_{t+N_{\text{future}}}||^2 \tag{4.4}$$

where $\theta$ are trainable model parameters, $v_{t+N_{\text{future}}}$ is the ground truth, and $f(\cdot)$ denotes the model's prediction. The following sub-sections provide details of the model at the level of the individual spatial and temporal convolutional blocks.

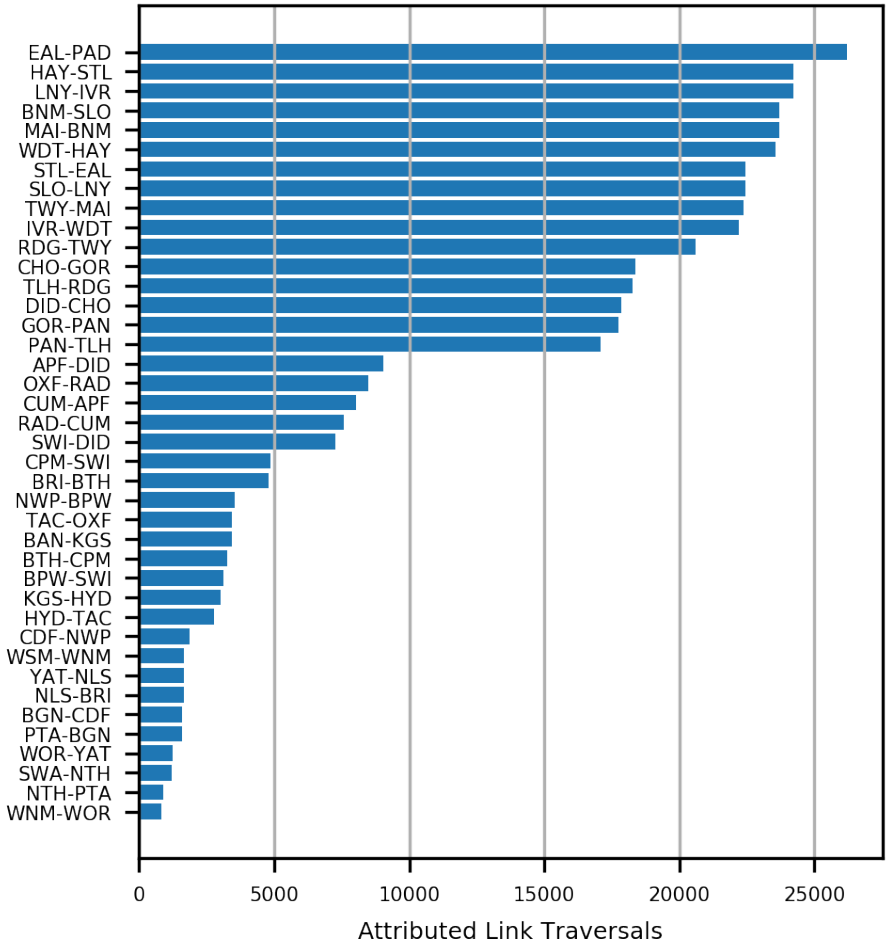


Figure 4.4: The set of links considered in the line graph $\mathcal{L}$ with the number of times they were traversed during the 2016-2017 data period. Each link name takes form AAA-BBB, where AAA / BBB are the station codes of the originating / terminating stations of the link, respectively.

## Convolution in the Spatial Dimension

The ST-Conv blocks of the STGCN architecture leverage graph convolutions to capture spatial relationships in the data. Spectral graph theory provides one method (i.e., the graph Fourier transform) for generalizing the convolution operation for graph-structured data. The analysis focuses on the eigenvalues of the normalized graph Laplacian matrix, given as,

$$L = I_N - D^{-1/2} A D^{-1/2} \tag{4.5}$$

where $I_N \in R^{N \times N}$ is the N-dimensional identity matrix which adds self-loop

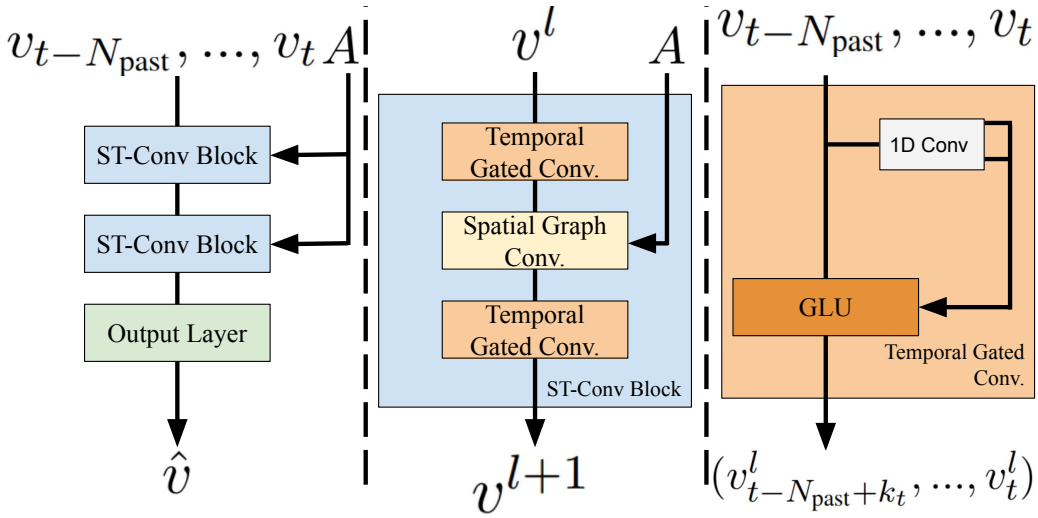


Figure 4.5: The STGCN model as developed in [1]. The overall model architecture is shown on the left, the ST-Conv block in the middle, and the temporal gated convolution block on the right.

connectivity to the adjacency, $A \in R^{N \times N}$ is the graph adjacency matrix, and $D \in R^{N \times N}$ is the diagonal degree matrix of $A$ such that $D_{ii} := \sum_j A_{ij}$.

The graph convolution "$*_G$" is defined as the multiplication of the graph signal $x \in R^N$ with kernel $\Theta$, such that,

$$\Theta *_G x = \Theta(L)x = \Theta(U \Lambda U^T)x = U\Theta(\Lambda)U^T x \tag{4.6}$$

where the graph Fourier basis $U \in R^{N \times N}$ is the matrix of eigenvectors of the normalized graph Laplacian, $\Lambda \in R^{N \times N}$ is the diagonal matrix of eigenvalues of $L$, and kernel $\Theta(\Lambda)$ is a diagonal matrix. Note that we denote the convolution operation on any generic graph $G$, which in our implementation is a rail network line graph $\mathcal{L}$.

Computation of $\Theta$ requires $\mathcal{O}(n^2)$ operations, making it computationally inefficient for large-scale graphs. However [66] introduces an approximation that restricts the graph kernel $\Theta$ to the set of Chebyshev Polynomials, and [67] introduces as a first-order approximation for the graph kernel. Both of these approximations are utilized in the STGCN architecture, after being generalized for use with multi-dimensional tensors. For brevity we do not include the details of the approximations or the generalization of graph convolution in this paper; however, more details may be found in [68, 52, 69].

### Convolution in the Temporal Dimension

The ST-Conv blocks also leverage a convolution to capture temporal relationships in the data. Recurrent Neural Networks (RNNs) are often used for this purpose; however, these networks can be difficult to train due to the "vanishing gradient" problem. Additionally, recent papers [70] [71] have shown that a 1D convolution along the temporal dimension of data can be more effective than an RNN on shorter sequences, while at the same time being quicker to train. As shown in Figure 4.5 (right), the temporal convolutional layer of each ST-Conv block contains a 1D causal convolution with kernel of size $k_t$ and a gated-linear unit (GLU) nonlinear activation. Similar to the gating present in RNN models, namely LSTM and GRU, the nonlinear activation provides a gating which determines importance of past inputs on future predictions. The resulting temporal convolution is defined as,

$$\Gamma *_T Y = P \odot \sigma(Q) \tag{4.7}$$

where $P$ and $Q$ result from splitting the input of the temporal block along the "channels" dimension. Further details of the temporal convolution, including generalization to 3D tensors, are provided in [1].

### Spatio-Temporal Convolutional Block

The ST-Conv blocks are constructed by combining these graph and temporal convolutions to capture spatio-temporal behaviors. The $l$th ST-Conv block is then given as,

$$v^{l+1} = \Gamma_1^l *_T \mathrm{ReLU}(\Theta^l *_G (\Gamma_0^l *_T v^l)) \tag{4.8}$$

where $\Gamma_0^l$ and $\Gamma_1^l$ are the temporal kernels within block $l$, $\Theta^l$ is the spectral kernel of the graph convolution, and ReLU denotes a rectified linear unit activation.

## 4.2.5 Model Implementation

For the STGCN model, we use a spatial kernel of size $k_s = 5$ and a temporal kernels of size $k_t = 3$. We use the Chebyshev polynomial approximation of

the graph Laplacian, and the channels within each ST-Conv block take a bottleneck form such that the number of channels are given as Block 1 = (1, 32, 64), Block 2 = (64, 32, 128), and Output Block = (128, 1). We train each model for 25 epochs with a batch size of 100 using the ADAM optimizer and L2 loss with an initial learning rate $R = 0.001$. Finally, we implement a learning rate decay of $R \leftarrow 0.1R$ every 10 epochs.

We also implement a multi-layer perceptron (MLP) model for comparison. We use a 3-layer fully-connected model with a 1-node input, 100-node hidden layer, and 1-node output. We use the RELU activation for the first two layers, and a sigmoid activation on the model output. This model is trained for 25 epochs using the same optimizer and loss function as STGCN.

## 4.3 Results

We compare the STGCN model's performance to two common statistical methods: linear regression (LR) and MLP. Neither LR nor MLP explicitly model the connections of graph-structured data, so for each node in the graph we optimized a new model for delay prediction. Furthermore, neither LR nor MLP are designed for time series prediction, so the features of each input time step are appended to form a feature vector of size $((N_{\text{past}} * F) \times 1)$. We test each model under multiple $(N_{\text{past}}, N_{\text{future}})$ time step conditions to understand the flexibility of the STGCN model its sensitivity to the input sequence length. We use MAE and RMSE to evaluate our models. These metrics are calculated by first averaging over the nodes of the graph, then averaging over the number of sequences in the dataset, and finally averaging over a set of 5 replicates per model. The results of our experiments are presented in Table 4.3.

We find that STGCN outperforms the other considered models on all test conditions. This result is likely due to the model's ability to capture dependencies between neighboring nodes in the graph via the graph convolution operation, as well as its ability to capture temporal dependencies via the temporal convolution operation. The combination of these operations implicitly models nonlinear cascading delays in the rail network. Our results show that this deep learning architecture can be readily applied to leverage available rail network data and provides more accurate predictions than

Table 4.3: Accuracy Metrics on Rail Network Data

| Model | | |
|---|---|---|
| $N_{past} = 6$ | MAE (10 / 30 / 60 min) | RMSE (10 / 30 / 60 min) |
| LR | 0.304 / 0.365 / 0.36 | 0.69 / 0.834 / 0.847 |
| MLP | 0.341 / 0.362 / 0.364 | 0.966 / 0.915 / 1.096 |
| STGCN | **0.256 / 0.311 / 0.302** | **0.625 / 0.803 / 0.755** |
| $N_{past} = 12$ | MAE (10 / 30 / 60 min) | RMSE (10 / 30 / 60 min) |
| LR | 0.279 / 0.337 / 0.338 | 0.59 / 0.753 / 0.785 |
| MLP | 0.331 / 0.34 / 0.327 | 0.982 / 0.896 / 0.931 |
| STGCN | **0.25 / 0.282 / 0.27** | **0.539 / 0.713 / 0.669** |

classical statistical methods.

## 4.4 Discussion

While our results were calculated on a subset of the British rail network, the STGCN model can easily scale to larger graphs while still capturing local dependencies between nodes. This scalability is due to the use of graph convolutions, which allow the model to output predictions for every node simultaneously. In comparison, classical statistical models either require a model to be trained for every node or implicitly assume full information propagation amongst nodes in a multiple response formulation. We also found that, while training the STGCN model took longer than the other methods, the STGCN model still trains relatively quickly, requiring on average 20 seconds per epoch for this dataset using a computer with an AMD Ryzen Threadripper 2920X CPU and an NVIDIA RTX 2080 GPU.

This work presents a novel, graph-based formulation of the British rail network. This formulation allows us to aggregate the experienced delay of multiple trains into a single measure of delay, namely link attributed delay, during a time period along each link of track in the train network. By attributing delays to links of the rail network, we are able to globally model the rail network instead of modeling individual trains or rail stations. The utilization of global information allows the STGCN architecture to optimize its predictions on real-world operational data to implicitly model nonlinear cascading delays on the entire rail network simultaneously. We demonstrate the feasibility of such a global formulation to predict expected delays that

trains would experience traversing each link of the rail network. Experiments on real-world rail data show that this architecture provides more accurate predictions than classical statistical models due to its ability to capture both spatial and temporal dimensions of the data.

# Chapter 5 : Conclusions

This thesis presents two primary approaches toward developing predictive models of CI behaviors. In our first approach, we explore social sensing methods to model disaster impacts on power systems through the case study of Hurricane Sandy as it impacted the state of New York. We find weak to moderate cross-correlations between Twitter data and load forecast errors, along with statistical evidence for Granger causality in the data, suggesting that Twitter data may provide value towards predicting impacts of disasters on infrastructure systems. Based on these findings, we then develop time series forecasting methods to predict future impacts on the NYISO power system at the zonal level, utilizing Twitter and load forecast data as model inputs. We find that forecast models for certain zones, particularly those whose load forecast errors were most impacted by Sandy, can be improved by including Twitter data.

In our second approach, we develop a novel, graph-based formulation of the British rail network to model the nonlinear cascading delays on the rail network. Using this formulation, we utilize several machine learning approaches, namely the application of the STGCN architecture, to predict expected delays that trains would experience traversing each link of the rail network. Through experiments on real-world rail data, we find that this architecture provides more accurate predictions than other models due to its ability to capture both spatial and temporal dimensions of the data.

More broadly, our proposed methods can improve CI resilience by providing more insight into behaviors of CIs during disruption events. Recall that a definition of resilience for CIs is "the ability to prepare and plan for, absorb, recover from, and more successfully adapt to adverse events" [10]. Real-time inference of CI impacts, provided by models such as ARMA or STGCN, would increase the situational awareness of infrastructure operators and give them high resolution awareness of disruptions to the infrastructure as they unfold, allowing for faster actions to be taken to mitigate damages. Addition-

ally, social sensing acts to augment statistical resilience frameworks by acting as an additional source of information in the determination of CI network functionality loss and CI adaptability [72, 73].

While we examined ARMA for power infrastructure and STGCN for rail infrastructure, future work may include the exploration of models that offer other desirable qualities not examined in this work, such as interpretability and uncertainty quantification. For social sensing, future directions of research include the use of natural language processing techniques for improved feature engineering of social media data, as well as the investigation of gazetteering approaches to infer geolocation from tweets. Geolocation inference will enable collection of significantly more data, since only about one percent of tweets are geotagged [74]. For rail infrastructure delay prediction, future work includes a more thorough comparisons of GNNs with existing models in the railway literature and alternative problem formulations to predict delays on specific routes and more explicitly consider inbound and outbound traffic on the rail network. Further study of the causes and propagation of delay in the rail network should also be included and develop of our models for real-world deployment.

# Bibliography


[1] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," *IJCAI International Joint Conference on Artificial Intelligence*, vol. 2018-July, pp. 3634–3640, 2018.

[2] P. Z. Ruckart, A. S. Ettinger, M. Hanna-Attisha, N. Jones, S. I. Davis, and P. N. Breysse, "The flint water crisis: a coordinated public health emergency response and recovery initiative," *Journal of public health management and practice: JPHMP*, vol. 25, no. Suppl 1 LEAD POISONING PREVENTION, p. S84, 2019.

[3] D. Mendonça and W. A. Wallace, "Impacts of the 2001 world trade center attack on new york city critical infrastructures," *Journal of Infrastructure Systems*, vol. 12, no. 4, pp. 260–270, 2006.

[4] Electricity Consumers Resource Council, "The Economic Impacts of the August 2003 Blackout," 2004.

[5] E. M. Roche, "Critical foundations: Protecting america's infrastructures," 1998.

[6] T. McDaniels, S. Chang, K. Peterson, J. Mikawoz, and D. Reed, "Empirical Framework for Characterizing Infrastructure Failure Interdependencies," *Journal of Infrastructure Systems*, vol. 13, no. 3, pp. 175–184, 2007. [Online]. Available: http://ascelibrary.org/doi/10.1061/{%}28ASCE{%}291076-0342{%}282007{%}2913{%}3A3{%}28175{%}29

[7] T. C. Sharkey, S. G. Nurre, H. Nguyen, J. H. Chow, J. E. Mitchell, and W. A. Wallace, "Identification and Classification of Restoration Interdependencies in the Wake of Hurricane Sandy," *Journal of Infrastructure Systems*, vol. 22, no. 1, pp. 04 015 007–1–12, 2016. [Online]. Available: http://ascelibrary.org/doi/10.1061/{%}28ASCE{%}29IS.1943-555X.0000262

[8] V. U. Unnikrishnan and J. W. van de Lindt, "Probabilistic framework for performance assessment of electrical power networks to tornadoes," *Sustainable and Resilient Infrastructure*, vol. 1, no. 3-4, pp. 137–152, 2016. [Online]. Available: http://dx.doi.org/10.1080/23789689.2016.1254998

[9] M. Ouyang, "Review on modeling and simulation of interdependent critical infrastructure systems," *Reliability Engineering and System Safety*, vol. 121, pp. 43–60, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.ress.2013.06.040

[10] Committee on Increasing National Resilience to Hazards and Disasters, *Disaster Resilience: A National Imperative*. Washington, DC: The National Academies Press, 2012.

[11] P. Uday and K. Marais, "Designing Resilient Systems-of-Systems: A Survey of Metrics, Methods, and Challenges," *Systems Engineering*, vol. 18, no. 5, pp. 491–510, 2015.

[12] A. W. Righi, T. A. Saurin, and P. Wachs, "A systematic literature review of resilience engineering: Research areas and a research agenda proposal," *Reliability Engineering & System Safety*, vol. 141, pp. 142–152, 2015.

[13] N. Yodo and P. Wang, "Engineering resilience quantification and system design implications: A literature survey," *Journal of Mechanical Design*, vol. 138, no. 11, pp. 111 408–1–13, 2016.

[14] M. Koliou, J. W. van de Lindt, T. P. McAllister, B. R. Ellingwood, M. Dillard, and H. Cutler, "State of the research in community resilience: progress and challenges," *Sustainable and Resilient Infrastructure*, pp. 1–21, jan 2018.

[15] P. Gasser, P. Lustenberger, M. Cinelli, W. Kim, M. Spada, P. Burgherr, S. Hirschberg, B. Stojadinovic, and T. Y. Sun, "A review on resilience assessment of energy systems," *Sustainable and Resilient Infrastructure*, vol. 00, no. 00, pp. 1–27, 2019. [Online]. Available: https://doi.org/10.1080/23789689.2019.1610600

[16] K. Barker, J. H. Lambert, C. W. Zobel, A. H. Tapia, J. E. Ramirez-Marquez, L. Albert, C. D. Nicholson, and C. Caragea, "Defining resilience analytics for interdependent cyber-physical-social networks," *Sustainable and Resilient Infrastructure*, vol. 2, no. 2, pp. 59–67, 2017.

[17] F. H. Jufri, V. Widiputra, and J. Jung, "State-of-the-art review on power grid resilience to extreme weather events: Definitions, frameworks, quantitative assessment methodologies, and enhancement strategies," *Applied Energy*, vol. 239, pp. 1049–1065, 2019.

[18] M. McGranaghan, M. Olearczyk, and C. Gellings, "Enhancing distribution resiliency: Opportunities for applying innovative technologies," *Electricity Today*, vol. 28, no. 1, pp. 46–48, 2013.

[19] R. Nateghi, S. D. Guikema, and S. M. Quiring, "Comparison and validation of statistical methods for predicting power outage durations in the event of hurricanes," *Risk Analysis: An International Journal*, vol. 31, no. 12, pp. 1897–1906, 2011.

[20] A. Arab, A. Khodaei, Z. Han, and S. K. Khator, "Proactive recovery of electric power assets for resiliency enhancement," *Ieee Access*, vol. 3, pp. 99–109, 2015.

[21] R. Eskandarpour, A. Khodaei, and A. Arab, "Improving power grid resilience through predictive outage estimation," in *2017 North American Power Symposium (NAPS)*. IEEE, 2017, pp. 1–5.

[22] M. Panteli, P. A. Crossley, D. S. Kirschen, and D. J. Sobajic, "Assessing the impact of insufficient situation awareness on power system operation," *IEEE Transactions on power systems*, vol. 28, no. 3, pp. 2967–2977, 2013.

[23] X. Guan and C. Chen, "Using social media data to understand and assess disasters," *Natural Hazards*, vol. 74, no. 2, pp. 837–850, Nov 2014. [Online]. Available: https://doi.org/10.1007/s11069-014-1217-1

[24] T. Bodnar, M. L. Dering, C. Tucker, and K. M. Hopkinson, "Using large-scale social media networks as a scalable sensing system for modeling real-time energy utilization patterns," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 10, pp. 2627–2640, 2016.

[25] Y. Kryvasheyeu, H. Chen, N. Obradovich, E. Moro, P. Van Hentenryck, J. Fowler, and M. Cebrian, "Rapid assessment of disaster damage using social media activity," *Science Advances*, vol. 2, no. 3, pp. e1 500 779–e1 500 779, 2016. [Online]. Available: http://advances.sciencemag.org/cgi/doi/10.1126/sciadv.1500779

[26] N. LaLone, A. Tapia, C. Zobel, C. Caraega, V. K. Neppalli, and S. Halse, "Embracing human noise as resilience indicator: twitter as power grid correlate," *Sustainable and Resilient Infrastructure*, vol. 2, no. 4, pp. 169–178, oct 2017.

[27] R. M. P. Goverde, "A delay propagation algorithm for large-scale railway traffic networks," *Transportation Research Part C: Emerging Technologies; 11th IFAC Symposium: The Role of Control*, vol. 18, no. 3, pp. 269–287, 2010.

[28] Y. Yang, P. Huang, Q. Peng, J. Li, and C. Wen, "Statistical delay distribution analysis on high-speed railway trains," *Journal of Modern Transportation*, vol. 27, no. 3, pp. 188–197, Sep 2019.

[29] A. O. Sorensen, A. D. Landmark, N. O. E. Olsson, and A. A. Seim, "Method of analysis for delay propagation in a single-track network," *Journal of Rail Transport Planning and Management*, vol. 7, no. 1, pp. 77–97, 2017.

[30] B. Cule, B. Goethals, S. Tassenoy, and S. Verboven, "Mining train delays," J. Gama, E. Bradley, and J. Hollmén, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 113–124.

[31] R. Wang and D. B. Work, "Data driven approaches for passenger train delay estimation," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, Sep. 2015, pp. 535–540.

[32] W. Barbour, C. Samal, S. Kuppa, A. Dubey, and D. B. Work, "On the data-driven prediction of arrival times for freight trains on us railroads," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 2289–2296.

[33] E. Bosscha, "Big data in railway operations: Using artificial neural networks to predict train delay propagation," Ph.D. dissertation, Jun 2016.

[34] L. Oneto, E. Fumeo, G. Clerico, R. Canepa, F. Papa, C. Dambra, N. Mazzino, and D. Anguita, "Train delay prediction systems: A big data analytics perspective," *Big Data Research; Selected papers from the 2nd INNS Conference on Big Data: Big Data and Neural Networks*, vol. 11, pp. 54–64, 2018.

[35] P. Wang and Q.-p. Zhang, "Train delay analysis and prediction based on big data fusion," *Transportation Safety and Environment*, vol. 1, no. 1, pp. 79–88, 02 2019.

[36] L. Oneto, E. Fumeo, G. Clerico, R. Canepa, F. Papa, C. Dambra, N. Mazzino, and D. Anguita, "Advanced analytics for train delay prediction systems by including exogenous weather data," in *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 2016, pp. 458–467.

[37] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 303–314, 1989.

[38] K. Hornik, M. Stinchcombe, H. White et al., "Multilayer feedforward networks are universal approximators." *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

[39] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[40] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[41] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," California Univ San Diego La Jolla Inst for Cognitive Science, Tech. Rep., 1985.

[42] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[43] C. Lea, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks: A unified approach to action segmentation," in *European Conference on Computer Vision*. Springer, 2016, pp. 47–54.

[44] D.-X. Zhou, "Universality of deep convolutional neural networks," *Applied and computational harmonic analysis*, vol. 48, no. 2, pp. 787–794, 2020.

[45] A. M. Schäfer and H. G. Zimmermann, "Recurrent neural networks are universal approximators," in *International Conference on Artificial Neural Networks*. Springer, 2006, pp. 632–640.

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

[47] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *arXiv preprint arXiv:1812.08434*, 2018.

[48] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=ryGs6iA5Km

[49] H. Maron, H. Ben-Hamu, H. Serviansky, and Y. Lipman, "Provably powerful graph networks," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 2156–2167. [Online]. Available: http://papers.nips.cc/paper/8488-provably-powerful-graph-networks.pdf

[50] R. Sato, "A survey on the expressive power of graph neural networks," *arXiv preprint arXiv:2003.04078*, 2020.

[51] Y. Seo, P. Vandergheynst, and X. Bresson, "STRUCTURED SEQUENCE MODELING WITH GRAPH CONVOLUTIONAL RECURRENT NETWORKS," no. 2013, pp. 1–10, 2017.

[52] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in neural information processing systems*, 2016, pp. 3844–3852.

[53] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," *arXiv preprint arXiv:1707.01926*, 2017.

[54] N. Feng, S. N. Guo, C. Song, Q. C. Zhu, and H. Y. Wan, "Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting," in *AAAI*, vol. 30, no. 3, 2019, pp. 759–769.

[55] M. Kunz, B. Mühr, T. Kunz-Plapp, J. E. Daniell, B. Khazai, F. Wenzel, M. Vannieuwenhuyse, T. Comes, F. Elmer, K. Schröter, J. Fohringer, T. Münzberg, C. Lucas, and J. Zschau, "Investigation of superstorm Sandy 2012 in a multi-disciplinary approach," *Natural Hazards and Earth System Sciences*, vol. 13, no. 10, pp. 2579–2598, 2013.

[56] H. Wang, E. Hovy, and M. Dredze, "The Hurricane Sandy Twitter Corpus," in *AAAI Workshop on the World Wide Web and Public Health Intelligence*, 2015, pp. 20–24.

[57] E. Loper and S. Bird, "NLTK: The Natural Language Toolkit," in *Proceedings of the COLING/ACL 2006 on Interactive Presentation Sessions*, Sydney, Australia, 2006. [Online]. Available: http://arxiv.org/abs/cs/0205028 pp. 69–72.

[58] M. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–7, 1980. [Online]. Available: http://www.emeraldinsight.com/doi/10.1108/00330330610681286

[59] R. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*, 2nd ed. Melbourne, Australia: OTexts, 2018.

[60] R. H. Shumway and D. S. Stoffer, *Time Series Analysis and Its Applications: With R Examples*, 4th ed. Cham, Switzerland: Springer International Publishing, 2017.

[61] RRUKA, "Call for research, data sandbox: Improving network performance," Tech. Rep., Oct 31, 2019.

[62] Office Of Rail and Road, "Passenger and freight rail performance 2018-19 q4 statistical release," Tech. Rep., May 24 2019.

[63] L. Bleakley, A. Akinola, and R. Fullard, "Rdg information feeds developer pack," Oct 2019. [Online]. Available: https://www.nationalrail.co.uk/46391.aspx

[64] "Darwin data feeds," 2019. [Online]. Available: https://www.nationalrail.co.uk/100296.aspx

[65] Department of Transport, "England and wales 'top 10' overcrowded train services: Spring and autumn 2015," Tech. Rep., Jul 28 2016.

[66] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, 2011. [Online]. Available: http://dx.doi.org/10.1016/j.acha.2010.04.005

[67] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," in *Proceedings of the 5th International Conference on Learning Representations*, ser. ICLR '17, 2017.

[68] M. Henaff, J. Bruna, and Y. LeCun, "Deep Convolutional Networks on Graph-Structured Data," pp. 1–10, 2015. [Online]. Available: http://arxiv.org/abs/1506.05163

[69] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[70] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint*, 2016. [Online]. Available: http://arxiv.org/abs/1609.03499

[71] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1243–1252.

[72] E. J. Gilrein, T. M. Carvalhaes, S. A. Markolf, M. V. Chester, B. R. Allenby, and M. Garcia, "Concepts and practices for transforming infrastructure from rigid to adaptable," *Sustainable and Resilient Infrastructure*, vol. 00, no. 00, pp. 1–22, 2019. [Online]. Available: https://doi.org/10.1080/23789689.2019.1599608

[73] R. Guidotti, H. Chmielewski, V. Unnikrishnan, P. Gardoni, T. McAllister, and J. van de Lindt, "Modeling the resilience of critical infrastructure: the role of network dependencies," *Sustainable and Resilient Infrastructure*, vol. 1, no. 3-4, pp. 153–168, 2016. [Online]. Available: http://dx.doi.org/10.1080/23789689.2016.1254999

[74] S. Kumar, F. Morstatter, and H. Liu, *Twitter data analytics*. Springer, 2014.