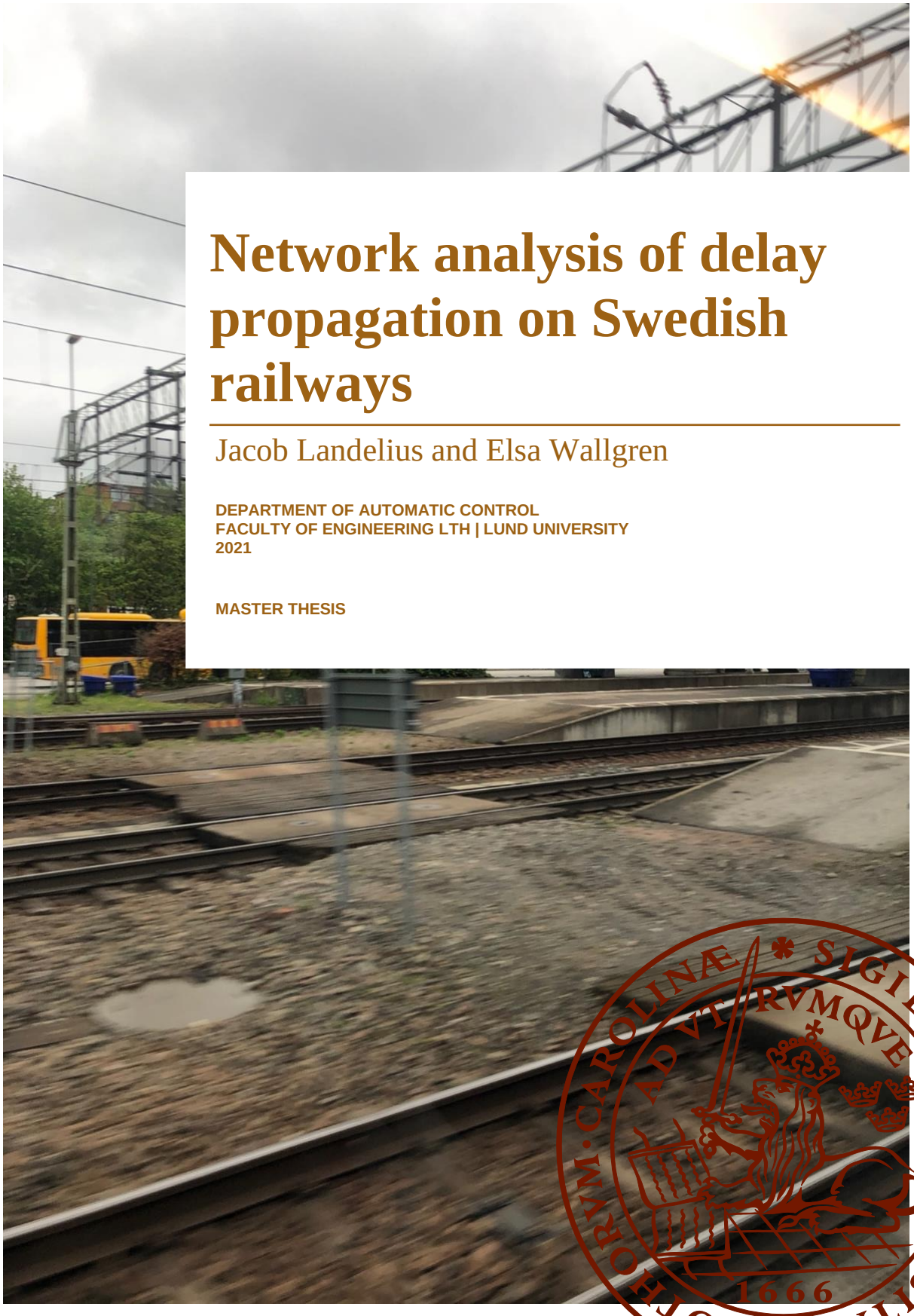

# Network analysis of delay propagation on Swedish railways

Jacob Landelius and Elsa Wallgren

DEPARTMENT OF AUTOMATIC CONTROL
FACULTY OF ENGINEERING LTH | LUND UNIVERSITY
2021

MASTER THESIS

MSc Thesis

Department of Automatic Control Lund University
Box 118
SE-221 00 LUND
Sweden



Lund 2021

**Abstract**

Travel on railway in Sweden has increased steadily over the past three decades and there are today twice as many passengers travelling by train as there were 30 years ago. The increasing awareness of environmental issues with other methods of transportation is likely to favor railway travel, and so the number of passengers is expected to continue to rise. As the number of passengers increase, so do the requirements on keeping trains on time. Delayed trains do not only cost money for train operators, but may also affect how likely we are to choose to travel by train. Understanding where and why delay occurs as well as how this delay might spread is therefore important not only from an economic point of view but from an environmental one as well.

This thesis shows that behavior of delay occurrence and propagation of delay in the Swedish railway network may be reproduced using an epidemic Susceptible-Infected-Susceptible (SIS) model with satisfying results. By optimizing the probability of a train carrying infection (delay) from an infected station to a susceptible one, the simulation can reproduce the level of delay over time as well as the geographic distribution of delay, thus capturing global as well as local delay behavior. The thesis further shows the necessity of heterogeneous delay propagation probabilities on edges in the network in order to reproduce real-world behavior.

Furthermore, the results indicate that re-scaling the nodal self-infection rate (spontaneous delay rate) improves the model and is needed when changing the number of delayed departures needed to make an infected state, i.e. marking a station as delayed. Simulations indicate that this nodal self-infection cannot be expressed using a linear function but varies non-linearly with the number of delayed departures. One explanation for this would be the possible dependence between self-infection rates on trains, which could be explained by the fact that some external factors giving rise to spontaneous delay might affect entire stations rather than individual departures. However, the effect of re-scaling the self-infection rate could also be obtained by increasing the recovery rate, why further analysis is needed in order to determine which parameter should be modified to what extent.

Lastly, results also indicate that the model may be used for estimating the impact on certain delay preventive measures by manually changing parameters for chosen stations or railway lines. This is valuable as it may give stakeholders a way of prioritizing projects and resources.

***Keywords***— networks, graph theory, delay, delay propagation, railway

# Acknowledgements

This thesis would not have been possible without the extensive support from the Department of Automatic Control at Lund University. We would especially like to extend our thanks to our supervisor Dr. Giacomo Como for sparking an interest in this topic to start with, for believing in our project since day one and for being there to guide us. We are also very grateful to Dr. Emma Tegling for endorsing the project early on, and to Alain Govaert for helping us to navigate this research field and for his invaluable feedback along the way. We feel very privileged for the amount of support and interest the Department has taken in our work, and are very grateful to have had several people to turn to for guidance.

We also thank Carl-William Palmqvist with Transport and Roads at Lund University for the collection and categorization of data used to conduct this analysis. His expertise and previous research have been a terrific resource when writing throughout the project. A similar thanks goes out to the Swedish Transport Authority, and especially to Dr. Anders Wigren, where we have been met with nothing but an eagerness to help and a curiosity as for what our project might entail.

Lastly, we thank Prof. Anders Rantzer, Head of the Department for Automatic Control at Lund University, for examining this thesis and improving its content. This thesis marks the end of our studies at the program for Industrial Engineering and Management at LTH, and we are both thankful for the years we have been able to spend in Lund and proud of the academic contribution that marks the end of them.

# Contents

# 1 Glossary

## Definitions

- Spontaneous delay: refers to delay caused by external factors, such as bad weather or personnel strike
- Propagated delay: refers to delay caused by other delay, i.e. a train might be delayed due to propagated delay if it is delayed by another train arriving or departing late
- Junctions: refers to stations connected by railway to more than two stations, i.e. nodes with degree $> 2$
- Endpoints: refers to stations connected to only one other stations, i.e. nodes with degree $= 1$
- Line stations: refers to stations connected to two other stations, i.e. nodes with degree $= 2$

## Notations

- $\mathcal{N}$: denotes the set of nodes
- $\mathcal{E}$: denotes the set of edges
- $\mathcal{G}$: denotes the graph made up of the tuple ($\mathcal{N}$, $\mathcal{E}$), or triple ($\mathcal{N}$, $\mathcal{E}$, $W$)
- $\mathcal{A}$: denotes the adjacency matrix connected to the graph $\mathcal{G}$
- $W$: denotes the weight matrix connected to the graph $\mathcal{G}$
- $S$: denotes the state matrix containing information about which states the node set $\mathcal{N}$ is in for each time step

## Parameters

- $\beta_{i,j}$: the edge-specific rate of propagated delay
- $\epsilon_j$: the train-node-specific rate of spontaneous delay
- $\delta_j$: the node-specific rate of recovery

# 2 Introduction

Travel on railway in Sweden has increased steadily over the past three decades, at the rate of approx. 2-3% per year, and there are today twice as many passengers travelling by train as there were 30 years ago. As the number of passengers has increased, so have the requirements on keeping trains on time (Palmqvist, 2019). According to the Swedish Transportation Authority, if defining a delay as a delayed departure or arrival of more than 5 minutes, approximately 90% of all trains meet their timetable (Trafikverket, 2020). This means that 10% of trains are delayed by more than 5 minutes.

These delays have an economic impact as well as an impact on the attractiveness of traveling by train and hence pose a threat to the conversion to railway travel from airway and road travel. Thus, the knowledge of how delays arise and propagate throughout the system is of great importance in order to predict, recover and prevent delays (Palmqvist, 2019).

Previous research indicate that this knowledge can be obtained through mathematical analysis as researchers have successfully characterized *delay propagation* in the railway system. One example of such an analysis can be found in the paper *Complex delay dynamics on railway networks from universal laws to realistic modelling* from 2018 by Monechi et al. (Monechi et al., 2018). In this study, the authors discuss a delay propagation mechanism on railways and state that their model is capable of reproducing the empirical distribution of delays measured in the data as well as the emergence of large congested areas, which supports the proposition that delay propagation can be successfully analyzed using mathematics.

Other attempts to model delay leverages the fact that transport systems can be mathematically represented using a graph. As it is not unnatural to assume (as this thesis will show) that delay behavior is linked to the underlying structure of the transport system (the graph), researchers have attempted to analyze this through applied network science. One such example is the research submitted by Li et al. in the paper *A Spectral Approach Towards Analyzing Air Traffic Network Disruptions* from 2019 (Li et al., 2019). In this paper, the authors apply Graph Signal Processing (GSP) techniques to the analysis of delay in flight networks. In the underlying graph, Li et al. define nodes as airports. Edges between nodes are thus represented by flights between these airports, in a way similar to how railway connects railway stations (which is leverages in this thesis). By assigning edges a weight equal to delay correlation between airport-pairs, the authors compute the weight matrix and model the graph representing the U.S. flight network.

In order to *track* propagated delay (over space and time), one might apply what is commonly referred to as *network epidemic models* to transport systems. These models are traditionally used to model the spread of infectious disease by assigning nodes (typically representing individuals in the conventional case) states. One example of such a model is the SIS (Susceptible-Infected-Susceptible) model, where nodes can take on one of two states: *Susceptible* or *Infected*. This model is typically employed to model a disease where an individual may be infected more than once, meaning that the individual could catch the disease, recover from it and then be infected anew.

This SIS model can provide original insights when applied to transport systems, as the states Susceptible and Infected may then correspond to a vehicle being "on time" or "delayed", according to some definition. In the paper *A Network Modelling Approach*

*to Flight Delay Propagation: Some Empirical Evidence from China* from 2019 (Wu et al., 2019), Wu et al. apply the SIS model to model delay propagation between flights in China. In this study, nodes are defined as flights which are connected to other flights by edges if they interact in space and time. The authors go on to argue that flight delay propagation and epidemic spreading have common features, as delays of upstream flights are the main cause for delays in downstream flights. Using the SIS approach, Wu et al. classify flights as either **Susceptible** (to delay) or **Infected** (delayed) and compute the edge-dependent probability of becoming infected ($\beta_{i,j}$) and the node-specific probability of recovering from delay ($\delta_i$). The delay propagation process is illustrated as per below.

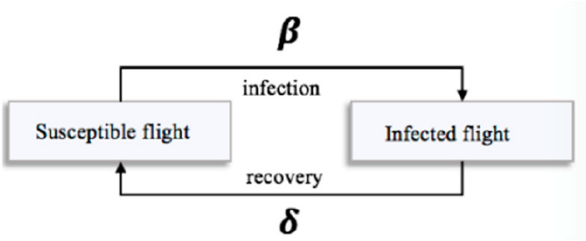


Figure 1: Cyclical epidemic processes between SIS model components $S$ and $I$ © (Wu et al., 2019)

A similar analysis on the US flight network is conducted by Ceria et al. and presented in the article *Modeling airport congestion contagion by heterogeneous SIS epidemic spreading on airline networks* from 2021 (Ceria et al., 2021).

Ceria et al., similarly to Wu et al., model delay congestion in a flight network using an SIS spreading process, but let nodes correspond to airports rather than individual flights. As in the study by Wu et al., the authors apply an SIS model with heterogeneous infection rates and recovery rates. The rationale behind this is that different airports may take different time to recover from delay congestion due to difference in infrastructure, the assumption being that larger airports (with more links to other airport) take less time to recover.

Whilst this analysis showed that delay propagation in airway systems could be analyzed through applied network science with the use of an SIS model, the model did not account for *exogenous* delay defined by Monechi et al., which was considered a flaw. In the SIS model, the exogenous delay would correspond to a nodal self-infection mechanism, allowing nodes to spontaneously alter their state from *Susceptible* to *Infected*. Such a mechanism was introduced by Van Mieghem et al. in the paper *Epidemics in networks with nodal self-infection and the epidemic threshold* from 2012. In this paper, Van mieghem et al. discuss a modified SIS model, referred to as the $\epsilon$-SIS model, where the parameter $\epsilon$ refers to the rate of nodal self-infection. This model reduces to the "classical" SIS model when $\epsilon = 0$.

Having reviewed this previous research, this thesis committed to examining whether a modified SIS model allowing nodal self-infection (an $\epsilon$-SIS model) could be applied to model the behavior and propagation of delay on the *Swedish railway system*. The belief was that this, if successfully implemented, could provide valuable insights into which parameters affect delay occurrence and propagation on both a global and local scale. These insights could then be employed to derive well-targeted and efficient

preventive and reactive measures, by simulating the results of such before-hand, to reduce overall as well as local delay in the railway system. This would in turn improve timetable accuracy and - over time - increase the attractiveness of and adaption to train travel in Sweden.

The results obtained would also be viable for any (fully connected) railway system and could be adapted to model delay behavior on other transport systems as well.

## 2.1 Thesis outline

In this thesis, the reader is first introduced to the glossary containing definitions and notations frequently used in the text. Next, the reader is introduced to an introduction of the thesis topic, including an introduction to the problem at hand and the proposed methodology to solve it. This is followed by a background on preliminaries tied to the methodology before moving on to research methodology, where data collection and processing is discussed.

Following this, the reader is provided with a graph representation of the Swedish railway and characteristics of it, before being introduced to the network epidemic model used to simulate railway delay propagation. Following this, the thesis results are presented, followed by a sensitivity analysis of the same and implications from these results are discussed in section 9. Lastly, the reader is provided with a thesis summary and an appendix containing additional information and results from initial simulation attempts.

# 3 Preliminaries

A network is defined as a set of interconnected objects (called nodes or vertices). The nodes are connected by edges (also called *links*). Network science is the study of networks as a representation of either symmetric relations or asymmetric relations between objects (Wellman, 1983). The theory is widely applicable and has been used to analyze topics including, but not limited to, social and social media networks, electrical power systems and failures on these, airway travel networks and traffic congestion (Barabási, 2013). A network may be represented mathematically by a graph.

## 3.1 Graph theory

### 3.1.1 Introducing graph notations

When illustrating a system with a graph, the graph $\mathcal{G}$ is generally defined as a tuple $(\mathcal{N}, \mathcal{E})$ or triple $(\mathcal{N}, \mathcal{E}, W)$, where:

- $\mathcal{N}$ is a finite set of nodes,
- $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ is a set of links, where $e = (i, j) \in \mathcal{E}$ indicates the existence of a link from node $i$ to node $j$, and
- $W \subseteq R_+^{\mathcal{N}\times\mathcal{N}}$ is a matrix describing the intensities, or *weights*, of each link.

### 3.1.2 Undirected/directed graph

A graph may be characterized in many different ways. For example, a graph may be undirected or directed. This can be explained using a graph where nodes illustrate traffic intersections and edges illustrate streets or roads connecting these intersections. Such edges could be either undirected, $e_{(i,j)} = e_{(j,i)}$ (traffic between two intersections runs in both directions), or directed $e_{(i,j)} \neq e_{(j,i)}$ (traffic between the two intersections is one-way only). A graph consisting entirely of undirected edges is called an *undirected graph*, whereas a graph with at least one directed edge is called a *directed graph*. More formally, a graph is undirected *if and only if* $A_{(i,j)} = A_{(j,i)}, \forall i, j \in \mathcal{N}$. Whether a graph is undirected or directed is important as it has implications on the complexity of computations available on the graph.

### 3.1.3 Unweighted/weighted graph

A graph may also be unweighted or weighted. In an unweighted graph, the value on the edge between two nodes is either 1 (if the two nodes are connected) or 0 (if they are not). In other words, the edge values are binary as are the entries in the adjacency matrix. In a weighted graph however, the edge may carry a weight. That is, the connection between two nodes may be given a weight connected to some feature. For example, in the example where nodes are traffic intersections and edges are roads connecting those intersections, the weight on a given edge might be the traffic capacity of the road represented by the edge. A weighted graph may be represented by a weighted adjacency matrix with non-binary entries.

### 3.1.4 Graph representation by adjacency matrix

Lastly, we introduce the concept of the adjacency matrix $\mathcal{A}$. This matrix is of dimensions $\mathcal{N} \times \mathcal{N}$ and contains the numeric characterization of the graph $\mathcal{G}$, meaning

that it holds information about whether or not two nodes are connected. For an unweighted graph, this matrix has binary entries. For a weighted graph, the adjacency matrix is often called the weight matrix, $W$, with entries corresponding to the weights of the edges. Hence, if node 1 and node 2 are connected by an *unweighted, undirected* edge, the entry $\mathcal{A}_{(1,2)} = \mathcal{A}_{(2,1)}$ in the *adjacency matrix* would be equal to 1. If the two nodes are not connected, it would be equal to 0. Similarly, if the two nodes were connected by a *weighted, undirected* edge, the entry $W_{(i,j)} = W_{(j,i)}$ would be equal to the weight on this edge. If the two nodes are not connected, the entry would again be zero. The adjacency matrix of an undirected graph is always symmetric.

For a *directed* graph, $W_{(i,j)}$ is not necessarily equal to $W_{(j,i)}$, and even if one of these entries is a non-zero value, the other could very well be zero. This is true for both unweighted and weighted directed graphs. In other words, the edges may be one-way links connecting node A to node B without also connecting node B to node A. This, in turn, means that the weight matrix $W$ of a directed graph is not symmetrical.

## 3.2 Network epidemic models

Network science has historically played a central part in the mathematical modelling of disease transmission. When simulating such transmission, a common tool is what is known network epidemic models. These are commonly known as SI, SIR and SIS models. In these graphs, nodes are assigned a state. These states may be Susceptible/Infected (SI), Susceptible/Infected/Recovered (SIR) or Susceptible/Infected/Susceptible (SIS) depending on the model used (in turn dependent on the nature of the infectious disease). For example, the transmission of a disease that can infect an individual only once would be modeled using an SIR graph, seeing that once individuals have recovered they cannot return to being susceptible to the disease. Similarly, a disease that may infect an individual more than once might be modeled using an SIS graph (Barabási, 2013).

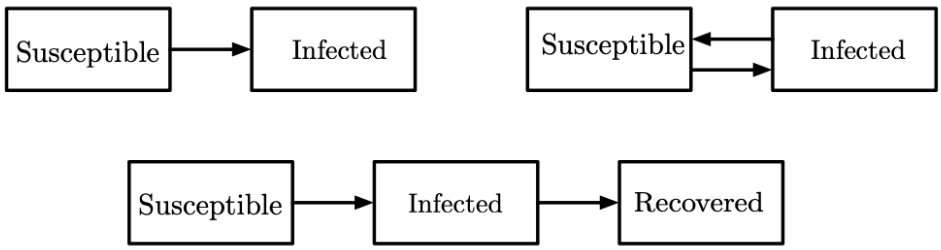


Figure 2: Basic models of epidemic spreading: $SI$, $SIR$ and $SIS$ © (Mei et al., 2017)

Each time step, the nodes change their states according to some pre-determined probability along with the state of the system. In a standard SIR model, $\beta$ is the probability that an infected node will infect a neighbouring node at any given time step. The probability of a susceptible node $i$ being infected at a time step $t$ is thus given by:

$$P(X_j(t+1) = 1 | X_j(t) = 0) = 1 - (1 - \beta)^k \tag{1}$$

where k is the number of infected neighbours of $i$ at time $t$. Additionally, the probability that an infected node will recover during a time step is given by $\delta$;

$$P(X_j(t+1) = 0 | X_j(t) = 1) = \delta \tag{2}$$

The SI/SIS/SIR graphs are known as pairwise interacting network systems and the states assigned to nodes belong to the finite space $\mathcal{A}$, with the dynamical process being Markovian. In the three-state scenario, for a network with $N$ nodes, the SIR model is described by a $3^N$-dimensional Markov process (Nowzari et al., 2016). Similarly, the SIS model presents a two-state scenario where the model is described by a $2^N$-dimensional Markov process.

# 4 Research methodology

## 4.1 Data description

As mentioned above, this study is based on data collected by the Swedish Transport Administration. This data was in part passed to the authors of this thesis from the Department of Civil Engineering at the Faculty of Engineering, Lund University.

The data includes all departures (excluding cancelled departures) from railway stations in Sweden during 2019. This traffic includes different kinds of railway transport including commute trains, regional trains, long-distance trains, airport trains and freight trains. For each departure (corresponding to a single row in the data), the data includes date and scheduled time of departure as well as actual departure time. It also includes station of departure, station of arrival, scheduled arrival time and actual arrival time.

Adding to this data, the Department of Civil Engineering has classified delayed trains and computed magnitude of delay, both at departure and at arrival, for each delayed train. Note that delay in the data set may be negative in the instance of a train departing or arriving before schedule.

Station data including geographical information such as coordinates was also requested and received from the Swedish Transport Authority through the department of Civil Engineering.

## 4.2 Data processing

Primarily, Python has been used to handle the data and building models. The rationale behind this is the vast selection of available libraries for everything from data crunching to network analysis. Many of these also offer tools to increase computational performance, e.g. by doing critical operations in C code, which is helpful in running simulations. Some of the Python libraries that have been used are Pandas (for handling of the data sets), NumPy and SciPy (for mathematical operations, such as those relating to vectors and matrices) and NetworkX (for building and visualizing graphs). Pure visualizations on maps included in the thesis are made in Tableau, with data pre-processed in Python.

The main scripts used for processing the data and running the simulations can be found on GitHub [1]. Note that these scripts do not contain all the code that was run during the project, but all code that was run used some variations of these scripts.

[1] https://github.com/JacobLandelius/Master-Thesis-Raw-Code

# 5 Graph representation of Swedish railway

The graph consists of two main components: *nodes*, which are objects in the network, and *edges*, which describe how these nodes are connected. In this paper, railway *stations* are viewed as nodes and railway *lines* are viewed as edges connecting these nodes. Hence, two nodes are connected by an edge if there exists railway infrastructure allowing for trains to travel between these stations. If one were to look at the Stockholm C-Södertälje-Norrköping part of the SJ express line, for example, the nodes Stockholm C and Södertälje would be connected by an edge whereas Stockholm C and Norrköping would not, as the stations are not directly connected.

When representing the Swedish railway network as a graph, the following notations were used:

- the set of *nodes*, $\mathcal{N}$, represents the set of Swedish railway stations
- the set of *edges*, $\mathcal{E}$, represents the railway connecting stations to one another[2]

In an early simulation set-up (see A), the graph was modelled as undirected and hence the weight matrix $W$ symmetric. In the final simulation presented under section 7 the graph was directed, i.e. $W_{i,j}$ was not necessarily equal to $W_{j,i}$ (see section 6).

## 5.1 Assumptions and approximations

This thesis make several assumptions of and approximations to the Swedish railway network. The most important ones are listed below.

### 5.1.1 Reduced network

In order to simplify computations and achieve a network that more accurately represents the dynamics behind delay propagation, the railway network is in this thesis reduced to a more compact network consisting only of nodes that are either "junctions" or "end stations". Since the model developed in this thesis treats infections on a station level, i.e. treats stations as infected entities that can infect other stations, this reduction of the network seems reasonable.

Junctions are defined as train stations with trains arriving and departing from *more than two* adjacent stations. End stations are defined as stations where trains arrive from and depart to only one other station, meaning that it is the final station on a railway line. In the reduced network, all other stations are excluded. This means that all stations, along with the railway binding them together, in between junctions and end stations will be represented as one single edge (one long railway).

More formally, the reduced network excludes all stations with a degree $deg(j) = 2$, but includes all others.

The reduction is illustrated by the line between the major stations Malmö C - Lund C. This line contains one additional stop (Hjärup station), which is a minor station (see figure 3).

[2] these edges are directed in the simulation set-up presented under section 6, but were undirected in an initial attempt included in section A.

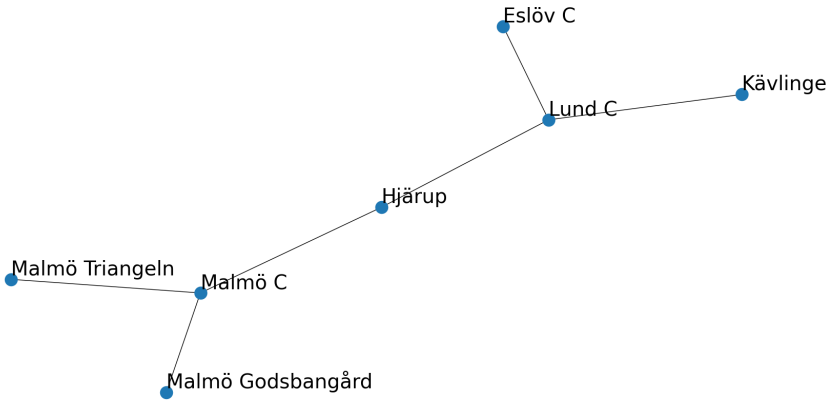


Figure 3: Original graph Malmö C - Lund C line

Since not all trains on the line actually stop at Hjärup, and given that the ones who do only stop there briefly, it seems intuitive that the station should have a much lower impact on the propagation of delay than the much larger hubs of Lund C and Malmö C. Thus, Hjärup station will, in the reduced network, not be treated as a station of its own, but rather be a part of the link between Malmö C and Lund C (see figure 4).

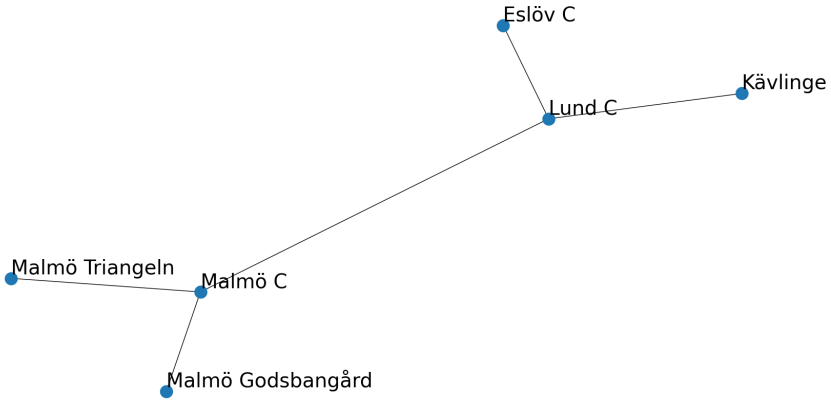


Figure 4: Reduced graph Malmö C - Lund C line

One important thing to note is that, while the reduced network excludes all nodes with $deg(j) = 2$ in the original network, there are a few nodes with $deg(j) = 2$ in the reduced network as well. This phenomenon occurs due to the presence of triangles in the original network, where one of the nodes in the triangle has $deg(j) = 2$ but the others have not. This is illustrated in figure 5 below.

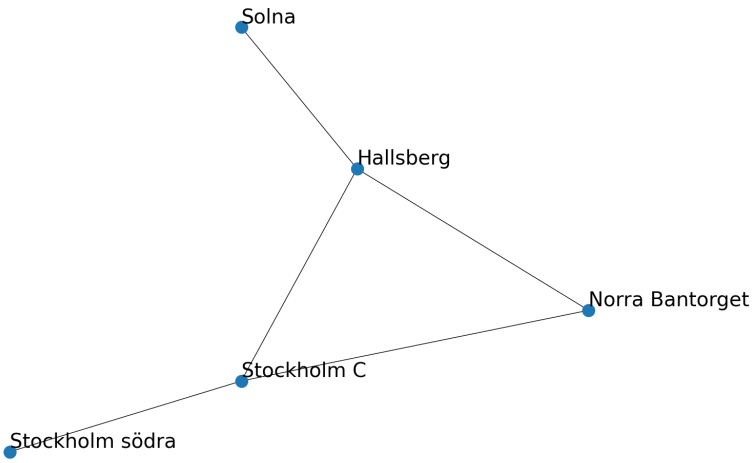


Figure 5: Original graph around Stockholm C

In the above graph, both Stockholm C and Hallsberg have the degree $deg(StockholmC) = deg(Hallsberg) = 3$. However, as Norra Bantorget has $deg(NorraBantorget) = 2$, it will be excluded in the reduced network, as illustrated in figure 6 below.

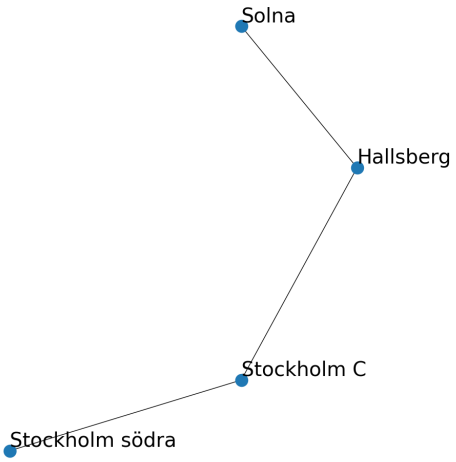


Figure 6: Reduced graph around Stockholm C

In this reduced graph, Stockholm C and Hallsberg have $deg(StockholmC) = deg(Hallsberg) = 2$.

One characteristic of the reduced network is that it increases the physical distance between the nodes in the network. The model simulates delay propagation in discrete time, and larger distances between stations allow for a larger time step to be used in the simulation, which in turn lowers the computational cost of certain algorithms.

A visualization of the reduced network containing only junctions and end stations (green circles) and edges (railway and minor stations) between them (black lines) in

comparison to the network including all stations (green circles) and railway (black lines) is included below[3].

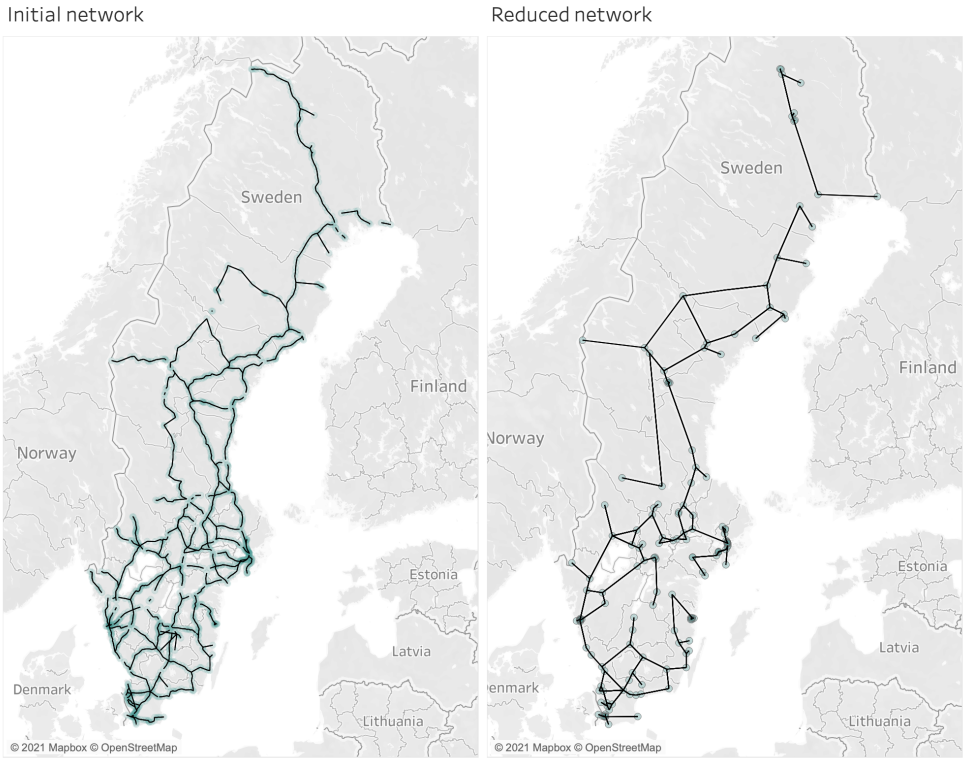


Figure 7: Visualization of reduced network

The non-reduced network, to the left in figure 7 above, contains 1246 nodes. The reduced network, to the right in the figure, contains 183 nodes. In other words, the reduced network is a reduction of 85% in size (measured as number of nodes).

One can also examine the implications on the network diameter when reducing the network to contain only junctions and end stations. The diameter, meaning the greatest distance between any node-pair in the network, for the initial network is 260. In comparison, the diameter of the reduced network is measured to 34. This implies a diameter reduction of 87%. The reader is reminded that the reduced network is a reduction of 85% in size (measured as number of nodes). Hence, the reduction in diameter seems proportional to the reduction in size. This suggests that the reduced graph respects the size-diameter relationship of the non-reduced graph.

There are other network characteristics, aside from size, that can be compared for the two networks. This sections treats two such characteristics for the two networks: *connectivity* and*tendency to cluster*.

If one examines the *connectivity* of both networks, one finds that both networks consists of only one connected component (there are no isolated sub-networks in the

[3]Note that in the data set some stations lacked geographic information, why the visualization is not entirely accurate. For example, the reduced network appears to consist of more than one connected component. It does not.

network) and thus both networks are *strongly connected*. This means that all nodes are connected via edges and hence one can reach any node in the network by starting at a random node and walking along a *path* of edges to the destination node. This is perhaps intuitive, as one can assume that all stations included in the data set (from which a train has departed from or arrived to) are connected via railway to at least one other station.

Although there is only one connected component in the two networks, nodes in the two networks may still have different tendencies to cluster as nodes can be connected to a different number of neighbor nodes within the connected component. The network tendency to cluster can be measured as the average clustering coefficient. For the initial network, the average clustering coefficient is 1.3%. For the reduced network, it is 8.7%. This indicates that the nodes in the reduced graph are more inclined to be part of a cluster, which is reasonable given that the average degree of nodes in the reduced network is higher (2.7) than for the non-reduced network (2.2) due to the fact that the proportion of junctions (high-degree nodes) is higher. A higher tendency to cluster would imply delay-dense areas where congestion is high when simulating delay propagation in the network. However, congested areas in the reduced network represents delay on many more stations, and propagation along many more edges, in the initial areas, why the increased tendency for delay to "cluster" is not considered to impact conclusions that can be drawn from simulation results.

The assumption is that simulation on the reduced network will maintain the delay mechanism of the initial network, which means that one can make computational cost savings whilst arriving at the same conclusions if running simulations on the reduced network rather than the initial one.

### 5.1.2 Restrictions on delay propagation

Another assumption affecting the results of the simulations in this study is the fact that delay can only propagate along railway. This means that a delayed station can only propagate its delay to adjacent stations. It cannot be excluded that a delayed railway station might propagate its delay in other ways as well. For example, there might exist a scenario where a traveller has the option of travelling either by regional or commute train to his or her destination. Perhaps these two trains depart from different stations, but account for a similar total travel time for the passenger. Should one of these stations then be delayed according to this thesis' definitions, it seems reasonable that the traveller might opt for departing from the non-delayed station. Should many travellers use this logic, it is impossible to exclude a scenario where delay does in fact propagate from one railway station to another one close-by, without the two being connected by railway.

The paragraph above describes how the model prevents delay from propagating between stations that are not connected via railway. On the other hand, however, the model also gives some liberty to delay propagation that may not match reality. In the model, a train departing from an infected station might infect a susceptible neighboring station during that specific time step. This is not always logical, seeing that travel time between stations may exceed that of a time step (30 min). For the sake of computations, however, this approximation was deemed necessary and the overall impact on results considered negligible. Also, the number of departures during a time step (30 min) was approximated as the number of hourly departures (for the specific hour) divided by two. These assumptions also have an impact on how the model allows for delay propagation.

### 5.1.3 Independence

Lastly, this thesis frequently makes assumptions regarding independence in terms of probability of delay occurrence. When omitting individual train departures from the analysis, this assumption is mostly in play when not taking into consideration the fact that the probability of spontaneous delay in many cases depends on geography. By this we mean that close-by stations might not be *independently* susceptible to spontaneous delay. Instead a spontaneous delay cause, such as extreme weather or a power failure, will affect the network regionally rather than just locally at one single stations. Hence, treating the probability of spontaneous delay as independent for all stations is a perhaps rather rough approximation that will affect results.

In the scenario where individual train departures affect the value of the $\beta_{i,j}$-parameter, this too is an assumption of independence that might not hold up. For trains departing from the same station at the same time step, these have the same probability of being spontaneously delayed. Again, some reasons for spontaneous delay affect entire stations and the assumption that some trains departing might be spontaneously infected whereas other might not is not entirely intuitive. A cause for spontaneous delay, such as a lasting signalling error for example, might affect an entire station, why *all* trains departing from that station during that time step should be spontaneously delayed (given that the signalling error is not solved within the time step). However, there are also causes for spontaneous delay that affect individual trains (e.g. a personnel issue), why the assumption of independent spontaneous delay is not always unreasonable. It is however important to note that this assumption does not always hold up and that it will affect simulations and may cause simulated results to differ from real ones.

Lastly, this thesis does not treat different types of trains differently in terms of how likely they are to be susceptible to delay. That is, all trains travelling on an edge between an infected and a susceptible node have the same probability of transmitting infection between the two nodes (stations). This is due to infection stemming from infected stations, not from infected (delayed) trains.

# 6 Epidemic model for railway delay propagation

The model of delay propagation described in this thesis is a modified epidemic model that applies the previously introduced SIS model theory with some modifications, such as nodal self-infection, inspired by preliminary research mentioned in section 2.

Some of the research papers previously presented use an SIS model to analyze delay propagation, but for airway traffic rather than railway traffic. Although there are a number of important differences between airway and railway travel, the SIS model can be applied to railway travel if slightly modified compared to the models used for airway travel. This is done by defining train stations as **Susceptible** or **Infected** depending on the level of delay at a station during a given time step.

## 6.1 Definition of states

One problem that arises when using railway stations as nodes is how to define a delayed station, as the notion of a delayed railway station is counter-intuitive. Naturally, delays arise on trains rather than stations. For the purpose of this network analysis, however, a definition of a delayed station (node) is needed. Note that the SIS model used is a $2^N$-dimensional Markov process, with $X_i(t) \in 0, 1 : X_i(t) = 0$ if the node is **susceptible** (not delayed) and $X_i(t) = 1$ if the station is **infected** (delayed).

The use of stations as nodes, and the defined states stations can venture into, is somewhat strengthened by the work of Carl-William Palmqvist with Transport and Roads at Lund University (Palmqvist, 2019). In his PhD thesis, Palmqvist found that "the average delay for a train on a line section between two stations is very close to zero, while at stations almost half of all stops take longer than scheduled, and the delays that do occur also tend to be greater at station stops than on line sections." He goes on to argue that "more than 90% of all delay time occurred at stations" (Palmqvist, 2019). Even though this of course relates to delayed trains, it supports the assumption on delay being able to propagate between adjacent stations as well as this thesis' focus on delay on stations, rather than edges.

In this study, an **infected** railway station is defined as a station for which the percentage of departing trains that are delayed exceeds $Y\%$. Consequently, a station is **susceptible** (not delayed) if the percentage of departing trains that are delayed (by more than five minutes) does not exceed $Y\%$.

The conventional definition of a delayed train is a train which departs more than five minutes late. Using that definition, public data shows that the overall punctuality in the railway network is 91% during 2019 (meaning 91% of trains departed on time) (Trafikverket, 2020). Approximating this threshold to 90%, this value is used as a baseline for the number of trains that can be delayed at departure from a station during a time step before it is considered delayed. In other words, any station in which the percentage of delayed trains in a given time step exceeds $Y = 10\%$ is considered delayed (infected) during that time step. As part of the thesis in general and the sensitivity analysis in particular, the model will be evaluated for other values of this threshold.

## 6.2 Definition of parameters

In the general SIS model, the network dynamics are defined by two homogenous parameters: $\beta$ and $\delta$, where $\beta$ is the parameter controlling the rate of infection and $\delta$

is the parameter controlling the rate of recovery. This paper, inspired by Monechi et al, introduces a third parameter, $\epsilon$, which corresponds to spontaneous (or exogenous) delay.

In the context of a railway network, there are two ways in which delays can occur: endogenous (propagated), and exogenous (spontaneous) as mentioned in the paper by Monechi et al. Endogenous, or propagated, delay, corresponds to the spread of disease in a regular SIS model and is controlled by the $\beta_{i,j}$ parameter. In this paper, $\beta_{i,j}$ is not a global variable, but is an edge-specific, heterogeneous variable. To make this characteristic clear, it is consistently referred to as $\beta_{i,j}$, which refers to the rate of infection on the link between the connected nodes $i$ and $j$.

Spontaneous delay, however, is not controlled by the $\beta_{i,j}$ parameter. In a traditional epidemic model, contagious disease does not spontaneously occur but requires a contaminator. Exogenous, or spontaneous, delay can occur at any time and does not require contact with an infected node to take place. This relates to delay caused by external factors, such as trains breaking down, trees falling over the rails, or bad weather. In order to capture this behavior in the model, the third parameter $\epsilon$ is introduced. Since it is a train-specific variable for a certain node j, it is consistently referred to as $\epsilon_j$, which is the rate of spontaneous infection on a train departing from node (station) $j$.

This means that for this thesis' model, some characteristics of the classical SIS model does not apply. For instance, a classic SIS model would reach an absorbing state: either all nodes would become susceptible over time (and the disease dies out) or all node become infected, which is called the endemic state (i.e. the disease never dies out) (Nowzari et al., 2016). This is not true for a model such as the one in this article which uses a third parameter ($\epsilon_j$) which constantly erupts the convergence towards an absorbing state.

Below follows a more detailed definition of the three parameters and their rationale.

### 6.2.1 Defining recovery rate ($\delta_j$)

Delta is a node-specific value defined as the probability of a node recovering during a time step. In other words, if a station $j$ is infected at time $t$, it will again be susceptible at time $t+1$ with probability $\delta_j$. More formally:

$$P(X_j(t+1) = 0 | X_j(t) = 1) = \delta_j \tag{3}$$

where $X_j(t)$ is the state of node $j$ at time $t$, $S_j(t) = 0$ denotes a susceptible state and $S_j(t) = 1$ denotes an infected state.

### 6.2.2 Defining self-infection rate ($\epsilon_j$)

As stated, $\epsilon$ is a node-specific value relating to the probability of a self-infection (or *spontaneous* infection). In this thesis, individual train departures are taken into account when determining the effective probability of self-infection at a certain time. More precisely, the value $\epsilon_j$ corresponds to the probability that a *single train* leaving from station $j$ at any time $t$ is delayed. It is reminded that the definition of a delayed node is that more than $Y\%$ of departures during a time step are delayed.

The probability of station $j$, susceptible at time $t$, being self-infected at time $t$ is then:

$$P(X_j(t+1) = 1 | X_j(t) = 0)_{self} = P(F_j(t) > L_j(t)) \tag{4}$$

where $F_j(t)$ is the number of trains that are delayed at departure at time $t$. In other words, the probability of a station becoming delayed can be seen as the probability of $L_j(t)$ successes in $d_j(t)$ draws, where the probability of success is $\epsilon_j$. Here, $d_j(t)$ is the number of trains departing from station $j$ at time $t$, and $L_j(t) = Y * d_j(t)$ where $Y$ is the minimum share of trains that are required to depart delayed in order for the station to be considered infected. More formally, the probability of self-infection is drawn from the hypergeometric distribution:

$$P(X_j(t+1) = 1 | X_j(t) = 0)_{self} = 1 - \sum_{k=0}^{L_j(t)} \binom{d_j(t)}{k} \epsilon_j^k (1-\epsilon_j)^{d_j(t)-k} \tag{5}$$

where $X_j(t)$ is the state of node $j$ at time $t$, $X_j(t) = 0$ denotes a susceptible state and $X_j(t) = 1$ denotes an infected state.

### 6.2.3 Defining edge-specific infection rate ($\beta_{i,j}$)

As stated, the edge infection rates are heterogeneous, meaning each link has an individual probability of infection, $\beta_{(i,j)}$.

The probability of a node $j$ being infected at time step $t$ then depends on the value of $\beta_{i,j}$ on its adjacent edges as well as the number of trains travelling on those edges during the specific time step. For the susceptible node $j$ with $N$ infected neighbors belonging to the set of infected nodes $I$, where $D_{j,i}(t)$ is the number of trains travelling on the edge between an infected neighbor $i$ and the susceptible node $j$ at time step $t$, the specific probability of propagated infection for this node was computed as:

$$P(X_j(t+1) = 1 | X_j(t) = 0)_{propagation} = 1 - \prod_{i \in I} (1 - \beta_{i,j})^{d_{i,j}(t)} \tag{6}$$

where $X_j(t)$ is the state of node $j$ at time $t$, $X_j(t) = 0$ denotes a susceptible state and $X_j(t) = 1$ denotes an infected state.

The value $\beta_{i,j}$ can thus be defined as: the probability of a *single* train, leaving from the infected node $i$ to the susceptible node $j$, carrying that infection from node $i$ to node $j$.

The $d_{i,j}(t)$ values were extracted from the data. They contain the average number of trains departing from station $i$ during weekday $d \in (0, 6)$ and hour $h \in (0, 23)$. Since the time step size is 30 minutes, the number of trains travelling between node $i$ and node $j$ at time $t$ is then approximated as the number of trains leaving from node $j$ during the weekday and hour at time $t$, divided by 2, divided by the degree of node $i$:

$$d_{i,j}(t) = (d(i, d(t), h(t))/2)/deg(i) \tag{7}$$

As an example, one can consider the time step where the weekday and hour at time $t$ is Friday ($d = 4$) and 11, respectively. During this specific hour there are, on average, 30 trains leaving from node $i$, which has $deg(i) = 3$ edges. The number of trains leaving from $i$ to *any* of its adjacent stations $j$ during time $t$ is then approximated as:

$$d_{i,j}(t) = (d(i, 4, 11)/2)/3 = 30/2/3 = 5 \tag{8}$$

The rationale behind this definition of the edge-specific probability of infection is presented below.

If a station $j$ is in a state in which it has a high level of delay, there is a certain probability of that state spreading to neighbouring stations. Not only would delayed departures end up delayed at arrival (and subsequently at departure) at neighbouring stations as well, but there are other factors which might add to this propagation. For example, delayed arrivals might cause train to have to wait for passengers who are changing trains on the specific station, etc.

By including the number of departures from infected neighboring stations to the susceptible station when modelling the $\beta_{i,j}$-values, the simulation takes individual train departures, and subsequently traffic levels, into account. Specifically, each train departing from an infected neighbor $i$ to a susceptible node $j$ during a time step $t$ may infect the susceptible node $j$ with a probability ($\beta_{i,j}$). The overall probability of a susceptible node becoming infected is then increased with the number of trains travelling to this node during the specific time step. This is logical, as an infected station from which very few trains leave will likely not have as big of an impact on surrounding stations as one from which a large number of trains is leaving.

Additionally, there is a variance in the level of delay with respect to the weekday and hour of the day. More specifically, delays (as a portion of total departures) are less common during nighttime and during weekends. This can in large be attributed to the lessened traffic during these time periods. Thus, including traffic levels in the model makes it possible to capture some of the seasonality of delays that is present in the real railway network.

## 6.3 Estimation of parameters from data

### 6.3.1 Estimating recovery rate ($\delta_j$)

The value of $\delta_j$ was, like $\epsilon_j$, extracted from the data sets provided.

For the relevant simulation time period, the data was re-sampled into 30-minute intervals (same as the ones used in the simulation). For each of these intervals, the state of each station (susceptible or infected) was computed. If a station got infected, it was given a station-specific timer. Once the station had recovered, this timer indicated how many time steps the station had spent being infected before recovering. This yielded, for every station, a sequence of integers describing the duration of infection. For each station, the average of this sequence was used as an approximation of the number of time steps needed for a station $j$ to recover ($\lambda_j$), and thus the probability of recovery for a node during a single time step, $\delta_j$, was calculated as:

$$\delta_j = 1/\lambda_j \tag{9}$$

### 6.3.2 Estimating self-infection rate ($\epsilon_j$)

The $\epsilon_j$-value is, as stated, the node-specific probability of spontaneous delay (for a train departing from the specific station), and was extracted from data supplied by the Swedish Transport Authority.

The Swedish Transportation Authority attributes codes to each delayed train, encoding the (probable) reason for that train being delayed. One of these codes is *"Disturbed by other train"*, which is the only delay code relating to *propagated* delay rather than exogenous delay.

In calculating $\epsilon_j$, this data was used to calculate the station-specific incidence of delays occurring *without* the aforementioned delay code. In other words, the share of delayed trains at a certain station which *did not* have the delay code *"Disturbed by other train"* was calculated. This was then divided by the total number of trains (delayed and non-delayed) leaving from that station. The resulting value was then interpreted as the probability of a train leaving from station $j$ being delayed, without that delay being caused by other trains, or more relevantly, $\epsilon_j$.

### 6.3.3 Estimating edge-specific infection rate ($\beta_{i,j}$)

Since there is no feasible way to derive the $\beta_{i,j}$ values directly from the data, these were treated as an unknown variable that needed to be solved for using an optimization algorithm, which is described below.

The value of $\beta_{i,j}$, for all edges $i, j \in \mathcal{E}$, started off with a generic start value ($\beta_{i,j} = 0.2$). 20 simulations were then run, using these generic start values along with the static values of $\epsilon_j$ and $\delta_j$. For these simulation, the average *fraction of total delay* (called the *delay score*, or $R_j$) that each station accounted for was calculated:

$$\hat{R_j} = \frac{\sum_{t=0}^{t=T} X_j(t)}{\sum_{t=0}^{t=T} \sum_{k \in N} X_k(t)} \tag{10}$$

where $X_j(t)$ denotes the state of node $j$ at time $t$.

If the delay score of node $j$ was *higher* in the simulations than the real delay score extracted from the data set, $R_j$, ($\hat{R_j} > R_j$) all $\beta_{i,j}$ values on links going in to node $j$ were *reduced* by a certain increment. If the delay score of node $j$ was lower, the in-bound $\beta_{i,j}$ values were increased with the same amount. This was done for all nodes. More formally, the optimization problem can be stated as:

$$\min_{\beta_{i,j}} \quad \frac{\sum_{t=0}^{t=T} X_j(t)}{\sum_{t=0}^{t=T} \sum_{k \in \mathcal{N}} X_k(t)} \tag{11a}$$

$$\text{s.t.} \quad \beta_{ki} \in (0, 1) \tag{11b}$$

If the sign of the difference between $\hat{R_j}$ and $R_j$ was different after iteration $i$ than it was during iteration $i-1$ (meaning the algorithms "overshoots"), the increment was halved. Thus, the algorithm was a node-specific, simple, bisection method.

Some things are worth noting with this approach. Firstly, since the values on the nodes going *in* to node $j$ are optimized separately from the nodes going *out* of it, the resulting matrix is asymmetrical, thus making the graph undirected.

Secondly, the algorithm is very slow, and since no analysis was made beforehand on the convexity of the problem, is not guaranteed to converge. However, the trials conducted showed the algorithm actually did converge, albeit slowly. More details on the result of the optimization is presented under results(7).

Lastly, the reader should know that the way in which $\beta_{i,j}$ was computed was one significant alteration made during the research process. The results of previous attempts can be found in Appendix, together with the reason for computing $\beta_{i,j}$ as described in this section.

### 6.3.4 Summary

To summarize, this paper uses the following parameters:

- $\delta_j$: Corresponds to the probability of recovery of node $j$ during a time step
- $\epsilon_j$: Corresponds to the probability of self-infection for a train departing from node $j$
- $\beta_{i,j}$: Corresponds to the probability of infection, i.e. the rate of infection on the edge between node $i$ and node $j$

The total probability of a node $j$, susceptible at time $t$, becoming infected at time $t+1$ is then:

$$P(X_j(t+1) = 1 | X_j(t) = 0) = 1 - (1 - \epsilon_j) \prod_{i \in I} (1 - \beta_{i,j})^{d_{i,j}(t)} \tag{12}$$

where $X_j(t)$ is the state of node $j$ at time $t$, $X_j(t) = 0$ denotes a susceptible state and $X_j(t) = 1$ denotes an infected state.

# 7 Results

## 7.1 Section outline

As this section is quite lengthy, the reader is provided with a brief overview of how results will presented in order to improve the reading experience. In this section, the reader is first introduced to the evaluation metrics with which the simulation performance is evaluated. The model is evaluated on its ability to reproduce both global as well as local delay behavior, why metrics for both are introduced. Following this, the convergence on the optimization algorithm previously introduced is discussed.

The reader is then introduced to the simulation results. The idea here is for the reader to understand the rationale behind changes in the parameters going into the model, why these sub-section are presented in chronological order. In each section, a table describing the simulation set-up is included, which is there to guide the reader as to which parameters are altered in each simulation.

In the section, the results from running the simulation using the already introduced parameters are introduced first. Then, for reasons given in detail in the text, the spontaneous delay rate parameter $\epsilon_j$ is examined in detail and results from running the simulation using a re-scaled version of this parameter are presented. This is considered the final simulation set-ups, and an analysis of the simulation implications follows next.

Following the presentation of the final simulation and potential implications (which are discussed in detail in section 9), a sensitivity analysis of the model and simulation set-up is included. This sections discussess how the model performs under different circumstances than the ones it is built on.

Last in the section the reader will find a brief paragraph relating to the issue of overfitting as well as a paragraph on how suitable the model is for predicting future delay. All sub-sections may be found in the table of contents on page 4.

## 7.2 Evaluation metrics

In the context of this thesis, it was deemed important that the model could replicate both the global and the local behaviour of the system. *Global* behaviour refers to the overall level of delay in the system, while *local* behaviour refers to station-specific levels of delay.

### 7.2.1 Metrics with regards to global behaviour

Performance with regard to global behaviour was evaluated using two main metrics: the mean portion of stations in a delayed state over the entire simulation period, $\bar{Y}$, compared to the same value from the real data, as well as the Mean Absolute Scaled Error ($MASE_Y$). The Mean Absolute Scaled Error, first proposed by Hyndman and Koehler in 2006, has many advantages to other forecasting measures. In the context of this thesis, there are three main advantages worth noting:

- It is *scale invariant*, meaning it can be meaningful to compare the $MASE_Y$-scores of model prediction on processes with different means.
- Predictable behaviour as $y(t) \to 0$

- Interpretability. A $MASE_Y$ score $< 1$ means that the model performs better in-sample than the naive predictor: y(t) = y(t-m) where m is the *seasonal period*. A score $> 1$ means it performs worse than the naive predictor.

To more formally introduce these measures and their formulas, we first define the portion of stations delayed at a given time step $t$ as:

$$Y(t) = \frac{\sum_{i \in \mathcal{N}} X_i(t)}{|\mathcal{N}|} \tag{13}$$

where $|\mathcal{N}|$ is the number of nodes in the graph.

The two primary global performance metrics are then:

$$\bar{Y} = \frac{\sum_{t=0}^{T} \hat{Y}(t)}{T} \tag{14}$$

and:

$$MASE_Y = \frac{\frac{1}{T} \sum_{t=0}^{T} |Y(t) - \hat{Y}(t)|}{\frac{1}{T-m} \sum_{t=m+1}^{T} |Y(t) - Y(t-m)|} \tag{15}$$

where $\hat{Y}(t)$ denotes the *simulated* value and $Y(t)$ denotes the real value, $T$ is the total number of time periods in the simulation set, and $m$ is the seasonal period of the data.

### 7.2.2 Metrics with regards to local behaviour

Performance with regards to local behaviour was evaluated by comparing the *delay score* obtained from the simulation, $R_j$, described in 6. More precisely, the mean of the absolute difference between the simulated $\hat{R}_j$ values and the real ones, $R_j$, was calculated:

$$MAE_R = \frac{\sum_{j \in \mathcal{N}} |R_j - \hat{R}_j|}{|\mathcal{N}|} \tag{16}$$

In addition, the ranking of the $\hat{R}_j$ values were compared to the real ranking. This was done using two measurements used for comparing rankings: *Kendall's tau* and the *Spearman's rank-order correlation coefficient*. Kendall's tau was calculated using the SciPy implementation (SciPy, 2021a), based on the idea presented by Kendall (Kendall, 1938):

$$\tau = \frac{P - Q}{\sqrt{(P + Q + T)(P + Q + U)}} \tag{17}$$

where P is the number of concordant pairs, Q the number of discordant pairs, T the number of ties only in the simulated ranking, and U the number of ties only in the real ranking. If a tie occurs for the same pair in both the simulated ranking and the real one, it is not added to either T or U.

Spearman's rank-order correlation coefficient, $r_s$, was calculated using the SciPy implementation (SciPy, 2021b), based on literature by Zwillinger and Kokoska (Zwillinger and Kokoska, 1999):

$$r_s = \frac{cov(rg_X, rg_Y)}{\sigma_{rg_X}\sigma_{rg_Y}} \tag{18}$$

where $rg_X$ and $rg_Y$ denotes the raw scores of simulated $\hat{R}_j$ values and real $R_j$ values, respectively, converted to ranks.

For both Kendall's tau and Spearman's r, a value of 1 indicates perfect agreement between the two rankings, a value of 0 indicates no correlation, and a value of -1 indicates perfect disagreement.

## 7.3 Convergence of optimization algorithm

The optimization algorithm used for finding the edge-specific rates of infection (described in 6) performed adequately and, given enough iterations, showed convergence. Since the algorithm optimized every $\beta_{i,j}$, $i, j \in \mathcal{E}$ separately (6), a global metric was used to measure the convergence. This metric was denoted $MAE_R$ and is described in the previous subsection. As a reminder, it was calculated by taking the mean of the absolute value of all of the differences between the simulated delay score, $\hat{R}_j$, and the real delay score, $R_j$, for every node $j \in \mathcal{N}$:

$$MAE_R = \frac{\sum_{j\in\mathcal{N}} |\hat{R}_j - R_j|}{|\mathcal{N}|} \tag{19}$$

where $|\mathcal{N}|$ is the total number of nodes. The convergence of the algorithm with respect to the $MAE_R$-value is shown below:

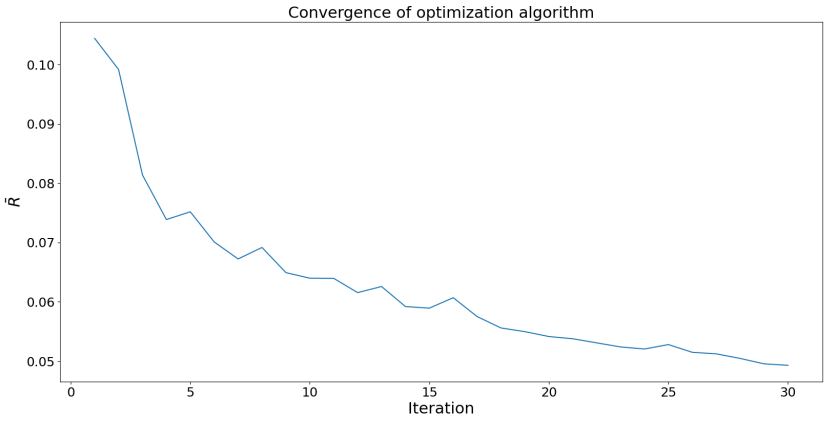


Figure 8: Convergence of optimization algorithm

While the graph indicates that the algorithm was still converging after 30 iterations, various trials concluded that it did so very slowly, and thus running it for more iterations was not considered worth the extra running time. Since, for each iteration,

the simulation is run 20 times (and then averaged), the simulation was run a total of 600 times, which was already time-consuming.

## 7.4 Simulation using optimized $\beta_{i,j}$-values

For the purpose of orientation amongst the different simulations, the simulation set-up is summarized in table 1 below.

| Name | Parameter used |
|---|---|
| Rate of recovery | $\delta_j$ |
| Rate of spontaneous delay | $\epsilon_j$ |
| Rate of propagated delay | $\beta_{i,j}$ |
| Time step | 30 min |
| Time period | March 2019 |

Table 1: Initial simulation set-up

The first optimization attempt yielded the following distribution of $\beta_{i,j}$ values:

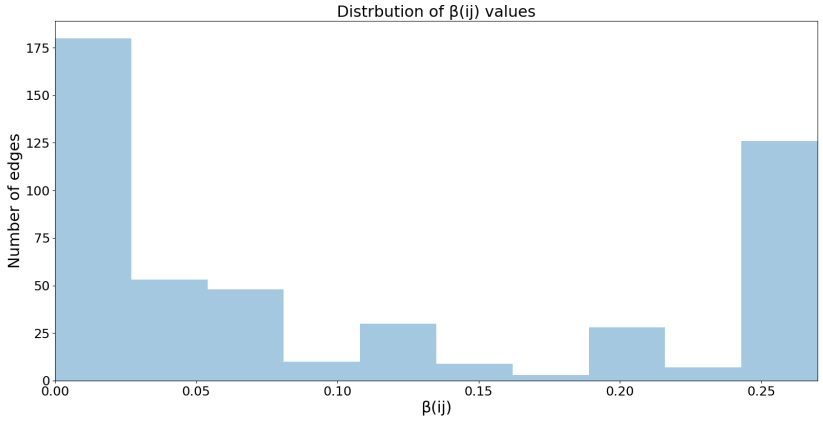


Figure 9: Distribution of $\beta_{i,j}$ values after 30 iterations

| Mean | Std. deviation | Max | Min |
|---|---|---|---|
| 0.108 | 0.109 | 0.270 | 0.00 |

Table 2: Measures of the $\beta_{i,j}$ values

Using these values of $\beta_{i,j}$, 20 simulations were run and then averaged. When visualizing the delay level over time, the following plot was obtained.

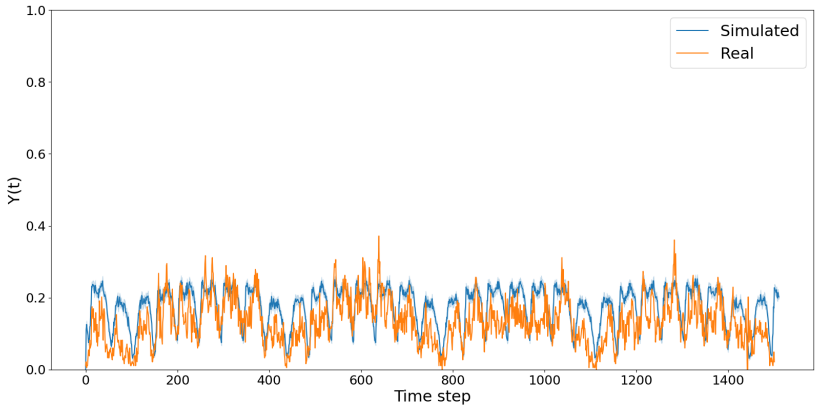


Figure 10: Delay level over time for initial simulation with 20 simulations

In the figure above, $Y(t)$ is, as is described under *Evaluation metrics* (7), the share of stations in a delayed state at time step $t$. Looking at the figure, there is a clear discrepancy between the simulation and the real data, indicating that the simulation does not reproduce global delay behavior with very satisfying results.

If one zooms in on figure 10 above, the 95% confidence interval is visual as the light blue area surrounding the dark blue line (the average of 20 simulations). As is clear from figure 11, the variance in-between simulations is not very large.

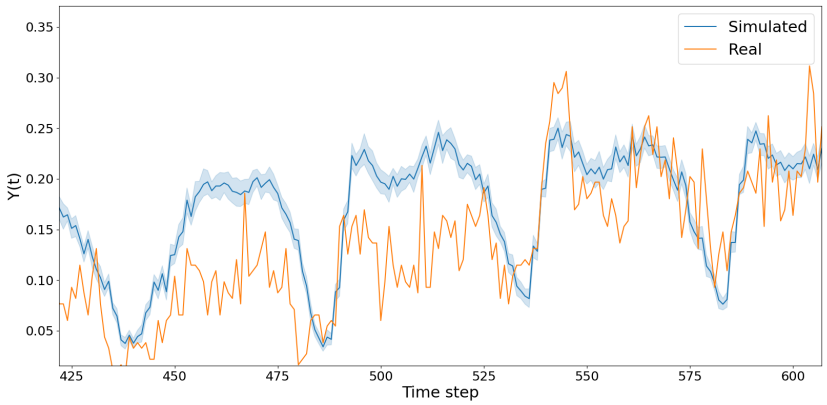


Figure 11: Figure 10 zoomed in, showing confidence interval

Looking at the quantified goodness-of-fit, found in table 27 below, the previously mentioned discrepancy between simulation and real data is obvious, given the too high mean. It is also evident from the $MASE_Y$ that the model performs poorly, as a measure of $3.153 > 1$ means it performs much worse than the naive predictor.

| **Real mean** | **Simulated mean** | $MASE_Y$ |
|---|---|---|
| **0.130** | 0.175 | 1.270 |

Table 3: Global performance metrics

When visualizing local performance (ranking of delay scores), the following graph is obtained.

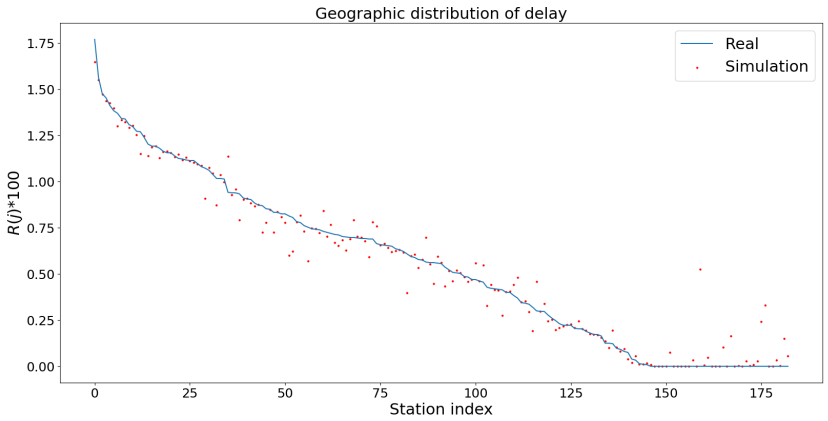


Figure 12: $R$ values, simulated vs real data

In figure 12 above, the blue line illustrates the real $MAE_R$ scores whereas red dots indicate the same values for the simulation, all multiplied by a factor 100. The local performance evaluation metric $MAE_R$ can be interpreted as the average vertical distance from the blue line to the red dots. The graph indicates a very small discrepancy and overall satisfying result. The goodness-of-fit is quantified using previously introduced measures and may be found in table 12 below.

| **Spearman's r** | **Kendall's tau** | $MAE_R$ |
|---|---|---|
| 0.976 | 0.901 | 0.0410 |

Table 4: Local performance metrics

Table 12 shows high values for both Spearman's R Correlation Coefficient as well as Kendall's Tau Distance. Adding to this, the Mean Absolute Error is small, strengthening the conclusion that the simulation of geographical distribution of delay was satisfactory.

When looking at both global and local metrics, it was evident that the simulation, whilst reproducing local delay behavior with satisfying results, was unable to do so for global delay behavior.

## 7.5 Simulation with $\beta_{i,j} = 0$

In order to investigate whether the too high level of delay was a result of an issue with the probabilities of propagating delay, the simulation was re-run with all $\beta_{i,j}$-values

set to 0. The hypothesis was that, this simulation would determine the extent to which the delay propagation probabilities were responsible for the poor performance of the previous model.

The results of this new simulation follow upon table 5 includes information on this simulation set-up.

| **Name** | **Parameter used** |
|---|---|
| Rate of recovery | $\delta_j$ |
| Rate of spontaneous delay | $\epsilon_j$ |
| Rate of propagated delay | $\beta_{i,j} = 0$ |
| Time step | 30 min |
| Time period | March 2019 |

Table 5: Simulation using $\beta_{i,j} = 0$

Figure 13 shows the level of delay over time, when excluding the probability of delay propagating in-between stations.

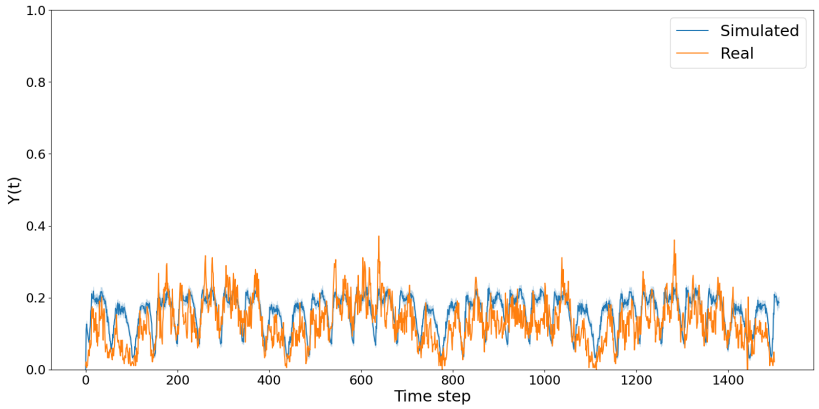


Figure 13: Delay level over time when simulating with $\beta_{i,j} = 0$

As indicated by the figure above, there is still a visible discrepancy between the simulation and the real data, indicating that the global performance is not greatly enhanced by setting $\beta_{i,j}$-values to 0. The goodness-of-fit is quantified in the table below.

| **Real mean** | **Simulated mean** | $MASE_Y$ |
|---|---|---|
| **0.130** | 0.160 | 1.092 |

Table 6: Global performance metrics, $\beta_{i,j} = 0$

Although table 6 indicates that the new simulation performs better, the level of delay is still too high. This instead indicated that the problem might lie with the node-specific self-infection rate, rather than in the edge-specific infection rate. In other

words, the new hypothesis was that $\epsilon_j$ parameters were too large[4].

Under this hypothesis, and under the assumption that this error affects all station equally, i.e. the size of the error being equal for the entire data set, a re-scaling of the $\epsilon_j$-values would produce the correct value and result in an accurate simulation of delay behavior and propagation.

In order to investigate whether a down-scaled value of $\epsilon_j$ might be appropriate, i.e. a smaller ratio of spontaneous delay versus propagated delay, the $\epsilon_j$-parameter was re-scaled globally using different values, denoted $\mu$. In other words, new values of $\epsilon_j$, $\epsilon_j^*$, were constructed by:

$$\epsilon_j^* = \epsilon_j * \mu, \quad \mu \in (0, 1) \tag{20}$$

The optimization algorithm was then run in the same way as before, yielding distinct $\beta_{i,j}$-values for each re-scaling value. The results of each of these re-scaling attempts are found in the next section.

## 7.6 Re-scaled rate of nodal self-infection rate ($\epsilon_j$)

When attempting to re-scale $\epsilon_j$, the simulation set-up was as follows, where $\mu$ takes on different values in the interval $\mu \in (0, 1)$:

| **Name** | **Parameter used** |
|---|---|
| Rate of recovery | $\delta_j$ |
| Rate of spontaneous delay | $\epsilon_j^* = \mu * \epsilon_j$ |
| Rate of propagated delay | $\beta_{i,j}$ |
| Time step | $t = 30$ min |
| Time period | March 2019 |

Table 7: Simulation using $\epsilon_j^* = \mu * \epsilon_j$

Firstly, the global performance of the model for different values of $\mu$ was investigated. In 14 it is shown that the real average level of delay lies around 13%, and that this measure is re-produced for $\mu = 0.65$, i.e a re-scaling of $\epsilon_j$ to 65% to its original value.

[4]Potential reasons for this are discussed in the *Sensitivity analysis* in section 7 and in the *Discussion* (section 9).

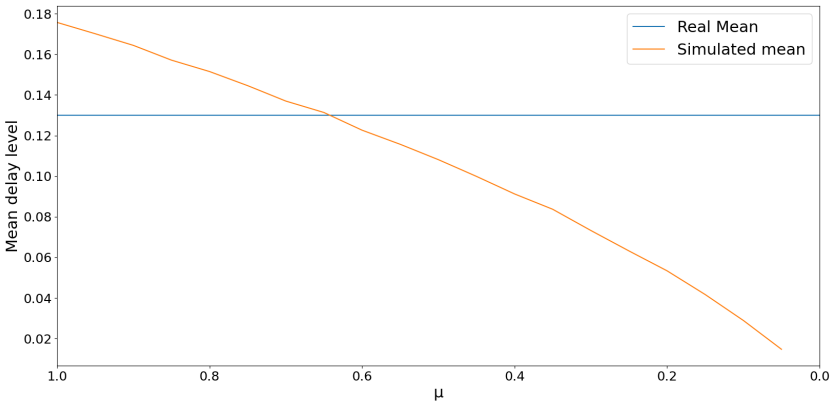


Figure 14: Mean level of delay for different levels of re-scaling of $\epsilon_j$

This result is further visualized in the figures 15 and 16, showing the delay level over time for different values of $\mu$:

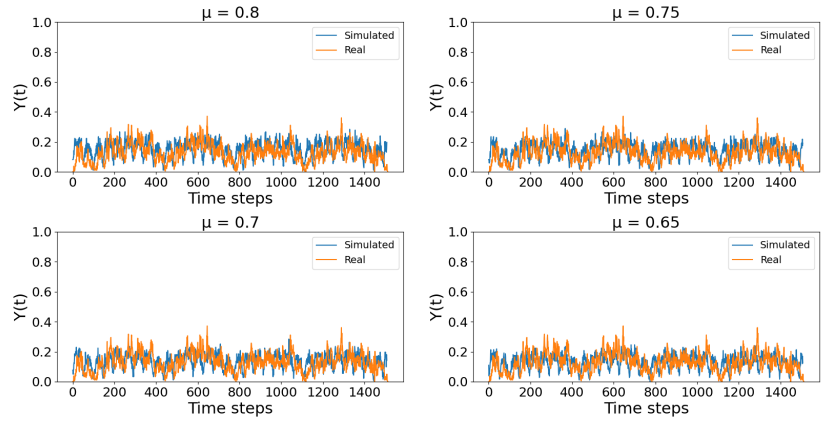


Figure 15: Real vs simulated level of delay over time for different values of $\mu$

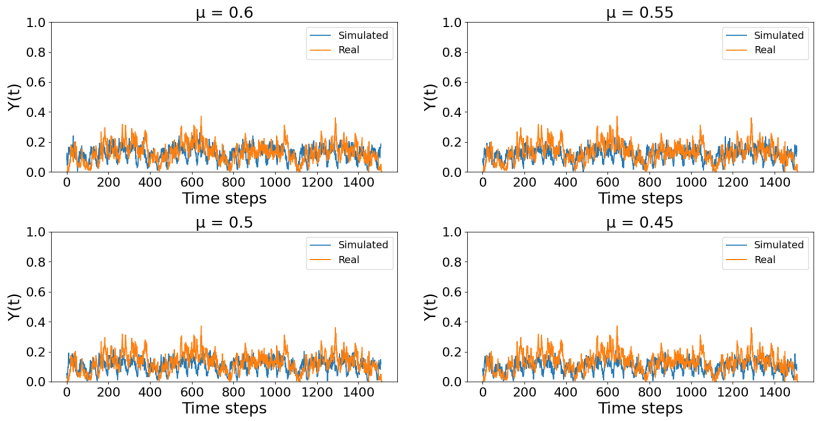


Figure 16: Real vs simulated level of delay over time for different values of $\mu$

When looking at the Mean Absolute Scale Error ($MASE_Y$) for different vaues of $\mu$, the plot indicates that $\mu = 0.6$ yields the best fit. This is slightly lower than the value suggested by figure 14, but very close. See figure 17 below.

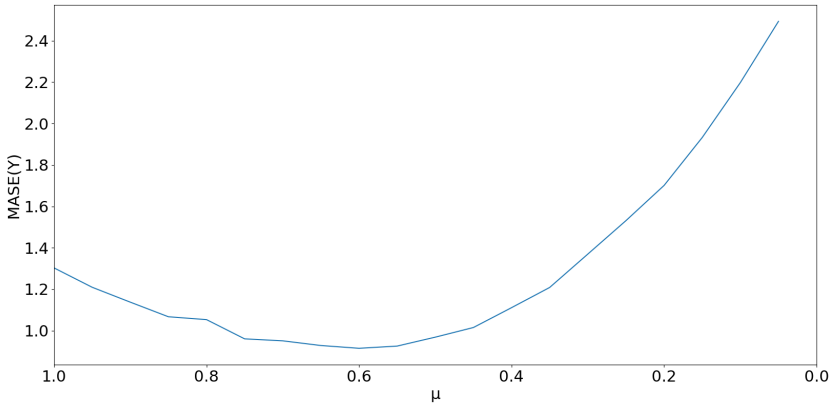


Figure 17: Mean Absolute Scaled Error for different choices of $\mu$

Secondly, the local performance of the model for different values of $\mu$ was investigated. Figure 18 shows the performance with regards to the Spearman's r and Kendall's tau ranking coefficients along with the Mean Absolute Error ($MAE_R$).

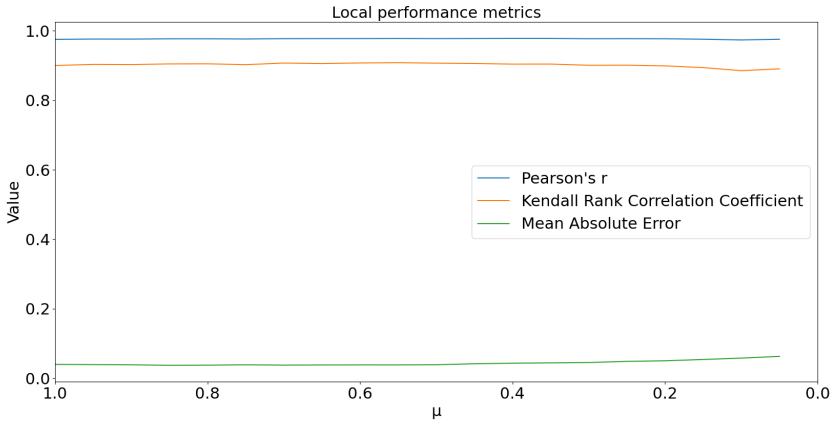


Figure 18: Local performance metrics for different values of $\mu$

While these metrics do not strongly suggest a value of $\mu = 0.6$ to be the optimal, the difference in local performance for the different values of $\mu$ was very small:

| $\mu$ | **Spearman's r** | **Kendall's tau** | $MAE_R$ |
|---|---|---|---|
| $\mu$ **= 0.6** | 0.978 | 0.907 | 0.038 |
| **Best** ($\mu$) | 0.978 (0.40) | 0.908 (0.55) | 0.037 (0.85) |
| **Worst** ($\mu$) | 0.974 (0.10) | 0.885 (0.10) | 0.063 (0.05) |

Table 8: Local performance metrics

As can be seen in the table above, local performance with $\mu = 0.6$ was close to the optimum. With these results, $\mu = 0.6$ was considered the optimal value when considering both global and local behaviour.

The re-scaled $\epsilon$, using the re-scaling parameter $\mu = 0.6$, will be referred to as $\epsilon^*$, where

$$\epsilon_j^* = \epsilon_j * 0.6 \tag{21}$$

## 7.7 Simulation using optimal $\epsilon_j^*$

Following the estimation of an optimal value of $\epsilon_j$, $\epsilon_j^*$, the resulting model was run for 20 simulation, averaged, and evaluated with the following set-up:

| **Name** | **Parameter used** |
|---|---|
| Rate of recovery | $\delta_j$ |
| Rate of spontaneous delay | $\epsilon_j^* = 0.6 * \epsilon_j$ |
| Rate of propagated delay | $\beta_{i,j}$ |
| Time step | $t = 30$ min |
| Time period | March 2019 |

Table 9: Simulation using $\epsilon_j^* = 0.6 * \epsilon_j$

Figure 19 and table 10 show the distribution and measures, respectively, of $\beta_{i,j}$ values yielded from the optimization algorithm.

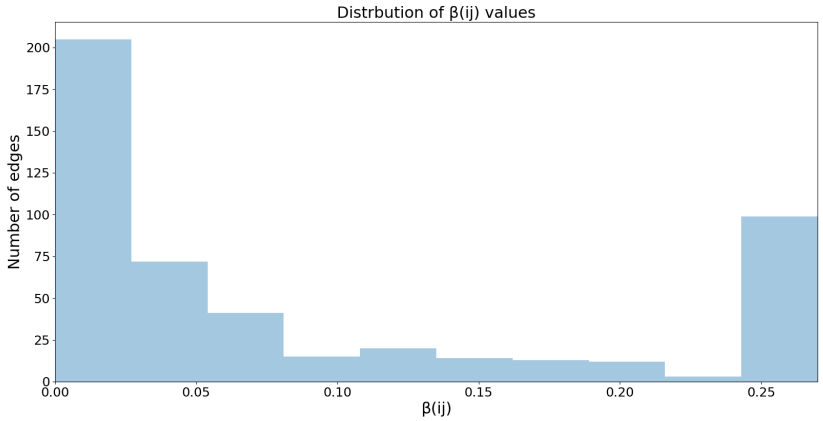


Figure 19: Distribution of $\beta_{i,j}$ values

| **Mean** | **Std. deviation** | **Max** | **Min** |
|---|---|---|---|
| 0.089 | 0.102 | 0.270 | 0.00 |

Table 10: Measures of the $\beta_{i,j}$ values

Both the mean of the $\beta_{i,j}$ values and the standard deviation are lower than in the original model, while both the maximum and minimum $\beta_{i,j}$ values remain the same.

Figure 20 and table 11 show the level of delay over time and the evaluation metrics for the simulations.

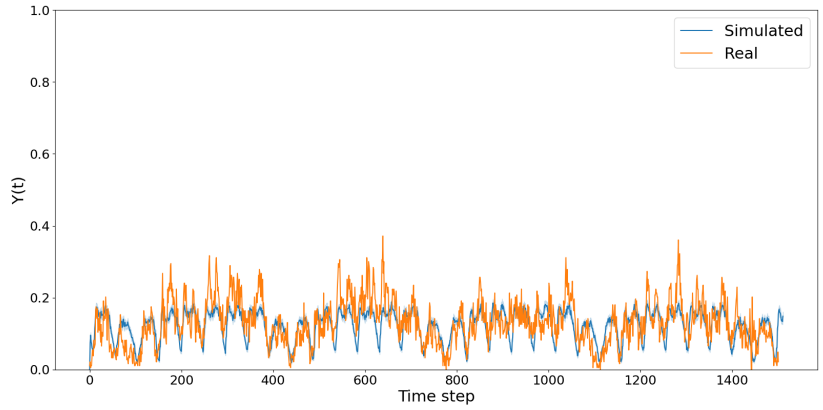


Figure 20: Level of delay over the simulation period

| **Real mean** | **Simulated mean** | $MASE_Y$ |
|---|---|---|
| **0.130** | 0.123 | 0.908 |

Table 11: Global performance metrics when simulating with $\epsilon_j^* = 0.6 * \epsilon_j$

Figure 20 shows that model seems to be good at capturing the seasonality and overall level of delay of the process, while table 11 shows that global behaviour is greatly improved when running the simulation with the re-scaled $\epsilon_j$-values. The fact that the model does not capture some of the high peaks of the real process might be due to the fact that the simulation plot in figure 20 is the average of 20 simulations, while the real line represents just one simulation period. A better fit might have been obtained had the real data been an average over a number of time periods as well.

The local performance of the model is visualized in figure 21:

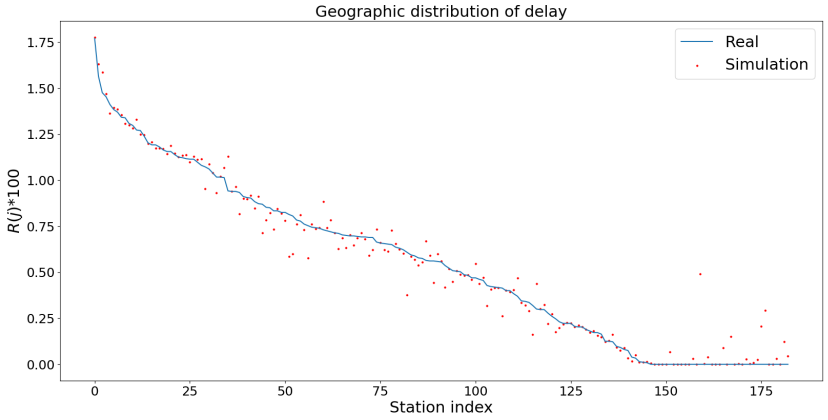


Figure 21: R values, simulated vs real data

The local performance was also improved, and the goodness-of-fit is quantified as before and is found in table 12 below:

| **Spearman's Rank Corr. Coeff.** | **Kendall Tau Rank Distance** | $MAE_R$ |
|---|---|---|
| 0.977 | 0.906 | 0.0397 |

Table 12: Local performance metrics

These results further support the conclusion that the model can reproduce both local and global behaviour of the system when using the re-scaled $\epsilon_j$ value, $\epsilon_j^*$.

The geographic distribution of frequently delayed stations is illustrated in figure 22 below. Note that the darkest circles correspond to the most frequently delayed stations.

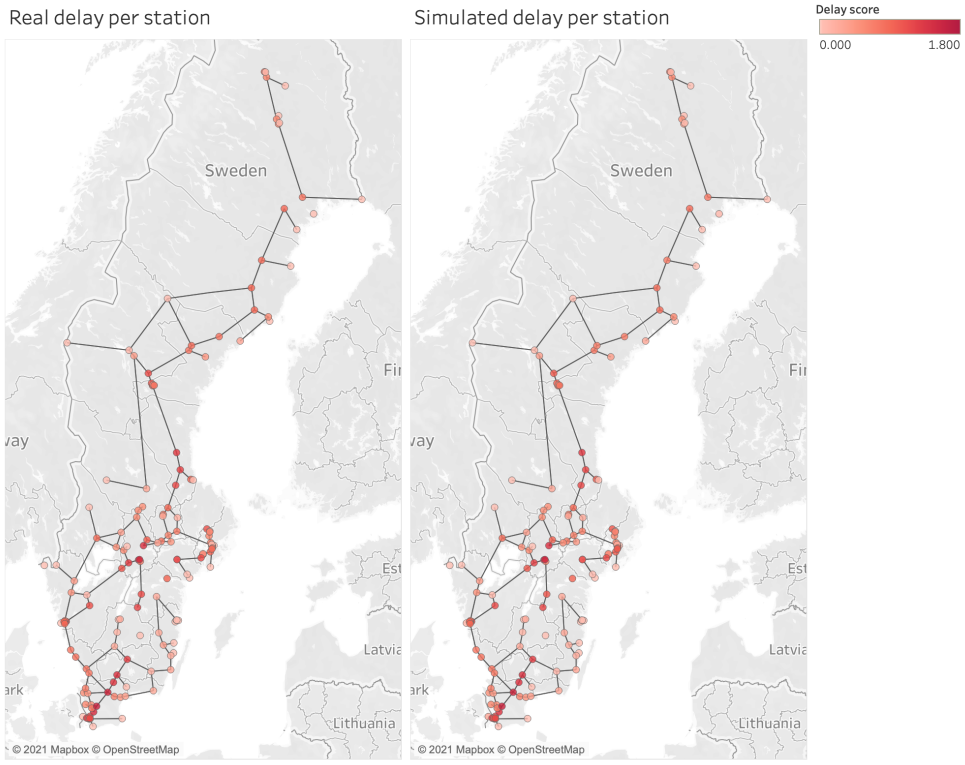


Figure 22: Real and simulated delay scores visualized

Looking at figure 22, the two maps are practically identical to the eye. This indicates that the simulated delay score is close to the real delay score. One can also note that there does not seem to exist an apparent geographical cluster of heavily delayed stations, although one could argue that there exists "micro clusters" in the graph as some stations with higher delay scores are adjacent.

Looking at the delay score of individual stations, one can derive the "top list" for delay in Sweden. The 10 most delayed station, including each station's individual delay score, according to both simulation and reality, are shown below:

| **Real ranking** | **Real score** | **Simulated ranking** | **Simulated score** |
|---|---|---|---|
| Hallsbergs Pbg | 1.77 | Hallsbergs Pbg | 1.77 |
| Eslöv | 1.56 | Eslöv | 1.63 |
| Hallsbergs Rbg | 1.48 | Hallsbergs Rbg | 1.56 |
| Hässleholm C | 1.45 | Hässleholm C | 1.46 |
| Lund C | 1.41 | Tälle | 1.38 |
| Tälle | 1.38 | Hovsta | 1.38 |
| Hovsta | 1.37 | Lund C | 1.35 |
| Älmhult | 1.34 | Älmhult | 1.35 |
| Lernacken | 1.34 | Göteborg Sävenäs | 1.33 |
| Osby | 1.31 | Lernacken | 1.30 |

Table 13: 10 most delayed stations according to delay score

As indicated by the figure 22 above, the two lists are similar. Although ranking for some stations differ, only one station included in the real top list (Osby) is missing from the simulated top list.

It is also shown in the table above that although the ranking of stations differs a bit, the difference in real and simulated delay score for the top 10 station is quite small.

When looking at the optimized $\beta_{i,j}$-values, figure 23 shows these as a heat map, where dark edges symbolize a higher probability of delay propagation (a higher $\beta_{i,j}$-value) on these lines. The value on each edge in the figure below is the optimized $\beta_{i,j}$-value for that edge, **multiplied with the average number of departures per time step**. This is for traffic intensity to have an impact on the visualized delay propagation probability, as infection will spread more easily on edges where a larger number of trains travel each time step. The reader should note that in the figure below, the edges appear to be undirected. In the simulation, these edges would in fact have different propagation probabilities depending on the direction of the edge, why this is an over-simplification and is purely for the sake of making the visualization more easy on the eye.

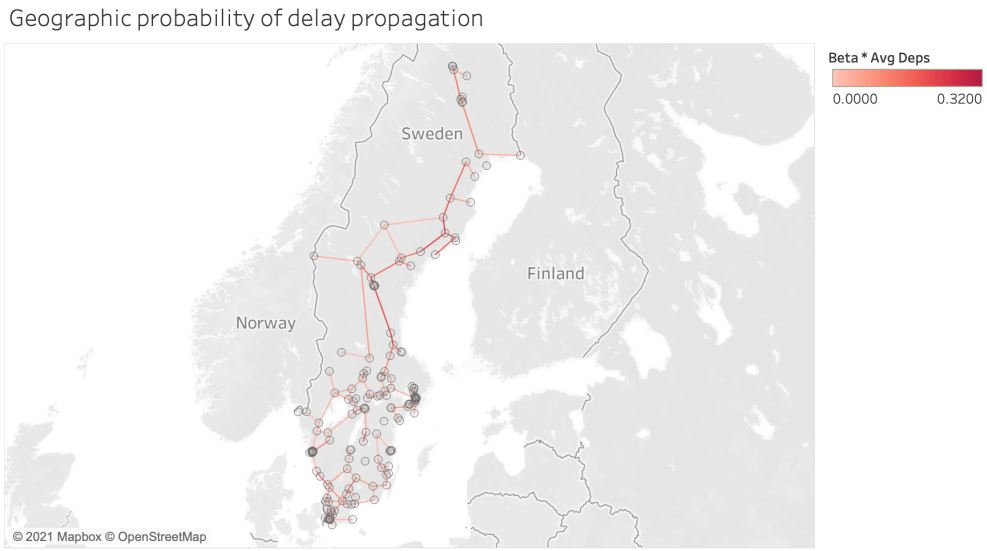


Figure 23: Geographic delay propagation probability

The figure above seems to show a slight inclination for propagated delay probability (times the number of average departures on an edge) to be higher on edges in the northern region. This is not entirely unexpected as freight traffic is more dense in the area, and freight trains are themselves more often delayed than other trains.

## 7.8 Predictive power of model

One potential use of the model is to predict both the global and local behaviour of the system, i.e. predict *where* and *to what extent* delays will occur in the future.

As the parameters of the model are estimated on data from March 2019, it would seem intuitive to use the model to predict delays for March 2020. However, due to the Covid-19 pandemic, March 2020 is not a "normal" month but rather deeply affected by the reduced travelling during that time, and so the outcome of this prediction was not expected to be valuable. Instead, the model was used for predicting delays for the four weeks following the March 2019, i.e. the first four weeks of April 2019.

As is described under *Evaluation Metrics*, the Mean Absolute Scaled Error ($MASE_Y$) compares the performance of the model to that of a naive predictor, and is thus a good measure of the predictive power of the model.

Figure 24 below is the predicted level of delay vs the real one for April 2019.

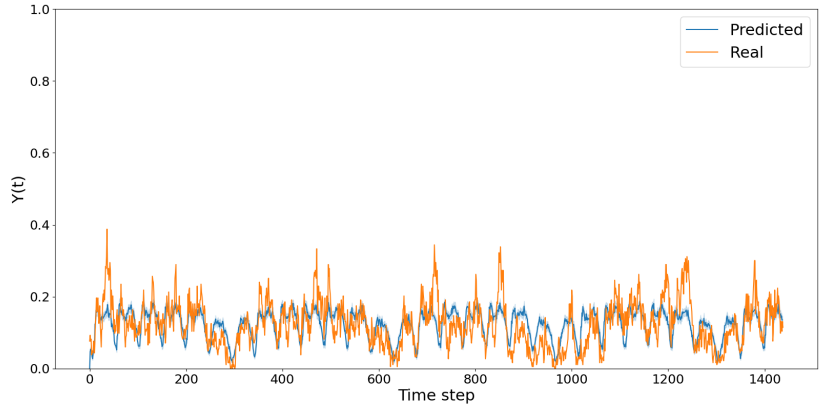


Figure 24: Delay level over time, April 2019

The goodness-of-fit of the simulation visualized in figure 24 is quantified as before by computing the mean delay as well as the Mean Absolute Scaled Error, see table below.

| **Real mean** | **Predicted mean** | $MASE_Y$ |
|---|---|---|
| **0.122** | 0.124 | 0.827 |

Table 14: Global performance metrics

When looking at the visualized performance of the model's ability to predict local delay behavior, the following figure is obtained.

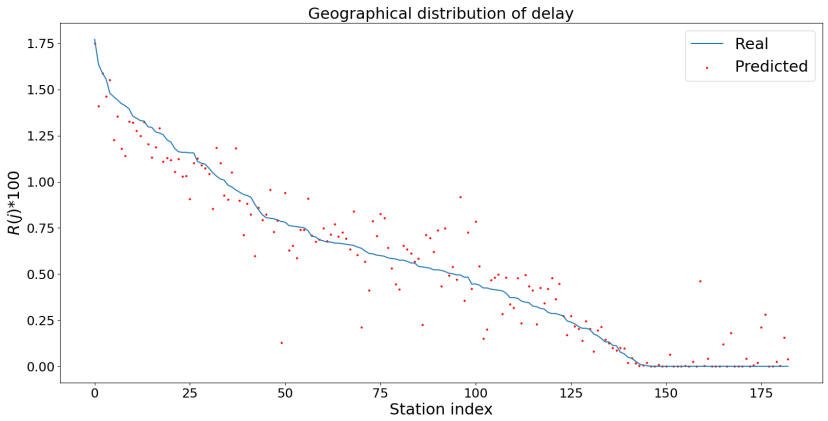


Figure 25: $R_j$ values for the prediction

Quantitatively, the goodness-of-fit can be expressed using the previously introduced Spearman's R Correlation Coefficient and Kendall's Tau Distance, as well as the Mean Absolute Error. The values can be found in table 15.

| Spearman's r | Kendall's tau | MAE ($MAE_R$) |
|---|---|---|
| 0.944 | 0.823 | 0.0860 |

Table 15: Local performance metrics

As seen in the figures and tables above, the model performs adequately on the prediction time period. On a global scale, it still performs better than the naive predictor (as indicated by the fact that $MASE_Y < 1$). The local performance on the prediction time period is not as good as in the training time period (March 2019), but still largely replicates the geographical distribution of delay. Overall, the prediction power of the model was deemed good, and it was hypothesised that it would perform even better had the ideal prediction time period (March 2020) been usable.

## 7.9 Analyzing effect of preventive measures

The following subsection discusses the potential of the model to determine the theoretical effects of targeted preventive measures on critical points in the network.

### 7.9.1 On stations

One potential use of the model is to analyze the effect of preventive measures on stations, i.e. what effect a change in the station-specific parameter $\epsilon_j$ has on those stations and the systems as a whole. One hypothesis is that reducing the spontaneous delay rate, $\epsilon_j$, on the nodes that are delayed most often, i.e. has the highest delay score, would have a large effect on the network as a whole. To investigate this, the top 10 stations in terms of delay score were reduced by 50%. This was done in order for the station to have a *lower percentage of late departures*. The top 10 stations according to the simulation were:

| Station | Simulated score |
|---|---|
| Hallsbergs Pbg | 1.77 |
| Eslöv | 1.63 |
| Hallsbergs Rbg | 1.56 |
| Hässleholm C | 1.46 |
| Tälle | 1.38 |
| Hovsta | 1.38 |
| Lund C | 1.35 |
| Älmhult | 1.35 |
| Göteborg Sävenäs | 1.33 |
| Lernacken | 1.30 |

Table 16: Top 10 delayed stations according to simulation

The simulation set-up, when altering the rate of sponaneous delay on these stations, is summarized below.

| Name | Parameter used |
|---|---|
| Rate of recovery | $\delta_j$ |
| Rate of spontaneous delay | *-50% for for critical stations* |
| Rate of propagated delay | $\beta_{i,j}$ |
| Time step | $t = 30$ min |
| Time period | March 2019 |

Table 17: Simulation with preventive delay measures on critical stations

When running the simulation with these settings, the probability of spontaneous delay for trains departing from critical stations was, as mentioned, reduced with 50%. The overall delay was then decreased for both critical stations as well as for the system at large. This is summarized in the table below. Note that $\epsilon_j$ is only reduced for critical stations.

| | |
|---|---|
| **Reduction of $\epsilon_j$** | 50.0% |
| **Decreased delay on critical stations** | 37.1% |
| **Decrease of overall delay** | 5.01% |
| **Decrease of delay on non-critical stations** | 0.37 % |

Table 18: Results of preventive measures on critical stations

The table above shows that a reduction of 50% of the spontaneous delay rate reduces delay on these stations with approx. 37%, whereas the overall system delay is decreased by approx. 5% and *unaltered* (non-critical) stations by approx. 0.4%. This means that the leverage on the overall system is 0.4%, which seems moderate but may still be significant on a larger scale. The reader should also note that this simulation only alters the spontaneous rate of infection, whereas the rate of propagated delay is left untouched.

In order to analyze how the choice of stations affect the result, the simulation was carried out once again, this time reducing the rate of spontaneous delay for 10 "average" stations, i.e. 10 stations with a delay score close to the average[5]. The simulation set-up is summarized below.

| Name | Parameter used |
|---|---|
| Rate of recovery | $\delta_j$ |
| Rate of spontaneous delay | *-50% for average stations* |
| Rate of propagated delay | $\beta_{i,j}$ |
| Time step | $t = 30$ min |
| Time period | March 2019 |

Table 19: Simulation with preventive delay measures on average stations

The subsequent results are found in the table below. Note that $\epsilon_j$ is only reduced for average stations.

[5]see Apendix for table

| **Reduction of $\epsilon_j$** | 50% |
|---|---|
| **Decreased delay on average stations** | 39.5% |
| **Decrease of overall delay** | 2.06% |
| **Decrease of delay on unaltered stations** | 0.06% |

Table 20: Results of preventive measures on average stations

The results above indicate that the effect on overall delay when decreasing the rate of spontaneous delay on *average* stations is about as large as when decreasing the rate on the stations with the highest delay scores. One reason for this may be that this simulation does not alter the probability of propagating delay on edges, why stations that are often delayed themselves may not always propagate delay onto other stations. This would in turn mean that preventive measures on these stations do not have a great impact on the overall system.

Given the results from this analysis it may be stated that delay score may not be the best indicator of which stations to focus on when allocating preventive measures to reduce overall delay.

### 7.9.2 On lines

The above analysis treats the effects of changing variables tied to stations. Another interesting analysis would be to investigate how line-specific variables, i.e. the spread of delay $\beta_{i,j}$, affects the system.

One hypothesis would be that lines with a high $\beta_{i,j}$ value and high traffic levels could be important to the delay of the network as a whole. To investigate this, the 10 lines with the highest delay propagation probabilities were extracted from the model. Again, these were computed as the lines with the highest values looking at the edge-specific $\beta_{i,j}$-values multiplied by the average number of trains travelling on the line. The relevant lines are presented in the table below.

| **Line** | **$\beta_{i,j}$ * average departures** |
|---|---|
| Ulriksdal - Helenelund | 3.33 |
| Älvsjö - Flemingsberg | 2.50 |
| Älvsjö - Årstaberg | 2.50 |
| Flemingsberg - Södertälje Syd Övre | 1.55 |
| Flemingsberg - Södertälje Hamn | 1.55 |
| Malmö C - Hyllie | 1.50 |
| Tomteboda Övre - Stockholm Södra | 1.37 |
| Hyllie - Malmö C | 1.27 |
| Hyllie - Lernacken | 1.27 |
| Hyllie - Svågertorp | 1.27 |

Table 21: Top 10 congested lines with high risk of delay propagation

In order to analyze the effect of preventive measures, the $\beta_{i,j}$-values for these 10 lines were reduced by 50%. The simulation set-up is found below.

| Name | Parameter used |
|---|---|
| Rate of recovery | $\delta_j$ |
| Rate of spontaneous delay | 0.6*$\epsilon_j$ |
| Rate of propagated delay | -50% on critical lines |
| Time step | $t = 30$ min |
| Time period | March 2019 |

Table 22: Simulation with preventive delay measures on critical lines

When running the simulation with this new set of $\beta_{i,j}$-values, the overall delay in the system was decreased by 53%. This is summarized in the table below. Note that by adjacent stations, what is referred to is the destination station of the altered edge.

| | |
|---|---|
| **Reduction of critical $\beta_{i,j}$** | 50% |
| **Decrease on adjacent stations** | 1.04% |
| **Decrease on non-adjacent stations** | -0.054% |
| **Decrease of overall delay** | 0.038% |

Table 23: Results of preventive measures on critical lines

The results above indicate that by reducing the $\beta_{i,j}$-values on the 10 edges with the highest probability of delay propagation multiplied with average number of departures, the overall delay is diminished by 0.038%. This change does not indicate a tremendous leverage on the entire system. However, it is possible that the amount of traffic, i.e. the average number of departures, is not an interesting factor when identifying critical lines.

In order to measure the effect of targeting high-risk lines, the same simulation was run when the probability of delay propagation on 10 lines with $\beta_{i,j}$-values close to the average (0.0717)[6] was decreased by the same percentage. The simulation set-up is summarized as:

| Name | Parameter used |
|---|---|
| Rate of recovery | $\delta_j$ |
| Rate of spontaneous delay | 0.6*$\epsilon_j$ |
| Rate of propagated delay | -50% on average lines |
| Time step | $t = 30$ min |
| Time period | March 2019 |

Table 24: Simulation with preventive delay measures on average lines

The table below indicates the system results for this simulation. Note that the change in delay on non-adjacent stations is positive, meaning that delay increases.

[6] see Apendix for table

| **Reduction of average $\beta_{i,j}$** | 50% |
|---|---|
| **Decrease on adjacent stations** | 2.29% |
| **Decrease on non-adjacent stations** | -0.144% |
| **Decrease of overall delay** | -0.01% |

Table 25: Results of preventive measures on average lines

The results above indicate that preventive measures on edges with high values of the product of delay propagation probability and average number of departures are much more effective than measures on edges with average values for the same product.

However, as with preventive measures on stations in the previous sub-section, the effect on overall delay does not seem very large even when targeting *critical lines*. Three things are however worth noting with this simulation and its results: firstly, there are 494 edges in the reduced system why this simulation only adjusts values on roughly 2% of edges. Secondly, this simulation only adjusts values on edges with a high value of the *product* of delay propagation probability and average number of departures. This means that this simulation may alter the value on an edge with a very low $\beta_{i,j}$ but a very high number of departures. On the other hand, however, should departures not be taken into account the simulation would run the risk of adjusting values on edges with barely any traffic, which would indicate an even smaller leverage on the overall system. Thirdly, the effect on adjacent stations is a lot higher. This could indicate that if more strategic edges were targeted, the effect on adjacent stations could be vital in lowering the overall delay. In summary, further analysis is needed to identify which edges are the most important when allocating preventive measures.

# 8 Sensitivity analysis

The following section discusses the simulation performance when changing certain variables of the model. Such an analysis not only provides a good basis for gauging the robustness of the model with regards to changes in the underlying definitions and assumptions, but also serves to give a better understanding of the relationship between the input and output variables of the model.

In this section, the model performance is investigated when changing variables such as the definition of an infected state, the size of the time step and the time period on which the simulation takes place.

## 8.1 Reducing number of variables

To investigate how the different variables used in the model affect the results, the model was run several times, changing one variable at a time. More precisely, the model was run replacing the heterogeneous $\beta_{i,j}$, $\epsilon_j$, and $\delta_j$ variables with a homogeneous variable, $\bar{\beta}$, $\bar{\epsilon}$, and $\bar{\delta}$, respectively (the average of the heterogeneous variable). The simulation set-ups were then:

| **Average $\beta_{i,j}$ model Parameters** | **Average $\epsilon_j$ model Parameters** | **Average $\delta_j$ model Parameters** |
|---|---|---|
| $\delta_j$ | $\delta_j$ | $\delta_j = \bar{\delta}$ |
| $\epsilon_j^* = 0.6 * \epsilon_j$ | $\epsilon_j^* = \bar{\epsilon^*}$ | $\epsilon_j^* = 0.6 * \epsilon_j$ |
| $\bar{\beta}$ | $\beta_{i,j}$ | $\beta_{i,j}$ |
| $t = 30$ min | $t = 30$ min | $t = 30$ min |
| March 2019 | March 2019 | March 2019 |

Table 26: Simulation set-ups

Figure 26 visualizes delay level over time for these simulations.

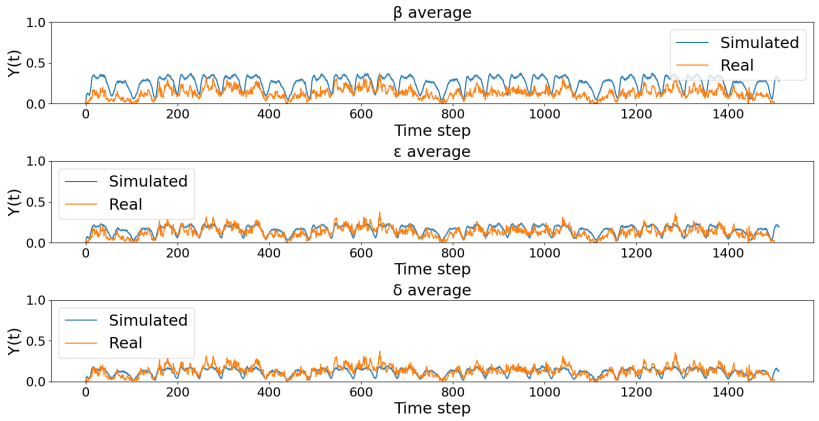


Figure 26: Delay level over time, different models

| **Model** | **Real mean** | **Simulated mean** | $MASE_Y$ |
|---|---|---|---|
| $\beta$ | **0.130** | 0.257 | 2.732 |
| $\epsilon$ | **0.130** | 0.158 | 1.011 |
| $\delta$ | **0.130** | 0.121 | 0.856 |

Table 27: Global performance metrics

From the above plot and table, it can be concluded that the heterogeneity of $\beta_{i,j}$ seems to have the strongest impact on the global performance of the model, whilst that of $\delta_j$ seems to have the lowest.

Figure 27 and table 28 shows the local performance, i.e. the geographic distribution of delay for the three modified models.

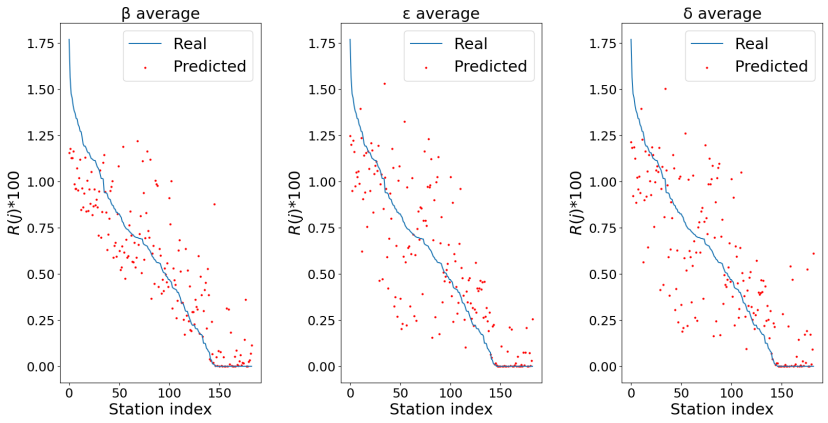


Figure 27: Geographic distribution of delay

| **Model** | **Spearman's r** | **Kendall's tau** | $MAE_R$ |
|---|---|---|---|
| $\beta$ | 0.875 | 0.705 | 0.164 |
| $\epsilon$ | 0.831 | 0.649 | 0.188 |
| $\delta$ | 0.783 | 0.598 | 0.208 |

Table 28: Local performance metrics

While the heterogeneity of all parameters is important in order for the model to reproduce local behaviour, the $\delta_j$ parameter seems to be the most influential, which is unexpected. This may indicate that $\delta_j$ should be modified rather than $\epsilon_j$ in the initial attempt.

## 8.2 Re-defining states

The first part of the sensitivity analysis consists of determining how the simulation performed when re-defining the definition of an infected state. As a reminder, the original definition of an infected state was a station where more than 10% of departures were delayed (departed more than five minutes after their scheduled time) during a time step. In the sensitivity analysis, this threshold was increased from 10% to 20%, 50%, 80%, and 100%. Since the value of the node-specific recovery rate, $\delta_j$, is extracted from the data using the specific definition of an infected state, new values for $\delta_j$ were calculated for each threshold. The values of $\epsilon_j$ are not extracted using the definition threshold, and therefore remained unchanged. For each threshold, the values of $\beta_{i,j}$ were determined using the same optimization algorithm as in previous sections.

In figures 28 and 31, along with tables 29 and 30 below, one can see how the performance of the model changes with the threshold.

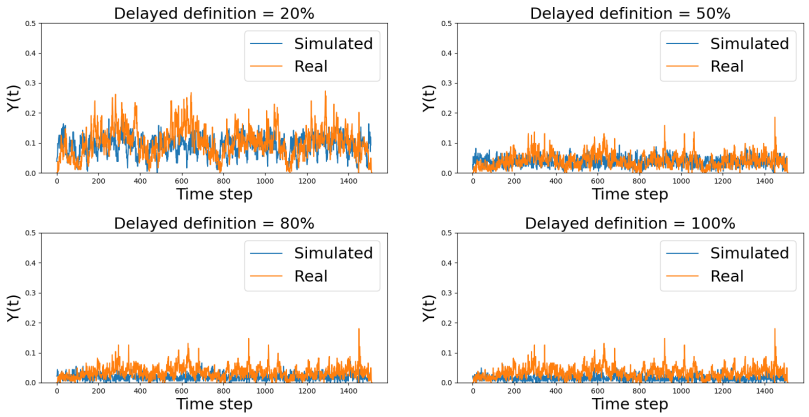


Figure 28: Level of delay over time for different thresholds of infection

| Threshold | Real mean | Simulated mean | $MASE_Y$ |
|---|---|---|---|
| **20%** | **0.103** | 0.090 | 0.975 |
| **50%** | **0.041** | 0.037 | 0.983 |
| **80%** | **0.034** | 0.019 | 1.126 |
| **100%** | **0.034** | 0.015 | 1.233 |

Table 29: Global performance metrics with different thresholds

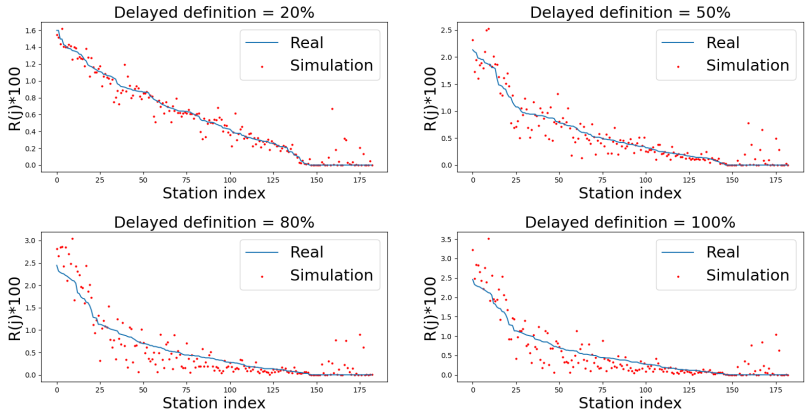


Figure 29: Geographic distribution of delay for different thresholds of infection

| Threshold | Spearman's r | Kendall's tau | $MAE_R$ |
|---|---|---|---|
| **20%** | 0.967 | 0.879 | 0.057 |
| **50%** | 0.906 | 0.783 | 0.125 |
| **80%** | 0.835 | 0.702 | 0.188 |
| **100%** | 0.832 | 0.699 | 0.194 |

Table 30: Local performance metrics with different thresholds

As the figures and tables above show, the performance of the model is markedly worse when using a higher threshold. Although the model still performs fairly well both globally and locally, the potential reasons for the worsened performance with a higher threshold carry important implications.

Firstly, it is noted that the re-produced delay level is *too low*. Since the $\delta_j$ values are extracted from the data using the individual thresholds, and the $\beta_{i,j}$ values, as seen previously, doesn't have as much impact on the overall delay level as $\epsilon_j$, this might indicate that the issue is with the $\epsilon_j$ values. It is reminded that the *effective probability* of a node getting self-infected (spontaneously delayed) at a given time is a function of the $\epsilon_j$ and the departures from station $j$ at time $t$, $d_j(t)$.

One possible explanation is thus that the assumption that the $\epsilon_j$ value is independent for all trains leaving from station $j$. This assumption is not reasonable under all circumstances, which is discussed in more detail in section 9.

## 8.3 Model performance on different season

Another important part of evaluating the model robustness was to run the simulation on a data set from a different season (in this case September 2019) and evaluate the performance. The only information collected from the September data is the train schedule, i.e. the average number of departures between stations for each time step, the rest of the variables are estimated from the March 2019 data. Thus, the model set-up can be summarized as:

he simulation can be summarized as:

| Name | Parameter used |
|---|---|
| Rate of recovery | $\delta_j$ |
| Rate of spontaneous delay | $\epsilon_j^* = 0.6 * \epsilon_j$ |
| Rate of propagated delay | $\beta_{i,j}$ |
| Time step | $t = 30$ min |
| Time period | September 2019 |

Table 31: Simulation using September departures

Figure 30 and table 32 below show global performance of the model:

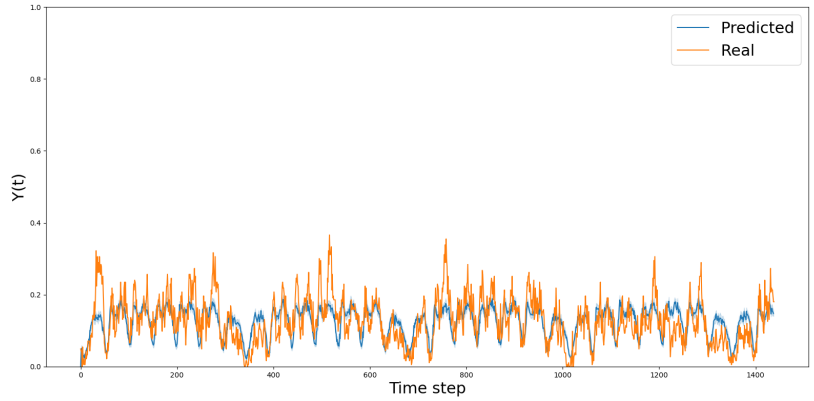


Figure 30: Delay level over time, simulation of September 2019

| **Real mean** | **Predicted mean** | $MASE_Y$ |
|---|---|---|
| **0.128** | 0.126 | 0.946 |

Table 32: Global performance metrics

The above results show that the model captures the mean of the process well and still outperforms the naive predictor, albeit with less than when predicting April 2019 (which is to be expected).

Figure 31 and table 33 below show the local performance of the model:

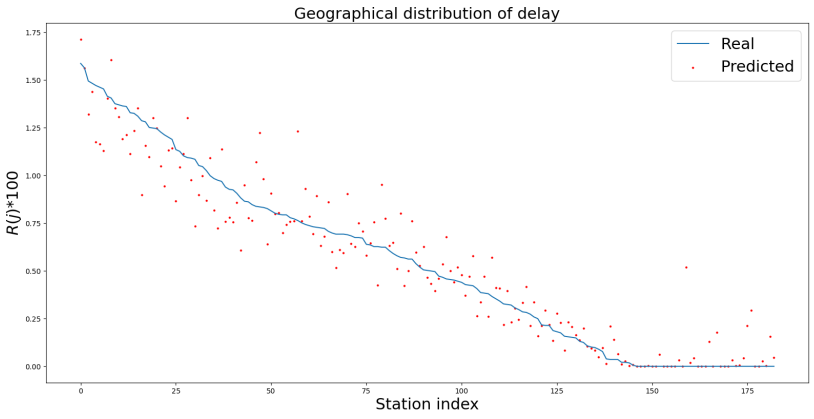


Figure 31: Geographic distribution of delay, simulation of September 2019

| **Spearman's r** | **Kendall's tau** | $MAE_R$ |
|---|---|---|
| 0.956 | 0.837 | 0.0922 |

Table 33: Local performance metrics

Once again, the performance is not as good as on April 2019, which is to be expected. However, as with the global performance, the local performance of the model is still robust on the September 2019 data. This indicates that the model captures some underlying characteristics of the system well.

## 8.4 Changing size of time step

Another important aspect was to determine how sensitive the model was to the size in time step. The reader is reminded that the original time step was 30 minutes. In this section, the simulation was run with a time step of 15 and 60 minutes respectively, i.e. smaller and larger than the original time step. As all variables needed to be extracted anew from the data in order to run the model with a different time step, this analysis focuses on the viability on the underlying structure of the model rather than the performance of the variables previously extracted from the data.

### 8.4.1 Smaller time step

For the smaller, 15-minute time step, the model was fitted for its $\beta_{i,j}$ values and the best choices of the re-scaling parameter of $\mu$ was determined. While the optimal value of $\mu$ with regards to global behaviour was equal to the one in the original model ($\mu = 0.6$)[7], its discrepancy with the optimal $\mu$ with regards to local behaviour was greater than before (optimal $MAE_R$ was obtained at $\mu = 1.0$)[8]. The reason for this discrepancy is not fully understood, and is discussed in more detail under section 9.

To visualize the model performance with the 15 min time step size, simulations were run with $\mu = 0.6$, yielding the following simulation set-up:

| Name | Parameter used |
|---|---|
| Rate of recovery | $\delta_j^{15}$ |
| Rate of spontaneous delay | $\epsilon_j^* = 0.6 * \epsilon_j$ |
| Rate of propagated delay | $\beta_{i,j}$ |
| Time step | $t = 15$ min |
| Time period | March 2019 |

Table 34: Simulation using time step of 15 min

Figures 32 and 33, as well as tables 35 and 36 below the global and local performance of the model.

[7] see Appendix for figure
[8] see Appendix for figure

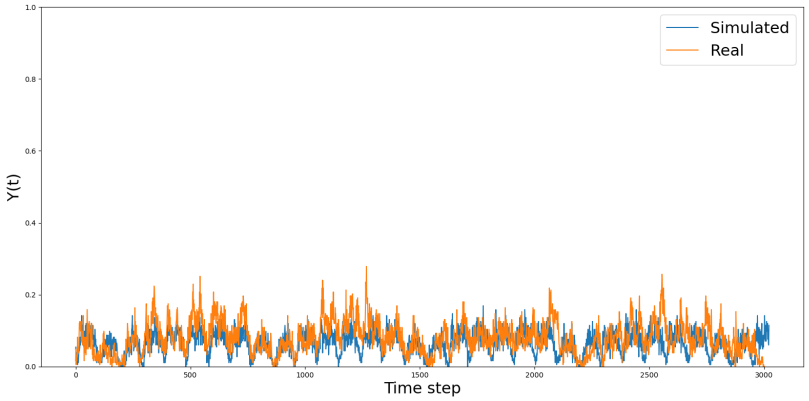


Figure 32: Delay level over time with time step = 15 min

| **Real mean** | **Predicted mean** | $MASE_Y$ |
|---|---|---|
| **0.08** | 0.065 | 1.0287 |

Table 35: Global performance metrics

Figure 32 and table 35 above indicate that the reproduction of global delay behavior is not as accurate when simulating on a smaller time step. The $MASE_Y$ is increased from 0.886 to 1.008 compared to the final simulation on a 30 min time step. The predicted mean is also slightly off.

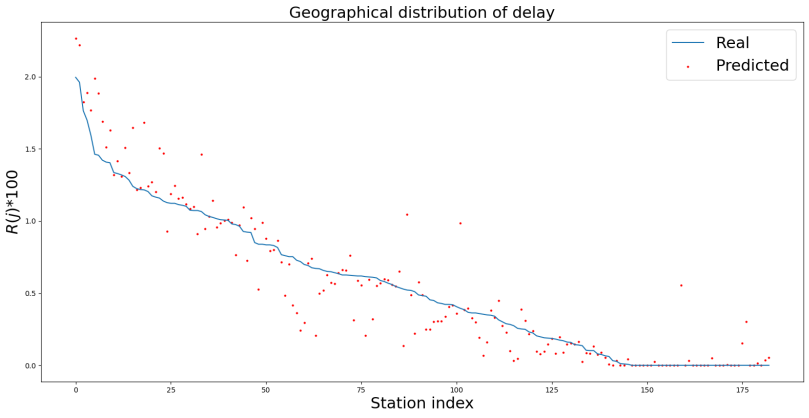


Figure 33: Geographic distribution of delay with time step = 15 min

| **Spearman's r** | **Kendall's tau** | $MAE_R$ |
|---|---|---|
| 0.947 | 0.832 | 0.104 |

Table 36: Local performance metrics

The reproduction of local delay behavior is not as accurate as the original model either, according to figure 33 and table 36.

### 8.4.2 Larger time step

For the 60-minute time step, the model was again fitted for its $\beta_{i,j}$ values and the best choices of the re-scaling parameter of $\mu$ was determined. For this time step, $\mu = 0.6$ again seems a reasonable choice, in agreement with the original model[9]. The model thus was run using the following set-up:

| Name | Parameter used |
|---|---|
| Rate of recovery | $\delta_j^{60}$ |
| Rate of spontaneous delay | $\epsilon_j^* = 0.6 * \epsilon_j$ |
| Rate of propagated delay | $\beta_{i,j}$ |
| Time step | $t = 60$ min |
| Time period | March 2019 |

Table 37: Simulation using time step of 60 min

Figures 34 and 35, as well as tables 38 and 39 below the global and local performance of the model.

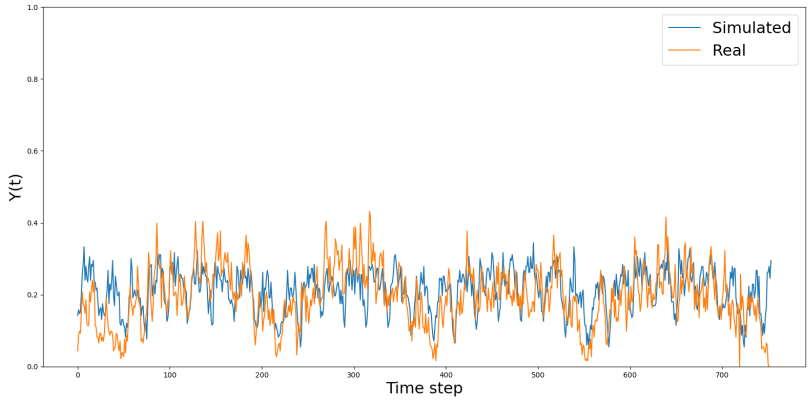


Figure 34: Delay level over time, time step = 60 min

| Real mean | Predicted mean | $MASE_Y$ |
|---|---|---|
| **0.193** | 0.212 | 0.852 |

Table 38: Global performance metrics

Figure 34 and table 38 indicate that the simulation of global delay behavior with the larger time step is not as good as with the original simulation, seeing that the predicted mean is somewhat higher than the real mean. However, this is not a great

[9] see Appendix for figure

difference and the overall look of the plot indicates that the simulation with the larger time step captures seasonality in the data.

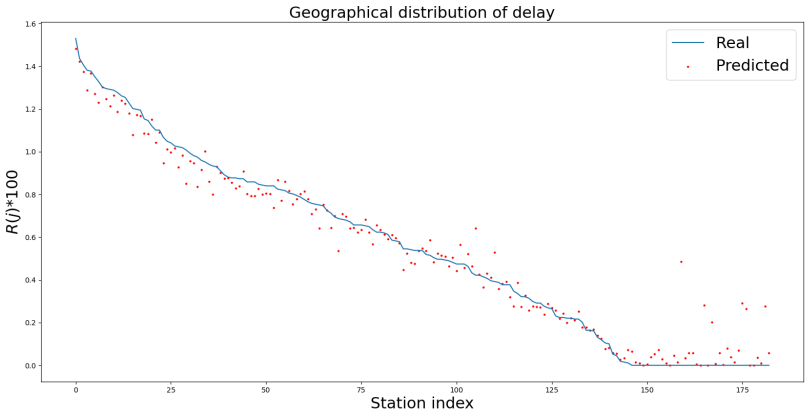


Figure 35: Geographic distribution of delay, time step = 60 min

| **Spearman's r** | **Kendall's tau** | $MAE_R$ |
|---|---|---|
| 0.977 | 0.905 | 0.0462 |

Table 39: Local performance metrics

According to the figure 35 and table 39, the model still performs very well in reproducing local delay behavior with just a slightly larger $MAE_R$. The visible discrepancy in the figure above is much less than for the simulation with the smaller time step. Overall, the simulation with a larger time step seems to perform well.

When looking at the simulation with both a smaller and larger time step, it seems the model performance remains strong when increasing the time step size from 30 to 60 minutes, but are negatively impacted by a *decrease* in time step size from 30 to 15 minutes. One hypothesis is that this would depend on the average travel time in-between stations in the reduced network (approx. 30 min) and that 15 minutes is a too short amount of time for the simulation to perform well as distances are greater in the reduced network compared to the initial one. This needs to be investigated further in order to determine the true cause for the difference in performance on the different time steps.

# 9 Discussion

## 9.1 Successful simulation using epidemic model

Figure 20 and figure 21 indicate that, using the chosen set of parameters and simulation set-up, an $\epsilon$-SIS model can in an effective way reproduce the occurrence of delay over time as well as the geographic distribution of delay (measured as ranking of stations in terms of delay). In other words, the SIS model captures both global and local delay behavior and the simulation attempt can be considered successful. One should note, however, that the results depend on the approximations made to the mechanisms controlling delay occurrence and that these may not hold up under all circumstances. One such assumption is the definition of a delayed station, as the notion of a delayed station may be considered counter-intuitive and an over-simplification of reality.

Under these assumptions, however, the simulation does produce results similar to the ones produced by the real data. The reader is reminded of the purpose of this thesis, and the authors believe the results are strong enough to suggest that a modified SIS model can in fact be used to model some characteristics of delay and delay propagation in the Swedish railway system.

In order for the reader to get a general feel for the simulation, a visualization of the infection process over four time steps (each time step is 30min) is shown in figure 36 below. Note that green nodes indicate susceptible nodes whereas red nodes indicate infected nodes.

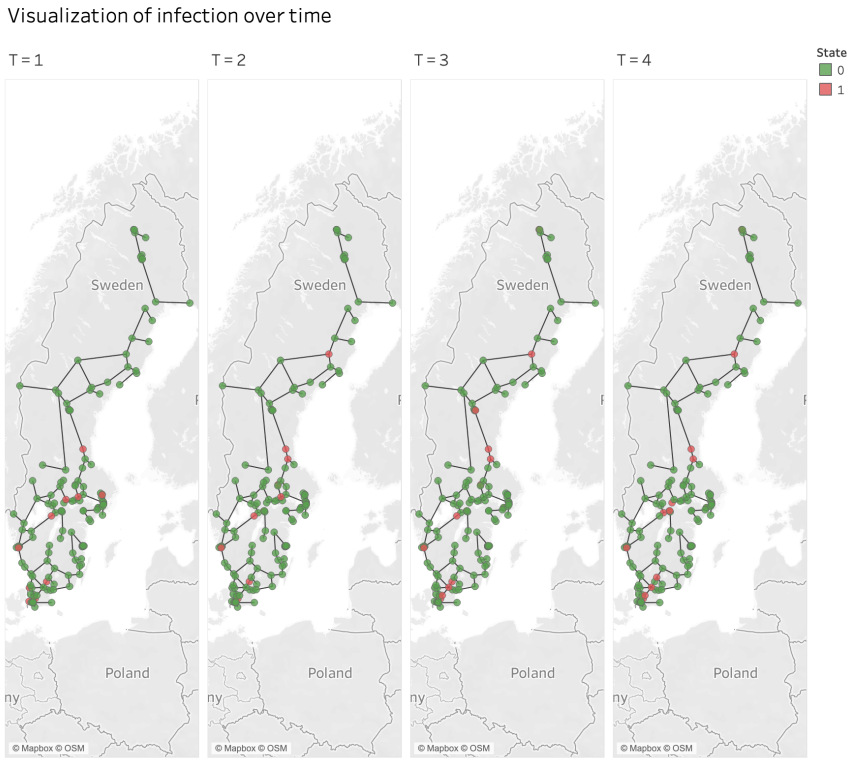


Figure 36: A visualization of the infection of stations over time

## 9.2 The necessary down-sizing of self-infection rate

Initially, the necessary down-sizing of $\epsilon$ (spontaneous delay rate/self-infection rate) in order to re-produce the overall delay level was interpreted as an implication that the problem might lie with the data from the Swedish Transport Authority. In this data, many fields are inserted manually. One of the activity codes used for marking the cause for delay is "delayed by another train", which would indicate propagated delay. In a scenario where this activity code is used too few times, the bulk of the delay would be considered to have arisen spontaneously and this would in turn result in a higher value of the $\epsilon$-parameter. This is not hard to imagine as propagated delay may sometimes be very hard to distinguish. For example, a power failure on a single line may delay a train. When other trains are then delayed due to this, it is difficult to determine whether the cause for their delay is the power failure or if they are disturbed by the *first* delayed trains, which would indicated that they are delayed due to propagated delay. One should also note that many delays that are not assigned an activity code, meaning that the reason for their delay is not classified.

However, should an error in the marking of activity codes (reasons for delay) be the only explanation for why the re-scaling of $\epsilon$ is necessary, it is expected that this down-sizing should be global and independent of things such as the size of the time step. As suggested by the sensitivity analysis presented under section **??**, this is not true as the simulation performance is worsened when the size of the time step is reduced.

Further, by the look of the plots in figure 28, the model performs worse when the definition of a delayed station is changed. According to these plots, a more accurate level of delay would potentially be obtained by a lower down-sizing of $\epsilon$ for higher percentages of delayed departures required to make a station infected. One hypothesis for this phenomenon is that the $\epsilon_j$ values are not independent of each other.

In this thesis, the $\epsilon_j$-value for station $j$ is considered an independent variable for all departures, meaning that all departures during a time step run the same risk of being spontaneously infected independently of one another. This assumption only holds up in a scenario where external factors are train-specific, such as involving a personnel issue or a malfunction with a specific train. In all scenarios where the entire station is affected by external factors, the train-specific probability of being spontaneously delayed should no longer be independent. This would in turn explain the downsizing required of the $\epsilon_j$ in order to produce satisfying results when raising the threshold for the definition of a delayed station.

This down-sizing of $\epsilon$ and whether it does in fact follow a non-linear curve requires further investigation before it can be properly proven. It is however a hypothesis which would explain the somewhat poor performance of the model when re-defining states (and perhaps also when changing the time-step), why the initial results presented in section 7 still seem promising and are not over-shadowed by issues presented in the sensitivity analysis.

## 9.3 Practical use of model

A model that in an accurate way reproduces delay behavior on the Swedish railway network can be used for several things. Firstly, it can be a valuable tool in identifying target points for preventive measures aiming to reduce delay in the system, whether globally or locally. The potentially efficient targeting of stations with a heavy load of freight traffic is an example of how the model may be used to identify critical points.

The model may also be used to stress test the system. By manually reducing or increasing the amount of delay on specific stations, stakeholders may learn how this will affect overall delay levels. This is similar to the analysis presented under section 7, where preventive measures on certain stations and lines are investigated.

Furthermore, the model may be used to predict delay occurrences. Using only the set schedule as an input, one could predict where and to what extent delays will occur in the future. The best usage of this would likely be on a year-to-year basis, such as using March 2019 to predict March 2020. Unfortunately, as discussed in 7, the training data in this thesis was from 2019, and the Covid-19 pandemic in 2020 and 2021 reduced the possibilities for determining the performance of the model on a year-to-year basis. However, the results in predicting both one month ahead (April 2019) and 6 months ahead (September 2019) showed promising results.

Finally, the model developed in this thesis is a mechanistic one. Granted, when using a mechanistic model the initial effort to program a simulation tool is quite high. Luckily, the authors of this thesis has dedicated that time. Once the simulation tool is developed, the cost of ownership is low, as is the effort needed for model calibration.

## 9.4 Potential issues with the model

As with all models, this one is not perfect. While capturing what is believed to be central aspects of delay behavior on the Swedish Railway Network, there are many things the model does not capture. Some of these things the authors are aware of, whereas others they are not. Before discussing specific issues with the model, it is worth mentioning that the authors' prior expertise on train traffic was limited. Whilst experts in the field have been consulted for this thesis, there are still many specifics of train traffic and delay management that presumably are not taken into account in the model set-up nor treated in the report.

In this thesis, the main concern is that approximations and assumptions made are over-simplistic and do not hold up on a greater scale. This is a balancing act between making necessary approximations and assumptions in order to be able to run the simulation at all, but at the same time tending to the integrity of the real mechanism in order for the model to be valuable.

### 9.4.1 Choice of data

One limitation to the model is the data which the model was built on. While data for all of 2019 was made available, the simulation was built and optimized using about one month worth of data (2019-02-28 through 2019-04-01), comprised of 2659405 departures (rows of data). The reason for this was that some of the algorithms performed in this thesis would be too computationally heavy to perform on a full year's worth of data. In addition, the amount and quality of the data for the month used led to the conclusion that one month was more than enough to achieve significant results. In retrospect, however, the *graph* should have been constructed from the yearly data. When the junctions and end stations was determined from the March 2019 data, there were 183 of them. When the March 2019 data was replaced with September 2019 data to perform the sensitivity analysis, it was revealed that there was a handful of stations which had arrivals and departures in the September data set but not in the March data set. This in turn meant that there were stations in the entire 2019 data set that should be included in the reduced network but that were not

included due to the fact that they were not represented amongst "active" stations in March. This affected the overall network and thereby results. Had there been more time, simulations would have been re-run on the updated network.

Adding to this, the 2019 data is believed to be representative for any other year. As the year of 2020 was affected by the Covid-19 pandemic, 2019 is considered the most recent "normal" year and hence the it was deemed reasonable to use data from this year for simulations. One should also note that the data set does not include cancelled departures. If there exist delays large enough to cause cancellations (or cancellations for other reasons), this implication cannot be drawn from the analysis.

### 9.4.2 Potential overfitting

Lastly, as discussed in the sensitivity analysis, a potential issue with the model, given the number of variables, is that of overfitting. As the recovery rate, self-infection rate and rate of propagating delay are all determined on a station based and edge based level respectively, this means that the number of variables in the model is equal to *the number of stations* * 2 + *the number of edges* * 1 + $\mu$, in this case $183 * 2 + 494 * 1 + 1 = 861$. Granted, it can be argued that the rate of recovery and the self-infection rate are all extracted from data and hence should not negatively impact the credibility of the model. This reasoning would argue that the model thus contains "only" 495 variables (the optimized $\beta_{i,j}$-values and $\mu$).

However, the performance of the model on the prediction data indicated that overfitting was not a pressing issue. The issue with overfitting is usually that the model performs well on the training data but poorly on the testing data. Since the $\epsilon$-SIS model used in this thesis performs similarly both on the testing data (September 2019) and the prediction data (April 2019) compared to the training data (March 2019), this was not an issue.

## 9.5 Further work

In order to improve this model, there are several things that may be done, many of them mentioned in the previous section. As an example, the simulation would presumably be more accurate if run on the entire network instead of the reduced one, preferably with a smaller time step.

Further, investigations into the dependency of the $\epsilon_j$ variable could yield improved performance. As mentioned previously, it is hypothesised that the reason for the model not performing as well with a different definition of a delayed station and a different time step could be down to this characteristic in the $\epsilon_j$ variable. However, as mentioned this poor performance could also be an outcome of too low $\delta_j$-values, why this should be analyzed further before drawing any conclusions.

Another, potentially interesting, addition to the method would be to replace the binary state space (infected, susceptible) with the actual level of delay. In other words, the state space would not be either 0 or 1, but rather all values between 0 and 1. This would potentially yield a more dynamic and detailed model.

Should these changes be implemented, it seems likely that the model's predictive power would be improved, making the model a more valuable tool for the Swedish Transport Authority, train operators as well as other stakeholders.

Lastly, with further work the model could presumably be altered in order to investigate the effects on temporarily closing stations or lines for infrastructure work. This

could be a valuable tool for project management and re-routing of traffic.

# 10 Summary

This purpose of this thesis was to examine whether a modified SIS model, traditionally used to model the spread of disease, could be used to model delay behavior and delay propagation in the Swedish railway system.

With the use of data from the Swedish Transport Authority, pre-processed by the Department for Civil Engineering at Lund University, a model was constructed and simulations using different simulation set-up were run.

The results obtained show that, when using an optimization algorithm to determine values of edge weights in the graph, as well as re-scaling the frequency of spontaneous delay in the model, the modified SIS model re-produces real-world delay behavior on both a global and a local scale. This is shown by the results indicating a re-produced level of delay over time close to that of the real data, as well as an accurate ranking of *which* stations were the most, and the least, delayed.

Furthermore, the results indicate that re-scaling the nodal self-infection rate (spontaneous delay rate) improves the model and is needed when changing the number of delayed departures needed to make an infected state, i.e. marking a station as delayed. Simulations indicate that this nodal self-infection cannot be expressed using a linear function but varies non-linearly with the number of delayed departures. One explanation for this would be the possible dependence between self-infection rates on trains, which could be explained by the fact that some external factors giving rise to spontaneous delay might affect entire stations rather than individual departures. However, the effect of re-scaling the self-infection rate could also be obtained by increasing the recovery rate, why further analysis is needed in order to determine which parameter should be modified and to what extent.

Results also indicate that the model may be used for estimating the impact on certain delay preventive measures by manually changing parameters for chosen stations or lines. This is valuable as it may give stakeholders a way of prioritizing projects and resources. Presumably, the method could also be tweaked to estimate the effects of infrastructure projects resulting in temporary closing of stations or lines, which could be a valuable tool when planning work or re-routing traffic.

Further analysis is needed in order to determine whether the SIS model is a valuable tool not only to re-produce delay behavior but to predict when and where delays will occur. Possibly, one could explore a spectral analysis similar to that of Li et al. to identify and predict unexpected delay patterns.

# References

Barabási, A.-L. (2013). Network science. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1987):20120375.

Ceria, A., Köstler, K., Gobardhan, R., and Wang, H. (2021). Modeling airport congestion contagion by heterogeneous sis epidemic spreading on airline networks. *Plos one*, 16(1):e0245043.

Kendall, M. G. (1938). A new measure of rank correlation. *Biometrika*, 30(1/2):81–93.

Li, M. Z., Gopalakrishnan, K., Balakrishnan, H., and Pantoja, K. (2019). A spectral approach towards analyzing air traffic network disruptions. In *13th USA/Europe Air Traffic Management Research and Development Seminar*.

Mei, W., Mohagheghi, S., Zampieri, S., and Bullo, F. (2017). On the dynamics of deterministic epidemic propagation over networks. *Annual Reviews in Control*, 44:116–128.

Monechi, B., Gravino, P., Di Clemente, R., and Servedio, V. D. (2018). Complex delay dynamics on railway networks from universal laws to realistic modelling. *EPJ Data Science*, 7(1):35.

Nowzari, C., Preciado, V. M., and Pappas, G. J. (2016). Analysis and control of epidemics: A survey of spreading processes on complex networks. *IEEE Control Systems Magazine*, 36(1):26–46.

Palmqvist, C.-W. (2019). *Delays and timetabling for passenger trains*. PhD thesis, Lund University.

SciPy (2021a). scipy.stats.kendalltau.

SciPy (2021b). scipy.stats.spearmanr.

Trafikverket (2020). Punktlighet på järnväg. Statistik 2021:2, [Online].

Wellman, B. (1983). Network analysis: Some basic principles. *Sociological theory*, pages 155–200.

Wu, W., Zhang, H., Feng, T., and Witlox, F. (2019). A network modelling approach to flight delay propagation: Some empirical evidence from china. *Sustainability*, 11(16):4408.

Zwillinger, D. and Kokoska, S. (1999). *CRC standard probability and statistics tables and formulae*. Crc Press.

# A Appendix

## Initial simulation set-up

In an early attempt, inspired by Li et al. (see Introduction), the weights on edges in the network (the $\beta_{i,j}$-values) were given as the correlation between the delays at different stations. Time series over the simulation period containing whether a station was delayed or not during the time steps were created, and the correlation coefficients (Pearson's r) was calculated between the time series of adjacent stations. The resulting network with correlation values as weights on edges is visualized in figure 37 below.

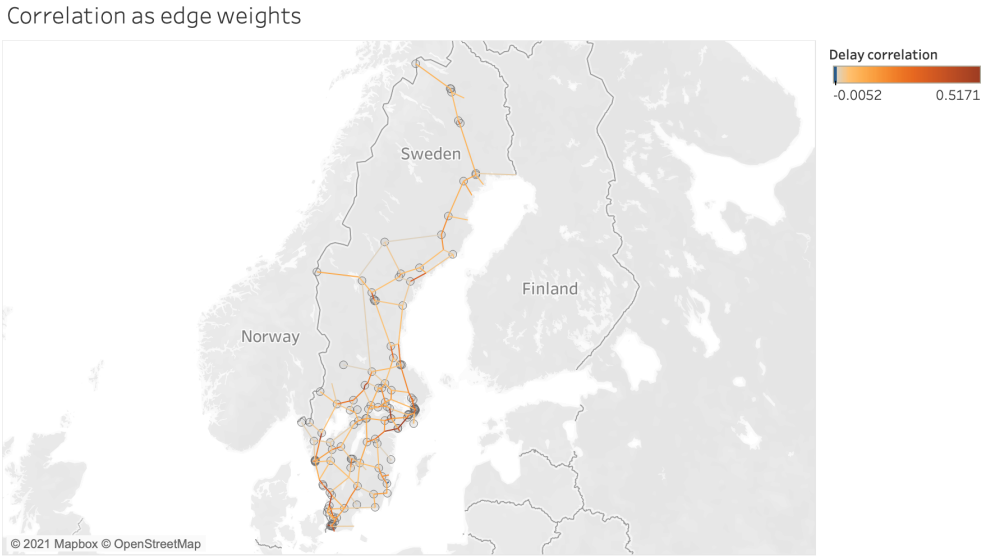


Figure 37: Visualization of network with correlation as edge weights

The node-specific probabilities of infection via propagation were then computed from these weights (which were assumed to be the edge-dependent $\beta_{i,j}$-values), as per below. Note that $I$ denotes the set of infected nodes.

$$P(S_{t+1}^j = 1 | S_t^j = 0)_{propagation} = 1 - \prod_{i \in I}(\beta_{i,j}) = 1 - \prod_{i \in I}(1 - W_{i,j}) \tag{22}$$

When assigning edge weights this way, the resulting graph was **undirected** as delay correlation does not differ depending on direction.

The rate of spontaneous delay, $\epsilon$ was harder to compute for this trial as individual train departures were not taken into consideration. Instead, the parameter $\epsilon$ was approximated, saying that the percentage of train departures that were late due to spontaneous delay should be proportional to the percentage of stations that can be assumed to be "delayed" due to spontaneous delay. In other words, a *station's* epsilon value is the same as that of a *train's* epsilon value. This is a very rough approximation and may be the main reason for the poor performance of the simulation set-up, why this attempt was later disregarded. However, the simulation was carried out under this assumption and initially the $\epsilon$ parameter was not re-scaled.

Furthermore, as the number of departures were not taken into account, the difference in train activity was estimated using static parameters. These parameters shifted the probability of propagated depending on weekday and hour of day. Based on data from the Swedish Transport Authority, Monday-Friday were treated equally whereas the probability of weekend delay propagation was down-scaled to 80% of that of weekdays. Similarly, all hours of the day were treated equally apart from 1AM-4AM where traffic, and thereby the probability of delay propagation, was assumed to be 0.

The following graph was obtained when comparing the percentage of delayed stations in each time step for real versus simulated data, using parameters as described above.

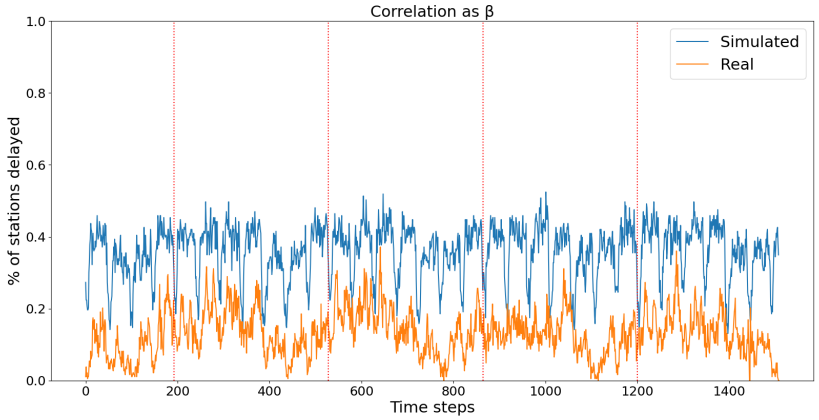


Figure 38: Percentage of stations that are delayed, simulation vs real data

The figure indicates that although the model captures periodicity of delay per time step, the average level of infected nodes (delayed stations) per time step is greater fro the simulated data than for the real data. This suggests an issue with the choice of $\beta_{i,j}$ or $\epsilon$.

When looking at the geographic distribution of delay, by looking at the ranking of stations in terms of delay frequency, the following graph was obtained. This shows a discrepancy between the two data sets. This indicates that the simulation using this particular set-up does not re-produce delay behavior at a local level, as it does not infect the "right" stations. This means that the level of delayed stations follows a pattern similar to that of the real world (albeit the level of delay is increased), but that it is not the same stations that are delayed when comparing model and reality.

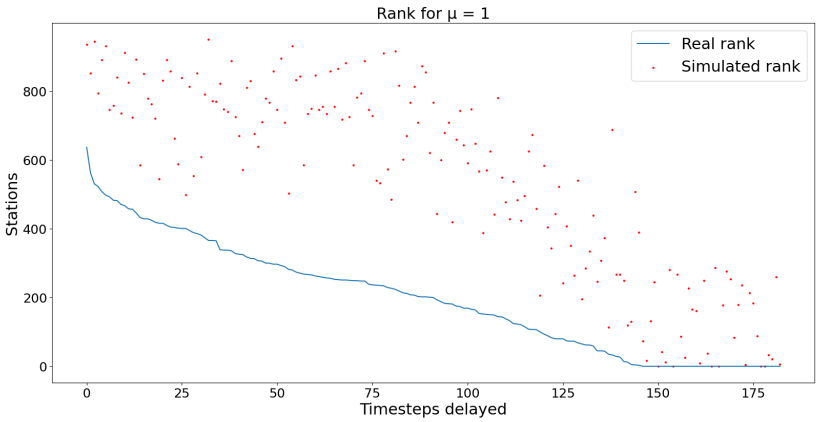


Figure 39: Ranking of stations in terms of delay frequency per station, simulation vs real data

As an issue with $\epsilon$ was later discovered and the parameter was re-scaled (see 6, this simulation was also run with a re-scaled version (using re-scaling parameter $\mu = 0.6$) of $\epsilon$. The reader is reminded that for this attempt, $\epsilon$ for a station is still approximated to $\epsilon$ of a train, i.e. the probability that a stations is infected by spontaneous delay is equal to that of a train being delayed spontaneously. The simulation using the re-scaled $\epsilon$ produced the following graphs.

Figure 40 shows the percentage of delayed stations over time.

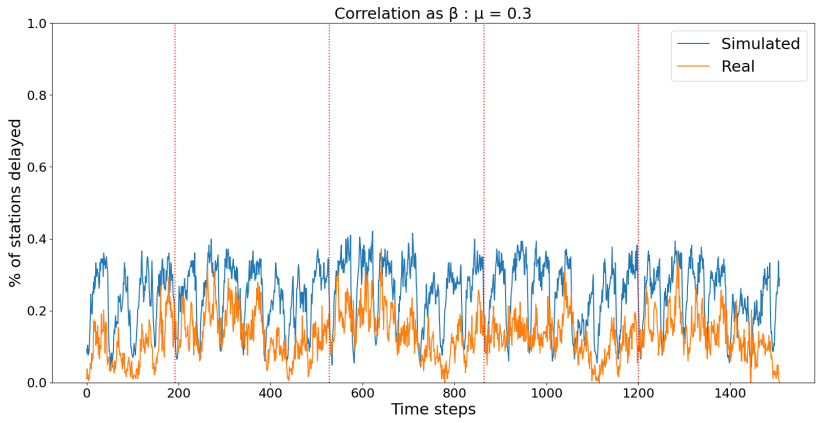


Figure 40: Ranking of stations in terms of delay frequency per station with re-scaled $\epsilon$, simulation vs real data

Figure 41 shows the geographic distribution of delay as a ranking of stations in terms of delay frequency when using the re-scaled $\epsilon$.

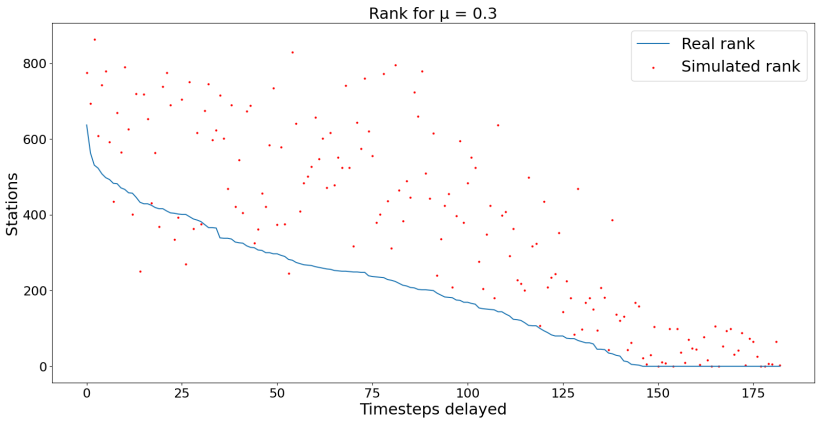


Figure 41: Ranking of stations in terms of delay frequency per station with re-scaled $\epsilon$, simulation vs real data

The discrepancy in all figures led to the conclusion that the set-up using delay correlation as edge weights was not satisfactory neither in re-producing the ratio of infected nodes over time nor in re-producing the geographic distribution of delay.

A fault that was found in this set-up, that accounted for the re-computation of $\beta_{i,j}$ as described in the thesis, was the lack of consideration for the number of departing trains from infected (delayed) neighboring stations to the susceptible one. That, in combination with the realization that edge weights were not easily computed, sparked the idea of optimizing the *probability that a train departing from a delayed station transfers this delay to the next station*, i.e. the edge weights. By then considering the total number of departing trains from an infected neighboring station, the total probability of the infection spreading from an infected node to a susceptible neighboring node was computed.

## Additional figures and tables

### 10 average stations

| Station | Delay score |
|---|---|
| Malmö C | 0.566 |
| Råtsi | 0.553 |
| Landskrona Östra | 0.541 |
| Teckomatorp | 0.532 |
| Jädersbruk | 0.520 |
| Grängesberg | 0.518 |
| Gimonäs | 0.516 |
| Huvudsta | 0.514 |
| Ställdalen | 0.504 |
| Strömtorp | 0.501 |

Table 40: 10 average stations according to delay score

### 10 average lines

| Line | Delay score |
|---|---|
| Hällnäs - Vännäs | 0.151 |
| Karlberg - Stockholm C | 0.152 |
| Kalmar Södra - Blomstermåla | 0.152 |
| Ånge - Sundsvalls Västra | 0.153 |
| Ånge - Ångebyn | 0.153 |
| Snyten - Avesta Krylbo | 0.154 |
| Södertälje Syd Övre - Flemingsberg | 0.162 |
| Nynäshamn - Älvsjö | 0.163 |
| Frövi - Ställdalen | 0.164 |
| Helsingborg Gbg - Teckomatorp | 0.164 |

Table 41: 10 average lines according to $\beta_{i,j}$-values

### Figures for changing time step to 15 min

Mean Absolute Scaled Error for delay level over time:

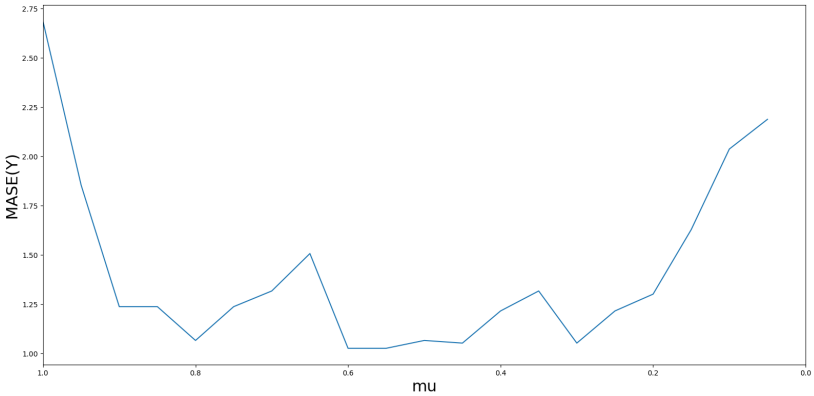


Figure 42: Mean Absolute Scaled Error for different re-scaling of $\mu$, time step = 15 min

Mean Absolute Error for geographic distribution:

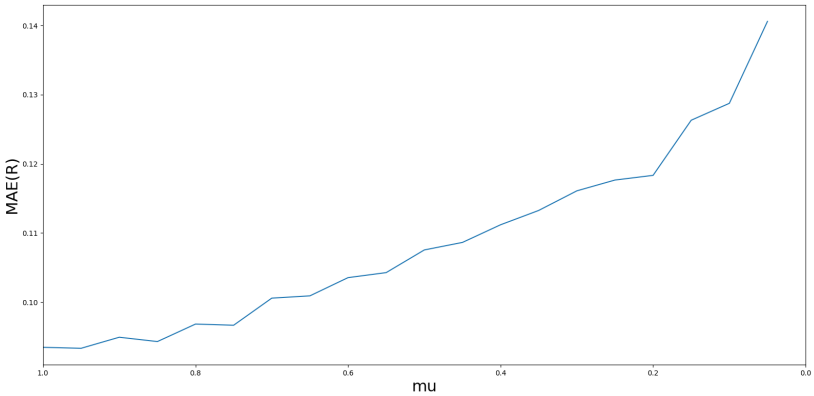


Figure 43: Mean Absolute Error for different values of $\mu$, time step = 15 min

**Figures for changing time step to 60 min**

Mean Absolute Scaled Error for delay level over time:

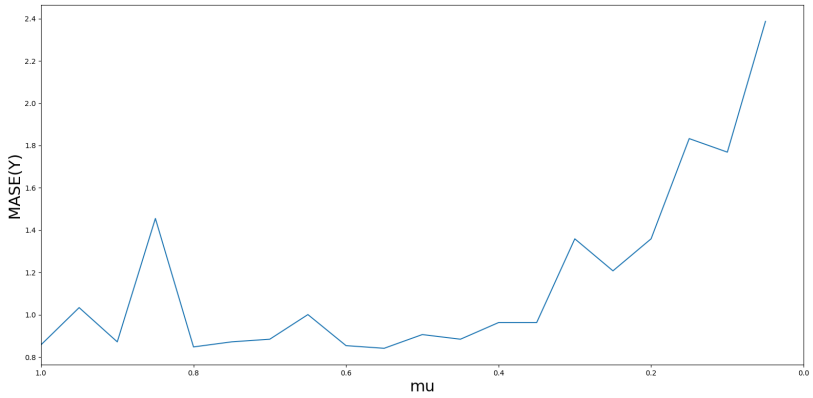


Figure 44: Mean Absolute Scaled Error for different re-scaling of $\mu$, time step = 60 min

Mean Absolute Error for geographic distribution:

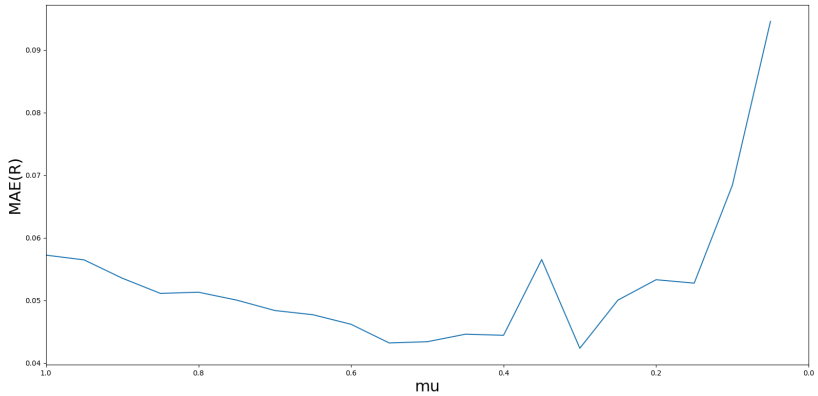


Figure 45: Mean Absolute Error for different values of $\mu$, time step = 60 min

**Lund University**
**Department of Automatic Control**
**Box 118**
**SE-221 00 Lund Sweden**

*Document name*
MASTER'S THESIS

*Date of issue*
June 2021

*Document Number*
TFRT-6137

*Author(s)*
Jacob Landelius
Elsa Wallgren

*Supervisor*
Carl-William Palmqvist, Transport and Roads, Lund University, Sweden
Emma Tegling, Dept. of Automatic Control, Lund University, Sweden
Giacomo Como, Dept. of Automatic Control, Lund University, Sweden
Anders Rantzer, Dept. of Automatic Control, Lund University, Sweden (examiner)

*Title and subtitle*
Network analysis of delay propagation on Swedish railways

*Abstract*

Travel on railway in Sweden has increased steadily over the past three decades and there are today twice as many passengers travelling by train as there were 30 years ago. The increasing awareness of environmental issues with other methods of transportation is likely to favour railway travel, and so the number of passengers is expected to continue to rise. As the number of passengers increase, so do the requirements on keeping trains on time. Delayed trains do not only cost money for train operators, but may also affect how likely we are to choose to travel by train. Understanding where and why delay occurs as well as how this delay might spread is therefore important not only from an economic point of view but from an environmental one as well.

This thesis shows that behaviour of delay occurrence and propagation of delay in the Swedish railway network may be reproduced using an epidemic Susceptible-Infected-Susceptible (SIS) model with satisfying results. By optimizing the probability of a train carrying infection (delay) from an infected station to a susceptible one, the simulation can reproduce the level of delay over time as well as the geographic distribution of delay, thus capturing global as well as local delay behavior. The thesis further shows the necessity of heterogeneous delay propagation probabilities on edges in the network in order to reproduce real-world behavior.

Furthermore, the results indicate that re-scaling the nodal self-infection rate (spontaneous delay rate) improves the model and is needed when changing the number of delayed departures needed to make an infected state, i.e. marking a station as delayed. Simulations indicate that this nodal self-infection cannot be expressed using a linear function but varies non-linearly with the number of delayed departures. One explanation for this would be the possible dependence between self-infection rates on trains, which could be explained by the fact that some external factors giving rise to spontaneous delay might affect entire stations rather than individual departures. However, the effect of re-scaling the self-infection rate could also be obtained by increasing the recovery rate, why further analysis is needed in order to determine which parameter should be modified to what extent.

Lastly, results also indicate that the model may be used for estimating the impact on certain delay preventive measures by manually changing parameters for chosen stations or railway lines. This is valuable as it may give stakeholders a way of prioritizing projects and resources.

*Keywords*
networks, graph theory, delay, delay propagation, railway

*Classification system and/or index terms (if any)*

*Supplementary bibliographical information*

*ISSN and key title*
0280-5316

*ISBN*

*Language*
English

*Number of pages*
1-70

*Recipient's notes*

*Security classification*