A model of station vulnerabilities towards delay propagation

Laetitia Molkenboer

TU Delft

A model of station vulnerabilities towards delay propagation

by

Laetitia Molkenboer

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on January 23, 2025.

Student number:	4553934	
Project duration:	February 12, 2024 – January 9, 2025	
Thesis committee:	Prof. O. Cats,	TU Delft, chair
	Asst. Prof. F. Schulte,	TU Delft, first supervisor
	Asst. Prof. Y. Zhu,	TU Delft, second supervisor

Front image by Population Dynamics Research Centers (2021).
An electronic version of this thesis is available at http://repository.tudelft.nl/.

TU Delft

Preface

With this report, my thesis and, therefore, my time at the TU Delft is about to end. It is strange to go from following classes, doing group work, taking exams, and finally writing my thesis to leaving my life in Delft behind. However, there is one thing for certain, which is that throughout my bachelor's in Technische Bestuurskunde and my master's in Transport, Infrastructure and Logistics, my passion for the world of transportation only got reinforced.

First, I would like to thank the members of my graduation committee. Starting with Oded Cats, thank you for offering me this research topic. It challenged me in all the right ways and I thoroughly enjoyed working on it. Also, Frederik Schulte, thank you for asking the hard questions during the weekly Friday meetings, which pushed me to move forward. Lastly, Yongqiu Zhu, you have truly been invaluable to my entire thesis. You always responded so quickly to my emails and always had time to meet with me. Thank you for brainstorming with me about the model, being critical of my thinking progress, and offering your knowledge and expertise.

Second, I have to thank some important people in my life. My family and boyfriend have been big supporters along the way. Thank you for allowing me to vent, for listening, for offering me advice, and for pushing me till the end.

Enjoy reading this report!

Laetitia Molkenboer
Delft, January 2025

Executive Summary

Metro networks serve an important societal purpose. However, operational challenges due to increased ridership and the growth of metro systems have surfaced. One of these challenges is the impact of primary delays, which, while local at first, could spread to other parts of the network. Network operators aim to prevent these delay propagations and minimize their impact when they occur, to ensure that they do not negatively affect travelers' experiences. Researchers have not yet identified any causal relationships between network structure, topological analysis, and delay propagation in the scientific literature. Consequently, the concept of vulnerability has been introduced in the scientific literature, which in this study is defined as the exposure of a public transport (PT) network to disruptions and the ability of the PT network to cope with these disruptions. Nonetheless, most of these studies do not consider delay propagation or take metro stations as their focus point when researching vulnerability. Moreover, the network operator's point of view has been neglected, while they are responsible for delay recovery strategies and prevention measures. Through epidemic models, delays and their impact have been studied successfully for air and public transportation and, thus, is a promising approach. However, when epidemic models have been applied to metro networks, congestion is always the focus point and, therefore, the passenger perspective. Therefore, the research objective is to see if the Susceptible-Infectious-Susceptible (SIS) model is suitable to model delay propagation in a metro network through its ability to reproduce the vulnerability of metro stations for specific instances.

Using the SIS model and scientific literature a model is composed. From the data, instances of delay propagations were identified and grouped based on the station and direction. Only two groups could be formed. These groups of instances were randomly split into 80% training and 20 % testing. The mean squared error (MSE) was used as a training performance indicator. Then, a differential evolution algorithm used data from the Washington Metro Network to train and test the model. For the testing, the MSE was used in combination with a comparison of the average vulnerability of the considered stations determined from the data and by the model. The vulnerability values as determined from the data for group 1 are presented in Figure 1.

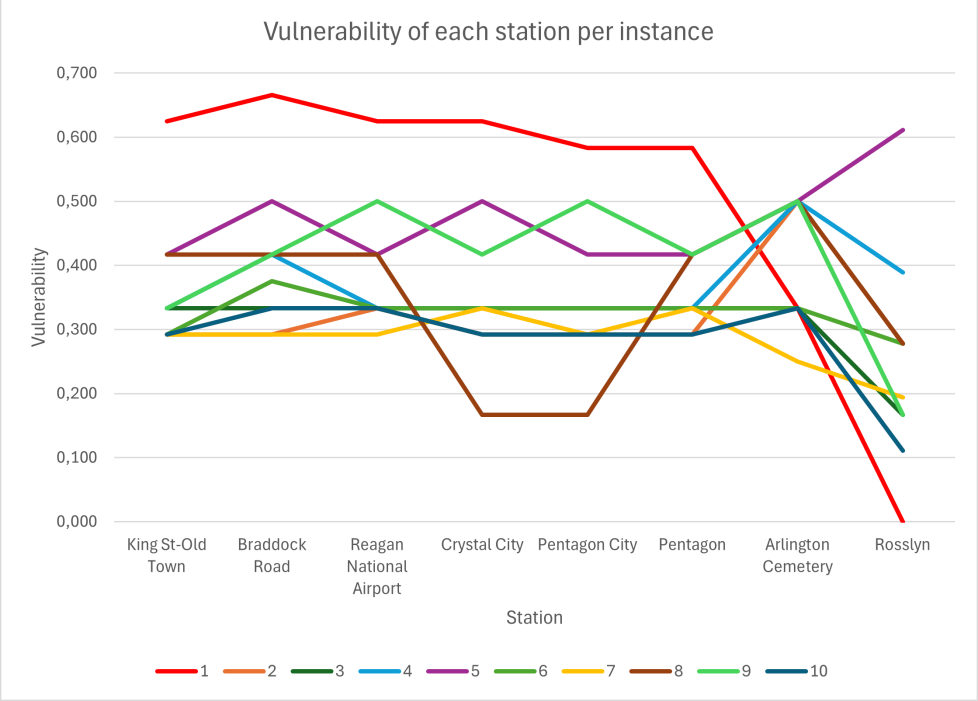


Figure 1: Vulnerability of each station per instance from group 1 in order of stations reached by the trains.

The analysis shows that train delay vulnerabilities generally decrease with distance from the primary delay station, but this trend is inconsistent across instances. Factors such as randomly delayed trains, arrival times near delay thresholds, and differing numbers of trains considered at each station affect the calculations. Additionally, similar primary delays, such as in instances 3 and 8, show differing vulnerability trends, suggesting other factors also play a role in how the delay propagates.

A training MSE of 0.022 was obtained for the first group of instances and the testing MSE for both testing instances was 0.004. While the testing results for group 1 reveal that the model was able to produce values close to the ones from the data (an error of less than one train), the model consistently underestimated station vulnerabilities, with data averages higher than model averages. Differences between predicted and actual vulnerabilities varied by station, with most stations underestimated. This underestimation likely stems from the training process, where the algorithm compensated for low recovery rates rather than low infection rates, resulting in higher recovery rates and lower overall vulnerability predictions. This trend persisted because the testing instances had similar vulnerability values to the training instances.

For group 2 the vulnerability values as determined from the data are presented in Figure 2.

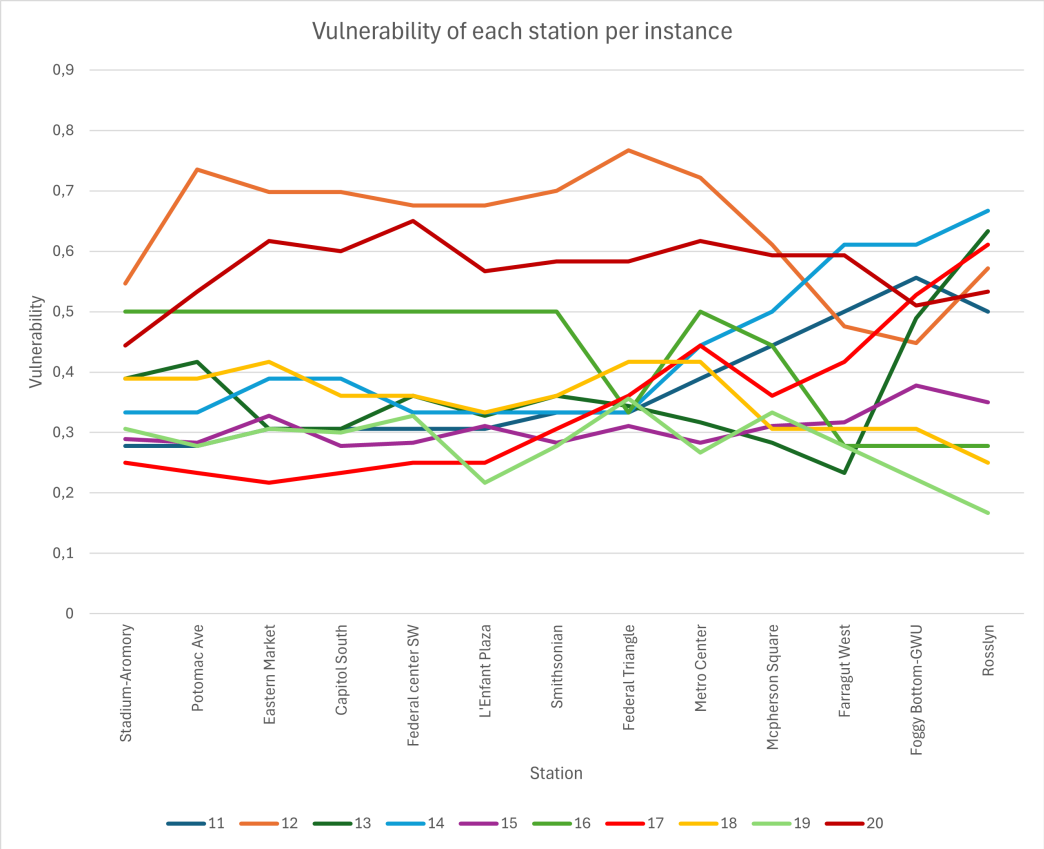


Figure 2: Vulnerability of each station per instance from group 2 in order of stations reached by the trains.

Group 2 stations exhibit a wider range of vulnerability values and greater fluctuations compared to group 1. The high traffic density in this part of the network hinders delayed trains from recovering, contributing to a delayed downward trend in vulnerabilities. Unexpected upward trends in vulnerability, such as for instances 11, 13, 14, and 17, highlight inconsistencies. Unlike group 1, where similar instances behaved differently, group 2 shows similar behavior across instances despite differing delay characteristics. The higher traffic density also increases the likelihood of random or minor delays, emphasizing that additional factors influence vulnerability trends.

The second group showed a MSE training performance of 0.043, and the testing instances have a MSE of 0.018 and 0.027. The testing MSE values translate to an error of around one train on average. In addition, the model overestimated the vulnerabilities compared to the data. This overestimation is likely due to the wider range of vulnerability values in group 2 ([0.25, 0.55]), which makes it harder for

the model to find a single parameter that fits all values effectively. Stations with higher vulnerabilities in the training data influenced the model, leading it to overestimate vulnerabilities for stations with lower-than-average values in the testing instances. The sensitivity of the model to the choice of training and testing instances requires further investigation to improve performance.

While the model initially shows promising results for the two groups, the values trained for the parameters indicate the training algorithm is slightly overfitting the model. One of the reasons for this overfitting is the mismatch in the order of magnitude of important model components. Also, the model is sensitive to the instances used for training and testing, which means the model's robustness could be improved. Furthermore, the model has one sensitive parameter, which causes the model to either overestimate or underestimate station vulnerabilities.

This research gives a starting point for developing a model inspired by epidemic models to study delay propagations in a metro network. Network operators can use this model to be proactive by targeting those stations that are the most vulnerable for certain delay propagation instances. The model also allows for a reactive response when the delay propagation has happened, through an investigation of the recovery of certain stations. Also, even if not enough data is available to train the model, information about stations can be obtained using the model components. Furthermore, using a SIS-inspired model to study delay propagations and the role of the vulnerability of stations helped fill in a gap in the literature. The model showed it can reproduce the vulnerability values with some limitations kept in mind. In the future, the research focus should be on improving the model through the introduction of more factors such as time of day. Also, the relationships between the different model components should be improved such that the problem of the order of magnitudes of the model equations is solved. Moreover, the model should be applied to a different network to get a better understanding of its performance. However, enough data, more than could be used in this research, is important to evolve this model.

Contents

1 Introduction

The problem is described first in section 1.1. Hereafter, the research objectives and questions are presented in section 1.2. The approach to answering the research questions is explained in section 1.3. Lastly, the structure of this thesis is presented in section 1.4.

1.1. Problem description

The metro has been and will continue to be an important mode of transport due to its social and environmental benefits (Redman et al., 2013). Over the years, the metro has faced several challenges such as the growth of cities due to urbanization, leading to increased ridership. At the same time, many metro systems have grown in distance (km of infrastructure) and number of stops (UITP, 2022). As a result, the pressure has intensified on metro systems generating more operational challenges. One of these operational challenges that needs attention is delays. Passengers possibly will not return if they experience delays too often while they are important to make public transportation (PT) networks financially feasible. Delays initially have a local impact but could spread to other parts of the network. Therefore, network operators are interested in limiting the effects of a delay and preventing the delay from happening in the first place. A concept that is tied to both aspects is delay propagation. Preventing the propagation of a delay from one vehicle to another means limiting the impact of the first delay and preventing a knock-on delay from happening. To take measures to prevent and respond to delay propagation, how and why delays propagate in metro networks must be studied.

Network structure and topological analysis have been used to study delay propagation. However, studies that have only looked into the influence of the network topology and structure failed to find any causal relations (Cats & Hijner, 2021; Wang et al., 2020; Yap & Cats, 2020). Therefore, the concept of vulnerability was introduced to be able to take more factors into account when studying delay impact and propagation. Some common keywords to describe vulnerability are susceptibility and serviceability (Berdica, 2002; Hong et al., 2022; Pan et al., 2021). Combining these keywords, vulnerability is the exposure of a public transport network to disruptions (susceptibility) and at the same time the ability of the PT network to handle these disruptions (serviceability) (Yap & Cats, 2020). Examples of the factors studied with the introduction of vulnerability are passenger flow distribution (Eltved et al., 2021; Szymula & Bešinović, 2020; Yap & Cats, 2022) as well as factors such as station-level characteristics (Zhang et al., 2021) and line operations (Malandri et al., 2018). Using these factors it has been found, for example, that the most vulnerable links/lines are those often crowded due to high passenger flows (Shi et al., 2019a; Sun & Guan, 2016; Yap & Cats, 2020; Yap et al., 2018b), with mainly the outflows at stations influencing the vulnerability (Zhang et al., 2020).

While advancements have been made in public transportation research with the introduction of vulnerability, this type of research does not always include delay propagation. Moreover, there has not been much attention to the vulnerability of metro stations and their role in delay propagation, even though stations play a central role in delay recovery strategies, such as holding trains, short-turning services, or adjusting schedules. Also, effective use of station infrastructure and staff, and knowing which stations to prioritize to devise prevention measures is crucial in mitigating delay propagation. Therefore, this study aims to fill in the knowledge gap of how to model delay propagation in a metro

network and the role the vulnerability of a metro station plays in this propagation.

For other transport modes, epidemic models have been used to fill in this knowledge gap as the spreading of diseases and propagation of delays show similarities. This method has proven to be a promising method in the railways (Dekker et al., 2022; Gurin et al., 2020; Monechi et al., 2018), air transportation (Baspinar & Koyuncu, 2016; Ceria et al., 2021; Wu et al., 2019) and metro (Jia et al., 2022; Shi et al., 2019b; Wang et al., 2019; Zeng & Li, 2018). Nonetheless, the studies for the metro have been solely focused on congestion propagation, the passenger perspective, neglecting the operator perspective. Therefore, the extent to which epidemic models can be used to model delay propagation from the operator's perspective is an interesting research direction.

1.2. Research objective and questions

The main goal of this research is to see if the Susceptible-Infectious-Susceptible (SIS) model is suitable to model delay propagation in a metro network through its ability to reproduce the vulnerability of metro stations for specific instances. It is also important to be critical under which conditions SIS is a suitable model and the limitations of this method. This goal is achieved by answering the following research question:

How can the SIS model effectively be utilized to produce the effects of delay propagation in metro networks, particularly through the models' ability to capture the vulnerabilities of metro stations for specific instances?

The main research question is jointly answered by the following sub-questions:

1. How can the traditional SIS model be adapted to accurately represent the characteristics and dynamics of delay propagations within a metro network?
2. Which model configuration produces the best results given performance metrics and keeping computational efficiency in mind?
3. Which combination of parameter values results in the model reproducing the vulnerabilities the best given a specific group of instances?
4. How sensitive is the model to changes in parameter values and data input?

1.3. Approach

The heterogeneous SIS model is adapted to a metro network context to answer the main research question and the sub-questions. All the subquestions focus on the heterogeneous SIS model adaptation and its implementation. The adaptation makes it possible to model the station vulnerability for specific delay propagation instances. These instances are based on the historical data of an existing network. To see which model configuration yields the best results, they are tested and compared based on performance metrics such as Mean Squared Error (MSE). To test the chosen model configuration its vulnerability results are compared to the vulnerabilities calculated using the historical data of a network for two cases using MSE as well as other metrics. Finally, experiments are done with the parameter values to test the model sensitivity.

1.4. Thesis structure

The structure of this thesis is as follows: Chapter 2 goes into more detail about the existing scientific literature on the topic of delay propagation and vulnerability. Then, Chapter 3 explains the methodology. The application of the model to the network case study and data processing are discussed in Chapter 4 as well as the experiments done to choose a model configuration. The results and an analysis of the results based on the chosen model configuration follow in Chapter 5. The thesis concludes in Chapter 6 with a conclusion, a discussion of this study, and recommendations for the future.

2

Literature review

This chapter discusses how this research fits into the scientific literature. First in section 2.1 the literature on network topology and vulnerability regarding delay propagation is discussed. Then, in section 2.2 previous applications of epidemic models in the field of transportation are highlighted.

2.1. Influence of network topology and vulnerability on delay propagation

Researchers have tried to find a relation between the structure of a public transportation network and its topological characteristics, and how a delay propagates. It is said that stations' locations within the network, connectivity, and criticality all play crucial roles in determining the extent and distribution of delays (Wang et al., 2020; Yap & Cats, 2020). One study found using the Bayesian Network method that while a delay caused by a disruption has the most impact on passenger delays at metro stations nearby on the same line direction, a correlation of delays between stations that are further apart exists (Cats & Hijner, 2021). The researchers argue that the constrained infrastructure as well as passengers changing their behavior is the cause of this correlation. They also suggest using epidemic models to capture both the spatial and temporal propagation of disruption, which their Bayesian method could not do. Thus, correlations were found, but no clear causation. Other studies tried to find more clear relations. Still, they only concluded that centrality measures are not indicative of the impact of disruptions on network performance (Malandri et al., 2018) and that transfer stations do not play a role in the disruption impact (Wang et al., 2020). Therefore, the causal relationship between topological features and the disruption impact is unknown, and topological features alone cannot lead to a good interpretation of the disruption impact.

Another perspective in the literature that has been adopted is assessing the role of vulnerability of nodes and links in propagating delays. This other way of looking at disruption impact and its propagation goes beyond conventional topological analysis and there are a few reasons for doing so. One of the reasons is that some of the studies using conventional topology analysis neglect passenger flow distribution, which could lead to underestimation or overestimation of the vulnerability of the network (Lu & Lin, 2019; Xu & Chopra, 2022). Secondly, depending on the day and traveler type (e.g. leisure versus work) the passenger flows are affected differently during and after a disruption, meaning the vulnerability can also differ throughout time (Eltved et al., 2021; Yap & Cats, 2022). This fluctuation in vulnerability was confirmed by another study (Xiao et al., 2018). Also, the effects of unplanned disruptions on passenger behavior can deviate from planned scenarios (Yap et al., 2018a), which is not something topology analysis can take into account. Furthermore, the effects of disruption were found to be heterogeneous across metro stations and dependent on its location in the network as well as other station-level characteristics, meaning that only using topology analysis would fail to acknowledge other factors planning a role (Zhang et al., 2021). The last reason to study vulnerability in tandem with topology is that initial failure propagates faster at the functional level (flow distribution) than at the structural level (network topology) (Chen et al., 2023), and, hence, only considering the structural level would give a limited picture of the problem. Moreover, both levels have different sources of vulnerabilities (Chopra et al., 2016). Therefore, only using topology analysis paints an incomplete picture of the

disruption and its impact, and research should go beyond using the concept of vulnerability.

Stations, the links between them, and the whole network have been considered in studies about delay propagation due to vulnerability. Studies have found that the most vulnerable links or lines are the ones that are often crowded due to the combination of relatively high passenger flows (as compared to other modes such as the bus and tram) and high disruption exposure (Shi et al., 2019a; Sun & Guan, 2016; Yap et al., 2018b). The influence of demand on the vulnerability of links was also confirmed by Szymula and Bešinovic (2020) as they found that which links are considered the most critical is highly demand dependent. In their work, critical links are those links that cause the most unfavorable consequences in terms of network performance, which was defined as dependent on physical topology and service characteristics (e.g. timetables, passenger demand). Similar results were obtained by a different study where stations were clustered according to their expected criticality into five categories (Yap & Cats, 2020). Criticality in their research meant: ”The degree of disruption exposure for an individual stop or link and the impact of a disruption occurring at that stop or link” (Cats et al., 2016). They found using machine learning that high train frequencies and passenger volumes were indicators of the most critical stations as well as whether the station was also a terminal/transfer station. This result is different from the previously mentioned research by Wang et al. (2020). However, that research only considered the network topology and not the criticality of stations. When only the network topology is considered transfer stations do not seem to play a role in delay propagation, but if the criticality of stations is considered transfer stations are important.

Some studies have tried to introduce more nuance into vulnerability research. In one study, the influence of passenger flows has been researched by separating the effects of the inflows and outflows on the vulnerability of a node. According to this study, passenger in-flows have a negligible impact on the vulnerability of an Urban Rail Transit (URT) station, while the vulnerability of the station will decrease as the out-flows increase (Zhang et al., 2020). This study, however, does not consider delay propagation explicitly except that neighboring stations might suffer from increased inflow. Also, the definition of vulnerability has been researched with more nuance. For example, Cats and Jenelius (2018) examined the effect of partial capacity degradation on PT network vulnerability, highlighting the importance of considering varying levels of disruption severity in vulnerability assessments as it could affect the vulnerability value. Also, most of the studies have defined vulnerability as a single value. However, a different approach was taken by Ermagun et al. (2023) in their study by defining vulnerability as a range of values for a complete metro network. Their results show that with a 1% increase in the ratio of links to nodes, the vulnerability increases by 0.50%. Similarly, a 1% increase in the ratio of the number of links to the maximum possible number of links decreases the vulnerability by 0.03%. These studies show a wide range of more nuanced vulnerability research has been done.

2.2. Epidemic models in transportation literature

There is already a foundation in the literature for other transport modes using the method of epidemic models. While epidemic models have mostly been used in the fields of epidemiology and communication, researchers in the transportation field have also slowly taken an interest in its potential applicability. There are three reasons for this interest. Firstly, there are similarities between the spread of diseases and the propagation of delays. Where individuals can infect others with disease, delays from one station can also ”infect” other stations. Secondly, factors playing a role in the delay propagation can be taken into account by an epidemic model. Thirdly, using epidemic models allows for a proactive response to the vulnerabilities in the network instead of reactive. These three reasons highlight the potential of using epidemic models as a method of modeling delay propagation and, as a result, vulnerability.

For metro, railway, and airline/air route networks research has already been done on the delay propagation using an epidemic model. First, the research in the context of railways is discussed, followed by the airline and airspace networks. Finally, how epidemic models have been used in metro research is examined.

2.2.1. Railway

A few papers discuss delay propagation in railways using epidemic models. The paper by Dekker et al. (2022) explores the phenomenon of delay propagation in railway networks using a diffusion-like spreading model, considering factors such as train schedules, infrastructure capacity, and network topology. The paper investigates how delays in one part of the railway network can spread and affect other parts.

This type of model, just like the SIS model, also describes the spreading of a delay throughout a system. However, a shortcoming of the diffusion model is that it is much more difficult to capture heterogeneity, while the SIS model used in this research can do so. The stations were modeled as homogeneous nodes in the mentioned study, which the authors also discussed as a shortcoming. Another study used a modified Susceptible-Infectious-Recovered (SIR) model to help quantify the propagation of delays of five cargo and four passenger trains in Ukraine (Gurin et al., 2020). In this study, 'susceptible' refers to trains prone to delays, 'infectious' entities represent the spread of these delays to neighboring trains, and "recovered" entities denote the delays being resolved. Even though Ukraine railways operate on a periodic schedule, it is common for cargo trains to leave their station of origin when they are ready instead of at the designated time (Gurin et al., 2020), increasing the complexity of the modeling, but this situation is not common. One last example of epidemic models used to model delay propagation in the railways is a study done by Monechi et al. (2018). They analyzed German and Italian railways by designing laws about the spreading of delays. The SIS model inspired this approach but it did not consider the recovery of the station or the station's ability to handle a delay. Also, the propagation probability was assumed to be uniform throughout the network neglecting any operational conditions. By using the SIS model in this research station's abilities to handle a delay and operational conditions can be taken into account.

2.2.2. Air transportation

In the field of air transportation epidemic models have been used to model delay propagation between airports as well as between flights. An example of such a study is the one by Baspinar and Koyuncu (2016). They defined a model for flights and airports, which used the SIS model characteristics to approximate the system dynamics under disrupted conditions. In the flight-based model, the SIS model was used to look at individual flights and how these flights could infect each other with delays. They then tried to define the collective behavior of airports using the flight-to-flight interactions in the airport-based model, which was modeled as a metapopulation. In a similar study, flight delays were also researched with the SIS model (Wu et al., 2019). A difference with the previously presented study is that the study by Wu et al. (2019) was done from an airline network perspective. The study by Baspinar and Koyuncu, on the other hand, was done with flight movements, disregarding the different airlines the flights belong to. They showed in their study that the propagation probability is network-related and varies across routes. Factors playing a role in this variation are flight frequencies at airports, route distances, and the propagated delay time. Smaller airports are largely affected by buffer times, while larger airports are mostly affected by flight movements. Also, lower network connectivity means more flights/airports are protected from delays, but aircraft utilization is also lower in comparison to better connected airports. In another study, the vulnerability of airports was modeled using a heterogeneous SIS model, which is very similar to the aim of this study (Ceria et al., 2021). They found that the vulnerability is the largest at airports whose strength in the airline network is neither too small nor too small. Airports with a low strength are often not as well connected to the other airports and so have to deal less with delays. Stronger airports, on the other hand, have many resources to minimize delays from propagating. Airports that are not as strong, but better connected than the weak airports are, therefore, less prepared to handle unexpected large disruptions. However, what has been done for airports is not directly applicable to metro networks, because the dynamics of the networks are different.

2.2.3. Metro

Epidemic models have also been used in metro research to study the propagation of delays due to crowding and congestion. For example, in the study by Jia et al. (2022) they adapted the SIS model to estimate the risk probability of crowding. They modeled the recovery and infection rate with the incoming and outgoing passenger flows at nodes and, thus, heterogeneous. Their research studied the changing risk probability of crowding over two hours. Some of their findings are that the majority of the network will be affected by crowding within half an hour as it starts to happen at some stations. Afterwards, the propagation speed will slow down. Also, the propagation strength diminishes with distance and the transfer nodes are impacted most significantly (Jia et al., 2022). However, the focus is on crowding propagation and not delay propagation specifically. A different study also looked into passenger flow congestion (Shi et al., 2019b). The study aimed to see if two control strategies could relieve the congestion pressure by reducing the infection rate and increasing the recovery rate. The

sensitivity of the propagation as a reaction to the control measures was also tested. They found that demand control measures helped deal with serious congestion. They used the SIR model, which does not allow infected stations to become infected again later with congestion after they recovered. However, in the context of this research, it should be possible for a station to be infected multiple times by different trains. A third study also looked into congestion propagation with intervention under emergency-caused delays specifically and, therefore, neglecting other types of delays (Wang et al., 2019). They based their passenger flows on the regret minimization theory for which they collected data through a state preference study. A difference was found in the congestion propagation for peak and non-peak hours. During peak hours a secondary propagation of congestion might be possible, while during non-peak hours there is only one propagation that has a short duration and influences a smaller range of nodes (Wang et al., 2019). A fourth study looked into congestion propagation for oversaturated conditions using the SIR model (Zeng & Li, 2018). Similar to the other studies Zeng and Li tried to quantify the rate at which the congestion propagates between different metro lines. Six influential factor groups were identified including passenger flow, train headway, and station capacity. All studies discussed for the metro used a type of epidemic model to model congestion propagation and, therefore, mainly focused on passenger flows, while this study will take the metro trains and their delay propagation as the perspective. Even though it is important to look at passenger flows, the network operator perspective should not be neglected to understand the challenges in daily operations and the influence of decisions on the network system.

2.3. Contribution

Previously the literature on delay propagation and vulnerability as well as the use of epidemic models in transportation studies have been presented. It is important to indicate what the position of this study is in the literature and its scientific contribution.

All the literature presented in section 2.1 is summarized in Table 2.1, including the articles that do not consider vulnerability. Table 2.1 also shows where this research categorizes within the literature.

Table 2.1: Overview of articles published on vulnerability in public transportation, as presented in this literature review.

		Method						DP	Network component	Real case
Reference	Focus	PM	ML	AM/DM	S	O	EM			
Cats & Hijner (2021)	Metro network	√						Y	Nodes	√
Cats & Jenelius (2018)	Multi-modal network			√				N	Links	√
Chen et al. (2023)	Multi-modal network		√		√			Y	Network	
Chopra et al. (2016)	Metro network				√			N	Network, nodes, links	√
Eltved et al. (2021)	Railway line		√					N	Line	√
Ermagun et al. (2023)	Metro network				√			N	Nodes, links	√
Lu & Lin (2019)	Multi-modal network				√			N	Nodes	√
Malandri et al. (2018)	Multi-modal network			√	√			Y	Network	√
Shi et al. (2019a)	URT network				√			N	Nodes	√
Sun & Guan (2016)	Metro network			√	√			N	Line	√
Szymula & Bešinović (2020)	Railway system			√		√		N	Links	√
Wang et al. (2020)	Metro network				√			N	Network, nodes	√
Xiao et al. (2018)	Metro network				√			N	Network, links	√
Xu & Chopra (2022)	Metro network			√				N	Network	√
Yap & Cats (2020)	Metro network		√					Y	Nodes	√
Yap & Cats (2022)	Multi-modal network		√					N	Network	√
Yap et al. (2018a)	Multi-modal network			√				N	Network	√
Yap et al. (2018b)	Metro/light-rail network			√	√			N	Links	√
Zhang et al. (2020)	URT network				√			N	Nodes	√
Zhang et al. (2021)	Metro network		√					N	Nodes	√
This work	Metro network						√	Y	Nodes	√

PM = probabilistic model; ML = machine learning; AM/DM = assignment/demand model; S = simulation; O = optimization; DP = delay propagation considered; Y = Yes; N = No

Research on delay propagation in public transportation networks has advanced significantly, with studies emphasizing passenger perspectives, network structure, and vulnerability assessments. While there is agreement in the literature that the network structure does influence delay propagation (Cats & Hijner, 2021; Wang et al., 2020; Yap & Cats, 2020), clear causal relations are yet to be found. Therefore, research on disruptions and their impact should go beyond conventional topology analysis. By considering factors such as passenger flow distribution (Eltved et al., 2021; Yap & Cats, 2022), studies have looked into the complex interplay between network structure, passenger behavior, and vulner-

ability. Exploring the vulnerability of network nodes and links also offers a nuanced understanding of disruption impact and delay propagation within public transportation networks, supplementing conventional topological analysis. Even so, the role vulnerability of individual nodes plays when looking specifically at delay propagation has been neglected in metro network research. Two papers in this literature review also considered the combination of delay propagation and stations. Nevertheless, the study done by Cats and Hijner (2021) and Yap and Cats (2020) focused on passenger delays, while this study will look at metro train delays. The focus on delays from the operational perspective helps to understand why and how delays happened, identify network components that need improvement to maintain reliability, and implement mitigation strategies effectively.

Research directions that have been advised are epidemic models to capture the delay propagation of disruptions, which is a method that focuses on nodes (Cats & Hijner, 2021). How epidemic models have been used before to research delay propagation was explored in section 2.2. First studies using epidemic models to model railway delay propagations were discussed. The problems of these studies are that they do not take the heterogeneity of stations into account, research uncommon situations, or fail to take operational conditions into account. The use of epidemic models in these studies can, therefore, not be applied to metro networks. While epidemic models in air transportation have been used to model delay propagation between airports and flights, the network dynamics are too different and their approach does not apply to metro networks. Lastly, epidemic models have been used to study metro networks. Nonetheless, the passenger perspective dominates these studies as congestion has been the main research topic.

This literature review showed which research gaps need to be filled. More insight is desired into how to model delay propagation in a metro network from the operator perspective using epidemic models and the role of vulnerability of metro stations in this propagation.

3

Methodology

This chapter explains how the previously defined research goals and questions are answered. As mentioned before, the main goal of this research is to see if the SIS model is a suitable method to model delay propagation in a metro network through its ability to reproduce the vulnerability of metro stations in a specific instance.

The definition of vulnerability from Yap and Cats (2020) is used in this research. Their definition combined definitions from Rodriguez-Nunez and Garcia-Palomares (2014) and Oliveira et al. (2016). Vulnerability is defined in their paper as the exposure of a PT network to disruptions and the ability of the PT network to cope with these disruptions. Therefore, the definition consists of two components, which are highlighted in this research through the model.

To achieve this goal several steps have to be taken, which are shown in Figure 3.1. First the instance selection criteria are explained in section 3.1, which will help understand how and why the instances were chosen for training and testing. Then in section 3.2 it is demonstrated how the vulnerability calculations using the data are done. That section is followed by an introduction to the SIS model and the adaption of it for this research in section 3.3. Finally, section 3.4 discusses how the model is trained and tested using metrics that compare the results from the vulnerability calculations using the data (3.2) and the training results of the model (3.3).

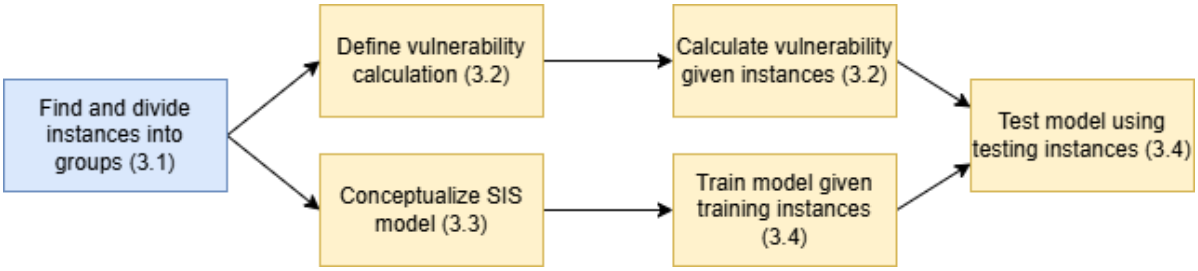


Figure 3.1: The methodology steps. Blue boxes indicate steps that have to do with the input and the yellow boxes indicate processes with this input. The numbers in the boxes correspond to the section in which the step is discussed.

3.1. Instance selection criteria

This research focuses only on the delays that lead to delay propagation and whose effects can be seen across multiple stations. Hence, not every delay found in the data fits the needs of this research. Therefore, selection criteria are needed to select instances from the data. An instance refers to a moment in the data when a delay propagation happened.

For a delay to be considered primary and propagated five conditions have to be met:

1. The primary train is delayed;
2. The train behind the primary train is delayed;

3. The train in front of the primary delayed train is not delayed;
4. The primary train was not delayed at the previous station;
5. The train behind was not delayed at the previous station.

The first two conditions are needed to ensure that propagation of the delay from the primary to the train behind could be happening. The third condition helps ensure that the primary delayed train is the first to be delayed and that the propagated delays come from this primary delayed train. Also, both trains cannot be delayed at the previous station (conditions four and five) to ensure propagation did happen. Sometimes the train behind is already delayed for a few stations and then when the train in front of it delays as well, it may seem like delay propagation, while it is not. To illustrate the conditions Figure 3.2 show examples of how the conditions are met or not met.

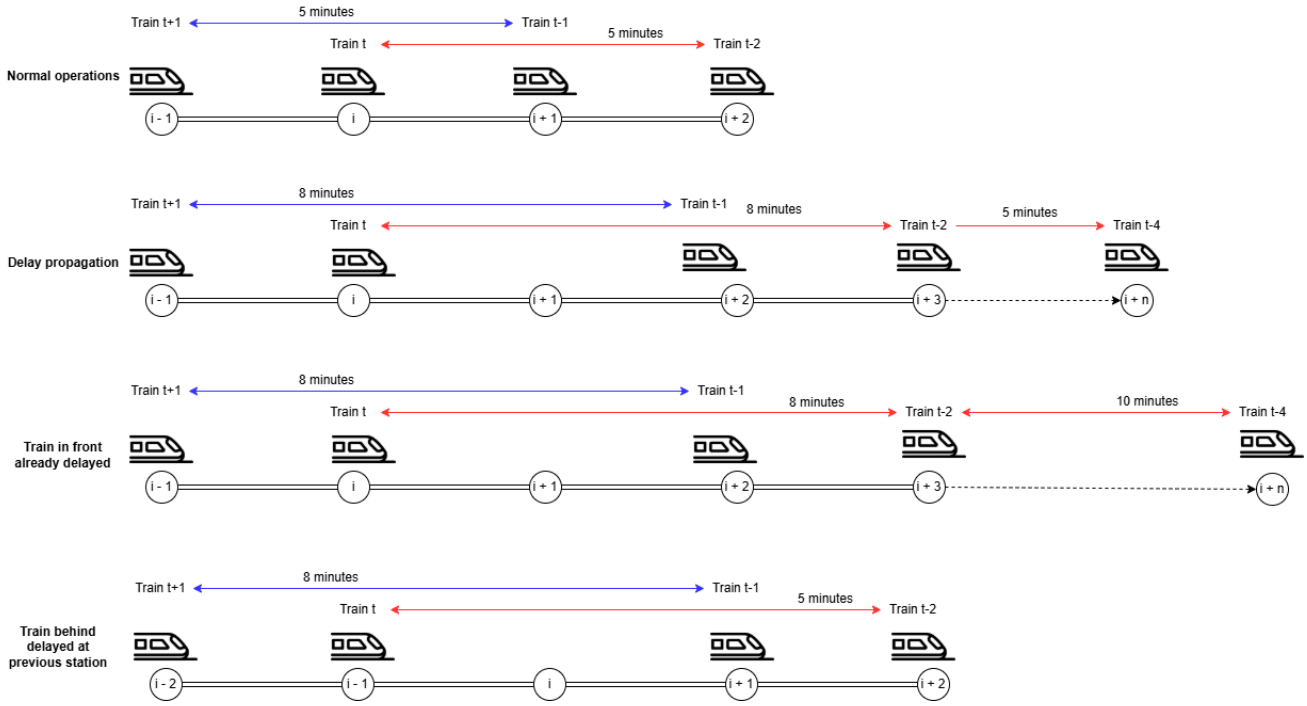


Figure 3.2: Examples showing whether the delay propagation conditions are fulfilled. The colored lines show trains belonging to the Blue or Red line and the arrow with the time represents the headway between the two trains. Note that the stations have changed in the last example.

The first example shows normal operations, so the conditions are not met. In the second example, trains t and $t-1$ are delayed which can be inferred from the increased headway. Also, they were both not delayed at the previous station and the train in front of train t was not delayed either because the headway was still five minutes per normal operations. In the third example train $t-2$, which is the train behind train $t-4$, is already delayed. Train t can, therefore, not be the primary delayed train anymore, because it could have been influenced by the delay of train $t-2$. The last example shows that train $t-1$ was already delayed at station $i-2$ and so even though train t might become delayed at station i, no delay propagation officially could happen, because train $t-1$ did not become initially delayed because of train t.

For each chosen instance of delay propagation, the following elements are collected from the data:

- The primary delay station
- The trains directly and indirectly affected by the primary delay
- The scheduled headway of the delayed trains
- The delay duration of the primary delay

- The direction of the primary delay
- The line of the primary delayed train
- The arrival time of the train in front of the primary delayed train at the primary delay station
- The arrival time of the last affected train at the last considered station

The last two points help set the boundaries on the time window w of an instance. The time window w of each instance is defined as the arrival time of the previous train at the primary station till the last train affected by the delay at the primary delay station reaches the station that all affected trains across all instances reach with a delay. To help understand Figure 3.3 shows an example of the determination of the end time.

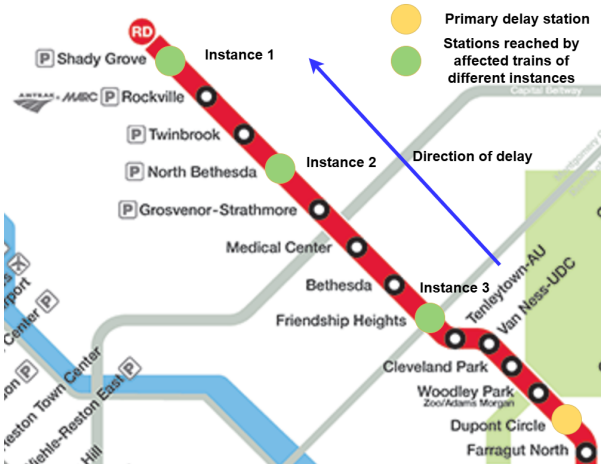


Figure 3.3: Example of how the end of the time window of each instance is determined. The yellow dot represents the primary delay station and the green dots are the stations affected trains of different instances have reached.

In Figure 3.3, the primary delay originates at the Dupont Circle station, but the affected trains of each instance are still delayed at different stations. However, they all cross the same station with a delay: Friendship Heights. This station is the last one to be considered for this instance. Consequently, the arrival time of the final affected train at that station is set as the end time for that instance rounded up to the nearest minute. The rationale for focusing on a selection of affected stations rather than the entire line is that some stations experience delays only for certain instances. This results in significant variability in the delay impact at those stations, posing challenges in effectively capturing these variations and training the model to produce accurate results. Further details can be found in subsection 4.3.1.

The aim is to find several groups of similar instances to train the model. Instances are alike when the primary delay starts at the same station and travels in the same direction. They must start at the same station and in the same direction to investigate properly the consequences of a specific delay. Delays starting at different stations or in other directions could have different effects and are, therefore, not comparable. The instances in a group are similar, while the instances in different groups must differ in primary delay station, affected lines, and delay direction so that the model can be tested on multiple parts of the case study network. The model configuration is the same for each group, but the training on different parts of the network will result in different trained values for the parameters. These trained values are, thus, tailored to the specific characteristics of each group.

3.2. Calculation of station vulnerabilities using the data

The vulnerabilities of the stations using the data are calculated by looking at the train movements at each station. These train movements are all the trains passing through a station during the time window w in the direction of the primary delay.

The vulnerability of each metro station is determined by looking at the number of delayed trains at a station i using the same infrastructure k in the direction of the primary delayed train in this time window w. The line on which the primary delay originates and the lines sharing infrastructure section k with

the primary delay line are considered. Consequently, depending on the station and the instance, the vulnerability is only determined by a selection of the lines running through the station. The vulnerability of a station is determined per infrastructure section k at a station because it is assumed that only trains that use the same infrastructure can propagate delays to each other. Trains using other infrastructure at the same station should not affect the vulnerability calculations. An example of multiple independent infrastructure sections is when stations have multiple levels with tracks and platforms.

Also, only the trains in the direction of the primary delay are considered, because it is assumed that delays only propagate to the trains behind the primary delayed train as delays have the most impact on metro stations nearby on the same line direction (Cats & Hijner, 2021). In other words, no cross-platform delay propagation is considered in this research. It was chosen to look at the arrival delay and not the departure delay because if a train arrives delayed at the next station the train is either the primary delayed train or a delay was propagated to this train. When a train departs with a delay, it does not mean it will propagate the delay if it can make up its delay.

Therefore, to determine if a train t is delayed, the time difference in arrival at station i between two consecutive trains of the same line, t and $t-1$, is calculated and represented as $b_{t,t-1}$. Typically, for trains traveling directly behind one another on the same line and in the same direction, this time difference should be approximately equal to the scheduled headway $m_{t,t-1}$ between trains t and $t-1$. However, due to potential variability in service, deviations in headway between two trains could occur without impacting overall network performance. A delay threshold parameter, denoted as δ, is introduced to account for these variations. This parameter is predetermined based on the specific characteristics of the case study network. If the time difference between the arrival of trains t and $t-1$ exceeds the sum of the scheduled headway $m_{t,t-1}$ and the delay threshold δ, the train t is considered delayed. In such cases, the variable a_{tiw}, which indicates whether train t was delayed during time window w at station i, is set to 1. The accompanying equation to determine if a train is delayed is shown in Equation 3.1.

$$a_{tiw} = \begin{cases} 1 & \text{if } b_{t,t-1} \geq m_{t,t-1} + \delta \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

The vulnerability of station i is the ratio of delayed trains to all observed trains T_{ikw} stopping at a station i using infrastructure k in the time window w of the instance in the direction of the primary delay. Mathematically, the vulnerability of station i translates to Equation 3.2.

$$v_{ik,data} = \frac{\sum_{t=1}^{T_{ikw}} a_{tiw}}{|T_{ikw}|} \tag{3.2}$$

The definition of each set, variable, and parameter is summarized in Table 3.1.

Table 3.1: This table shows the mathematical notation of sets, parameters, and variables used in the vulnerability calculations from the data.

Sets and indices		
N	set of stations	$i \in N,\ j \in N$
T_{ikw}	set of observed trains in the direction of the delay	$i \in N,\ t \in T_{ikw},\ T_{ikw} \subseteq W$
W	set of observed train movements during time window w	$w \in W$
Variables		
$v_{ik,data}$	vulnerability of infrastructure k at station i calculated from the data	$0 \leq v_{ik,data} \leq 1$
a_{tiw}	whether train t arrives at station i with delay or not in time window w	$a_{tiw} \in \{0, 1\}$
$b_{t,t-1}$	difference between the arrival times of trains t and $t-1$	[seconds]
Parameters		
$m_{t,t-1}$	scheduled headway between trains t and $t-1$	[seconds]
δ	delay threshold	[seconds]

3.3. Susceptible-infectious-susceptible model of delay propagations

This section explains the standard SIS model in more detail in subsection 3.3.1. Knowing the basics, how the traditional SIS model is transformed into a model that can be applied to estimate the vulnerabilities of stations in metro networks is explained in subsection 3.3.2.

3.3.1. SIS model basics

The Susceptible-Infected-Recover (SIR) model was the first epidemiological model of its kind (Kermack & Mckendrick, 1927). The Susceptible-Infected-Susceptible (SIS) model was developed at the same time as the SIR model and is used to study epidemics of diseases to which individuals could not become immune. While the applicability of these models has mostly been in the fields of epidemiology and communication, section 2.2 has shown that the applications of these models now also serve transportation.

The basics of the SIS model define that each individual in a system is either in the susceptible to a disease state or infected by the disease state. When an individual comes into contact with an infected individual, it can transfer from the susceptible to the infected state. If this contact happens, the infection rate governs whether an individual becomes infected. After an individual becomes infected it has a recovery rate and will eventually transfer back to the susceptible state. The infection and recovery processes are independent of each other. This process continues until one of two conditions are met: 1. the system reaches an equilibrium, meaning the amount of individuals in each state stays the same or 2. each individual is in the susceptible state as the disease has disappeared. If the model is called heterogeneous, it means that the individuals are not the same and, therefore, have varying characteristics that different model components take into account.

There are more versions of the SIR and SIS model. For example, the Susceptible-Exposed-Infected-Recover (SEIR) model is an example where individuals also have an exposed state, meaning they first go to the exposed state and then the infected state. Another example is that individuals can enter (birth) or leave (death) the system. The standard heterogeneous SIS model is used in this research because the model versions mentioned earlier do not apply. Furthermore, as stated in chapter 2, the effects of disruption were found to be heterogeneous across metro stations (Zhang et al., 2021) and so creating a heterogeneous SIS model as compared to a homogeneous SIS model is more realistic. The translation of the heterogeneous SIS model to a metro network is explained in subsection 3.3.2.

3.3.2. SIS mathematical model for a metro network

As mentioned before, the SIS model is used in this research to create the model. The stations are the individuals who are susceptible or infected. Stations can become infected by trains arriving with a delay. In this research the model is heterogeneous, meaning each station recovers at a different rate from the delay.

Table 3.2 shows all the different sets and variables used in the mathematical model in this research. How this notation is used in the formulation of the model can be found in equations 3.3 through 3.9 with the reasoning behind the formulations of each model component.

Network graph A network graph is created to resemble the stations and their connections. There are a few options to choose from for the metro network representation. One of these options is the L-space. The L-space is a network graph where the nodes are stations and all stations directly adjacent to each other and connected by a service are connected by an edge (Derrible & Kennedy, 2011; Von Ferber et al., 2009). This representation resembles the physical network but makes no distinction between different lines. Another option is the P-space, which also represents the stations as nodes. However, each station is connected to the stations that can be reached without a transfer (Derrible & Kennedy, 2011; Von Ferber et al., 2009). The P-space has been called the ”space-of-service” (Luo et al., 2020) and simplifies the network by focusing only on the relationships between stations. An example of both representations is shown in Figure 3.4.

Table 3.2: This table shows the mathematical notation of sets, parameters, and variables used in the model.

Sets and indices		
N	set of stations	$i \in N, j \in N$
K_i	set of independent sections of infrastructure for station i	$k \in K_i, i \in N$
L	set of lines in network	$l \in L$
E	set of edges that fulfill one of three conditions described in paragraph Link weight	$e \in E$
P	set of primary delay stations	$p \in P, P \subseteq N$
S	set of stations indirectly connected to a station i through a transfer station	$s \in S, S \subseteq N$
Variables		
w_{ij}	weight of link between stations j and i	[minutes]
$h(p,j)$	number of stations between primary delay station p and station j in real network	[stations]
i_{ik}	infection rate of section k at station i	[-]
r_{ik}	recovery rate of section k at station i	[-]
$v_{ik,model}$	vulnerability of section k at station i based on the model	[-]
Parameters		
g_j	indicates whether a station j is a transfer station or not	$g_j \in \{0,1\}$
u_{ki}	number of tracks in section k at station i	[tracks]
d_{ki}	traffic density of section k at station i	$[\frac{trains}{hour}]$
h_{lki}	scheduled headway of line l running on section k at station i	[minutes]
θ	heterogeneity of recovery rate	[-]
α	primary delay duration of instance	[minutes]
γ	delay propagation scalar	[-]
a_e	number of lines allocated to an edge e	[lines]
z_{ik}	to be trained constant for the infection rate of station i	[-]
c_{ik}	to be trained constant for the recovery rate of station i	[-]

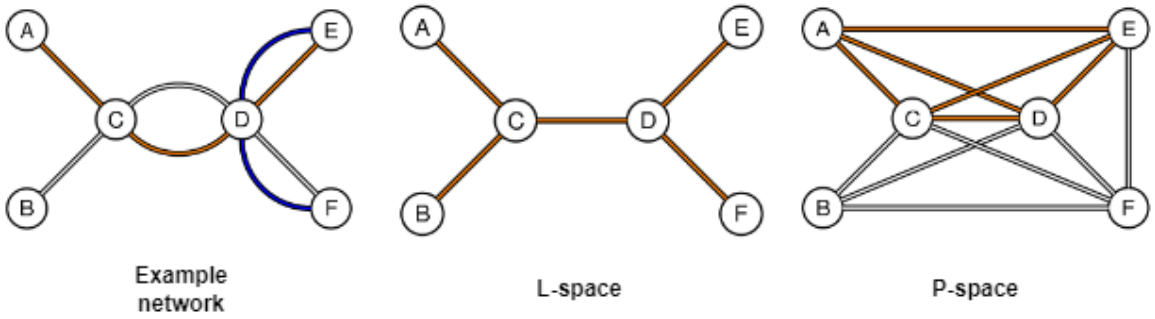


Figure 3.4: Possible representations of a PT network. For the example network the colors resemble the lines. In the L-space network, lines are not considered separately. In the P-space network, the colors do represent the lines, with in some cases an overlap if multiple lines run over the same track (Von Ferber et al., 2009).

In Figure 3.4, the blue line seems to disappear, but the P-space makes no distinction between lines. For that reason, the blue line was 'absorbed' into the red and white edges. The P-space representation allows for a more elaborate analysis of specific routes and the influence of transfer stations on the network. In other words, the P-space offers a more service-relevant network representation. Also, the P-space highlights the relationships between stations in terms of their connectedness, which works well with the exposure part of the vulnerability definition used in this research. As the perspective of this research is from the operational point of view, it was decided to use a non-directional P-space network representation. However, the L-space network representation is used once for the link weight adjustment calculations, described in the paragraph Link Weight.

Link weight A weight is assigned to each link in the network. The mathematical definition of this link weight is shown in Equation 3.3, where j and i represent two connected nodes in the P-space. The infection rate is partially defined by the link weight, which is explained in more detail in the paragraph

Infection rate below.

$$w_{ij} = \frac{1}{t_{ij}} \tag{3.3}$$

The link weight is the inverse of the travel time t_{ij} between nodes i and j. The idea behind this definition is that the propagation strength of a delay diminishes with increasing distance (Jia et al., 2022). Hence, the further away two stations are from each other, the less chance a delay starting at station j has to reach station i.

Every link is initially defined as shown in Equation 3.3, but depending on the instance used for model training w_{ij} is adjusted for specific edges. Three types of edges are adjusted by increasing their weight, with an example to clarify the different types shown in Figure 3.5:

1. Edges between the primary delay node and all other nodes in the direction of the delay on the same line (P-space);
2. Edges between the nodes that are in the direction of the delay and affected by the primary delay (L-space) ;
3. Edges that are part of the L-space path between two nodes and the path between the two nodes includes the primary delay node.

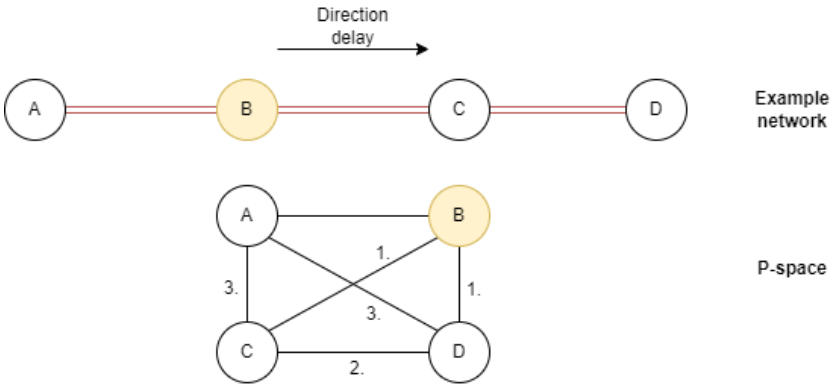


Figure 3.5: Example network and P-space showing which edge types weight would be adjusted with the numbers representing the described edge types. The yellow node is the infected node in the example.

The delay affects those edges and their edge weights must be adjusted to reflect the increased chance of infection given the instance. Type 1 and 2 edges are affected directly. Type 3 edges are affected when the train reaches the primary delay station and are, thus, only partially affected by the primary delay. The edge weights that are adjusted are part of set E. First, the weights of the type 1 and 2 edges in the P-space are changed using the formula shown in Equation 3.4.

$$w_{ij} = w_{ij} + \frac{w_{ij} * \alpha * a_e}{e^{h(p,j)*\gamma}} \tag{3.4}$$

The top part of the fraction represents the factors influencing the impact, while the lower part represents the effect of the delay diminishing with distance. The higher the duration of the primary delay α the more the weights are increased as a more severe delay causes more trouble (Cats & Jenelius, 2018; Marra & Corman, 2020). a_e represents the number of lines passing through a station. The complexity of operations increases with multiple train operation routes (Lu et al., 2021). Hence, if several lines go through a station, the link weights of the edges connected to the station should increase. Also, the further away a station is from the primary delay station, the less impact the delay will have (Jia et al., 2022). This is reflected by the $h(p, j)$, which defines how many nodes are between the primary delay node p and the currently considered node j in the L-space, capturing the space component of the delay propagation. The parameter γ is a scalar and helps capture the delay propagation characteristic of how quickly the propagation effect diminishes with time. This scalar is needed because it has been

found that the delay impact differs depending on the time of the day among other factors (Wang et al., 2019). This parameter is trained.

Finally, the edge weights in the P-space of type 3 are adjusted. The L-space path between, which includes the primary delay node, is determined. It is then calculated which percentage of the edges in the L-space path is before the primary delayed station and which after. The initial link weight is then multiplied by the two percentages. The link weight portion reflecting the edges after the primary delayed station is adjusted using Equation 3.4. That value is then added to the unaffected portion of the link weight. The consequence of Equation 3.4 is that where previously all link weights were between 0 and 1, links that are part of types 1-3 could now have a weight > 1. Therefore, all the edge weights are normalized to keep the infection and recovery rate within the approximately same range of values using Equation 3.5.

$$w_{ij} = \frac{w_{ij}}{\sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij}} \tag{3.5}$$

This normalization formulation helps ensure the relative importance of the different edges is preserved. Other normalization techniques such as the max-min method would force edges to be 0, which is not desired in this research.

Infection rate The infection rate is the exposure of a station to disturbances and, hence, represents the first part of the vulnerability definition given in section 1.1. All trains arriving at station i can carry a delay and infect the station. However, the further away a delay started, the more distance the delay had to travel to infect the station. As mentioned earlier, the propagation strength diminishes with increasing distance; hence, the chance of infection decreases with increasing travel time. At the same time, transfer stations receive many more trains than non-transfer stations as they serve multiple lines increasing the chance of infection (Lu et al., 2021). Moreover, stations connected to a transfer station can also be infected by delays from stations on other lines connected to the transfer station. This delay propagation between lines can only happen for trains using the same infrastructure, which is in line with the vulnerability calculation described earlier in Equation 3.2. Thus, while a station allows passengers to transfer to other lines, that does not make it a transfer station in this model if the other lines run on another infrastructure section k. There is no delay propagation between trains using different infrastructure at the same station. As a result, each section k of station i has an infection rate. This reasoning leads to the infection rate equation shown in Equation 3.6.

$$i_{ik} = \sum_{j}^{J} (w_{ji} + (w_{ji} * \sum_{s \epsilon S, s \neq j,i} w_{sj} * g_j)) + z_{ik} \tag{3.6}$$

In Equation 3.6 three types of stations are considered. The infection rate i_{ik} is determined for section k at station i. The stations j are the stations station i is directly connected to in the P-space, so no transfer is needed. The stations s are not directly connected to station i but can be reached by traveling through a transfer ($g_j = 1$) station j that uses the same infrastructure for both lines.

For each station i the directly connected stations j, which have a direct propagation route, are considered first. This direct propagation is represented with the first w_{ji} in the equation. Then, there is the possibility station i is connected to a transfer station j. The propagation through transfer stations is represented in the inner set of brackets. For each transfer station ($g_j = 1$) the link weights of stations s are summed, which are the stations that station i has no direct link with. The sum is then multiplied by the link connecting the transfer station to station i. g_j has to be 1 for a delay on another line to affect the infection rate of station i. After all the summations, a constant z_{ik} is added to the infection rate, which is a trained parameter, to account for any unobservable factors for that section k.

An example network showing how Equation 3.6 works is displayed in Figure 3.6. In this example, the station only has one section for simplicity.

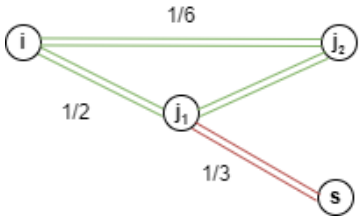


Figure 3.6: Example P-space network with a red and green line showing the difference between the nodes i, j_1 and j_2, and s. Nodes j_1 and j_2 can be reached directly from node i, while to reach node s, a transfer is required at node j_1.

In Figure 3.6 the infection rate is calculated for station i. From station i stations j_1 and j_2 can be reached without transfers as they are on the same green line, so these link weights are added directly. For station j_1 $g_j = 1$, because that station is also connected to the red line, which uses the same infrastructure. For that station j_1 the station s it is connected to is considered as a possible indirect source of delays. Hence, those link weights are also included in the infection rate of station i.

Recovery rate The recovery rate highlights the second part of the vulnerability definition given in section 1.1 and, therefore, is the ability of a station to cope with a disturbance. No instances are considered where rescheduling strategies such as short-turning or reordering were used. Therefore, how well a station i recovers after being infected is dependent in this model on two factors: 1. number of tracks and 2. the traffic density. The assumption is that more tracks result in a better recovery rate as operators have more flexibility in appointing trains to a track. Also, it has been found that the service frequency of a line influences the impact (Marra & Corman, 2020; Yap & Cats, 2020). A disturbance affects lines with higher service frequency more than lines with lower service frequency. There is less flexibility to use recovery strategies for stations with a higher traffic density. Accordingly, a higher traffic density leads to a lower recovery rate. The recovery rate is calculated for each independently operating section of infrastructure k at station i. The number of tracks u_{ki} and the traffic density d_{ki} might differ per section due to different serviced lines. Therefore, the traffic density and the flexibility of the infrastructure are considered per section at a station. All these considerations lead to the equation for the recovery rate presented in Equation 3.7.

$$r_{ik} = c_{ik} + \left(\frac{u_{ki}}{d_{ki}}\right)^{\theta} \tag{3.7}$$

In Equation 3.7 d_{ki}, the traffic density, is calculated using Equation 3.8. This equation calculates the number of trains stopping at a station per hour per infrastructure section k.

$$d_{ki} = \sum_{l}^{L} \frac{60}{h_{lki}} \tag{3.8}$$

In Equation 3.8 h_{lki} is the scheduled headway of the line l running on the set of tracks k at station i and, hence, the formula calculates the number of trains stopping at station i using infrastructure segment k per line l and sums those values. The scheduled headway is used instead of the actual headway because the actual headway might differ among trains, while they are expected to be the same. These headway differences are due to traffic operators changing the headways in response to the disturbance (Cadarso et al., 2013). The actual headways, therefore, do not necessarily paint an accurate picture of the situation at the time.

Combining Equation 3.7 and Equation 3.8, it is calculated per independent set of tracks k at a station i how much traffic passes, d_{ki} and divided by the number of tracks u_{ki}. Stations with more tracks are less vulnerable as the traffic can spread out more. Stations with a high traffic density are more vulnerable as they have to deal with more traffic and, hence, are less flexible.

θ is introduced in Equation 3.7 for the heterogeneity in the recovery rate of stations. A high value for this parameter leads to more heterogeneous recovery rates, while a low value makes the recovery rates more homogeneous across stations. This parameter is trained to see how heterogeneous the recovery rates of the stations are. Another parameter that is trained is c_{ik}. This constant should capture

any unobservable factors influencing the recovery rate, just like z_{ik} does for the infection rate. Both θ and c_{ik} are ≥0.

Vulnerability equation Equation 3.9 shows how the vulnerability of station i is calculated for a specific section k. This equation reflects the definition of vulnerability given earlier in section 1.1, where vulnerability is the difference between exposure to a disturbance (infection) and the ability to cope with the disturbance (recovery). To that end, the vulnerability equation of the model is also written in a way where the vulnerability is determined by the difference in infection and recovery, as shown in Equation 3.9.

$$0 = -r_{ik} * v_{ik,model} + (1 - v_{ik,model}) * i_{ik} \tag{3.9}$$

This formulation of the vulnerability works because a high recovery rate value would lead to a vulnerability value close to 0. Similarly, a high infection rate value would lead to a vulnerability value closer to 1. As a result of a vulnerability value per k, a station could have multiple vulnerability values if it has several infrastructure sections that are all part of the training.

3.4. Model training and testing metrics

The model also needs to be trained and tested. The parameters that are trained are summarized below:

- γ: parameter in the link weight equation capturing the diminishing effect of the delay propagation;
- z_{ik}: parameter in the infection rate equation correcting for any factors currently not considered;
- θ: parameter representing how heterogeneous the recovery rate of stations are in the recovery rate equation;
- c_{ik}: parameter in the recovery rate equation correcting for any currently not considered factors.

The training is done over a group of similar instances, previously explained in section 3.1. The instances are randomly split into two groups: 1. training instances and 2. testing instances. The first group is used to train the model and contains 80% of the instances. The second group tests the trained model and has the remaining 20% instances. The mean squared error is computed for each instance used in the training. This calculation is done based on the vulnerabilities calculated from the data and determined by the model, using the formula shown in Equation 3.10, where N is the number of all stations of all affected lines. The objective function used in the training is to minimize the sum of the MSE of all instances F, which is shown in Equation 3.11.

$$mse = \frac{\sum_{i=1}^{N} (v_{ik,data} - v_{ik,model})^2}{N} \tag{3.10}$$

$$min \sum MSE = \sum_{f=1}^{F} mse_f \tag{3.11}$$

After the training, the trained parameters are used as input for a testing instance to see if the trained parameter values lead to vulnerability values similar to the data values of the testing instance. Again the MSE is used to see how close the vulnerabilities from the model are to the vulnerabilities based on the data. This MSE, however, is calculated using only the considered stations. Additionally, the average vulnerabilities of the model and data are compared and the differences in vulnerability between the data and model are determined for each station. All of this is done per k if a station i has multiple infrastructure sections and they are part of the training, which follows Equation 3.9.

4

Application and case study description

This chapter sheds light on how the model will be applied to an existing metro network, the case study. First, the data itself and its processing as well as how the search for instances was conducted is introduced in section 4.1. Then, how the model was implemented and trained is described in section 4.2. Lastly, the experiments to make certain modeling decisions are discussed in section 4.3. The experiment results motivate the chosen model configuration.

4.1. Case study

This section discusses the case study data used and its processing. The Washington Metropolitan Area Transit Authority (WMATA) provided the data, which is about the Washington DC metro network displayed in Figure 4.1.

Figure 4.1: Map of the Washington Metropolitan Area metro network (Washington Metropolitan Area Transit Authority, 2022).

The Washington DC network currently consists of 98 stations, with the last 7 stations opened at the end of 2022. These stations are served by six different lines of which the Gray, Blue, and Orange share a portion of the infrastructure as do the yellow and green lines.

The available data ranges from 2019 to 2022. The year 2019 was chosen to exclude any influence of COVID-19 on the network and its operations. The headway was increased during COVID-19 as the number of passengers decreased. Larger headways also meant it was less likely for trains to propagate any delays leading to potentially not enough instances to consider in this research. Which data files were used and how is explained in subsection 4.1.1. Afterwards, it is explained in subsection 4.1.2 how the groups of instances were selected.

4.1.1. Data description

Two data files from WMATA were used to perform this research. These files are the Automatic Vehicle Location (AVL) data from 2019 and the station information data. The station information data was supplemented with information from the WMATA website. The upcoming paragraphs explain the processing of each data file.

Station information From the station information file, the station names and their station code are used in this research. While the station information includes all stations open right now, six stations on the Silver Line were not open yet in 2019. Therefore, these six stations were not considered in the study and were deleted from the station information file. Furthermore, some stations in the network have an upper and lower level, resulting in two station codes for those stations. These stations serve multiple lines, which are split between the levels. The WMATA website was used to see what the infrastructure looks like at those stations.

Automatic Vehicle Location data The Automatic Vehicle location data file contains information about when each train reached the stations on its route and at what time. This data file only contained information about August through December. However, a part of the Blue and Yellow lines were under maintenance in August, so this data was excluded. Also, there are data points for the station Potomac Yard in the Automatic Vehicle Location (AVL) data. This station was not open yet and so these data points were deleted. Moreover, the station codes in the AVL data for stations that previously had two station codes were changed to the one station code from the station information file. Then, the AVL data was used to determine the scheduled headway of each line at the time of the primary delay instances. Lastly, to make the vulnerability calculations easier all the train movements were sorted based on station, line, and direction.

4.1.2. Instance selection from the data

The data was searched for instances with a similar primary delay to train and test the model. The aim was to collect at least 10 similar instances that showed that delay propagation happened between the metro vehicles running on the same infrastructure, irrespective of their lines. The 2019 AVL dataset was divided into weeks. Then it was determined for each train of each day of each week if it propagated its delay to another train to make searching for instances easier using the conditions described in section 3.1. The primary delay had to be at least five minutes. A delay smaller than five minutes has a limited delay propagation effect. It is more interesting to consider the cases where the delay propagates and the affected trains stay delayed for a few stations. The delay could also not be a very high value, because a too-high delay duration would cause the traffic controllers to intervene with measures not modeled at the moment such as short-turning.

Also, maintenance work is done all the time on the network with station closures as a possible consequence. Hence, an important criterion for the instance selection was that all stations were open on the lines considered in the case to ensure a complete picture of the effects of the delay propagation on the considered lines.

Ultimately, two groups of instances were used to train and test the model. The first group has 10 instances with a primary delay starting at the King St-Old Town station. This group of instances contains both Yellow and Blue trains delayed in the direction of Greenbelt and Downtown Largo respectively. The stations considered are between King St-Old Town and Rosslyn, and King St-Old Town, and L'Enfant Plaza as shown in Figure 4.2.

Figure 4.2: Map showing which stations are considered for group 1. The arrows indicate the traveling direction of the considered train movements.

The second group also has 10 instances with the primary delay starting at Stadium-Armory station. Blue, Silver, and Orange trains are delayed in the direction of Franconia-Springfield, Wiehle-Reston East, and Vienna respectively. The stations between Stadium-Armory and Rosslyn are considered and shown in Figure 4.3.

Figure 4.3: Map showing which stations are considered for group 2. The arrow indicates the traveling direction of the considered train movements.

An overview of all groups with more information about the specific instances used in this research and how the groups were split can be found in Appendix B. The vulnerabilities were calculated using the data for each of the instances used in each group. Those values were used in the training or testing of the model through the MSE calculations. For the vulnerability calculations, the delay threshold was set to two minutes. Any delay smaller than two minutes is attributed to service variability. The vulnerabilities of the not-considered stations were forced to be 0. The stations cannot be removed from the model as they are needed to construct the graph, because the edge weights and infection rate values are calculated from the graph links. The groups chosen to test the model contain instances that only influence one section k at stations with the currently considered stations. Subsequently, the stations, which could have multiple vulnerability values due to several infrastructure sections, only have one vulnerability value.

4.2. Model implementation

A few steps were taken to train the model. The Washington network had to be created to calculate the infection and recovery rate of each station, which is explained in more detail in subsection 4.2.1. The settings of the model will be outlined in subsection 4.2.2.

4.2.1. Network creation

A P-space and L-space network graph had to be created to train the model parameters. This was done using the NetworkX library in Python. In previously done research, the needed P-space and L-space graphs were created using NetworkX and made available to any user (Cats et al., 2019). These P-space and L-space files were used to create the necessary network graphs. The labeling of the nodes in the L-space and P-space was not in the same order, so adjustments were made to make them the same. Each of the edges in the L-space and the P-space graph got the travel time between two nodes as an attribute. These travel times were obtained from the same source. The L-space travel times were directly obtained from the same dataset and the P-space travel times were calculated based on the L-space. Other attributes given to the nodes were whether they are transfer stations, meaning here passengers can transfer to a train from another line that runs on the same set of tracks and comes from the same direction but when leaving the station will travel to another station than the trains of the other lines. Example stations are Stadium-Armory (all lines use the same infrastructure, but the Orange line goes in another direction in the east direction) and Pentagon (the Blue and Yellow trains use the same infrastructure at the station, but split when they go north).

4.2.2. Training settings

After all the data was prepared and the vulnerabilities were calculated for each instance, the model was trained on each group of instances. In a group of instances, 80% were used to train the model and 20% to test it.

To train the model differential evolution from the Python package SciPy was used. SciPy is a global optimization algorithm designed to solve non-linear and non-convex problems (SciPy, n.d.). As the problem is non-linear and non-convex a traditional gradient-based method would struggle. The idea of the algorithm is that it maintains a population of candidate solutions that evolve over iterations. These candidate solutions are vectors of parameters. Each iteration the algorithm will mutate a randomly chosen candidate vector and recombine it with the current candidate vector to create a new vector. These two vectors will then compete and the vector with a better objection function value will enter the population. The algorithm will stop when a maximum number of iterations is reached or the desired precision is achieved.

To recap the following parameters were trained and they have the following bounds:

- γ: parameter in the link weight equation capturing the diminishing effect of the delay propagation. The bounds are [0, 2];
- z_{ik}: parameter in the infection rate equation correcting for any factors currently not considered. The bounds are [0, 0.5];
- θ: parameter representing how heterogeneous the recovery rate of stations are in the recovery rate equation. The bounds are [0.5, 2];
- c_{ik}: parameter in the recovery rate equation correcting for any factors currently not considered. The bounds are [0, 0.5].

The parameters γ and θ cannot be negative, because in the case of γ a delay always travels forward and θ is an exponent that should enlarge or decrease an effect. c_{ik} can also never be smaller than zero, because it is meant as a variable that captures any additional measures taken to improve the situation in the network. z_{ik} needs to be a positive value to capture the effects of a train having more effects on the operations than the infection rate would otherwise capture.

The range of the c_{ik} and z_{ik} was chosen because much larger values would lead to model overfitting. Instead of capturing the underlying patterns in the data, the training algorithm would try to find a combination of parameters making the model 0.0001 more accurate for the vulnerability values. A low value for these two parameters would mean the model has to barely adjust to get close to the

vulnerability values from the data. High values would mean that one or multiple factors are not taken into account yet by the model and the parameters have to adjust for this. The range for γ and θ was determined through experimentation. A low γ means the edge weights are barely adjusted and this parameter would have limited influence on the infection rate. A high value for γ means the edge weights are adjusted and the infection rate of stations with adjusted edges will increase. Similarly, θ must be at least 0.5 to force the model to use the parameter. Otherwise, the model makes the parameter θ 0 and tries to compensate everything with the constants c_{ik} and z_{ik}. If a high value is trained for θ the recovery rates of all the stations in the network are not similar at all and the model tries to increase these differences. A low value, on the other hand, would mean the recovery rates are forced to be similar and the difference between the stations is negligible.

4.3. Model configuration

The experiments described in this section were conducted to make model configuration decisions using the case study network. Each experiment involved modifying one specific aspect of the model while keeping others consistent with the basic model configuration. The basic model configuration, introduced in subsection 3.3.2, is summarized as follows:

- Only stations reached by all affected trains are considered during training;
- The network is represented as an undirected graph in the P-space.

An overview of all experiments and the corresponding model configurations is provided in Table 4.1. Each experiment tested variations of one model element to analyze its impact while retaining the shared characteristics of the basic model configuration. In other words, for experiment 1 and 2 the first model configuration corresponds with the basis model.

Table 4.1: An overview of all the experiments to make the model configuration choices. The basis model configuration is the same across experiments and corresponds with configuration 1 for experiments 1 and 2.

	Model configuration 1	**Model configuration 2**
Experiment 1 (subsection 4.3.1)	Stations reached by all affected trains considered in the training	All stations of all affected lines considered in the training
Experiment 2 (subsection 4.3.2)	Undirected graph of the P-space	Directed graph of the P-space

4.3.1. Station configuration

As briefly explained in section 3.2, a choice about which stations to include in the model training had to be made. The options were to either include the stations that each affected train passed with a delay or all the stations of the affected lines. In the first case, only a small selection of the stations would be included. The advantage of this smaller selection is that the vulnerability values range for these stations is small. Consequently, it will be easier for the training algorithm to find parameter values that fit these vulnerability values. A disadvantage of this approach is that a part of the delay propagation picture disappears. If a network operator uses this model in the future, they will not have the full picture of the delay propagation situation. However, the model might not be able to handle the situation where all stations are considered. As not every station is passed by delayed trains in each instance the range of vulnerability values is much larger. Hence, the trained parameter values must work for a wider range of values, which is much harder to do as the model parameters are designed to increase or decrease the infection rate and recovery rate values the same for each station across all instances.

This trade-off of both model versions was experimented with. For both models the same training settings were used, such that the only difference is the number of stations included in the training. The training results of this experiment can be found in Table 4.2. Again, the MSE values in Table 4.2 were calculated by Equation 3.11.

Table 4.2: Training results for experiment 1 using group 1: King St - Old Town.

	Stations reached by affected trains with delay	*All stations of affected lines*
MSE	0.022	0.141

The model performance with the configuration where only the stations reached by an affected train with delay is better. Its MSE value is over 0.100 lower than for the model configuration where all stations of all affected lines are included. To gain more insights into where this difference in performance comes from the trained values for the parameters are investigated. These values are presented in Table 4.3.

Table 4.3: Trained parameters for group 1: King St - Old Town when either a limited set of stations is considered or all stations of all affected lines.

	Stations reached by affected trains with delay		**All stations of affected lines**	
γ		0.449		0.324
θ		0.500		0.500
Stations	z_{ik}	c_{ik}	z_{ik}	c_{ik}
Arlington Cemetery	0.254	0.451	0.500	0.170
Braddock Road	0.390	0.007	0.093	0.240
Crystal City	0.264	0.248	0.254	0.398
King St-Old Town	0.303	0.495	0.125	0.344
L'Enfant Plaza	0.222	0.127	0.039	0.105
Pentagon	0.269	0.417	0.248	0.227
Pentagon City	0.412	0.457	0.115	0.158
Reagan National Airport	0.217	0.378	0.235	0.213
Rosslyn	0.150	0.358	0.243	0.433

The γ value is lower for the second configuration, which means the second configuration has lower infection rates. Also, for some stations where for the first model configuration the z_{ik} was much higher than the c_{ik} value, meaning during training it was decided to compensate the infection rate values more than the recovery rate values, for the second model configuration it is the opposite. An example of this is the Arlington Cemetry station. Therefore, the model makes other decisions depending on the model configuration.

While the model version where all stations are included gives a more complete picture of the situation, that model configuration does not perform well. For now, the model version with fewer stations is chosen, but the limitations of this approach are recognized.

4.3.2. Network graph configuration

Another model configuration consideration experimented with was whether the graph used in the calculations should be undirected or directed. To help the explanation, let us define two nodes, u and v, and two directed edges and an undirected edge between nodes u and v. Also, the delayed train travels from node u to node v.

With an undirected graph, stations u and v would add the edge weight of edge [u,v] to their infection rate. However, a delay is directional and the u station should not feel the same infection chance from that edge as station v. Node u should possibly not even feel the delay effects from that edge as the trains have already passed station u. With a directed graph node u would add the weight of the edge [v,u] to its infection rate, while node v would add the weight of the edge [u,v]. As the delay travels from node u to node v, the edge weight of edge [u,v] would be increased, while the edge weight of edge [v,u] would stay the same. Then node v would feel the effects of the delayed train traveling from u to v, while node u does not.

The same training settings were used for both model configurations, such that the only difference would be the undirected and directed graph. The training results are presented in Table 4.4 and determined by Equation 3.11.

Table 4.4: Training results for experiment 2.

	Undirected P-space graph	*Directed P-space graph*
MSE	0.022	0.019

The performance difference between the two model configurations is small. The model configuration with the directed P-space graph performs a bit better. Where the performance difference is coming from will be better understood by looking at the trained parameter values.

Table 4.5: Trained parameters for group 1: King St - Old Town when the graph is undirected and directed.

	Undirected graph		**Directed graph**	
γ		0.449		0.324
θ		0.500		0.500
Stations	z_{ik}	c_{ik}	z_{ik}	c_{ik}
Arlington Cemetery	0.254	0.451	0.438	0.071
Braddock Road	0.390	0.007	0.281	0.100
Crystal City	0.264	0.248	0.250	0.179
King St-Old Town	0.303	0.495	0.184	0.368
L'Enfant Plaza	0.222	0.127	0.303	0.435
Pentagon	0.269	0.417	0.373	0.249
Pentagon City	0.412	0.457	0.266	0.301
Reagan National Airport	0.217	0.378	0.358	0.207
Rosslyn	0.150	0.358	0.218	0.345

The γ value for the model configuration with the directed graph is lower than when the graph is undirected. Also, there is a clear change in trained values for both z_{ik} and c_{ik}, while the value θ has stayed the same, implying that any difference in model performance can be attributed to the other parameters. For example, when the graph is undirected the c_{ik} value for the station Arlington cemetery is very high. When the graph is directed, however, the value is almost 0. The opposite can be seen for the L'Enfant Plaza station, where the c_{ik} value is much higher for the directed graph model configuration than for the undirected. The training algorithm makes different choices when the graph is undirected versus directed. Whether the graph is undirected or directed will influence the infection rate of each station, leading to different trained values that fit the data the best.

The largest disadvantage of the directed graph approach can be explained by considering the King St-Old town station. At this station, the primary delay started for group 1. It would have, therefore, no incoming edges with the edge weight adjusted if the graph is directed. As a result, its infection rate will be too low to reflect that it also receives trains with a delayed arrival. The model will then use z_{ik} to increase its vulnerability. In the future, the model will be improved and the boundaries on the constants will be tightened. The structural disadvantage of the directed model version will persist. Therefore, the undirected graph model version was used for the remainder of this study.

5 Results

This chapter discusses the study results based on the chosen model configuration. First, an analysis of the data and the model itself is presented in section 5.1. Afterwards, the testing results of group 1 are discussed in section 5.2 followed by the training and testing results of group 2 in section 5.3. The results of both groups are compared in section 5.4. The sensitivity analysis is presented in section 5.5. Then, the benchmarking results are shown in section 5.6. The chapter ends with a summary of the findings in section 5.7. All instances belonging to group 1 have a unique number between 1 and 10, making it easier to refer to specific instances. Similarly, all group 2 instances have a unique number between 11 and 21.

5.1. Exploratory data and model analysis

The available data and the individual model component's results are explored in this section. Insights from the data are discussed in subsection 5.1.1. An exploratory analysis of the individual model components is given in subsection 5.1.2.

5.1.1. Insights from data

The months September through December were searched for suitable instances. After all the possible instances were collected, stations with a minimum of 10 primary delays resulting in delay propagations in one direction were searched for. A minimum of 10 delay propagations was required to have sufficient data to train and test with. How many instances were found per station is presented in Figure 5.1.

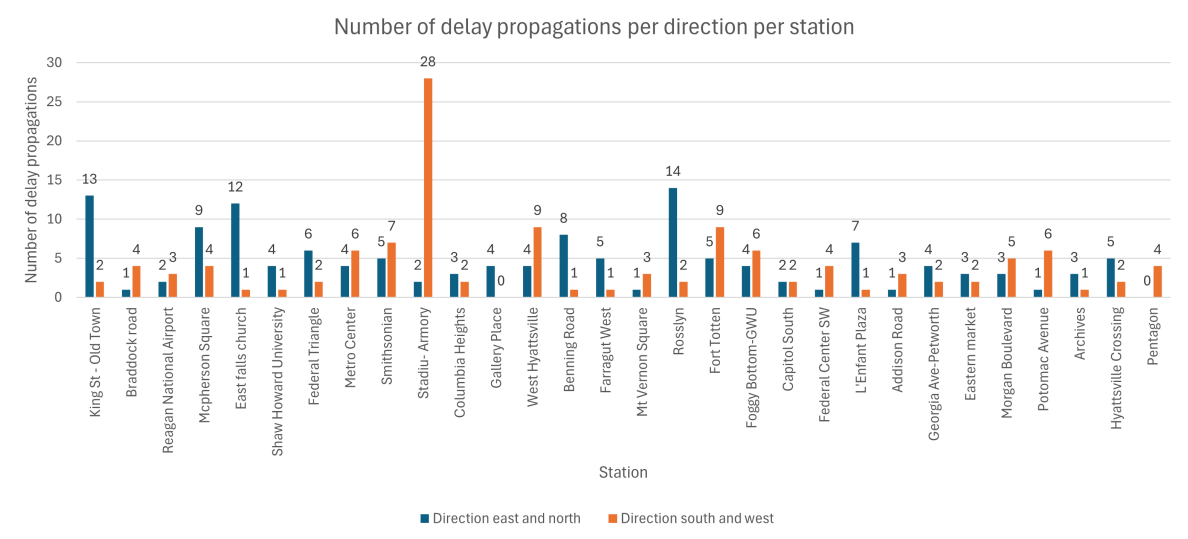


Figure 5.1: Graph showing the number of delay propagations per station and direction.

Stations with less than three delay propagations were not included in the graph for readability, mostly stations serving only the Red line. These stations are included in the statistics in the graphs presented later. Figure 5.1 shows very few stations have more than ten instances of delay propagation in one direction. While this low number could suggest that the Washington Metro network handles delay and possible propagations well, it also limits the data availability for this research. The only stations with more than 10 instances in one direction are the King St-Old Town station in the north direction (which is group 1), the East Falls Church station in the east direction, the Stadium-Armory station in the west direction (which is group 2) and the Rosslyn station in the east direction. Consequently, only four options were available to use in this research. The group of instances for Stadium-Armory is the only group in the west/south direction, so it was important to use this group of instances for direction diversity. The King St-Old town instances were chosen to diversify the lines considered in model training. The East Falls Church and Rosslyn station cover the same lines as the Stadium-Armory station.

It is interesting to note that the stations with the highest number of delay propagations are transfer stations, which matches the findings of the studies done by Cats et al (2016) and Lu et al. (2021). Figure 5.2 sums up all the delay propagations per line per direction.

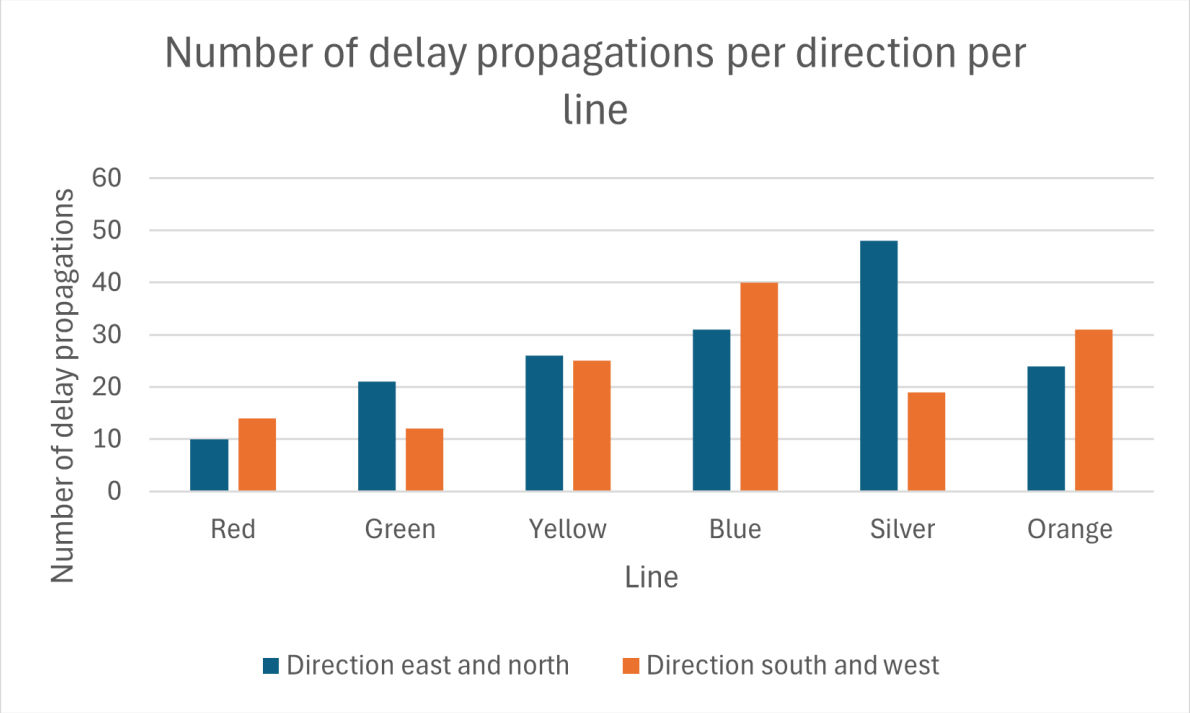


Figure 5.2: Graph showing the number of delay propagations per line and direction.

Figure 5.2 shows that the most delay propagations for the Green line happen in the northern direction. However, the Green line has much fewer delay propagations compared to the Yellow line. The Yellow line has to merge with/split from another line at three separate stations. Its operations are more complicated. The number of delay propagations in both directions for the Yellow line is almost the same. The number of delay propagations is higher in the south and west direction for the Blue line. The Blue line has to merge with another line at two stations in that direction, making operations more difficult as traffic density increases. The east direction dominates the Silver line delay propagations. In that direction, the Silver line becomes part of ever increasing traffic density. The Silver line first merges with the Orange line at East falls church station and the Blue line is added at Rosslyn, further increasing the traffic density that the Silver line is part of. The Orange line has a bit more delay propagations in the west direction. It does have to merge with the Blue and Silver lines all at once at the Stadium-Armory station, while in other parts of the network, it is just the Silver line it shares the infrastructure with.

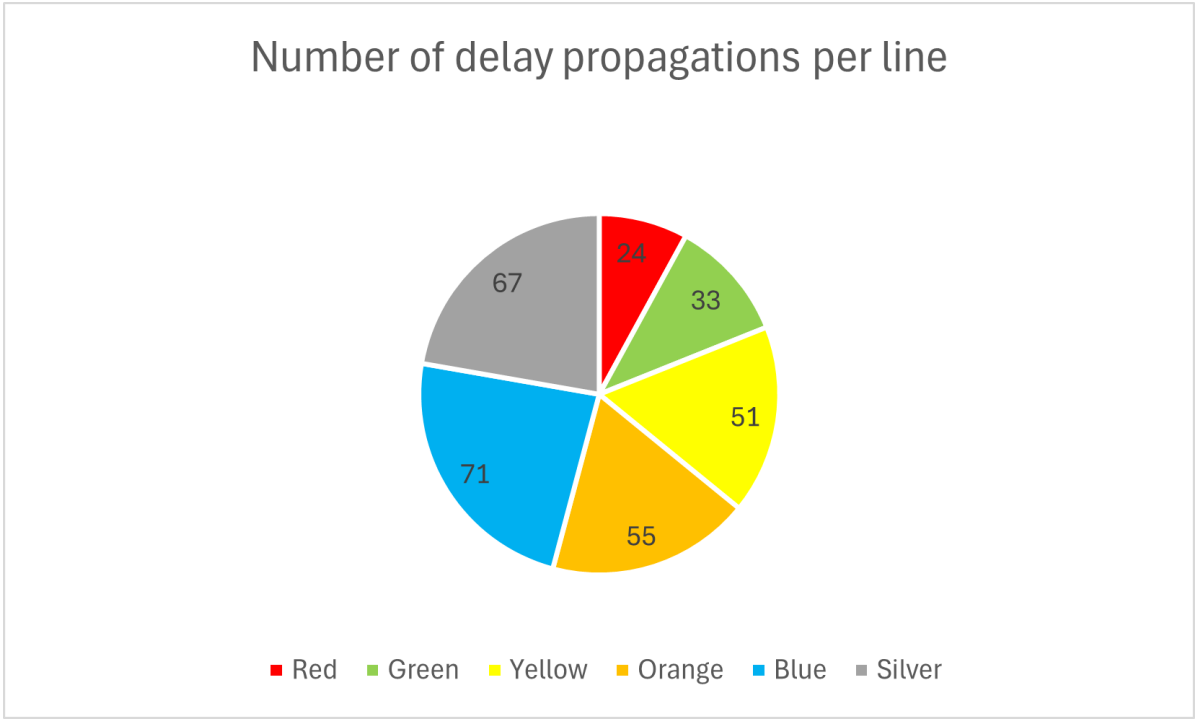


Figure 5.3: Graph showing the total number of delay propagations per line.

Figure 5.3 shows all the delay propagations summed up per line. It can be seen from Figure 5.3 that the Blue, Silver, and Orange lines have the most delay propagations. These lines also share most of their infrastructure. The Blue line, which shares most of its line with other lines, has the highest number of delay propagations, followed by the Silver and Orange lines. The Yellow line also shares most of its infrastructure but with fewer other lines compared to the Blue, Orange, and Silver lines. The Green line shares around half of its infrastructure and, thus, is second to last in the number of delay propagations. Having to share infrastructure with other lines, thus, means delays propagate more easily. This insight follows the findings from the research done by Lu et al. (2021) that multiple train routes make daily operations more complex. The Red line, which shares none of its infrastructure, has by far the least number of delay propagations. At the same time, this low number also means that the model could not be tested on this part of the network. A minimum of 10 instances was required to ensure enough instances could be used for training and testing. For none of the stations serving the Red line even close to 10 instances were found.

5.1.2. Analysis model components

It is also interesting to look at some model components. The infection and recovery rates of the case study network stations are analyzed in more detail below.

Infection rates The infection rates of all the stations in the Washington metro network are shown in Figure 5.4. The z_{ik} values are not added to the infection rate values yet and no edge weights were adjusted. As a result, the infection rates presented here represent the baseline values, calculated solely based on the initial edge weights of the undirected P-space graph, without incorporating any adjustments from parameters or specific instances. Additionally, for stations with multiple infrastructure sections only the highest infection rate value is displayed for readability.

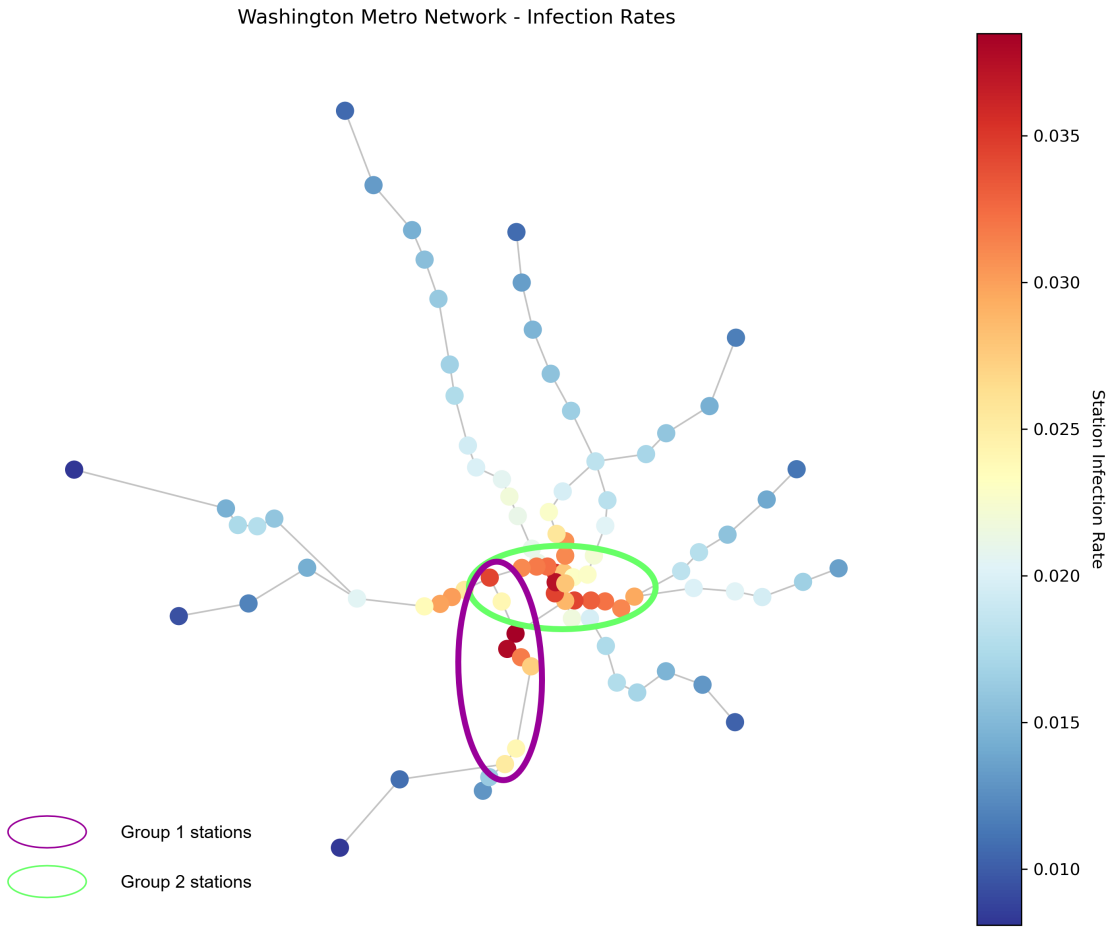


Figure 5.4: Graph showing baseline infection rates of each station in the case study network. For stations with multiple infrastructure layers, only one value is shown. The colored circles indicate the stations belonging to one of the two groups.

The transfer stations exhibit higher infection rates because the formulation of the infection rate amplifies how well stations are connected. Also, stations close to the transfer stations will have a higher infection rate because of their close connection to the transfer station. The central stations are well connected to most other stations and the travel times are low because of the central station location. The edge weights of these stations are among the highest in the graph. Stations on the outskirts of the network are less connected to the other stations and the travel times are higher resulting in lower edge weights. As a result, central stations have higher baseline infection rates compared to non-central stations. The further the station is from the center, the lower the infection rate.

A downside of the infection rate equation (Equation 3.6) is that when a primary delay starts near a terminal station a high z_{ik} is needed to increase the infection rate for that station above the terminal stations' infection rate. Another downside is that the stations near the end of lines often have a high recovery rate because fewer lines run through those stations. The z_{ik} might need to compensate for the infection rate of those stations with a high value to overcome this higher recovery rate and have a high vulnerability value. Depending on the boundary set for the parameters and the factors included in the model, the z_{ik} might or might not be able to compensate for the infection rate of those stations enough.

Recovery rates The stations in the middle of the network are expected to have a low baseline recovery rate because they have a high traffic density due to more lines serving those stations. Stations near the end of lines, on the other hand, are expected to have a high baseline recovery rate as they often serve only one line. These expectations are based on the recovery rate equation (Equation 3.7) without the parameters θ and c_{ik}. The baseline recovery rates of the Washington metro network for

instance 3 are presented in Figure 5.5. Also, for the stations with multiple recovery rate values, one value was chosen to show on the map for readability.

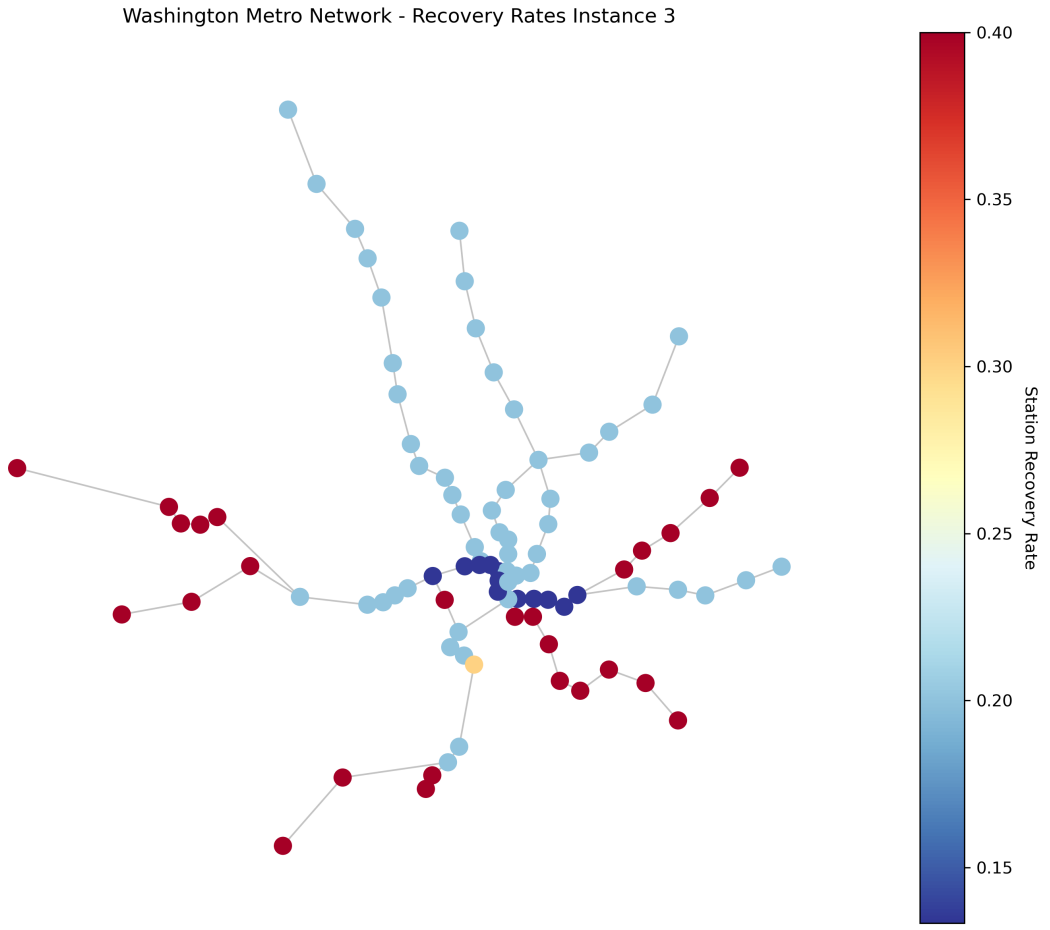


Figure 5.5: Graph showing recovery rates of each station for instance 3.

Stations serving the same lines have the same recovery rate, except for Reagan National Airport, which has three instead of two tracks. Also interesting to note is that the stations serving the Red line and the Green and Yellow lines have the same recovery rate, which are light blue nodes in the top half of the figure. The Red line operates independently on its tracks with a short headway (6 minutes), leading to a traffic density comparable to that of the Green and Yellow lines, which achieve together a similar traffic density despite having longer headways (12 minutes both). Traffic density is a very influential variable of the recovery rate. The scheduled headways at the time determine the traffic density, which means it can differ from instance to instance. To show how the headway influences the recovery rate, Figure 5.6 shows the recovery rates for instance 6.

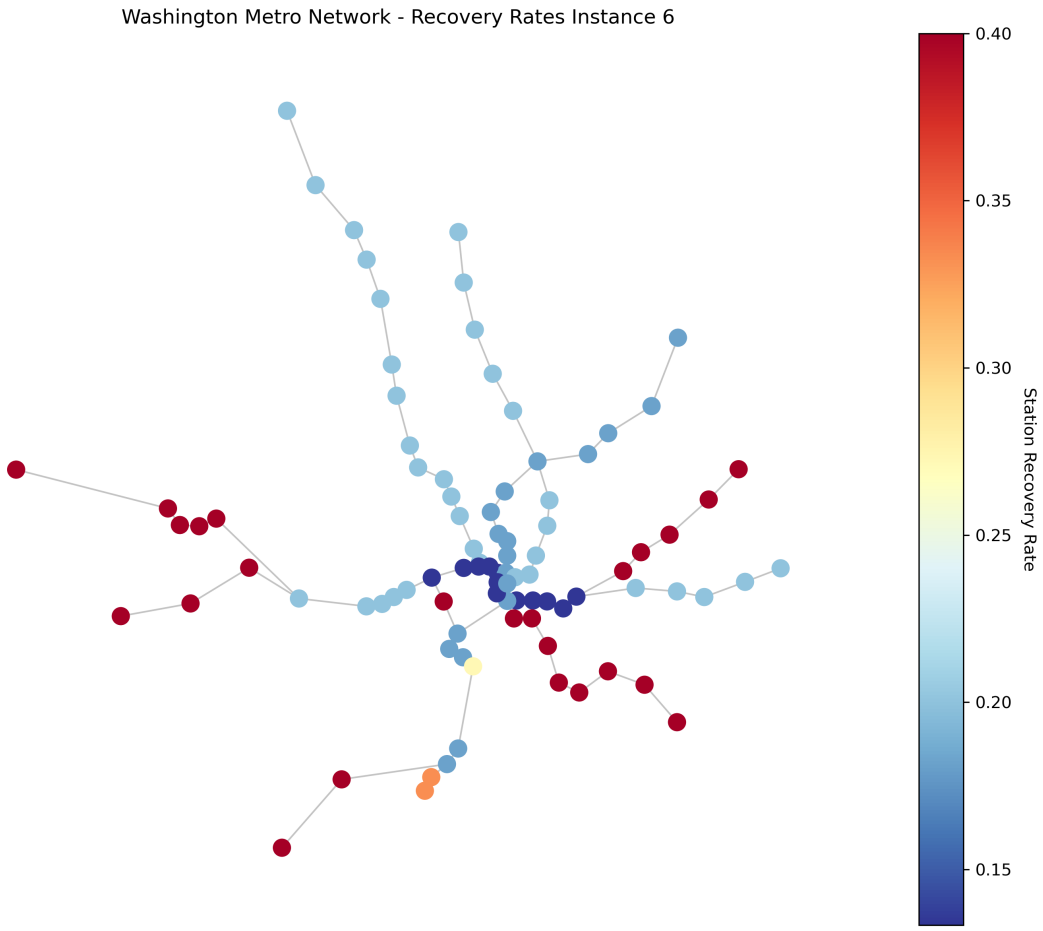


Figure 5.6: Graph showing recovery rates of each station for instance 6.

Figure 5.5 and Figure 5.6 show for most stations a very similar recovery rate. However, the influence of traffic density can be seen in some parts of the network. For example, in Figure 5.6 the stations from L'Enfant Plaza to Greenbelt have a lower recovery rate than in Figure 5.5. This difference is because the headway at the time of instance 6 was lower for the Yellow and Green lines. Therefore, the traffic density was higher at the time of instance 6 than at instance 3. The graph will then also be different for instances with completely different headways from instances 3 and 6. This difference in recovery rates will also influence the model training. The model might have to compensate more for one instance than for another, but the model needs to find a parameter fitting both instances. The recovery rates are, therefore, almost completely dependent on the traffic density at a station.

5.2. Results group 1: King street-Old Town station

In the two subsections coming up the results for group 1 will be presented, first the vulnerability calculation results from the data followed by the model.

5.2.1. Vulnerability results from the data for group 1

This group contains 10 instances. The vulnerability based on the data for each considered station was calculated for each instance. On average 9 trains are within the time window at each station and, hence, considered in the vulnerability calculations. These calculated vulnerabilities are shown in Table 5.1 and visualized in Figure 5.7.

Table 5.1: Vulnerability of each station calculated from the data per instance from group 1 based on the data.

	Instances									
Stations	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
King St - Old Town	0.625	0.292	0.333	0.333	0.417	0.292	0.292	0.417	0.333	0.292
Braddock Road	0.666	0.292	0.333	0.417	0.500	0.375	0.292	0.417	0.417	0.333
Reagan National Airport	0.625	0.333	0.333	0.333	0.417	0.333	0.292	0.417	0.500	0.333
Crystal City	0.625	0.292	0.333	0.333	0.500	0.333	0.333	0.167	0.417	0.292
Pentagon City	0.583	0.292	0.333	0.333	0.417	0.333	0.292	0.167	0.500	0.292
Pentagon	0.583	0.292	0.333	0.333	0.417	0.333	0.333	0.417	0.417	0.292
Arlington Cemetery	0.333	0.500	0.333	0.500	0.500	0.333	0.250	0.500	0.500	0.333
Rosslyn	0.000	0.278	0.167	0.389	0.611	0.278	0.194	0.278	0.167	0.111
L'Enfant Plaza	0.333	0.583	0.375	0.250	0.250	0.417	0.125	0.167	0.500	0.167

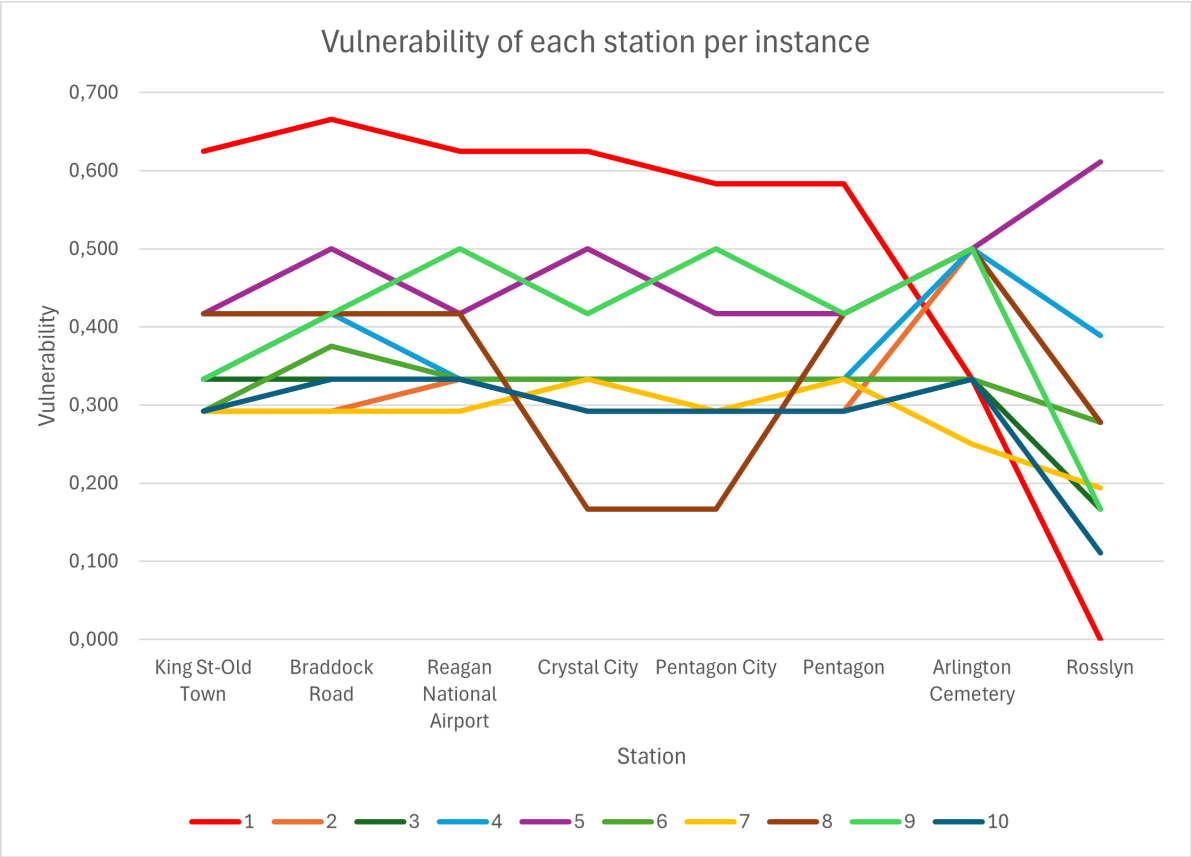


Figure 5.7: Vulnerability of each station per instance from group 1 in order of stations reached by the trains. The graph does not include L'Enfant Plaza station for visualization reasons.

The vulnerabilities were expected to decrease with increasing distance from the primary delay station. A delayed train is expected to catch up on its delay as it uses its buffer time. Most instances have a lower vulnerability for the Rosslyn station, the last considered station. Nevertheless, for some instances, the vulnerabilities fluctuate while for other instances the vulnerabilities stay similar. The only instance that follows the expected trend is the first instance. The reason why the instances do not follow the expected trend could be that randomly delayed trains are also included in the vulnerability calculations. As it takes some time for the primary delayed train and the other affected trains to reach the further stations, other trains will already have reached those stations in the time window. If those other trains are randomly delayed, they would influence the vulnerability calculations. Another explanation is that the arrival time of trains is just above or under the two-minute threshold, making it delayed at one station, but not at the other. An example showing both points can be found in the train

movement data of instance 2. This instance has a train arriving at L'Enfant Plaza station before the affected trains and is delayed by only 15 seconds. This train was not considered delayed at the previous station. As a consequence, this train will increase the vulnerability of the L'Enfant Plaza station, while it was unaffected by the primary delay train. One last explanation is that the number of trains used in the vulnerability calculations is not the same for each station. A train might arrive at a station a minute before or after the time window cutoff time.

Furthermore, more factors seem to influence the primary delay and its propagation effects. For instances 3 and 8, the primary delay is a Blue line train, which is five minutes delayed at around 11:20 am on a weekday and travels with a headway of 12 minutes. At first sight, these instances would be deemed very similar and, as a result, the influence of these primary delays would be expected to be similar. However, Figure 5.7 shows that the vulnerability trend for these two instances across stations is very different. While the vulnerabilities of the stations for instance 8 heavily fluctuate, the vulnerabilities of instance 3 stay relatively stable. Hence, other factors must explain this difference in delay propagation effects.

5.2.2. Model testing results group 1

In this subsection, only the testing results of group 1 are presented. The training results were presented in section 4.3 and are not repeated here. After the model was trained on eight out of 10 instances from group 1, it was tested on instances 3 and 6. The metrics for a first idea of how the model performed during testing are outlined in Table 5.2. The MSE values presented in the table are based on Equation 3.10 only.

Table 5.2: Testing metrics for group 1.

Metric	*Testing instance 1*	*Testing instance 2*
MSE	0.004	0.004
Average vulnerability data	0.319	0.336
Average vulnerability model	0.295	0.299

The MSE value for both testing instances is 0.004, indicating that the model's predicted vulnerability values for the stations deviate, on average, by no more than 0.07 ($\sqrt{0.004}$) from the actual data values. For this group 0.07 means the model was off by less than one train. The data and model averages differ for both instances, but the difference between them is small. For both instances, the average vulnerability of the data was higher than the model; hence, the model underestimated the vulnerabilities for these instances. To gain more insights as to why the model might have underestimated the values, the differences in vulnerability between the data and the model of each station must be examined. The difference between the model estimation and the vulnerability calculation based on the data are calculated using Equation 5.1.

$$\delta_{ik} = v_{ik,data} - v_{ik,model} \tag{5.1}$$

A negative value for δ_i means the model overestimated the vulnerability of station i and a positive one means the model underestimated the vulnerability. A value close to 0 is desired, because then the model found a value very similar to the vulnerability calculated from the data. The vulnerability differences are shown in Table 5.3.

Table 5.3: Comparison of vulnerabilities as determined from the data and by the model for group 1.

	Testing instance 1			Testing instance 2		
Stations	*Data*	*Model*	*Difference*	*Data*	*Model*	*Difference*
Arlington Cemetery	0.333	0.399	-0.066	0.333	0.397	-0.064
Braddock Road	0.333	0.265	0.068	0.375	0.277	0.098
Crystal City	0.333	0.288	0.045	0.333	0.294	0.039
King St-Old Town	0.333	0.235	0.098	0.292	0.240	0.052
L'Enfant Plaza	0.375	0.330	0.045	0.417	0.328	0.089
Pentagon	0.333	0.317	0.016	0.333	0.324	0.009
Pentagon City	0.333	0.275	0.058	0.333	0.281	0.052
Reagan National Airport	0.333	0.279	0.054	0.333	0.290	0.043
Rosslyn	0.167	0.266	-0.099	0.278	0.264	0.014

If one looks at Table 5.3 it makes sense why the average vulnerability from the model is lower than the average vulnerability based on the data for both testing instances. The model underestimates the vulnerabilities of almost all stations. Only the stations Arlington Cemetery and Rosslyn were slightly overestimated. Looking back at the training results presented in section 4.3, the algorithm chose during training to compensate the stations with a too-low recovery rate instead of the stations with a too-low infection rate as indicated by the higher values of the recovery rate parameters. This decision of the training algorithm means that the recovery rates were increased, leading to higher recovery rate values compared to the infection rate values, meaning lower vulnerability values. Hence, the testing instances results will also show underestimation if the vulnerability values of those instances are similar to the vulnerability values of the training instances, which they are in this case.

5.3. Results group 2: Stadium-Armory station

This section presents the results for the second group of instances. First, the results of the vulnerability calculations based on the data are discussed, followed by the training and testing results.

5.3.1. Vulnerability results from the data for group 2

This group also contains 10 instances. For group 2 the vulnerability calculations were based on 14 trains on average. The calculated vulnerabilities are shown in Table 5.4 and visualized in Figure 5.8.

Table 5.4: Vulnerability of each station calculated from the data per instance from group 2 based on the data.

	Instances									
Stations	11	12	13	14	15	16	17	18	19	20
Stadium-Armory	0.278	0.547	0.389	0.333	0.289	0.500	0.250	0.389	0.306	0.444
Potomac Ave	0.278	0.735	0.417	0.333	0.283	0.500	0.233	0.389	0.278	0.533
Eastern Market	0.306	0.698	0.306	0.389	0.328	0.500	0.217	0.417	0.306	0.617
Capitol South	0.306	0.698	0.306	0.389	0.278	0.500	0.233	0.361	0.300	0.600
Federal Center SW	0.306	0.676	0.361	0.333	0.283	0.500	0.250	0.361	0.328	0.65
L'Enfant Plaza	0.306	0.676	0.328	0.333	0.311	0.500	0.250	0.333	0.217	0.567
Smithsonian	0.333	0.700	0.361	0.333	0.283	0.500	0.306	0.361	0.278	0.583
Federal Triangle	0.333	0.767	0.344	0.333	0.311	0.333	0.361	0.417	0.356	0.583
Metro Center	0.389	0.722	0.317	0.444	0.283	0.500	0.444	0.417	0.267	0.617
Mcpherson Square	0.444	0.611	0.283	0.500	0.311	0.444	0.361	0.306	0.333	0.593
Farragut West	0.500	0.476	0.233	0.611	0.317	0.278	0.417	0.306	0.278	0.593
Foggy Bottom-GWU	0.556	0.448	0.489	0.611	0.378	0.278	0.528	0.306	0.222	0.510
Rosslyn	0.500	0.572	0.633	0.667	0.350	0.278	0.611	0.250	0.167	0.533

The range of vulnerability values is larger for group 2 than for group 1. Furthermore, the vulnerabilities of stations of group 2 also tend to fluctuate and not show a smooth trend. The middle part of the network, which the stations considered in group 2 are part of, has a high traffic density. Even if the trains would want to catch up on their delay, the high traffic density makes it hard to do so. The high traffic density would explain why it takes some time for most instances to show a downward trend of

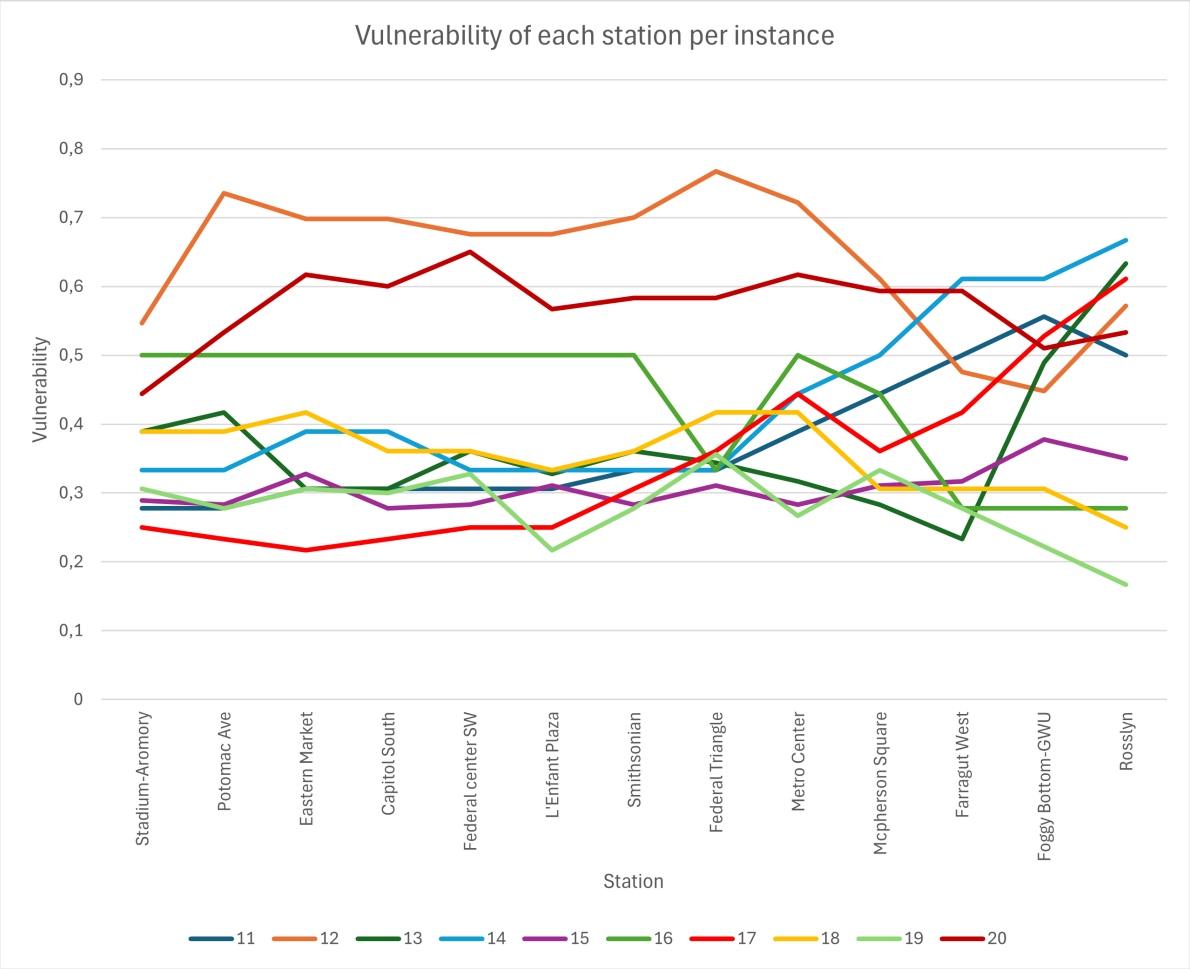


Figure 5.8: Vulnerability of each station per instance from group 2 in order of stations reached by the trains.

the station vulnerabilities. However, also for this group, some instances show an unexpected trend in station vulnerabilities. For example, instances 11, 13, 14, and 17 show an upward trend in vulnerability as the distance from the primary delay station increases. Two instances for group 1 showed different behavior while being very similar. Here four instances show similar behavior, but the primary delay does not start on the same line, around the same time, or pertain to trains with similar headways. Also, as the traffic density is higher in this part of the network, the chance of randomly or just delayed trains is higher than for group 1. Therefore, the group 2 results reiterate that more factors must be at play to explain the vulnerability trends.

5.3.2. Model training and testing results group 2

For group 2 no results have been presented yet as group 2 was not used for the experiments presented in section 4.3. Therefore, the training results are presented first, followed by the testing results.

The MSE value of the trained model is 0.043, which was calculated using Equation 3.11. Thus, 0.043 is the summed MSE for all 8 training instances. On average, the MSE per training instance is 0.005. Table 5.5 shows the trained parameter values for group 2.

Table 5.5: Trained parameters for group 2: Stadium-Armory in the west direction till Rosslyn.

γ	0.342	
θ	0.500	
Stations	z_{ik}	c_{ik}
Stadium-Armory	0.219	0.453
Potomac Ave	0.151	0.305
Eastern Market	0.118	0.220
Capitol South	0.151	0.162
Federal Center SW	0.320	0.305
L'Enfant Plaza	0.258	0.212
Smithsonian	0.306	0.195
Federal Triangle	0.348	0.256
Metro Center	0.436	0.242
Mcpherson Square	0.302	0.128
Farragut West	0.320	0.179
Foggy Bottom - GWU	0.440	0.247
Rosslyn	0.379	0.100

The trained value for γ is low, which means that the edge weights did not change much. The model again uses the lower bound for the θ parameter. Interestingly, the stations in group 2 have the highest baseline infection rates due to their central network location. Even so, the z_{ik} values for each station are quite high, which means that the algorithm still compensates quite a bit for the infection rate values. The baseline recovery rates of these stations are low due to the high traffic density. Still, the model does not need the upper bound set on c_{ik} to get closer to the vulnerabilities as calculated from the data. Therefore, the training algorithm chose to compensate for the infection rate more than the recovery rate values.

The trained parameters were tested on instances 15 and 19 from group 2. The performance of the model on these unseen instances is displayed in Table 5.6. The MSE values presented in Table 5.6 are based on Equation 3.10 only.

Table 5.6: Testing metrics for group 2.

Metric	*Testing instance 1*	*Testing instance 2*
MSE	0.018	0.027
Average vulnerability data	0.308	0.279
Average vulnerability model	0.424	0.414

The MSE values 0.018 for testing instance 1 and 0.027 for testing instance 2 show that the model performed worse for group 2 than group 1. Also, the MSE values indicate the predictions of the model are off on average by approximately two (out of 14) trains. Furthermore, while for group 1 the average vulnerability based on the model and data were close to each other, for group 2 there is more distance between the values for both instances. The worse performance of the model for group 2 could be explained by the larger range of vulnerability values for group 2. For group 1 most vulnerability values are in the range [0.30, 0.50]. For group 2, on the other hand, the range is [0.25, 0.55]. This larger range of vulnerability values makes it harder for the model to find a parameter that fits all those values. Consequently, the model will also perform worse than for a group of instances where the vulnerability values are more similar. Table 5.7 gives more insights into how the model performed for specific stations. These differences were calculated using Equation 5.1.

Table 5.7: Comparison of vulnerabilities as determined from the data and by the model for group 2.

	Testing instance 1			Testing instance 2		
Stations	*Data*	*Model*	*Difference*	*Data*	*Model*	*Difference*
Stadium-Armory	0.289	0.319	-0.030	0.389	0.413	-0.024
Potomac Ave	0.283	0.321	-0.038	0.389	0.457	-0.068
Eastern Market	0.328	0.383	-0.055	0.417	0.456	-0.039
Capitol South	0.278	0.279	-0.001	0.361	0.424	-0.063
Federal Center SW	0.283	0.293	-0.010	0.361	0.435	-0.074
L'Enfant Plaza	0.311	0.450	-0.139	0.333	0.540	-0.207
Smithsonian	0.283	0.428	-0.145	0.361	0.505	-0.144
Federal Triangle	0.311	0.388	-0.077	0.417	0.482	-0.065
Metro Center	0.283	0.502	-0.219	0.417	0.649	-0.232
Mcpherson Square	0.311	0.441	-0.130	0.306	0.413	-0.107
Farragut West	0.317	0.464	-0.147	0.306	0.453	-0.147
Foggy Bottom - GWU	0.378	0.564	-0.186	0.306	0.548	-0.242
Rosslyn	0.350	0.587	-0.237	0.250	0.592	-0.342

Where the model tends to underestimate the vulnerability for the instances in group 1, the model overestimates the vulnerability values for the testing instances of group 2. This overestimation makes sense because the vulnerabilities based on the data for all stations for these testing instances are low compared to the other instances. While the model does not work towards average vulnerabilities, it does try to find one parameter value that fits all. If the data vulnerabilities of the other training instances at a specific station are higher than those in testing instances 1 and 2, the model will favor fitting those stations. As a result, the trained parameter values fit instances with higher vulnerabilities better. The model will then overestimate the vulnerabilities of instances with lower-than-average vulnerabilities. It must be explored how sensitive the model is to which instances are used for training and which for testing.

5.4. Comparison group 1 and group 2

This section compares the training and testing results of groups 1 and 2. The trained parameters for both groups show similar trends. Both groups have a low γ value and $\theta = 0.500$. As a result, the edge weights for both groups were minimally adjusted, and the recovery rates across all stations were made more uniform. Group 1 exhibited a γ value that was 0.017 higher, leading to a larger adjustment in edge weights for group 1 compared to group 2. The stations in group 1 require more significant adjustments to become more vulnerable because group 1 stations are located farther from the network's center and have a lower baseline infection rate than those in group 2. Consequently, the model needs greater adjustments to achieve high vulnerability values as observed from the data for group 1 stations.

Even though the groups are about different parts of the network, the model shows similar performance. The largest difference between the two groups of instances is that where the vulnerabilities of the testing instances of group 1 were underestimated, in group 2 the model overestimates the vulnerabilities. The similar model training on both groups, even though they are different in many ways, begs the question of how this could be. One of the reasons for the similar behavior is the order of magnitude of the infection and recovery rates. The training algorithm used high values for the z_{ik} parameter to compensate because the values for the infection rate are so much smaller than for the recovery rate due to the normalization of the edge weights. Using the z_{ik} parameter, the infection rate values could come close to the recovery rate values. Even so, why the model performed the way it did has to be further investigated. Therefore, a sensitivity analysis was performed on different parts of the model, which is discussed in section 5.5.

5.5. Sensitivity analysis

While the model shows promising results, it is important to be critical of the conditions under which the model has this performance. The model sensitivity was tested in two ways: 1. its sensitivity to the parameter values and 2. its sensitivity to the instances used for training and testing. The first sensitivity analysis is discussed in subsection 5.5.1 and the second in subsection 5.5.2.

5.5.1. Sensitivity parameters

The model sensitivity was tested for the parameters γ and θ. First, the sensitivity to γ is presented, followed by θ. The sensitivity of both parameters was tested using the training results of group 1.

Sensitivity to γ The γ parameter is part of the equation used to adjust the weights of some of the edges in the P-space graph. A high sensitivity means the model could increase the infection rates too much and then overestimate the vulnerabilities, while a low sensitivity would mean that the z_{ik} parameter would have to make all the difference in the infection rate equation. The sensitivity analysis results are presented in Table 5.8 and Table 5.9.

Table 5.8: Testing metrics for sensitivity analysis of the γ parameter.

	γ = 0		γ = 0.45		γ = 1	
Metric	*Testing instance 1*	*Testing instance 2*	*Testing instance 1*	*Testing instance 2*	*Testing instance 1*	*Testing instance 2*
MSE	0.005	0.004	0.004	0.004	0.005	0.004
Average vulnerability data	0.319	0.336	0.319	0.336	0.319	0.336
Average vulnerability model	0.292	0.299	0.295	0.299	0.290	0.294

Table 5.9: Differences in vulnerability for the testing instances of group 1 for different γ values.

	γ = 0		γ = 0.45		γ = 1	
	Testing instance 1	**Testing instance 2**	**Testing instance 1**	**Testing instance 2**	**Testing instance 1**	**Testing instance 2**
Stations	*Difference*	*Difference*	*Difference*	*Difference*	*Difference*	*Difference*
Arlington Cemetery	-0.067	-0.058	-0.066	-0.064	-0.065	-0.065
Braddock Road	0.101	0.124	0.068	0.098	0.065	0.097
Crystal City	0.051	0.034	0.045	0.039	0.056	0.052
King St-Old Town	0.105	0.054	0.098	0.052	0.100	0.055
L'Enfant Plaza	0.030	0.082	0.045	0.089	0.050	0.090
Pentagon	0.011	-0.010	0.016	0.009	0.024	0.018
Pentagon City	0.052	0.030	0.058	0.052	0.071	0.066
Reagan National Airport	0.071	0.050	0.054	0.043	0.066	0.058
Rosslyn	-0.104	0.025	-0.099	0.014	-0.099	0.012

Based on Table 5.8, it can be concluded that the model's performance is largely insensitive to the γ parameter, with only slight decreases in performance observed when $\gamma = 0$ or $\gamma = 1$. These differences are small and primarily visible in the average vulnerability differences between the data and the model, which are higher for $\gamma = 0$ and $\gamma = 1$. A similar conclusion can be drawn from Table 5.9, where the vulnerability differences across γ values remain comparable. However, as shown in the third column of Table 5.15 and Table 5.5, the z_{ik} values are consistently high. These high z_{ik} values allow the model to compensate for the low infection rates resulting from the normalization of edge weights. This compensation mechanism makes z_{ik} the dominant factor in increasing infection rates, reducing the role of γ. Consequently, the model may overfit using z_{ik} rather than capturing unobserved factors. This overfitting is not very significant as evidenced by the low MSE values.

Sensitivity to θ Like γ, the θ parameter influences model performance. A high θ value means the stations' recovery rates are very heterogeneous, while a low value means stations have a similar recovery rate. The sensitivity of θ was tested by setting it to 0 and 1 and comparing those results with each other and the value from the training. The sensitivity analysis results are displayed in Table 5.10 and Table 5.11.

Table 5.10 shows how a low value for θ leads to underestimation of the vulnerabilities, while a high value leads to overestimation as indicated by the difference between the average vulnerability from the data and the model. When θ approaches 0, the second part of the recovery rate equation (Equation 3.7) becomes 1. Hence, the recovery rates of the stations will be $c_{ik} + 1$. The model will then underestimate the vulnerabilities because the infection rates can never be that high. Similarly, when θ

Table 5.10: Testing metrics for sensitivity analysis of the θ parameter.

	θ = 0		θ = 0.50		θ = 1	
Metric	*Testing instance 1*	*Testing instance 2*	*Testing instance 1*	*Testing instance 2*	*Testing instance 1*	*Testing instance 2*
MSE	0.019	0.022	0.004	0.004	0.008	0.005
Average vulnerability data	0.319	0.336	0.319	0.336	0.319	0.336
Average vulnerability model	0.197	0.197	0.295	0.299	0.378	0.384

Table 5.11: Differences in vulnerability for the testing instances of group 1 for different θ values.

	θ = 0		θ = 0.50		θ = 1	
	Testing instance 1	**Testing instance 2**	**Testing instance 1**	**Testing instance 2**	**Testing instance 1**	**Testing instance 2**
Stations	*Difference*	*Difference*	*Difference*	*Difference*	*Difference*	*Difference*
Arlington Cemetery	0.030	0.030	-0.066	-0.064	-0.166	-0.166
Braddock Road	0.157	0.198	0.068	0.098	-0.009	0.026
Crystal City	0.131	0.131	0.045	0.039	-0.023	-0.029
King St-Old Town	0.177	0.135	0.098	0.052	0.029	-0.019
L'Enfant Plaza	0.188	0.229	0.045	0.089	-0.045	-0.005
Pentagon	0.129	0.129	0.016	0.009	-0.088	-0.098
Pentagon City	0.146	0.146	0.058	0.052	-0.015	-0.022
Reagan National Airport	0.143	0.143	0.054	0.043	-0.043	-0.058
Rosslyn	-0.003	0.109	-0.099	0.014	-0.169	-0.058

approaches 1, the differences in recovery rates are enlarged. Most of the stations in group 1 have a fairly good recovery rate as they are located near the end of the Yellow and Blue lines. They are also all very similar and, consequently, enlarging these differences only leads to a similar decrease in recovery rate values with an overestimation of the vulnerability values as a consequence. This analysis is also reflected by Table 5.11. When $\theta = 0$ the model underestimates the vulnerabilities and when $\theta = 1$ the model overestimates the vulnerabilities. Therefore, the model is sensitive to the θ parameter.

5.5.2. Sensitivity training and testing instances

To test how sensitive the model performance is to which instances are used in the model training and testing, the instances from group 1 were used again. This time instead of having instances 3 and 6 used for testing, instances 1 and 9 were randomly chosen. Tabels 5.12 and 5.13 show the results of this sensitivity analysis.

Table 5.12: Testing metrics for sensitivity analysis of instance division for training and testing.

Metric	*Instance 2*	*Instance 9*
MSE	0.010	0.028
Average vulnerability data	0.350	0.417
Average vulnerability model	0.304	0.280

Table 5.13: Comparison of vulnerabilities as determined from the data and by the model for the testing instances used in the experiment.

	Instance 2			Instance 9		
Stations	*Data*	*Model*	*Difference*	*Data*	*Model*	*Difference*
Arlington Cemetery	0.500	0.403	0.097	0.500	0.362	0.138
Braddock Road	0.292	0.286	0.006	0.417	0.265	0.152
Crystal City	0.292	0.286	0.006	0.417	0.263	0.154
King St-Old Town	0.292	0.232	0.060	0.333	0.211	0.122
L'Enfant Plaza	0.583	0.310	0.273	0.500	0.293	0.207
Pentagon	0.292	0.350	-0.058	0.417	0.327	0.090
Pentagon City	0.292	0.284	0.008	0.500	0.265	0.235
Reagan National Airport	0.333	0.297	0.036	0.500	0.269	0.231
Rosslyn	0.278	0.287	-0.009	0.167	0.267	-0.100

The model performance is sensitive to the instances used for training and testing. Comparing Table 5.12 and Table 5.2 shows that the model performed better when instances 3 and 6 were used for testing and the rest for training. The MSE values and the distance between the average vulnerabilities of the data and model, when instances 3 and 6 are used for testing, are smaller. When considering the metrics, the model is able to predict on average the vulnerability values of instances 2 and 9 with an error of about one train. Even so, for instance 2 the model performs much worse for L'Enfant Plaza station because its vulnerability based on the data is much higher than the other instances. Similarly, the station vulnerabilities for instance 9 are also higher than average, leading to underestimation by the model. Therefore, the model is sensitive to which instances are used for training and testing, which mainly shows itself in the results of individual station vulnerabilities.

The model has a few key limitations. Firstly, the model sensitivity means that the model reacts badly to instances that are different from the other instances. Case in point are L'Enfant plaza for instance 2 and all stations of instance 9. Therefore, the model lacks robustness. Secondly, there is a dependence on the input data, which means that more factors need to be considered to decrease this dependence and improve how the model reacts to 'odd' instances. All the instances have a very similar primary delay, but the vulnerabilities of stations differ greatly. While some variations can be attributed to randomly delayed trains or trains barely being delayed, the rest is not modeled yet. Lastly, the model is currently overfitting, which is the result of its sensitivity and the equations. Solving the other limitations will help reduce the overfitting of the model.

5.6. Benchmarking

It is also important to benchmark the created model to see how well it performs compared to other similar models. As this model is the first of its kind, these other models do not exist. However, the model can be compared to another version of itself. Hence, the model presented in this study is compared to a model version where the constants z_{ik} and c_{ik} in the infection rate and recovery rate equations respectively are changed, which is explained in more detail below.

The infection rate and recovery rate equations currently have a station-specific constant to capture any factors currently not considered in the model. The model is benchmarked by creating two additional models:

- a model where the constants are not included;
- a model where the constants are made non-station-specific.

The advantage of (station-specific) constants is that any factors currently not included can be captured more accurately, especially as the effects of these factors might differ greatly across stations. The disadvantage, however, is that introducing constants means additional parameters have to be trained. Having to train all these additional parameters increases the running time of the model training.

To see how accurate the model would be with no constants and non-station-specific constants, these two model versions were trained using the instances of group 1. For all models, the same training settings were used. The only difference is if and how the constants were defined. Table 5.14 shows the model performance for all three model configurations, so including the model version used in this study for easier comparison. The MSE values in Table 5.14 are based on Equation 3.11.

Table 5.14: Training results for benchmarking.

	No z_{ik} and c_{ik}	*Non station-specific z_{ik} and c_{ik}*	*Station-specific z_{ik} and c_{ik}*
MSE	0.074	0.072	0.022

The model performs very similarly for the model configuration with no constants at all and non-station-specific constants. While the MSE value does go down, the difference is negligible. The model configuration where the parameters z_{ik} and c_{ik} are station-specific performs better with a MSE decrease of 0.05. To better understand how the model configurations performed, one also has to look at Table 5.15.

Table 5.15: Trained parameters for all three model configurations. For the constants c and z, a range of values obtained for the station-specific constants is presented.

	No constants	**Non-station specific**	**Station specific**
γ	0.358	0.355	0.449
θ	0.500	0.674	0.500
c	-	0.201	0.1 - 0.5
z	-	0.001	0.1 - 0.5

The γ parameter values shown in Table 5.15 indicate that γ was low for all three model configurations, which means that the edge weights were barely adjusted. In the case of the non-station-specific constants, the infection rate constant z was also low. Hence, during training with these model configurations and parameter bounds, higher values for γ and z, which would lead to higher infection rates, resulted in worse results. Instead, the compensation of stations with a too low recovery rate leads to the best training result.

The model configuration where the constants are station-specific outperforms the other configurations because the compensation happens on a station level instead of one value for all the stations. Therefore, there must be a station-specific factor that the model can only capture by making the constants station-specific. How much needs to be corrected to get to the right vulnerability value differs per station. Having the station-specific parameters means more tailored values can be used, as shown by the fact that almost the full range of the boundaries is used in the station-specific parameter case.

5.7. Policy implications

The results and findings of this study have several implications for WMATA and other metro network operators. While the model requires further refinement, it offers actionable insights into delay propagation and response strategies. Insights were obtained from the results about the Washington DC metro network and the model. These insights are listed below and briefly explained:

- **Delay propagations at transfer stations:** Most stations in the Washington network have only a few delay propagations in the studied time period. Only four stations have ≥ 10, all transfer stations. Overall it seems that WMATA needs to focus on the transfer stations, especially those where lines merge and split;
- **Shared infrastructure increases delay propagations:** Lines that share most of their infrastructure tend to have more delay propagations than lines that do not share any of their infrastructure. This insight also helps WMATA target specific lines and stations;
- **Fluctuating vulnerability:** With the traffic density fluctuating throughout the day, the vulnerability of the station will also fluctuate. WMATA, and other network operators, have to take these fluctuations into account when designing prevention and mitigation strategies;
- **Variation in delay propagation effects:** Although some of the primary delays across instances of group 1 are similar, station vulnerabilities differ significantly, and the model does not yet account for these differences adequately. Similarly, group 2 showed different instances with the same effects resulting in comparable vulnerabilities. More research is needed to uncover which additional factors need to be included to explain these differences;

- **Information from the baseline infection and recovery rates:** Even if little data is available and the model cannot be trained, the baseline infection and recovery rates still give information about the stations and how they relate to each other based on the factors that are included.
- **Model performance:** The MSE values for both training and testing for both groups of instances show promising results. While the trained values of the parameters indicate some problems that need to be resolved, the testing MSE values indicate that the model can predict the vulnerability values with an error of ≤1 train out of 10 for group 1 and approximately 2 out of 14 for group 2. These predictions showcase a promising future of using epidemic models in transportation research.
- **Model sensitivity:** The model was found to be sensitive to γ and which instances are used for training and testing. γ influences the heterogeneity of the recovery rates; hence, its value and sensitivity give information about the recovery rate of the stations and their relationship to each other. With regards to the instance sensitivity, mostly station vulnerabilities that could be considered outliers within their instance itself and/or across all instances, are not captured well by the model. Therefore, the model lacks robustness.
- **Model overfitting:** The model sensitivity, the order of magnitude of the infection and recovery rate, and the resulting dynamics between parameters γ and z_{ik}, causes the model to overfit slightly. Fixing the mentioned issues in the future will greatly help improve the model.

6

Conclusion and Discussion

This chapter consists of two parts: 1. the conclusion of the research where the main research question is answered and 2. a discussion of the results through the identification of implications, limitations, and future research directions.

6.1. Conclusion

This study aimed to fill the research gap of how delay propagation in a metro network could be modeled using the SIS model to reproduce the vulnerability of a metro station for specific instances. A model based on the SIS model was constructed and trained for several parameters using data about the Washington Metro network. First experiments were done to decide on the best model configuration. Then, the model was trained and tested on two parts of the Washington metro network. The first part is the stations between King St - Old Town, Rosslyn, and L'Enfant Plaza station in the direction of Rosslyn and L'Enfant Plaza station. The second part of the network studied is the stations between Stadium-Armory and Rosslyn station in the west direction. The model training and testing results for both groups were analyzed and the conditions under which the model produced the results were reflected upon through a sensitivity analysis. This was all done to answer the following main research question:

How can the SIS model effectively be utilized to produce the effects of delay propagation in metro networks, particularly through the models' ability to capture the vulnerabilities of metro stations for specific instances?

This research question is supported by several subquestions. First, the answers to these subquestions are given. Then, using those answers, the main research question is answered.

How can the traditional SIS model be adapted to accurately represent the characteristics and dynamics of delay propagations within a metro network?
The traditional SIS model was adapted through a literature study of factors contributing to station vulnerability and into the SIS model itself. A definition of vulnerability was followed that highlights the exposure to and how well stations cope with disruptions. The main elements of the SIS model were kept the same as they can be attributed to the two components of the vulnerability definition but transformed to a metro context. Therefore, the model used in this research consisted of stations, which have a chance of being infected and recovering again at a certain rate. Multiple factors were included to resemble the characteristics and dynamics of delay propagation in a metro network. The infection rate of metro stations is dependent on their network connectivity and the severity of the delay propagation and its effects. Moreover, the further the delayed train travels the less chance of infection it should have. The recovery rate is dependent on station and line characteristics. Stations with a high traffic density have a harder time recovering compared to stations with low traffic density. Also, stations with more tracks are more flexible and, hence, increase the recovery rate. The traditional SIS model can be adapted through the modeling of station and line characteristics in the infection rate and recovery

rate of stations.

Which model configuration produces the best results given performance metrics and keeping computational efficiency in mind?
Experiments were done to see which model configuration produced the best training results. These experiments were about the model itself and the context in which the model is used. One of these experiments is about whether constants z_{ik} and c_{ik} should be included in the infection and recovery rate definition respectively to capture any factors currently not modeled. If the constants are included, whether they would be station-specific or not also would have to be decided. Ultimately, it was decided to use the model configuration with the station-specific constants. Especially with how the delay propagation effects across different instances differ widely, it is too difficult for the model to capture these varying situations well without station-specific constants. The second experiment was about which stations to consider in the training and testing. There were two options: 1. the stations reached by all affected trains or 2. all the stations of the affected lines. The first option proved to be a better model configuration though it has its limitations. The last model configuration experimented with was whether the P-space graph should be directed or undirected. The two versions performed similarly, but there are problems with both. The directed graph has a structural issue due to the infection rate formulation, but the undirected graph is less realistic. Ultimately, the model configuration with station-specific constants, stations reached by all affected trains, and an undirected graph was chosen. However, the experiments revealed model problems that should be solved in the future.

Which combination of parameter values results in the model reproducing the vulnerabilities the best given a specific group of instances?
Two groups of instances were used to train and test the model. For both groups, the θ parameter got a value of 0.500, with the γ parameter having a value around 0.45. For z_{ik} and c_{ik} the training algorithm used almost the full range of the bounds. These values led for both groups to a MSE < 0.030. The model does perform better for group 1 than group 2. One of the reasons for this difference in performance is that the range of station vulnerabilities is larger for group 2 than for group 1. When the algorithm is training the parameter values of group 2, it needs to find values that fit a larger range, which is harder to do. The relatively low values for θ and γ in combination with the training algorithm using the full range of the boundaries for z_{ik} and c_{ik} suggest overfitting.

How sensitive is the model to changes in parameter values and input?
The sensitivity of the model was tested in three ways. The values of the parameters γ and θ were varied and which instances were used for training and testing of group 1 was changed. For $\gamma = 0$ and $\gamma = 1$, the performance barely changes from the trained value. Therefore, the model seems insensitive to γ. The model underestimates the vulnerabilities for $\theta = 0$ and overestimates the vulnerabilities for $\theta = 1$. The model is, therefore, sensitive to γ. Lastly, instead of instances 3 and 6, 2 and 9 were used to test the model. The model performs worse when instances 2 and 9 are used for testing. Thus, the model is sensitive to the input. This means that the model lacks robustness.

Now the main research question can be answered. The SIS model has proven that it can be utilized to produce the effect of delay propagation in metro networks through its ability to capture the vulnerabilities of metro stations for specific instances. Vulnerability definitions from the literature have shown that vulnerability contains two components: exposure to and the ability to cope with disturbances, which are highlighted through the infection and recovery rate respectively. The flexibility of the model allows the introduction of new factors easily. The balance between the infection rate and recovery rate gave way to a vulnerability equation that managed to come close to the values from the data. The SIS model adaptation and its utilization led to some promising results, also compared to other model configurations. However, the conditions under which these results were obtained leaves room for discussion.

6.2. Discussion

This discussion section starts with describing the limitations of this study in subsection 6.2.1. The future research directions are proposed in subsection 6.2.2, followed by possible applications in subsection 6.2.3.

6.2.1. Limitations

In this subsection, the limitations of this research are described. First, the limitations of the data and the calculations based on it are analyzed. The limitations of the model are discussed afterwards.

As discussed in subsection 5.1.1 the data caused some restrictions on this research. While the available data seemed large enough at first (four months), the actual number of delay propagations in this period was small. Enough delay propagations of a station in one direction especially to form a group of instances was hard. The small set of found delay propagations meant the level of diversity for the groups desired at the beginning of the research was not possible. For example, in the literature study, several factors were described that influence vulnerability, delay propagation, or both. An example is leisure versus work trips and the time of day (Eltved et al., 2021; Yap & Cats, 2022). Time was not included in this study as a factor, because the limited number of instances did not allow for the creation of groups covering different times of day while also having enough instances in a group. To find more delay propagations from the data more data is needed. Then, with enough instances, more factors can be included and their influence measured.

Furthermore, the current vulnerability calculation based on the data does not exclude delayed trains that were not affected by the primary delay. As a consequence, the vulnerabilities of stations might be higher than they would be if only affected trains were considered. If those randomly delayed trains are excluded somehow, the vulnerabilities of stations might show the expected trend. Then the question becomes how to exclude those trains, completely from the train movements at all stations or just the stations where the train is randomly delayed. With the first option, there is a problem. This randomly delayed train could also be the train in front of the primary delayed train. If that train is deleted, the primary delay train will not be seen as delayed anymore at every station. However, the second option also causes problems. Excluding trains at a few stations would change the number of trains considered per station. The number of trains could already be uneven, but excluding trains at some stations might only worsen this problem. The current calculation method is not great, but the other two options also have issues. More thought into this calculation is needed to tackle these problems.

Also, the amount of data needed to train the model is not necessarily large, but finding instances of delay propagation has proven to be difficult. Moreover, whenever the infrastructure of the network changes, new data has to be acquired, because the stations might react differently. The model is, therefore, very dependent on the quality and quantity of the available data. Obtaining the data to test the model on the changed/new parts of the network might take months or even years.

A limitation of the model is that the graph of the Washington Metro network is undirected. As the experiment described in subsection 4.3.2 showed, using a directed graph could improve the model. Nevertheless, with the current definition of the infection rate the model would structurally underestimate the vulnerability of the primary delay station. A directed graph, however, might improve the model as it represents the dynamics and direction of delay propagations better. Therefore, a limitation is the undirected graph, but to use a directed graph an improved definition of the infection rate would be needed.

The next limitation is the stations considered in the current model. subsection 4.3.1 outlines how considering all the stations of the line would give a more complete picture of the delay propagation, but as not every primary delay influences the whole line the vulnerability values range would be too large. Consequently, the current model only shows a partial picture of the influence of delay propagations on network operations. This limitation should be solvable, at least partially, by including more factors in the model, such that the model can overcome the differences between instances better.

Additionally, the order of magnitude of the recovery and infection rate values differs. Due to the normalization of the edge weights, the order of magnitude for the infection rate is smaller than the recovery rate. Without the normalization, the infection rate values would be disproportionally large after the edge weight adjustments. How the variables relate to each other and are quantified should be rethought to get the infection and recovery rate values in the same order of magnitude.

Lastly, because the instances in the groups did not necessarily behave similarly, the z_{ik} and c_{ik} parameters failed to capture any factors currently not part of the model. Instead, these parameters had to overcome the differences in vulnerability values of the instances.

6.2.2. Future research

Based on the presented results and limitations some directions for further research are proposed. Improvements can be made for both the approach to the data and the model.

The model needs improvement and several ways can be explored how to do this improvement. Given that more data is available in the future and more groups of instances can be created, other factors should be explored to include in the model. An example is the time of the day, found in other studies as an influential factor (Eltved et al., 2021; Xiao et al., 2018). Time is a factor that could influence vulnerability as headways and, thus, traffic density change throughout the day in reaction to fluctuating passenger flows. An experiment comparing two groups of instances where the only difference is peak and off-peak or weekday versus weekend days would be interesting to get a first idea of how it influences the current model. Then, ways of including time as a factor should be investigated. Including more factors should also help explain the different trends in propagation effects in the data.

Another reason to look for data spanning a longer time is that the model can also be tested on those parts of the network that only concern one line. In the Washington Metro network, this is the Red line. Due to a low number of delay propagations the Red line could not be studied at all. While it is a less complicated case compared to the studied parts, to get a complete picture of the model performance those parts should also be investigated.

Furthermore, with the availability of additional data, more groups of instances can be identified. This will allow deeper insights into how different parts of the network respond to delay propagation and how these groups differ in their behavior. For example, two groups with a primary delay station close to the other can be compared. This will give insights into if any generalizations are possible.

All the proposed future research directions will contribute to a deeper understanding of the sensitivity of the θ parameter. As the model is studied more extensively, with a focus on different parts of the network, the inclusion of additional factors, and the comparison of groups of instances, the true sensitivity of θ will become clearer. With an enhanced model, a broader range of θ values can be analyzed, revealing that the parameter may not be as sensitive as initially believed. Additionally, it may show that θ tends to assume similar values across various groups, suggesting that its sensitivity does not have much impact.

Another future research direction is applying the model to a different network. The station characteristics in the Washington Metro network are very similar. All stations have two tracks per infrastructure section except for the Reagan National Airport. Therefore, the influence of the number of tracks could barely be explored in this study. Using another network as a case study, which has more diversity in station characteristics, could help improve the model by uncovering which factors do make a difference and which do not. More diverse stations will also give more insights into the sensitivity of the θ parameter. As the stations used in the two groups are very similar, the value of θ now only homogeneously increases or decreases the recovery rate values of all stations. There is currently no conclusion on whether the differences in recovery rates between the stations should indeed be heterogeneous.

6.2.3. Applications

There are several recommendations for WMATA and other network operators on how to apply this study in the future. First, stations that are prone to significant delay propagations for a variety of disturbances can be identified and prioritized with the needed mitigation measures. It does not make sense for WMATA to have a proactive and reactive response for each station because that is expensive and probably unnecessary. Additionally, if factors such as peak and off-peak influence how the delays propagate and, thus, the vulnerability of stations, WMATA could incorporate these variations in their response. For instance, they might allocate more staff and resources to specific stations during peak hours. Also, since the model accounts for the station characteristics, it is easier for WMATA to see which characteristics are particularly sensitive and target those accordingly. This could include adding an extra track or reconsideration of which stations are serviced by which lines to reduce their susceptibility to delays.

With the insights provided by the model, WMATA can create adaptive response plans that vary based on the delay propagation instance. For example, instead of applying uniform strategies across all stations, operators can implement differentiated approaches tailored to specific stations' roles in delay propagation. Also, the WMATA staff can be trained using the model ensuring they are prepared and have the appropriate response when delay propagations occur in the network.

Besides the response of WMATA on an operational level, the findings can be integrated into the passenger information systems. As it is predicted how delays will propagate, passengers can be informed proactively which will help them to make more informed travel decisions and improve overall satisfaction.

The model applications are not limited to the Washington Metro network, let alone metro networks. In section 2.2 the use of epidemic models for transport modes such as the train was already described. Those studies used models that do not capture the heterogeneity of the stations or only considered a few trains. This model, which does consider heterogeneity and a time window-dependent number of trains, could be transferable to the train. A condition would be that the translation due to a different network dynamic is done well.

Bibliography

(2021). Population Dynamics Research Centers. https://popresearchcenters.org/research-highlights/page/4/

Baspinar, B., & Koyuncu, E. (2016). A data-driven air transportation delay propagation model using epidemic process models. *International Journal of Aerospace Engineering*, *2016*.

Berdica, K. (2002). An introduction to road vulnerability: What has been done, is done and should be done. *Transport Policy*, *9*, 117–127.

Cadarso, L., Marín, Á., & Maróti, G. (2013). Recovery of disruptions in rapid transit networks. *Transportation Research Part E: Logistics and Transportation Review*, *53*, 15–33.

Cats, O., & Hijner, A. (2021). Quantifying the cascading effects of passenger delays. *Reliability Engineering and System Safety*, *212*. https://doi.org/https://doi.org/10.1016/j.ress.2021.107629.

Cats, O., & Jenelius, E. (2018). Beyond a complete failure: The impact of partial capacity degradation on public transport network vulnerability. *Transportmetrica B: Transport Dynamics*, *6*, 77–96.

Cats, O., van Cranenburgh, S., Vijlbrief, S., Krishnakumari, P., & Massobrio, R. (2019, October). A curated data set of p-space representations for 51 metro networks worldwide. https://doi.org/10.4121/21316824

Cats, O., Yap, M., & van Oort, N. (2016). Exposing the role of exposure: Public transport network risk analysis. *Transportation Research Part A: Policy and Practice*, *88*, 1–14.

Ceria, A., Köstler, K., Gobardhan, R., & Wang, H. (2021). Modeling airport congestion contagion by heterogeneous sis epidemic spreading on airline networks. *Plos one*, *16*.

Chen, C., Wang, S., Zhang, J., & Gu, X. (2023). Modeling the vulnerability and resilience of interdependent transportation networks under multiple disruptions. *Journal of Infrastructure Systems*, *29*.

Chopra, S., Dillon, T., Bilec, M., & Khanna, V. (2016). A network-based framework for assessing infrastructure resilience: A case study of the london metro system. *Journal of The Royal Society Interface*, *13*.

Dekker, M. M., Medvedev, A. N., Rombouts, J., Siudem, G., & Tupikina, L. (2022). Modelling railway delay propagation as diffusion-like spreading. *EPJ Data Science*, *11*.

de Oliveira, E. L., da Silva Portugal, L., & Junior, W. P. (2016). Indicators of reliability and vulnerability: Similarities and differences in ranking links of a complex road system. *Transportation Research Part A: Policy and Practice*, *88*, 195–208.

Derrible, S., & Kennedy, C. (2011). Applications of graph theory and network science to transit network design. *Transport reviews*, *31*, 495–519.

Eltved, M., Breyer, N., Ingvardson, J. B., & Nielsen, O. A. (2021). Impacts of long-term service disruptions on passenger travel behaviour: A smart card analysis from the greater copenhagen area. *Transportation Research Part C: Emerging Technologies*, *131*.

Ermagun, A., Tajik, N., Janatabadi, F., & Mahmassani, H. (2023). Uncertainty in vulnerability of metro transit networks: A global perspective. *Journal of Transport Geography*, *113*.

Gurin, D., Prokhorchenko, A., Kravchenko, M., & Shapoval, G. (2020). Development of a method for modelling delay propagation in railway networks using epidemiological sir models. *Eastern-European Journal of Enterprise Technologies*, *6*.

Hong, W. T., Clifton, G., & Nelson, J. D. (2022). Rail transport system vulnerability analysis and policy implementation: Past progress and future directions. *Transport Policy*, *128*, 299–308.

Jia, C., Zheng, S., Qian, H., Cao, B., & Zhang, K. (2022). Analysis of crowded propagation on the metro network. *Sustainability*, *14*.

Kermack, M., & Mckendrick, A. (1927). Contributions to the mathematical theory of epidemics: Part ii. *Proceedings of the Royal Society A*, *138*, 55–83.

Lu, J., Ma, X., & Xing, Y. (2021). Risk factors affecting the severity of disruptions in metro operation in shanghai, 2013-2016. *Journal of Transportation Safety & Security*, *13*, 69–92.

Lu, Q., & Lin, S. (2019). Vulnerability analysis of urban rail transit network within multi-modal public transport networks. *Sustainability*, *11*.

Luo, D., Cats, O., & van Lint, H. (2020). Can passenger flow distribution be estimated solely based on network properties in public transport systems? *Transportation*, *47*, 2757–2776.

Malandri, C., Fonzone, A., & Cats, O. (2018). Recovery time and propagation effects of passenger transport disruptions. *Physica A: Statistical Mechanics and its Applications*, *505*, 7–17. https://doi.org/https://doi.org/10.1016/j.physa.2018.03.028

Marra, A. D., & Corman, F. (2020). From delay to disruption: Impact of service degradation on public transport networks. *Transportation Research Record*, *2674*, 886–897.

Monechi, B., Gravino, P., Di Clemente, R., & Servedio, V. D. (2018). Complex delay dynamics on railway networks from universal laws to realistic modelling. *EPJ Data Science*, *7*.

Pan, S., Yan, H., He, J., & He, Z. (2021). Vulnerability and resilience of transportation systems: A recent literature review. *Physica A: Statistical Mechanics and its Applications*, *581*.

Redman, L., Friman, M., Gärling, T., & Hartig, T. (2013). Quality attributes of public transport that attract car users: A research review. *Transport Policy*, *25*, 119–127.

Rodríguez-Núñez, E., & García-Palomares, J. C. (2014). Measuring the vulnerability of public transport networks. *Journal of transport geography*, *35*, 50–63.

SciPy. (n.d.). Differential_evolution. https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.differential_evolution.html

Shi, J., Wen, S., Zhao, X., & Wu, G. (2019a). Sustainable development of urban rail transit networks: A vulnerability perspective. *Sustainability*, *11*.

Shi, Z., Zhang, N., & Zhu, L. (2019b). Understanding the propagation and control strategies of congestion in urban rail transit based on epidemiological dynamics model. *Information*, *10*.

Sun, D., & Guan, S. (2016). Measuring vulnerability of urban metro network from line operation perspective. *Transportation Research Part A: Policy and Practice*, *94*, 348–359.

Szymula, C., & Bešinović, N. (2020). Passenger-centered vulnerability assessment of railway networks. *Transportation Research Part B: Methodological*, *136*, 30–61.

UITP. (2022, May). World metro figures 2021.

Von Ferber, C., Holovatch, T., Holovatch, Y., & Palchykov, V. (2009). Public transport networks: Empirical analysis and modeling. *he European Physical Journal B*, *68*, 261–275.

Wang, X., Yao, E., & Liu, S. (2019). Simulation of metro congestion propagation based on route choice behaviors under emergency-caused delays. *Applied Sciences*, *9*, 348–359.

Wang, Z., Ma, W., & Chan, A. (2020). Exploring the relationships between the topological characteristics of subway networks and service disruption impact. *Sustainability*, *12*. https://doi.org/https://doi.org/10.3390/su12103960

Washington Metropolitan Area Transit Authority. (2022). 2022-system-map. https://www.wmata.com/about/news/New-Silver-Line-Extension-Map.cfm

Wu, W., Zhang, H., Feng, T., & Witlox, F. (2019). A network modelling approach to flight delay propagation: Some empirical evidence from china. *Sustainability*, *11*.

Xiao, X., Jia, L., & Wang, Y. H. (2018). Correlation between heterogeneity and vulnerability of subway networks based on passenger flow. *Journal of Rail Transport Planning & Management*, *8*, 145–157.

Xu, Z., & Chopra, S. (2022). Network-based assessment of metro infrastructure with a spatial–temporal resilience cycle framework. *Reliability Engineering & System Safety*, *223*.

Yap, M., & Cats, O. (2020). Predicting disruptions and their passenger delay impacts for public transport stops. *Transportation*, *48*, 1703–1731. https://doi.org/https://doi.org/10.1007/s11116-020-10109-9

Yap, M., & Cats, O. (2022). Analysis and prediction of ridership impacts during planned public transport disruptions. *Journal of Public Transportation*, *24*, 83–106. https://doi.org/https://doi.org/10.1016/j.jpubtr.2022.100036.

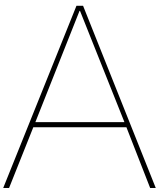
Yap, M., Nijënstein, S., & van Oort, N. (2018a). Improving predictions of public transport usage during disturbances based on smart card data. *Transport Policy*, *61*, 84–95.

Yap, M., van Oort, N., van Nes, R., & van Arem, B. (2018b). Identification and quantification of link vulnerability in multi-level public transport networks: A passenger perspective. *Transportation*, *45*, 1161–1180.

Zeng, Z., & Li, T. (2018). Analyzing congestion propagation on urban rail transit oversaturated conditions: A framework based on sir epidemic model. *Urban Rail Transit*, *4*, 130–140.

Zhang, J., Wang, Z., Wang, S., Luan, S., & Shao, W. (2020). Vulnerability assessments of urban rail transit networks based on redundant recovery. *Sustainability*, *12*.

Zhang, N., Graham, D., Hörcher, D., & Bansal, P. (2021). A causal inference approach to measure the vulnerability of urban metro systems. *Transportation*, *48*, 3269–3300. https://doi.org/https://doi.org/10.1007/s11116-020-10152-6



Scientific paper

The scientific paper starts on the next page.

A model of station vulnerabilities towards delay propagation

Laetitia E. Molkenboer
Delft University of Technology
Delft, the Netherlands

January, 2025

ABSTRACT

Metro networks face operational challenges due to increasing ridership and system growth, particularly in managing delay propagation. Epidemiology models have recently been an interesting method in transportation research for studying delays. This study, therefore, aims to see if the Susceptible-infectious-susceptible (SIS) model is suitable to help model delay propagation in a metro network through its ability to reproduce the vulnerability of metro stations for specific instances. Using data from the Washington Metro Network, delay propagation instances were grouped, and the model was trained and tested using a differential evolution algorithm. The results indicate that the vulnerability values as calculated from the data do not follow the expected trend. Also, the model can predict the vulnerability values for the first group more accurately than the second group. However, limitations such as underestimation and overestimation of station vulnerabilities, and sensitivity to training data and parameters were observed. These challenges stemmed from the dynamics between specific parameters, the mismatch in the order of magnitude of model components, and the lack of additional factors.

Keywords: Delay propagation, Vulnerability, SIS model, Metro stations

1 Introduction

The growth of cities due to urbanization has led to increased ridership and the growth of metro systems in distance (km of infrastructure) and number of stops (UITP, 2022). As a result, the challenge of delays and their propagation to other network parts has become more crucial. Network operators aim to prevent these delay propagations and minimize their impact when they occur, to ensure they do not negatively affect travelers' experiences. Therefore, how and why delays propagate in metro networks must be studied, so that network operators can take the appropriate measures to prevent and mitigate them.

Some studies have used network structure and topological analysis to study delay propagation. However, these studies failed to find any causal relations (Cats & Hijner, 2021; Wang et al., 2020; Yap & Cats, 2020). Another perspective in the literature that has been adopted is assessing the role of vulnerability of nodes and links in propagating delays. Vulnerability is often described in the literature with the words susceptibility and serviceability (Berdica, 2002; Hong et al., 2022; Pan et al., 2021). Combining these keywords, vulnerability is the exposure of a public transport network to disruptions (susceptibility) and at the same time the ability of the PT network to cope with these disruptions (serviceability) (Yap & Cats, 2020).

One of the reasons for considering vulnerability in addition to network topology is that passenger flow distribution has been neglected by most studies, which could lead to underestimation or overestimation of the vulnerability of the network (Eltved et al., 2021; Szymula & Bešinović, 2020; Yap & Cats, 2022). Furthermore, the passenger flows are affected differently during and after a disruption, because they depend on the day and traveler type (e.g. leisure versus work) (Eltved et al., 2021; Xiao et al., 2018; Yap & Cats, 2022). As a result, the vulnerability value can also fluctuate. In addition, the topology analysis does not consider that the disruption effects on passenger behavior differ depending on whether

the disruption was planned or unplanned (Yap et al., 2018a). Furthermore, the effects of disruption were found to be heterogeneous across metro stations and dependent on its location in the network as well as other station-level characteristics, meaning that only using topology analysis would fail to acknowledge other factors planning a role (Zhang et al., 2021). Lastly, at the functional level (flow distribution) initial failure propagates faster than at the structural level (network topology) (Chen et al., 2023), and, hence, only considering the structural level would give a limited picture of the problem. Moreover, both levels have different sources of vulnerabilities (Chopra et al., 2016). Other introduced factors are station-level characteristics (Zhang et al., 2021) and line operations (Malandri et al., 2018). Using these factors it has been found, for example, that the most vulnerable links/lines are those often crowded due to high passenger flows (Shi et al., 2019a; Sun & Guan, 2016; Yap & Cats, 2020; Yap et al., 2018b), with mainly the outflows at stations influencing the vulnerability (Zhang et al., 2020).

Although advancements have been made in public transportation research with the introduction of vulnerability, those studies do not always include delay propagation and the role vulnerability of metro stations plays in delay propagation. Therefore, this study aims to fill the knowledge gap of how to model delay propagation in a metro network through the model's ability to reproduce the vulnerability of metro stations for specific instances.

For several transport modes, epidemic models have been used to study delay propagation as the spreading of diseases and propagation of delays show similarities. This method has proven to be a promising method in the railways (Dekker et al., 2022; Gurin et al., 2020; Monechi et al., 2018). However, these studies use either an epidemic model that does not allow to take heterogeneity into account or failed to consider operational conditions. Also, for air transportation interesting studies have been done, but the translation to metro systems is difficult due to different network dynamics (Baspinar & Koyuncu, 2016; Ceria et al., 2021; Wu et al., 2019). Lastly, the studies for the metro have been solely focused on congestion propagation, the passenger perspective, neglecting the operator perspective (Jia et al., 2022; Shi et al., 2019b; Wang et al., 2019; Zeng & Li, 2018). Therefore, the extent to which epidemic models can be used to model delay propagation from the operator's perspective is an interesting research direction.

2 Methodology

This section explains how this study was performed. subsection 2.1 explains how the instances used in the study were selected. Then, in subsection 2.2 the equations to calculate the vulnerability values from the data are presented. Afterwards, the model itself is explained in subsection 2.3 followed by the training and testing metrics in subsection 2.4.

2.1 Instance selection criteria

Instances from the data must be found that show delay propagation happened and its effects are felt at multiple stations. Therefore, selection criteria are needed. The following five conditions had to be met for an instance to be included in the research:

1. The primary train is delayed;
2. The train behind the primary train is delayed;
3. The train in front of the primary delayed train is not delayed;
4. The primary train was not delayed at the previous station;
5. The train behind was not delayed at the previous station.

The first two conditions clarify if a delay propagation could have happened. Condition three helps ensure that the primary delay train is the first to be delayed. Conditions four and five ensure that the primary train became delayed at the primary delay station and the delay of the train behind was caused by the primary delay. The time window w of the instance determines which train movements are included. The time window w of each instance is defined as the arrival time of the previous train at the primary station till the last train affected by the delay at the primary delay station reaches the station that all affected trains across all instances reach with a delay. This definition means that not all stations on the affected lines are considered. The rationale for focusing on a selection of affected stations rather than the entire line is that some stations experience delays only for certain instances. This results

in significant variability in the delay impact at those stations, posing challenges in effectively capturing these variations and training the model to produce accurate results.

The found instances are grouped based on the primary delay station and direction. They must start at the same station and in the same direction to investigate the consequences of a specific delay. These groups then differ in primary delay station, affected lines, and delay direction so that the model could be tested on multiple parts of the case study network.

2.2 Calculation of station vulnerabilities using the data

The vulnerability of a station is calculated using the train movements during the time window w in the direction of the primary delay. These train movements all use the same infrastructure section k at the station in the direction of the primary delay, because delays have the most impact on metro stations nearby on the same line direction (Cats & Hijner, 2021). An example of multiple infrastructure sections is when a station has multiple levels with tracks and platforms. The infrastructure section k also limits the considered lines to the ones running on that infrastructure. Because only train movements in the direction of the primary delay are considered, no cross-platform delay propagation was considered.

To determine if a train t is delayed, the arrival time difference at station i between two consecutive trains of the same line, t and $t-1$, is calculated and represented as $b_{t,t-1}$. Normally, this difference is equal to the scheduled headway $m_{t,t-1}$ between trains t and $t-1$. However, due to potential variability in service, deviations in headway between two trains could occur without impacting overall network performance. A delay threshold parameter, denoted as δ, is introduced to account for these variations. Its value is dependent on the case study network. If $b_{t,t-1}$ exceeds the sum of the scheduled headway $m_{t,t-1}$ and the delay threshold δ, the train t is considered delayed. In such cases, the variable a_{tiw}, which indicates whether train t was delayed during time window w at station i, is set to 1. The accompanying equation to determine if a train is delayed is shown in Equation 1.

$$a_{tiw} = \begin{cases} 1 & \text{if } b_{t,t-1} \geq m_{t,t-1} + \delta, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

The vulnerability of station i is then the ratio of delayed trains to all observed trains T_{ikw} stopping at a station i using infrastructure k in the time window w of the instance in the direction of the primary delay. Mathematically, the vulnerability of station i translates to Equation 2.

$$v_{ik,data} = \frac{\sum_{t=1}^{T_{ikw}} a_{tiw}}{|T_{ikw}|} \tag{2}$$

2.3 SIS mathematical model for a metro network

The Susceptible-infectious-susceptible (SIS) model is used in this research. Furthermore, as stated in section 1, the effects of disruption were found to be heterogeneous across metro stations (Zhang et al., 2021) and so creating a heterogeneous SIS model as compared to a homogeneous SIS model is more realistic. The translation of the heterogeneous SIS model to a metro network is explained in sections 2.3.1 through 2.3.4.

2.3.1 Network Graph

An undirected P-space graph was created to represent the stations and the links between them. This type of graph has often been called the "space-of-service" (Luo et al., 2020). A weight is assigned to each link in the network. The propagation strength of a delay diminishes with increasing distance (Jia et al., 2022). Hence, the further away two stations are from each other, the less chance a delay starting at station j has to reach station i. Therefore, The link weight is the inverse of the travel time t_{ij} between nodes i and j. These considerations translate to Equation 3.

$$w_{ij} = \frac{1}{t_{ij}} \tag{3}$$

After the initial link weights are set, some of the edge weights are adjusted depending on the instance. For the following three types of edges, the weight is increased to reflect the temporarily increased chance of infection due to the primary delay of the instance, which is visualized in Figure 1:

1. Edges between the primary delay node and all other nodes in the direction of the delay on the same line (P-space);

2. Edges between the nodes that are in the direction of the delay and affected by the primary delay (L-space) ;

3. Edges that are part of the L-space path between two nodes and the path between the two nodes includes the primary delay node.

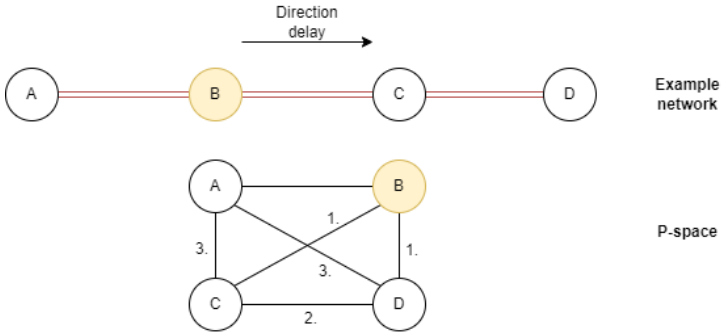


Figure 1: Example network and P-space showing which edge types weight would be adjusted with the numbers representing the described edge types. The yellow node is the infected node in the example.

The edges of types 1, 2, and 3 are part of set E. First, the type 1 and 2 edges were changed in the P-space using Equation 4.

$$w_{ij} = w_{ij} + \frac{w_{ij} * \alpha * a_e}{e^{h(p,j)*\gamma}} \tag{4}$$

The top part of the equation includes factors that influence the delay impact and the lower part reflects the delay effects diminishing with distance. The higher the primary delay duration α, the more the weights are increased as a more severe delay causes more trouble (Cats & Jenelius, 2018; Marra & Corman, 2020). a_e represents the number of lines passing through a station. More lines increase the delay impact because the complexity of operations increases with multiple train operation routes (Lu et al., 2021). $h(p, j)$ defines how many nodes are between the primary delay node p and the currently considered node j in the L-space and, hence, captures the space component of delay propagations. γ is a to-be-trained parameter that captures how quickly the propagation effect diminishes with time.

Finally, the type 3 edge weights are adjusted in the P-space. First, the L-space path, which includes the primary delay node, is determined. Then, it is calculated which percentage of the edges in the path is before the primary delayed station and which after. The initial link weight is then multiplied by the two percentages. The link weight portion reflecting the edges after the primary delayed station is adjusted using Equation 4. That value is then added to the unaffected portion of the link weight. When all the necessary edge weights are changed, all the edge weights are normalized using Equation 5.

$$w_{ij} = \frac{w_{ij}}{\sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij}} \tag{5}$$

2.3.2 Infection Rate

The infection rate of the model represents the susceptibility part of the vulnerability definition. All trains arriving at station i can infect the station if they are delayed, but the more distance the delayed train has to travel, the less infection chance it should have. Also, transfer stations receive many more trains because they serve multiple lines, increasing their chance of infection (Lu et al., 2021). Furthermore, stations linked to transfer stations could be infected by delays from trains on other lines, as long as the trains use the same infrastructure section k. The infection rate equation is presented in Equation 6.

$$i_{ik} = \sum_{j}^{J} (w_{ji} + (w_{ji} * \sum_{s \epsilon S, s \neq j, i} w_{sj} * g_j)) + z_{ik} \tag{6}$$

The infection rate i_{ik} for section k at station i is determined by summing the edge weights between station i and stations j, which are the station's station i is directly connected to. If station j is a transfer station (g_j), stations s also have to be considered given that the line connecting stations s and station j uses the same infrastructure as the line connecting station i to j.

2.3.3 Recovery Rate

The recovery rate of the stations in the model highlights the ability of a station to cope with a disturbance. The recovery rate of a station i is dependent on the number of tracks and the traffic density at the station because no instances where rescheduling strategies were used are considered. More tracks increase the station's flexibility because traffic operators can appoint the trains to more tracks. Also, the service frequency of a line influences the impact (Marra & Corman, 2020; Yap & Cats, 2020), where lines with higher service frequency are affected more than lines with lower service frequency. Stations with a higher traffic density are less flexible because more traffic has to be considered and, hence, have a lower recovery rate. The recovery rate is calculated for each infrastructure k at station i. The number of tracks u_{ki} and the traffic density d_{ki} might differ per section k due to different serviced lines. The recovery rate equation is presented in Equation 7.

$$r_{ik} = c_{ik} + \left(\frac{u_{ki}}{d_{ki}}\right)^{\theta} \tag{7}$$

In Equation 7 d_{ki} is calculated using Equation 8. This equation calculates the number of trains stopping at a station per hour per infrastructure section k using the scheduled headway h_{lki} of line l running on infrastructure section k at station i.

$$d_{ki} = \sum_{l}^{L} \frac{60}{h_{lki}} \tag{8}$$

θ is introduced in Equation 7 to see how heterogeneous the recovery rates of stations are. Also, the parameter c_{ik} captures any unobservable factors influencing the recovery rate, just like z_{ik} does for the infection rate. Both θ and c_{ik} are trained and ≥ 0.

2.3.4 Vulnerability equation

Equation 9 shows how the vulnerability of station i is calculated for a specific section k. This equation reflects the definition of vulnerability given earlier in section 1, where vulnerability is the difference between exposure to a disturbance (infection) and the ability to cope with the disturbance (recovery). As a result of a vulnerability value per k, a station could have multiple vulnerability values if it has several infrastructure sections that are all part of the training.

$$0 = -r_{ik} * v_{ik,model} + (1 - v_{ik,model}) * i_{ik} \tag{9}$$

2.4 Model training and testing metrics

In total four model parameters are trained:

- γ: parameter in the link weight equation capturing the diminishing effect of the delay propagation;
- z_{ik}: parameter in the infection rate equation correcting for any factors currently not considered;
- θ: parameter representing how heterogeneous the recovery rate of stations are in the recovery rate equation;
- c_{ik}: parameter in the recovery rate equation correcting for any currently not considered factors.

The training is done over a group of similar instances, which are randomly split into two groups: 1. training instances and 2. testing instances. The first group is used for the model training and contains 80% of the instances. The second group tests the trained model and has the remaining 20% instances. The mean squared error is computed for each instance used in the training. This calculation is done based on the vulnerabilities calculated from the data and determined by the model, using the formula shown in Equation 10, where N is the number of all stations of all affected lines. The objective function used in the training is to minimize the sum of the MSE of all instances F, which is shown in Equation 11.

$$mse = \frac{\sum_i^N (v_{ik,data} - v_{ik,model})^2}{N} \tag{10}$$

$$min \sum MSE = \sum_{f=1}^{F} mse_f \tag{11}$$

Additionally, the average vulnerabilities of the model and data are compared and the differences in vulnerability between the data and model are determined for each station. All of this is done per k if a station i has multiple infrastructure sections and they are part of the training.

3 Application

3.1 Case Study: Washington Metro network

This section discusses the case study data used and its processing. The Washington Metropolitan Area Transit Authority (WMATA) provided the data, which is about the Washington DC metro network displayed in Figure 2 and is about the year 2019. From this data two files were used: 1. station information and 2. the Automatic Vehicle Location data.

Figure 2: Map of the Washington Metropolitan Area metro network (Washington Metropolitan Area Transit Authority, 2022).

The Washington DC network currently consists of 98 stations, with the last 7 stations opened at the end of 2022. These stations are served by six different lines of which the Gray, Blue, and Orange share a portion of the infrastructure as do the Yellow and Green lines. From the data, instances were collected to train and test the model. At least 10 similar instances had to be found to form a group to ensure enough data to train with. Also, instances where the primary delay led to recovery strategies such as short-turning were not considered. Moreover, all the stations of the considered lines had to be open to ensure a complete picture of the effects of the delay propagation. Two groups of instances were formed. The first group contains instances where the primary delay starts at the King St-Old Town station. Both Yellow and Blue trains are delayed in the direction of Greenbelt and Downtown Largo respectively. The considered stations are those between King St-Old Town and Rosslyn, and King St-Old Town and L'Enfant Plaza as shown in Figure 3.

Figure 3: This figure shows the stations considered for group 1.

The second group concerns primary delays of Blue, Silver, and Orange trains in the direction of Franconia-Springfield, Wiehle-Reston East, and Vienna respectively. The considered stations are those between Stadium-Armory and Rosslyn and displayed in Figure 4.

Figure 4: This figure shows the stations considered for group 2.

For each of the instances, the vulnerability was calculated using the data. The delay threshold to determine whether a train was delayed was set to two minutes. The vulnerabilities of the not-considered stations were forced to be 0.

3.2 Implementation

A few steps were taken to implement the model. First, a P-space and L-space graph of the Washington DC metro network had to be created, which was done using the NetworkX library in Python. P-space and L-space files from previously done research were used to create the necessary graphs (Cats et al., 2019). From these data files, the traveltimes were also obtained and set as edge attributes. After the preparation of the data and the implementation of the model in Python, the groups of instances were used to train the model. For this training, the differential evolution algorithm of the SciPy Python package was used (SciPy, n.d.). The bounds on the parameters were as follows:

- γ: parameter in the link weight equation capturing the diminishing effect of the delay propagation. The bounds are [0, 2];
- z_{ik}: parameter in the infection rate equation correcting for any factors currently not considered. The bounds are [0, 0.5];
- θ: parameter representing how heterogeneous the recovery rate of stations are in the recovery rate equation. The bounds are [0.5, 2];
- c_{ik}: parameter in the recovery rate equation correcting for any factors currently not considered. The bounds are [0, 0.5].

4 Results

This section will analyze the training and testing results for the two formed groups in sections 4.1 and 4.2. Then, the sensitivity analysis results are presented in subsection 4.3.

4.1 Results Group 1: King St-Old Town station

This group contains 10 instances. For more information about the instances of group 1, see Appendix A. For each of these instances, the vulnerability was calculated using the data. On average 9 trains are within the time window at each station and, hence, considered in the vulnerability calculations. The vulnerability values are displayed in Figure 5.

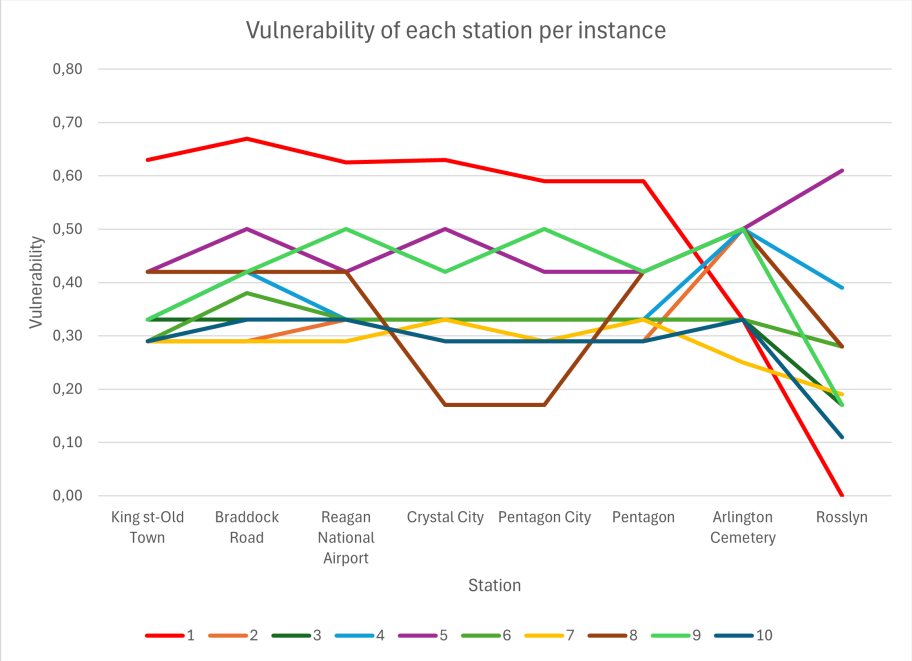


Figure 5: Vulnerability of each station per instance from group 1 in order of stations reached by the trains.

The vulnerabilities were expected to decrease with increasing distance from the primary delay station. A delayed train should catch up on its delay as it uses buffer time. The vulnerabilities for some instances fluctuate and in other cases, the vulnerabilities stay similar across stations. There are a few reasons for this behavior. Firstly, randomly delayed trains are also included in the vulnerability calculations. Trains in front of the primary delay could arrive at a station delayed in the time window, influencing the vulnerability calculations. Another explanation is that the arrival time of trains is around the two-minute threshold, making it delayed at one station, but not at the next. Lastly, the number of trains used in the vulnerability calculations is not the same for each station. A train might arrive at a station a minute before or after the time window cutoff time.

Furthermore, more factors seem to influence the primary delay and its propagation effects. For instances 3 and 8, the primary delay is a Blue line train, which is five minutes delayed at around 11:20 am on a weekday and travels with a headway of 12 minutes. At first sight, these instances are very similar, and the influence of these primary delays is expected to be similar. However, Figure 5 shows that the vulnerability trend for these two instances across stations is very different. While the vulnerabilities of the stations for instance 8 heavily fluctuate, the vulnerabilities of instance 3 stay relatively stable. Hence, other factors must explain this difference in delay propagation effects.

The model was then trained using eight out of 10 instances. The MSE value for the training is 0.022. The trained parameters are presented in Table 1.

Table 1: Trained parameters for group 1: King St

γ		0.455
θ		0.500
Stations	z_{ik}	c_{ik}
Arlington Cemetery	0.254	0.451
Braddock Road	0.390	0.007
Crystal City	0.264	0.248
King St-Old Town	0.303	0.495
L'Enfant Plaza	0.222	0.127
Pentagon	0.269	0.417
Pentagon City	0.412	0.457
Reagan National Airport	0.217	0.378
Rosslyn	0.150	0.358

The low γ value indicates that the edge weights were barely adjusted. For θ the lower bound was found as the best value, meaning the stations were made homogeneous. The large range of values used for the z_{ik} and c_{ik} suggest that the training algorithm chose to compensate with those parameters mostly, instead of using γ and θ. The trained parameters were then tested on instances 3 and 6 from group 1, whose results are shown in Table 2.

Table 2: Testing metrics for group 1.

Metric	*Testing instance 1*	*Testing instance 2*
MSE	0.004	0.004
Average vulnerability data	0.319	0.336
Average vulnerability model	0.295	0.299

The MSE values of 0.004 for both testing instances mean the model was able to predict the vulnerability values of these stations with a margin of less than one train. The difference between the average vulnerability determined from the data and the model is also small for both testing instances. However, for both instances the average vulnerability as determined from the model is higher suggesting underestimation by the model. To better understand this underestimation the difference per station was determined using Equation 12.

$$\delta_{ik} = v_{ik,data} - v_{ik,model} \tag{12}$$

Using Equation 12 the difference was calculated for each k, if applicable, and each station i. These results are displayed in Table 3.

Table 3: Comparison of vulnerabilities as determined from the data and by the model for group 1.

	Testing instance 1			**Testing instance 2**		
Stations	*Data*	*Model*	*Difference*	*Data*	*Model*	*Difference*
Arlington Cemetery	0.333	0.399	-0.066	0.333	0.397	-0.064
Braddock Road	0.333	0.265	0.068	0.375	0.277	0.098
Crystal City	0.333	0.288	0.045	0.333	0.294	0.039
King St-Old Town	0.333	0.235	0.098	0.292	0.240	0.052
L'Enfant Plaza	0.375	0.330	0.045	0.417	0.328	0.089
Pentagon	0.333	0.317	0.016	0.333	0.324	0.009
Pentagon City	0.333	0.275	0.058	0.333	0.281	0.052
Reagan National Airport	0.333	0.279	0.054	0.333	0.290	0.043
Rosslyn	0.167	0.266	-0.099	0.278	0.264	0.014

For almost every station the vulnerability is underestimated. Only the stations Arlington Cemetery and Rosslyn were slightly overestimated. Looking back at Table 1, the underestimation is caused by the training algorithm that chose to mostly compensate the stations with a too low recovery rate instead of stations with a too low infection rate. The recovery rate values are then higher than the infection rates, leading to lower vulnerability values. As the vulnerability values of the testing instances are similar to the average vulnerability values of the training instances, the testing instances will also be underestimated.

4.2 Results group 2: Stadium-Armory station

Group 2 contains 10 instances. Approximately 14 trains are within the time window at each station and, hence, are considered in the vulnerability calculations. For more information about the instances of group 2, see Appendix A. The calculated vulnerabilities are visualized in Figure 6.

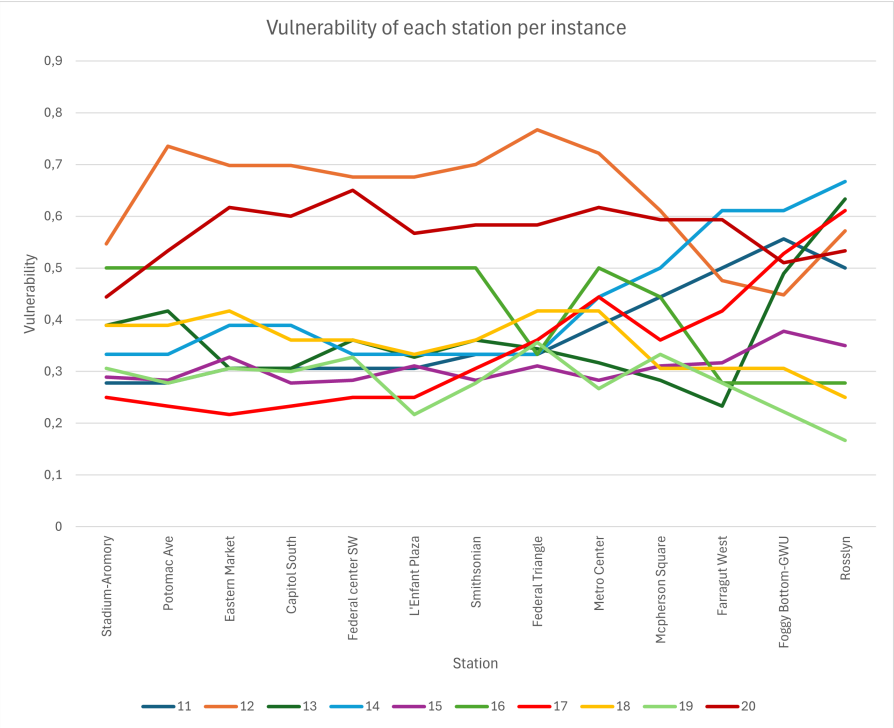


Figure 6: Vulnerability of each station per instance from group 2 in order of stations reached by the trains.

The vulnerability values of the stations in group 2 have a larger range than group 1. Furthermore, these instances also fluctuate instead of showing the expected decreasing trend. The stations considered for group 2 are in the busiest part of the network. If the traffic density is high, it is more difficult for the trains to catch up on their delay. Also, instances 11, 13, 14, and 17 show almost identical behavior but are very different from each other. For these four instances the line, time of primary delay, and scheduled headways are different. Therefore, the group 2 results reiterate that more factors must be at play to explain the vulnerability trends. The MSE value of the trained model for group 2 is 0.043. Table 4 shows the trained parameter values for group 2.

Table 4: Trained parameters for group 2: Stadium-Armory in the west direction till Rosslyn.

γ	0.342	
θ	0.500	
Stations	z_{ik}	c_{ik}
Stadium-Armory	0.219	0.453
Potomac Ave	0.151	0.305
Eastern Market	0.118	0.220
Capitol South	0.151	0.162
Federal Center SW	0.320	0.305
L'Enfant Plaza	0.258	0.212
Smithsonian	0.306	0.195
Federal Triangle	0.348	0.256
Metro Center	0.436	0.242
Mcpherson Square	0.302	0.128
Farragut West	0.320	0.179
Foggy Bottom - GWU	0.440	0.247
Rosslyn	0.379	0.100

Also for group 2, the trained value for γ is low and for θ is the lower bound. Due to the central location of the group 2 stations in the network, they have high baseline infection rate values (so without considering z_{ik}), but the training algorithm still chose to use high values for z_{ik} for most stations. Similarly, while the recovery rates are low for the stations in this part of the network due to the high traffic density, the training algorithm does not use the upper bound.

These trained parameters were tested on instances 15 and 19. Table 5 indicates how well the model performed on these unseen instances.

Table 5: Testing metrics for group 2.

Metric	*Testing instance 1*	*Testing instance 2*
MSE	0.018	0.027
Average vulnerability data	0.308	0.279
Average vulnerability model	0.424	0.414

The model performed worse for the second testing instance than for the first. Also, the MSE values are higher than the values of group 1. The testing MSE values indicate the model was off on average by approximately one train. Furthermore, the distance between the averages as determined by the data and the model is larger than group 1. Group 2 does have a larger range of vulnerability values, which means it is more difficult to find parameter values that fit this range, which could partially explain the worse performance compared to group 1. Table 6 gives more insights into how the model performed for specific stations. These differences were calculated using Equation 12.

Table 6: Comparison of vulnerabilities as determined from the data and by the model for group 2.

	Testing instance 1			**Testing instance 2**		
Stations	*Data*	*Model*	*Difference*	*Data*	*Model*	*Difference*
Stadium-Armory	0.289	0.319	-0.030	0.389	0.413	-0.024
Potomac Ave	0.283	0.321	-0.038	0.389	0.457	-0.068
Eastern Market	0.328	0.383	-0.055	0.417	0.456	-0.039
Capitol South	0.278	0.279	-0.001	0.361	0.424	-0.063
Federal Center SW	0.283	0.293	-0.010	0.361	0.435	-0.074
L'Enfant Plaza	0.311	0.450	-0.139	0.333	0.540	-0.207
Smithsonian	0.283	0.428	-0.145	0.361	0.505	-0.144
Federal Triangle	0.311	0.388	-0.077	0.417	0.482	-0.065
Metro Center	0.283	0.502	-0.219	0.417	0.649	-0.232
Mcpherson Square	0.311	0.441	-0.130	0.306	0.413	-0.107
Farragut West	0.317	0.464	-0.147	0.306	0.453	-0.147
Foggy Bottom - GWU	0.378	0.564	-0.186	0.306	0.548	-0.242
Rosslyn	0.350	0.587	-0.237	0.250	0.592	-0.342

The vulnerability values of the group 2 testing instances are overestimated. This overestimation is not unexpected, because the vulnerability values as calculated from the data are low for these testing instances compared to the training instances. The training algorithm will favor fitting the largest group of similar instances. If those instances have higher vulnerability values, the training algorithm will try to find parameter values that fit those instances. Instances with lower vulnerability values will then be overestimated, highlighting the need for more factors to be included in the model.

4.3 Sensitivity analysis

The results of groups 1 and 2 show similar results even though they are about different parts of the network. Both groups have a low γ and θ value, meaning edge weights were minimally adjusted and the recovery rates were made more homogeneous. The largest difference between the two groups is that the vulnerabilities of the group 1 testing instances were underestimated, while for group 2 they were overestimated. The similar model performance of these two very different groups of instances begs the question of how this could be. One of the reasons for the model behavior is that the order of magnitude of the infection and recovery rate are too different. The baseline infection rate values are much smaller than the baseline recovery rates. Therefore, the infection rate values have to be more compensated to get close to the recovery rate values.

A sensitivity analysis was performed to investigate the model behavior in more detail. The sensitivity of the parameters γ and θ (subsubsection 4.3.1), and which instances are used for training and testing (subsubsection 4.3.2) was explored.

4.3.1 Sensitivity parameters

The model sensitivity was tested for the parameters γ and θ. First, the sensitivity to γ is presented, followed by θ. The sensitivity of both parameters was tested using the training results of group 1.

Sensitivity to γ The γ parameter adjusts the weights of some of the edges in the P-space graph. A high sensitivity means the model could increase the infection rates too much and then overestimate the vulnerabilities, while a low sensitivity would mean that the z_{ik} parameter would have to make all the difference in the infection rate equation. The sensitivity analysis results are presented in Table 7 and Table 8.

Table 7: Testing metrics for sensitivity analysis of the γ parameter.

	$\gamma = 0$		$\gamma = 0.45$		$\gamma = 1$	
Metric	*Testing instance 1*	*Testing instance 2*	*Testing instance 1*	*Testing instance 2*	*Testing instance 1*	*Testing instance 2*
MSE	0.005	0.004	0.004	0.004	0.005	0.004
Average vulnerability data	0.319	0.336	0.319	0.336	0.319	0.336
Average vulnerability model	0.292	0.299	0.295	0.299	0.290	0.294

Table 8: Differences in vulnerability for the testing instances of group 1 for different γ values.

	$\gamma = 0$		$\gamma = 0.45$		$\gamma = 1$	
	Testing instance 1	**Testing instance 2**	**Testing instance 1**	**Testing instance 2**	**Testing instance 1**	**Testing instance 2**
Stations	*Difference*	*Difference*	*Difference*	*Difference*	*Difference*	*Difference*
Arlington Cemetery	-0.067	-0.058	-0.066	-0.064	-0.065	-0.065
Braddock Road	0.101	0.124	0.068	0.098	0.065	0.097
Crystal City	0.051	0.034	0.045	0.039	0.056	0.052
King St-Old Town	0.105	0.054	0.098	0.052	0.100	0.055
L'Enfant Plaza	0.030	0.082	0.045	0.089	0.050	0.090
Pentagon	0.011	-0.010	0.016	0.009	0.024	0.018
Pentagon City	0.052	0.030	0.058	0.052	0.071	0.066
Reagan National Airport	0.071	0.050	0.054	0.043	0.066	0.058
Rosslyn	-0.104	0.025	-0.099	0.014	-0.099	0.012

Based on Table 7, when $\gamma = 0$ or $\gamma = 1$ the model performs worse, but the difference is negligible. Therefore, the model appears to be insensitive to γ. Also, from Table 8 it can be seen that the vulnerability differences are similar across different values for γ. Moreover, from the third column of Table 1 and Table 4 it can be seen that the values for z_{ik} are all quite high. The z_{ik} parameter can compensate for the low infection rate of stations much more easily and accurately than the γ parameter can. These compensations are needed because the infection rate values are very small due to the normalization of the edge weights. The fact that the model uses z_{ik} for this compensation instead of mostly γ, suggests overfitting and that the parameter fails to capture unobserved factors. Therefore, while the model seems insensitive to γ, the training algorithm uses z_{ik} to overfit, because z_{ik} can more accurately help approach the values from the data than γ.

Sensitivity to θ A high θ value means the stations' recovery rates are very heterogeneous, while a low value means stations have a similar recovery rate. The sensitivity of θ was tested by setting it to 0 and 1 and comparing those results with each other and the value from the training. The sensitivity analysis results are displayed in Table 9 and Table 10.

Table 9 shows how a low value for θ leads to underestimation of the vulnerabilities as the average vulnerability for the data is higher. Similarly, a high θ value leads to overestimation. When θ approaches 0, the second part of the recovery rate equation (Equation 7) becomes 1. Hence, the recovery rates of the stations will be $c_{ik} + 1$. The model will then underestimate the vulnerabilities because the infection rates

Table 9: Testing metrics for sensitivity analysis of the θ parameter.

	$\theta = 0$		$\theta = 0.50$		$\theta = 1$	
Metric	*Testing instance 1*	*Testing instance 2*	*Testing instance 1*	*Testing instance 2*	*Testing instance 1*	*Testing instance 2*
MSE	0.019	0.022	0.004	0.004	0.008	0.005
Average vulnerability data	0.319	0.336	0.319	0.336	0.319	0.336
Average vulnerability model	0.197	0.197	0.295	0.299	0.378	0.384

Table 10: Differences in vulnerability for the testing instances of group 1 for different θ values.

	$\theta = 0$		$\theta = 0.50$		$\theta = 1$	
	Testing instance 1	**Testing instance 2**	**Testing instance 1**	**Testing instance 2**	**Testing instance 1**	**Testing instance 2**
Stations	*Difference*	*Difference*	*Difference*	*Difference*	*Difference*	*Difference*
Arlington Cemetery	0.030	0.030	-0.066	-0.064	-0.166	-0.166
Braddock Road	0.157	0.198	0.068	0.098	-0.009	0.026
Crystal City	0.131	0.131	0.045	0.039	-0.023	-0.029
King St-Old Town	0.177	0.135	0.098	0.052	0.029	-0.019
L'Enfant Plaza	0.188	0.229	0.045	0.089	-0.045	-0.005
Pentagon	0.129	0.129	0.016	0.009	-0.088	-0.098
Pentagon City	0.146	0.146	0.058	0.052	-0.015	-0.022
Reagan National Airport	0.143	0.143	0.054	0.043	-0.043	-0.058
Rosslyn	-0.003	0.109	-0.099	0.014	-0.169	-0.058

can never be that high. Similarly, when θ approaches 1, the differences in recovery rates are enlarged. Most of the stations in group 1 have very similar recovery rates. Thus, enlarging these differences only leads to a similar decrease in recovery rate values with an overestimation of the vulnerability values as a consequence. This analysis is also reflected by Table 10. When $\theta = 0$ the model underestimates the vulnerabilities and when $\theta = 1$ the model overestimates the vulnerabilities. Therefore, the model is sensitive to the θ parameter.

4.3.2 Sensitivity training and testing instances

To test how sensitive the model performance is to which instances are used in the model training and testing, the division of training and testing instances was changed for group 1. This time instead of having instances 3 and 6 used for testing, instances 1 and 9 were randomly chosen. Tabels 11 and 12 show the results of this sensitivity analysis.

Table 11: Testing metrics for sensitivity analysis of instance division for training and testing.

Metric	*Instance 2*	*Instance 9*
MSE	0.010	0.028
Average vulnerability data	0.350	0.417
Average vulnerability model	0.304	0.280

Table 12: Comparison of vulnerabilities as determined from the data and by the model for the testing instances used in the experiment.

	Instance 2			**Instance 9**		
Stations	*Data*	*Model*	*Difference*	*Data*	*Model*	*Difference*
Arlington Cemetery	0.500	0.403	0.097	0.500	0.362	0.138
Braddock Road	0.292	0.286	0.006	0.417	0.265	0.152
Crystal City	0.292	0.286	0.006	0.417	0.263	0.154
King St-Old Town	0.292	0.232	0.060	0.333	0.211	0.122
L'Enfant Plaza	0.583	0.310	0.273	0.500	0.293	0.207
Pentagon	0.292	0.350	-0.058	0.417	0.327	0.090
Pentagon City	0.292	0.284	0.008	0.500	0.265	0.235
Reagan National Airport	0.333	0.297	0.036	0.500	0.269	0.231
Rosslyn	0.278	0.287	-0.009	0.167	0.267	-0.100

Comparing tables 2 and 3 to 11 and 12 show that when instances 3 and 6 were used for testing instead of 2 and 9, the model performed better. The MSE was lower and the difference between the averages was smaller. Also, the vulnerability value of L'Enfant Plaza for instance 2 is much higher than for the other instances, causing the model to underestimate heavily. Similarly, the station vulnerabilities for instance 9 are also higher than average, leading to underestimation by the model. Therefore, the model is sensitive to which instances are used for training and testing.

The sensitivity analysis uncovered a few key limitations. Firstly, the model reacts poorly to testing instances that are different from the instances used in training. Hence, the model struggles to generalize to atypical instances. Secondly, the model heavily depends on specific information from the data. For example, the vulnerabilities of the group 1 instances show that while they are similar, the calculated vulnerabilities vary. Therefore, these variations are under-modeled. Lastly, the model sensitivity, the order of magnitude of the infection and recovery rate, and the resulting dynamics between parameters γ and z_{ik} causes the model to overfit slightly. Addressing these issues, the overfitting will be reduced and the model will be able to generalize better to unseen instances.

4.4 Benchmarking

As this model is the first of its kind, it is not possible to benchmark this model using literature. However, the model can be compared to another version of itself. Hence, the model presented in this study is compared to a model version where the constants z_{ik} and c_{ik} in the infection rate and recovery rate equations respectively are changed. Therefore, the model is benchmarked by creating two additional models:

- a model where the constants are not included;
- a model where the constants are made non-station-specific.

The advantage of (station-specific) constants is that any factors currently not included can be captured more accurately, especially as the effects of these factors might differ greatly across stations. The disadvantage, however, is that introducing constants means additional parameters have to be trained, which increases the running time of the model training.

The two additional model configurations were trained using the instances of group 1. For all models, the same training settings were used. Table 13 shows the model performance for all model configurations, including the model version used in this study for easier comparison. The MSE values in Table 13 are based on Equation 11.

Table 13: Training results for benchmarking.

	No z_{ik} and c_{ik}	*Non station-specific z_{ik} and c_{ik}*	*Station-specific z_{ik} and c_{ik}*
MSE	0.074	0.072	0.022

The model performs similarly for the model configuration with no constants at all and non-station-specific constants. While the MSE value does go down, the difference is negligible. The model configuration where the parameters z_{ik} and c_{ik} are station-specific performs better with a MSE decrease of 0.05. To better understand how the model configurations performed, one also has to look at Table 14.

Table 14: Trained parameters for all three model configurations. For the constants c and z, a range of values obtained for the station-specific constants is presented.

	No constants	**Non-station specific**	**Station specific**
γ	0.358	0.355	0.449
θ	0.500	0.674	0.500
c	-	0.201	0.1 - 0.5
z	-	0.001	0.1 - 0.5

The γ values in Table 14 indicate minimal adjustment of edge weights across all model configurations. For non-station-specific constants, the infection rate constant z was also low. Higher γ and z values, leading to increased infection rates, produced worse results. Instead, the best training outcomes arose from compensating for stations with low recovery rates.

Station-specific constants outperform other configurations by enabling station-level compensation instead of a single value for all stations. This suggests a station-specific factor that can only be captured with tailored parameters. The model benefits from this flexibility, as evidenced by the broader use of parameter boundaries in the station-specific case.

5 Conclusion

This study aimed to fill the research gap of how delay propagation in a metro network could be modeled using the SIS model as inspiration to reproduce the vulnerability of a metro station for specific instances. A model based on the SIS model was constructed and trained for several parameters using data about the Washington Metro network. The model training and testing results for both groups were analyzed and the conditions under which the model produced the results were reflected upon through a sensitivity analysis. The results indicate that the model can reproduce the vulnerabilities of instances for two different groups with an accuracy of less than one train for the first and one train for the second. Also, the benchmarking showed that the current model performs well compared to other model configurations. At the same time, the sensitivity analysis showed the model's sensitivity to outliers. Therefore, the model lacks robustness. Furthermore, there is a dependence on the input data, which means that more factors need to be considered to decrease this dependence and improve how the model reacts to 'odd' instances. Lastly, the model is currently overfitting, which is the result of its sensitivity and the model dynamics. Solving the other limitations will help reduce the overfitting of the model.

Even so, this first model allows network operators to identify the vulnerable stations in the network, under which circumstances they are vulnerable, and which factors play a role in this vulnerability. This information allows them to have a more targeted approach when delay propagations do happen. Instead of uniform strategies across all stations, WMATA can create adaptive response plans that vary depending on the delay propagation instance. Also, to prevent delay propagations the network operator can be more proactive with targeted maintenance and infrastructure upgrades.

For future research, it is important to tackle the issues mentioned in the sensitivity analysis. The inclusion of additional factors should help decrease the variation now seen across similar instances. Also, reconsidering some of the model equations and the vulnerability calculation of the data will help limit the overfitting. Another direction could be the application of the model to additional parts of the WMATA network or a completely different network to further investigate how the model should be developed. Application to another network will also help uncover the sensitivity of factors that could not be explored in this study due to how the Washington network was built. For example, all stations in the WMATA network have two tracks, except for the Reagan National Airport station, which means it was not possible to test the added value of this factor.

Acknowledgements

The author would like to thank the Washington Metropolitan Area Transit Authority for providing the data that made this study possible. This paper was written as part of the thesis project for the MSc in Transport, Infrastructure & Logistics at Delft University of Technology, under the supervision of prof. dr. O. Cats, dr. F. Schulte and dr. Y. Zhu.

References

Baspinar, B., & Koyuncu, E. (2016). A data-driven air transportation delay propagation model using epidemic process models. *International Journal of Aerospace Engineering*, *2016*.

Berdica, K. (2002). An introduction to road vulnerability: What has been done, is done and should be done. *Transport Policy*, *9*, 117–127.

Cats, O., & Hijner, A. (2021). Quantifying the cascading effects of passenger delays. *Reliability Engineering and System Safety*, *212*. https://doi.org/https://doi.org/10.1016/j.ress.2021.107629.

Cats, O., & Jenelius, E. (2018). Beyond a complete failure: The impact of partial capacity degradation on public transport network vulnerability. *Transportmetrica B: Transport Dynamics*, *6*, 77–96.

Cats, O., van Cranenburgh, S., Vijlbrief, S., Krishnakumari, P., & Massobrio, R. (2019, October). A curated data set of p-space representations for 51 metro networks worldwide. https://doi.org/10.4121/21316824

Ceria, A., Köstler, K., Gobardhan, R., & Wang, H. (2021). Modeling airport congestion contagion by heterogeneous sis epidemic spreading on airline networks. *Plos one*, *16*.

Chen, C., Wang, S., Zhang, J., & Gu, X. (2023). Modeling the vulnerability and resilience of interdependent transportation networks under multiple disruptions. *Journal of Infrastructure Systems*, *29*.

Chopra, S., Dillon, T., Bilec, M., & Khanna, V. (2016). A network-based framework for assessing infrastructure resilience: A case study of the london metro system. *Journal of The Royal Society Interface*, *13*.

Dekker, M. M., Medvedev, A. N., Rombouts, J., Siudem, G., & Tupikina, L. (2022). Modelling railway delay propagation as diffusion-like spreading. *EPJ Data Science*, *11*.

Eltved, M., Breyer, N., Ingvardson, J. B., & Nielsen, O. A. (2021). Impacts of long-term service disruptions on passenger travel behaviour: A smart card analysis from the greater copenhagen area. *Transportation Research Part C: Emerging Technologies*, *131*.

Gurin, D., Prokhorchenko, A., Kravchenko, M., & Shapoval, G. (2020). Development of a method for modelling delay propagation in railway networks using epidemiological sir models. *Eastern-European Journal of Enterprise Technologies*, *6*.

Hong, W. T., Clifton, G., & Nelson, J. D. (2022). Rail transport system vulnerability analysis and policy implementation: Past progress and future directions. *Transport Policy*, *128*, 299–308.

Jia, C., Zheng, S., Qian, H., Cao, B., & Zhang, K. (2022). Analysis of crowded propagation on the metro network. *Sustainability*, *14*.

Lu, J., Ma, X., & Xing, Y. (2021). Risk factors affecting the severity of disruptions in metro operation in shanghai, 2013-2016. *Journal of Transportation Safety & Security*, *13*, 69–92.

Luo, D., Cats, O., & van Lint, H. (2020). Can passenger flow distribution be estimated solely based on network properties in public transport systems? *Transportation*, *47*, 2757–2776.

Malandri, C., Fonzone, A., & Cats, O. (2018). Recovery time and propagation effects of passenger transport disruptions. *Physica A: Statistical Mechanics and its Applications*, *505*, 7–17. https://doi.org/https://doi.org/10.1016/j.physa.2018.03.028

Marra, A. D., & Corman, F. (2020). From delay to disruption: Impact of service degradation on public transport networks. *Transportation Research Record*, *2674*, 886–897.

Monechi, B., Gravino, P., Di Clemente, R., & Servedio, V. D. (2018). Complex delay dynamics on railway networks from universal laws to realistic modelling. *EPJ Data Science*, *7*.

Pan, S., Yan, H., He, J., & He, Z. (2021). Vulnerability and resilience of transportation systems: A recent literature review. *Physica A: Statistical Mechanics and its Applications*, *581*.

SciPy. (n.d.). Differential$_e$*volution*. https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.differential_evolution.html

Shi, J., Wen, S., Zhao, X., & Wu, G. (2019a). Sustainable development of urban rail transit networks: A vulnerability perspective. *Sustainability*, *11*.

Shi, Z., Zhang, N., & Zhu, L. (2019b). Understanding the propagation and control strategies of congestion in urban rail transit based on epidemiological dynamics model. *Information*, *10*.

Sun, D., & Guan, S. (2016). Measuring vulnerability of urban metro network from line operation perspective. *Transportation Research Part A: Policy and Practice*, *94*, 348–359.

Szymula, C., & Bešinović, N. (2020). Passenger-centered vulnerability assessment of railway networks. *Transportation Research Part B: Methodological*, *136*, 30–61.

UITP. (2022, May). World metro figures 2021.

Wang, X., Yao, E., & Liu, S. (2019). Simulation of metro congestion propagation based on route choice behaviors under emergency-caused delays. *Applied Sciences*, *9*, 348–359.

Wang, Z., Ma, W., & Chan, A. (2020). Exploring the relationships between the topological characteristics of subway networks and service disruption impact. *Sustainability*, *12*. https://doi.org/https://doi.org/10.3390/su12103960

Washington Metropolitan Area Transit Authority. (2022). 2022-system-map. https://www.wmata.com/about/news/New-Silver-Line-Extension-Map.cfm

Wu, W., Zhang, H., Feng, T., & Witlox, F. (2019). A network modelling approach to flight delay propagation: Some empirical evidence from china. *Sustainability*, *11*.

Xiao, X., Jia, L., & Wang, Y. H. (2018). Correlation between heterogeneity and vulnerability of subway networks based on passenger flow. *Journal of Rail Transport Planning & Management*, *8*, 145–157.

Yap, M., & Cats, O. (2020). Predicting disruptions and their passenger delay impacts for public transport stops. *Transportation*, *48*, 1703–1731. https://doi.org/https://doi.org/10.1007/s11116-020-10109-9

Yap, M., & Cats, O. (2022). Analysis and prediction of ridership impacts during planned public transport disruptions. *Journal of Public Transportation*, *24*, 83–106. https://doi.org/https://doi.org/10.1016/j.jpubtr.2022.100036.

Yap, M., Nijënstein, S., & van Oort, N. (2018a). Improving predictions of public transport usage during disturbances based on smart card data. *Transport Policy*, *61*, 84–95.

Yap, M., van Oort, N., van Nes, R., & van Arem, B. (2018b). Identification and quantification of link vulnerability in multi-level public transport networks: A passenger perspective. *Transportation*, *45*, 1161–1180.

Zeng, Z., & Li, T. (2018). Analyzing congestion propagation on urban rail transit oversaturated conditions: A framework based on sir epidemic model. *Urban Rail Transit*, *4*, 130–140.

Zhang, J., Wang, Z., Wang, S., Luan, S., & Shao, W. (2020). Vulnerability assessments of urban rail transit networks based on redundant recovery. *Sustainability*, *12*.

Zhang, N., Graham, D., Hörcher, D., & Bansal, P. (2021). A causal inference approach to measure the vulnerability of urban metro systems. *Transportation*, *48*, 3269–3300. https://doi.org/https://doi.org/10.1007/s11116-020-10152-6

Appendix A

Table 15 and Table 16 show the instances used for the training and testing of the model belonging to each group. The bold instances are the ones used in the testing, while the non-bold instances are used for the training.

Table 15: Data first group, end station = last station train delayed, Y = Yellow, B = Blue.

Instance	Line of primary delay	Delay duration primary delay	Time window	Headway [s]
1	Yellow	6	14:07-14:47	Y: 480 B: 720
2	Blue	5	16:24-17:00	480
3	Blue	5	11:10-11:53	720
4	Blue	6	20:12-20:56	720
5	Blue	7	10:07-10:52	900
6	Yellow	5	13:48-14:30	Y: 480 B: 700
7	Blue	5	15:36-16:10	480
8	Blue	5	11:04-11:48	720
9	Yellow	7	11:01-11:41	720
10	Yellow	5	16:20-16:55	480

Table 16: Data second group, B = Blue, O = Orange, S = Silver.

Instance	Line of primary delay	Delay duration primary delay	Time window	Headway [s]
11	Blue	13	13:33-14:27	720
12	Blue	7	17:30-18:49	480
13	Silver	15	08:58-09:53	B: 720 O: 600 S: 660
14	Orange	7	11:02-11:51	720
15	Orange	10	19:11-20:01	B: 480 O: 360 S: 540
16	Blue	8	22:53-00:03	1200
17	Blue	5	15:02-15:44	480
18	Orange	7	12:33-13:45	900
19	Blue	8	16:52-17:33	480
20	Orange	6	18:33-19:40	480

B

Instances

Group 1 Table B.1 shows the instances used for the training and testing of the model belonging to group 1. The bold instances are the ones used in the testing, while the non-bold instances are used for the training.

Table B.1: Data first group, end station = last station train delayed, Y = Yellow, B = Blue.

Instance	Line of primary delay	Delay duration primary delay	Time window	Headway [s]
1	Yellow	6	14:07-14:47	Y: 480 B: 720
2	Blue	5	16:24-17:00	480
3	Blue	5	11:10-11:53	720
4	Blue	6	20:12-20:56	720
5	Blue	7	10:07-10:52	900
6	Yellow	5	13:48-14:30	Y: 480 B: 700
7	Blue	5	15:36-16:10	480
8	Blue	5	11:04-11:48	720
9	Yellow	7	11:01-11:41	720
10	Yellow	5	16:20-16:55	480

Group 2 Table B.2 displays all the instances part of group 2. Again the bold instances were used in the testing and the non-bold instances for the model's training.

Table B.2: Data second group, B = Blue, O = Orange, S = Silver.

Instance	Line of primary delay	Delay duration primary delay	Time window	Headway [s]
11	Blue	13	13:33-14:27	720
12	Blue	7	17:30-18:49	480
13	Silver	15	08:58-09:53	B: 720 O: 600 S: 660
14	Orange	7	11:02-11:51	720
15	Orange	10	19:11-20:01	B: 480 O: 360 S: 540
16	Blue	8	22:53-00:03	1200
17	Blue	5	15:02-15:44	480
18	Orange	7	12:33-13:45	900
19	Blue	8	16:52-17:33	480
20	Orange	6	18:33-19:40	480