

RESEARCH

Modelling railway delay propagation as diffusion-like spreading

Mark M. Dekker^{1,2*}, Alexey N. Medvedev³, Jan Rombouts⁴, Grzegorz Siudem⁵ and Liubov Tupikina^{6,7}

Abstract

Railway systems form an important means of transport across the world. However, congestions or disruptions may significantly decrease these systems' efficiencies, making predicting and understanding the resulting train delays a priority for railway organisations. Delays are studied in a wide variety of models, which usually simulate trains as discrete agents carrying delays. In contrast, in this paper, we define a novel model for studying delays, where they spread across the railway network via a diffusion-like process. This type of modelling has various advantages such as quick computation and ease of applying various statistical tools like spectral methods, but it also comes with limitations related to the directional and discrete nature of delays and the trains carrying them. We apply the model to the Belgian railways and study its performance in simulating the delay propagation in severely disrupted railway situations. In particular, we discuss the role of spatial aggregation by proposing to cluster the Belgian railway system into sets of stations and adapt the model accordingly. We find that such aggregation significantly increases the model's performance. For some particular situations, a non-trivial optimal level of spatial resolution is found on which the model performs best. Our results show the potential of this type of delay modelling to understand large-scale properties of railway systems.

Keywords: Complex Networks; Spreading on Networks; Railways; Scaling; Coarse graining; Diffusion

1 Introduction

Railway systems are of vital importance for transporting passengers and goods. The trains in these systems travel via predefined schedules that allow for highly efficient utilization of the routes and tracks. Temporal deviations from such scheduled operations are commonplace. They take the form of *delays* and decrease the system's efficiency. Small delays are often absorbed by built-in buffers and therefore do not have effects on larger scales [1, 2]. However, from time to time, logistic disruptions — often caused by external factors like weather — lead to congestion or even a large-scale stand-still, with detrimental costs to society and economy [3, 4, 5, 6].

The above shows the importance of better understanding of delay propagation and its prediction. A large variety of delay propagation models exists, and the choice of the approach depends on a number of questions related to, among other factors, the spatial focus, availability of data and the delay severity. For

example, when aiming to accurately predict the delay in a geographically confined area, there are high-performing statistical models [7, 8]. However, such statistics generally only work accurately in circumstances where delay is not too severe — as per definition these highly delayed scenarios are exceptional. Also, when upscaling to larger areas, long-range interactions and associated correlations come into play which may be difficult to account for when using average statistics. Larger scales and more highly delayed scenarios are therefore often analyzed with machine learning or big-data approaches [9, 10], but at the cost of understanding cause-and-effect or fine spatial resolution. Alternatives to such purely data-driven methods can be found in models where mechanisms of delay propagation are explicitly implemented. For example, Monechi et al. analyzed the German and Italian railways and found a set of 'laws' that drive the spreading of delays [11], analogous to epidemic spreading models. Of course, the infrastructure networks underlying the dynamical processes in any of the mentioned models play an important constraining role. However, this information is already embedded in the schedules and therefore less discussed in the context of delay sim-

*Correspondence: m.m.dekker@uu.nl

¹m.m.dekker@uu.nl, Department of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands

Full list of author information is available at the end of the article

ulations. However, the role of railway network topology is addressed by various scholars in relation to resilience properties [12, 13, 14]. Most models are based on the schedules of the railway system, commonly using trains as agents that have the potential to carry delays. The perspective of delays as a properties of discrete trains or events can be found in many analytical models [15, 16, 17, 18, 2], using either deterministic or stochastic techniques to derive future delays from past information. Because of the abundance of this perspective in existing delay propagation models, we refer to the view of delays as properties of discrete trains or events as the ‘traditional view’. In contrast, one could also view delays as variables associated not to trains, but to the nodes (stations) and edges of the railway network, which stay in the same position. How delay spreads between these nodes does not have to be described in terms of discrete trains and events, but instead a description may rely solely on general (or even system-wide) quantities such as the network topology and schedule. One can make the analogy of fluid dynamics: while traditionally, delays are treated as Lagrangian particles (i.e., following the trains as the fluid carrying the particles), we propose to treat delays from an Eulerian point of view (i.e., determining incoming and outgoing delays in a fixed spatial frame). This is also discussed in [19]. This is the basis of the model proposed in this paper.

The traditional view of delays as discrete quantities of explicitly modelled trains or events is useful because it allows for tracking expected routes of delays along the train’s trajectories explicitly. In other words, given that you know that delay is in the system at location A , it is unlikely to spread in all possible directions from A , but more likely to follow a particular direction that is dependent on which trains are exactly affected. One only knows this direction if discrete train units (and their trajectories) are explicitly included in the model. But there are also disadvantages of such models. One limitation is that many such models rely on many statistics in addition to mere schedule information. For example, if trains A , B and C are simulated explicitly, the interactions of all their events and relative magnitudes of their delays have an impact on each other’s delays. These relations need to be well studied using for example neural networks [20, 10] or probability updating [21, 22]. Another limiting consideration of treating delays as discrete quantities is the spatial scale. In confined systems, the mechanisms of delay propagation and their parameters can be well-defined, as in [7]. Defining all such interactions on a country-wide scale is generally more more complex, due to potential long-range correlations.

In this paper we propose to treat delays not as bound by discrete trains or events, but rather as continuously spreading across the infrastructure network. The spreading between nodes of the network is weighted by properties of the system. The intuition behind these models is that on average — in a ‘mean-field approximation’ — these parameters drive the overall direction of delay propagation. We refer to this way of treating delay propagation as ‘diffusion-like spreading’. Small-scale accuracy is traded for larger scale accuracy: when looking at a micro-scale or individual trains, we expect this non-traditional way of dealing with delays to be less accurate than more detailed models, but on a large scale, we expect the performance of such a model to increase. As is shown in section 2, the model contains only simple schedule information (e.g., train frequencies and travel times) rather than complicated statistics, and all model information is embedded in a single matrix, which makes analysis of the system’s properties easy. The mentioned reasons motivate us to write this paper on delay propagation as a diffusion-like spreading mechanism. We apply our proposed model to the Belgian railways as a case study to discuss when and how it is advantageous to use such models.

An important aspect of delay propagation in general is the spatial scale and resolution of the analysis. High resolution (‘micro-scale’) modelling allows for explicit simulation of infrastructure capacity issues, the role of speed gradients or the identification of station-specific properties, for example. Low resolution, but large-scale (‘macro-scale’) modelling captures the impact of long-range interactions related to resource allocation [2], the impact of long train lines [9] or other system-wide properties. Many models lie between these extremes. Diffusion-like models should typically be regarded as having a lower resolution but working well on a larger scale, because of the earlier mentioned trade of small-scale accuracy for larger-scale accuracy. Spatial resolution is often expressed by treating railway infrastructure as a network, consisting of nodes (geographical locations) and edges (connections between them). At the highest spatial resolution, the nodes are certain control points in stations and tracks, where train activities are logged [2]. More commonly is a slightly aggregated version of this, namely the more coarse passenger stations [14, 15, 11]. Lower resolutions are obtained when constructing regions that correspond to groups of stations — so-called ‘clusters’, on which we elaborate later. Larger geographical areas in lower resolutions combine existing delays from higher resolutions and are treated as one unit. Choosing the correct level of spatial aggregation is an important consideration to make when assessing the viability of the diffusion-like model.

When discussing spatial aggregation, it is important to define how higher levels of aggregation are derived from the lower ones. In particular: how do we join stations and tracks together into larger and coarser regions? A large amount of complex network literature is devoted to this question of *clustering*, and clustering methods come in many forms in various applications [23]. For example, graph or connectivity-based methods emphasize how connections and topology lead to a natural aggregation of nodes into larger groups. This can be quantified by the so-called modularity of the partition, first proposed by Newman [24]. Various clustering algorithms based on modularity optimisation exist, such as the Louvain method [25].

Spectral clustering focuses on properties of the eigenspace of the Laplacian or model-relevant matrices. A third common method for clustering any — also non-networked — data is K-means [26, 27], which defines centroids and groups nodes based on their respective distances to these cluster centroids (also known as Voronoi iteration [28]), given a definition of ‘distance’ between nodes. This method has been used in the context of transportation before, albeit mostly to characterise statistical space (rather than actual stations and physical space) [29, 30]. An important aspect of K-means, in contrast to for example the Louvain method, is that it requires the specification of the number of desired clusters (K) up front, which can be both advantageous and disadvantageous. However, the freedom of choosing K turns out to be useful when analysing our diffusion-like delay model. This, together with the fact that K-means is a well known and commonly used method, motivates us to use K-means with geographical distance to cluster the stations in our paper. By choosing the number of clusters, we vary the spatial aggregation level. We will compare the performance of the diffusion-like model on each of these levels.

In summary, the aim of this paper is to discuss the usefulness of treating delay propagation as a diffusion-like spreading mechanism. We propose a model doing so in section 2. We apply the model to the example case of the Belgian railways and discuss the data and methodology for this in section 3. Section 4 discusses the results of a toy model, its performance on different types of disrupted situations, and the overall performance of the model. Here we discuss in what cases the diffusion-like aspect of the model is beneficial and what we can learn about the Belgian railways using this framework. We end with a summary and several conclusive remarks in section 5.

2 Model

In this section we introduce our diffusion-like model. We start by defining the delay variable and set up the

equations that describe its evolution over time. We continue by discussing how this model can be generalized to any spatial scale. For a detailed derivation of the model, see Appendix A. Tab. 1 summarizes the variables and parameters of the model.

2.1 General concepts

The main idea behind the model is to define the delay on fixed locations, and to describe the evolution of this delay distribution over time using macroscopic parameters such as train frequencies and travel times. While delays are inherent attributes of trains (i.e. agents), we aggregate the delays on passenger stations (i.e. nodes), as the impact of disruptions can mostly be felt at level of stations rather than being a problem of individual trains. This aggregation of delays onto stations means that we lose some of the finer details on which delays belong to which train. However, it will allow us to use tools for studying dynamical processes on networks: a delay is associated to each node, and its evolution is determined by the coupling of nodes through edges. For ease of notation, we will use the terms ‘station’ and ‘node’ interchangeably even though some nodes are actually junctions and not stations. We denote the delay of a station i at time t by $D_i(t)$. This variable is defined as the sum of the delays of all trains that are moving *towards* station i at time t :

$$D_i(t) = \sum_{T \in \mathcal{T}(i,t)} d_T(t), \quad (1)$$

where $\mathcal{T}(i,t)$ is the set of trains moving to station i (i.e. the very next station they cross will be i , whether they stop there or not) at time t and $d_T(t)$ denotes the delay carried by train T at time t . We consider two ways in which the value of D_i can change over time:

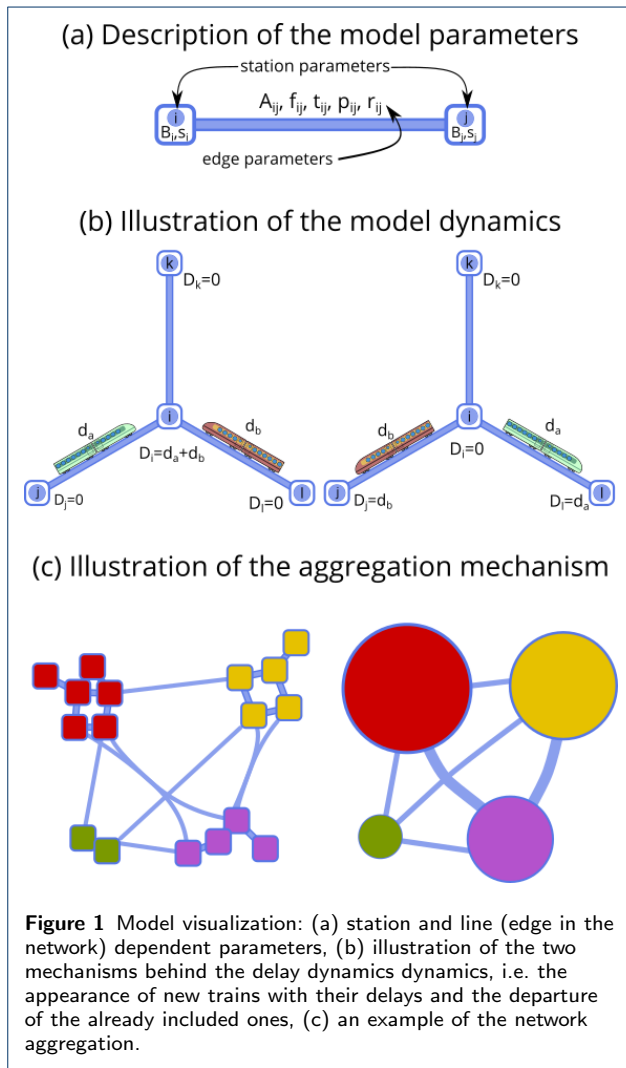
- 1 A train, which was previously moving towards another station j , reaches j and is now moving towards i . Therefore, its delay is now added to D_i .
- 2 A train, which was moving towards i , reaches i and either moves further towards another station or ends its trajectory. Therefore, its delay is removed from D_i .

The delay of station i at the next time step — we refer to this as $D_i(t + \Delta t)$, with Δt being the time step size — is dependent on the delays in various locations at the previous time step, not only $D_i(t)$. Thus, we write the relation between the delays between two consecutive time steps using a delay vector

$\vec{D} = (D_1, D_2, \dots, D_N)^T$, where N is the total number of nodes:

$$D_i(t + \Delta t) - D_i(t) = \underbrace{F_{1,i}(\vec{D})}_{\text{New incoming trains towards } i} - \underbrace{F_{2,i}(\vec{D})}_{\text{Arrival of trains at station } i} \quad (2)$$

with $F_{1,i}$ describing how the delay at station i changes over a time step Δt by means of the first term above (the addition of delay), and $F_{2,i}$ likewise by the second term above (the removal of delay). In the next section we express both these functions $F_{1,i}$ and $F_{2,i}$ in terms of several parameters and $\vec{D}(t)$. An illustration of the model and its terms is given in Fig. 1.



2.2 Diffusion model equations

The first term ($F_{1,i}$) sums the delays carried by all trains that start moving towards i in the interval $[t, t + \Delta t]$:

$$F_{1,i}(\vec{D}) = \sum_{\substack{\text{Trains } T \text{ that started moving to } i \\ \text{at } \tau \in [t, t + \Delta t]}} d_T(\tau).$$

This sum can be rewritten as a sum over the neighbors of i . By making a number of assumptions, like approximating the fraction of trains in each direction by the relative frequency (a full derivation can be found in Appendix A), we can rewrite the delay of a train moving to a station j as function of the delay of that station D_j and express how many of the trains arriving at a station j continue to i . This leads to:

$$F_{1,i}(\vec{D}) = \Delta t \sum_{j \in \mathcal{N}_{in}(i)} p_{ji} B_j D_j(t). \quad (3)$$

Here $\mathcal{N}_{in}(i)$ is the set of stations j that have an edge to i . The parameter p_{ji} is the probability that a train that reaches station j will continue towards station i , and is computed as follows:

$$\begin{aligned} p_{ji} &= P(\text{to } i | \text{from } j) \\ &= P(\text{to } i | (\text{from } j \ \& \ \text{do not end at } j)) \cdot P(\text{do not end at } j) \\ &= \frac{f_{ji}}{\sum_{\ell \in \mathcal{N}_{out}(j)} f_{j\ell}} \cdot \left(1 - \text{Probability that train has end station at } j\right) \\ &= r_{ji}(1 - s_j), \end{aligned} \quad (4)$$

where $\mathcal{N}_{out}(j)$ is the set of stations to which there is an edge from j . The value of p_{ji} is equal to a multiplication of two factors. The first (denoted by r_{ji}) is the probability that if a train reaches j and it does not end its journey there, it will then continue towards i . Note that we consider this probability to be independent of where the train came from: we do not consider any memory in this process. The value is calculated as the frequency of trains going from j to i divided by the frequency of all outgoing trains from j . The second factor in Eqn. 4 (denoted by $1 - s_j$) is the probability that a train that arrived at j ends its journey there, for example because it is the terminus. The variable B_i in Eqn. 3 is a station-dependent parameter, defined as

$$B_i = \frac{\sum_{\text{edges } e \text{ to } i} f_e}{\sum_{\text{edges } e \text{ to } i} f_e t_e} = \frac{\sum_{\ell \in \mathcal{N}_{in}(i)} f_{\ell i}}{\sum_{\ell \in \mathcal{N}_{in}(i)} f_{\ell i} t_{\ell i}}, \quad (5)$$

where f_e denotes the frequency of trains on edge e , and t_e corresponds to the time a train takes to cross edge e . The parameter B_i has units of time^{-1} and can

therefore be interpreted as a rate. The inverse of B_i is the average time of edges towards i , weighted by their frequency. A high value of B_i corresponds to a station with incoming short edges with high frequency. Intuitively, B_i can be thought of as a station's train turnover rate.

The second term of Eqn. 2 ($F_{2,i}$) counts the delays of trains that reach station i and therefore remove their delays from D_i . We express $F_{2,i}$ as follows (for details see Appendix A):

$$\begin{aligned} F_{2,i}(\vec{D}) &= \sum_{\substack{\text{Trains } T \text{ that reached } i \\ \text{at } \tau \in [t, t + \Delta t]}} d_T(\tau) \\ &= \Delta t B_i D_i(t). \end{aligned} \quad (6)$$

The term only depends on the delay $D_i(t)$ at station i at the previous time step, and the previously mentioned parameter B_i . The delay loss at a station can be interpreted as an exponential process with rate B_i .

The contributions F_1 and F_2 are expressed in terms of the delay state vector \vec{D} and in terms of various railway parameters (summarized in Tab. 1). Filling in these two terms into Eqn. 2 gives the full expression for the evolution of the delay D at any station i :

$$D_i(t + \Delta t) - D_i(t) = \Delta t \left[\sum_{j \in \mathcal{N}_{\text{in}}(i)} p_{ji} B_j D_j(t) - D_i(t) B_i \right].$$

We can simplify the sum over the neighbours of i by using the railway network's adjacency matrix A , which has entries $A_{ji} = 1$ if there is an edge from station j to station i and entries zero elsewhere:

$$\frac{D_i(t + \Delta t) - D_i(t)}{\Delta t} = \sum_j A_{ji} B_j D_j(t) p_{ji} - D_i(t) B_i.$$

Here, the sum goes over all nodes j . This equation can be written in matrix form using \vec{D} as a column vector. Moreover, we can take the limit $\Delta t \rightarrow 0$. This leads to the expression

$$\frac{d\vec{D}(t)}{dt} = \mathbf{G} \cdot \vec{D}(t). \quad (7)$$

The above equation contains the core model matrix \mathbf{G} , an $N \times N$ matrix defined as follows (δ_{ij} is the Kronecker delta):

$$G_{ij} = A_{ji} p_{ji} B_j - \delta_{ij} B_j. \quad (8)$$

All of the dynamics of the model are encapsulated in the matrix \mathbf{G} .

2.3 Model aggregation to clusters of stations

In this paper, we aim to describe how well our model describes real delay propagation patterns. One variable in this analysis is the level of spatial aggregation at which we simulate the model. In the previous section we explained the model where each node of the network consists of a single station or junction. However, the same principles can be applied to a network where nodes correspond to a group of such stations. The method we use to group stations into clusters is explained in Sec. 3.4. Here, we discuss how the model parameters for the full-resolution model based on individual stations can be translated into a lower resolution version. The discussed aggregation process is very similar to network of networks idea known in the networks literature [31, 32, 33].

Above, each delay variable D_i corresponds to one node of the network. This is achieved by transforming delays on trains to delays on stations via Eqn. 1. This is already a form of coarse-graining the delay dynamics. Now, we assume that the original railway network of N stations is divided into K clusters (or groups of stations). We indicate stations with lowercase letters (i and j) and clusters with uppercase letters (I and J). The clusters naturally form a network: an edge between clusters I and J exists if there is at least one station i in I and one j in J such that there is an edge between i and j in the original network. We define the function \mathcal{C} from stations to clusters such that $\mathcal{C}(i)$ is the cluster to which station i belongs. Let $D_I(t)$ denote the total delay of all trains moving to any station in cluster I at time t , either from inside the cluster, or coming from other clusters. An equation for the evolution of this delay can be derived in the same way as we did above for stations. The delay D_I can change when trains start towards any station in this cluster, or when trains arrive at a station in this cluster. The main difference with the non-clustered case (above) is the fact that in the clustered case, self-loops in the network appear. This is because trains moving to a station in a cluster — and thus adding to the cluster's delay — can reach that station, and then continue to another station in the same cluster, again adding to the cluster's delay.

While the equations in the clustered case are the same as in the non-clustered case, the parameters such as frequencies and travel times are now defined on edges between clusters. We explain a method to express these cluster parameters in terms of their non-clustered counterparts (i.e., the ones in Tab. 1). We

Variable	Description
$D_i(t)$	Delay at station i and time t
A_{ji}	Adjacency matrix of the railway network
B_i	Train turnover rate of station i
$d_T(t)$	Delay carried by train T at time t
f_{ij}	Train frequency from stations i to j
\bar{t}_{ij}	Average travel time from station i to j
p_{ji}	Fraction of trains to j that continues to i
r_{ji}	Fraction of trains to j that continue to i if they do not end at j
s_j	Fraction of trains that end at station j
$\mathcal{T}(i, t)$	Set of trains moving to station i at time t
$\mathcal{N}_{\text{out}}(j)$	Set of stations to which there is an edge from j
$\mathcal{N}_{\text{in}}(j)$	Set of stations from which there is an edge towards j
N	Amount of stations
δ_{ij}	Kronecker delta ($\delta_{ij} = 1$ if $i = j$, and 0 otherwise)

Table 1 Overview of the model variables and parameters.

start with the total frequency f_{IJ} and weighted averaged travel time t_{IJ} of trains between two clusters I and J . We define them as

$$f_{IJ} = \sum_{i \in I} \sum_{j \in J} f_{ij} \quad (9)$$

$$t_{IJ} = \frac{\sum_{i \in I} \sum_{j \in J} \bar{t}_{ij} f_{ij}}{\sum_{i \in I} \sum_{j \in J} f_{ij}}. \quad (10)$$

These definitions are intuitive: the total frequency of trains between two clusters is the sum of the frequencies on edges going from a station in the first to a station in the second cluster. The travel time is the weighted average of the travel times of the edges going from the first to the second, weighted by their frequency.

Next, we need to define the stopping probability s_I for a cluster I . In order to do this, we define the station parameter q_i as the probability that a train which arrives in the cluster $\mathcal{C}(i)$, arrives at station i . In a way, it indicates how important station i is in its cluster, measured by the total frequency of all incoming trains to that station. The quantity is approximated as follows:

$$q_i = \frac{\sum_{j \in \mathcal{N}(i)} f_{ji}}{\sum_{j \in \mathcal{C}(i)} \sum_{\ell \in \mathcal{N}(j)} f_{\ell j}}. \quad (11)$$

Note that the values q_i are weights of stations whose sum is one. Each station is weighted by the frequency of incoming trains. Next we use the quantities q_i to estimate the stopping probability s_I for cluster I :

$$s_I = \sum_{i \in I} s_i q_i. \quad (12)$$

One can interpret the stopping probability formula using the following formula:

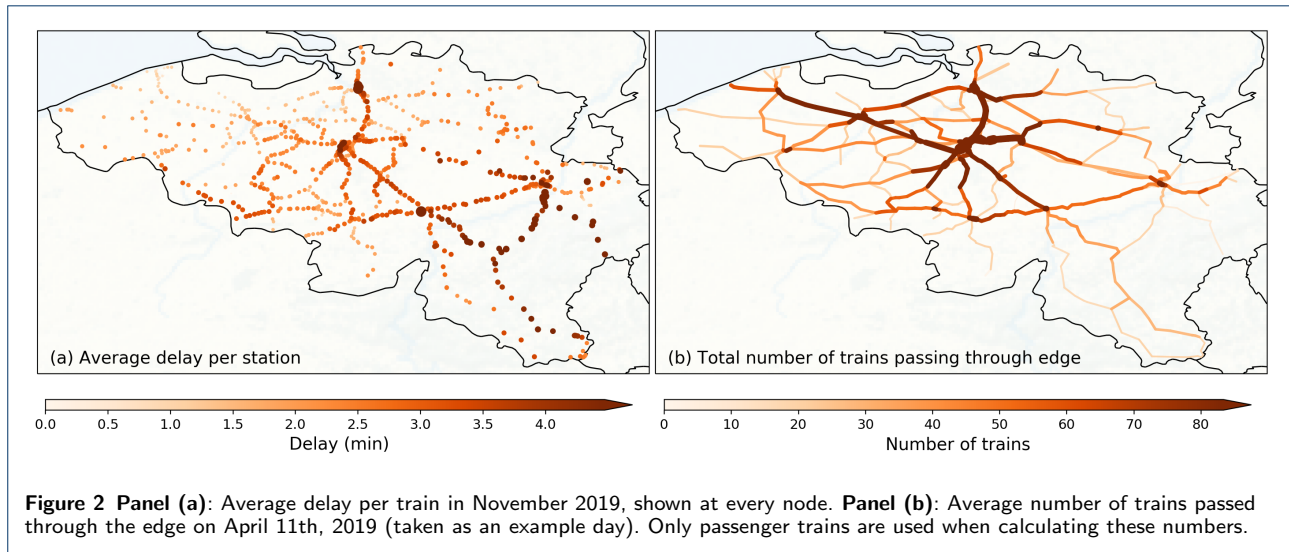
$$s_I = \sum_{i \in I} P(\text{stops in } i | \text{arrived in } i) P(\text{arrives in } i | \text{arrives in } I)$$

Using this approach, we can set up a model for any clustering of the original network using only the parameters of the full network. We can thus compute the matrix \mathbf{G} (Eqn. 7 and 8) for each clustered case. We denote such matrices of the clustered model by \mathbf{G}_c .

An additional possibility, which we do not discuss further, is to define a clustered model directly, without relying on the parameters of the full network. In this case, the frequencies, average travel times and stopping probabilities need to be directly measured from data.

2.4 Model considerations

There are a number of important assumptions we used in our model (see also Appendix A). Because we aggregate delays from trains onto stations, we lose a lot of details, such as origin-destination information of trains. In the derivation of the model, a delay ‘arriving’ at a station is subsequently spread out and propagated to all neighbors of that station, based on a fixed weighting of the outgoing edges. However, in real railway systems, there is a high correlation between where the delay comes from and where it goes to, and memory effects can be important. Our model is expected to work better on lower spatial resolution, on scales where a lot of trains and train routes contribute to the dynamics of a single node, such that trains picking a random direction constitute a decent approximation to the real dynamics, which on the detailed level is inherently schedule-based and not random. Furthermore, the delays in our model are treated as variables smoothly varying in time and space. In reality, delays which are localized in space are of a discrete nature: a single train can be delayed, and when the train has ‘passed’ a station, the delay suddenly disappears from this station. This means that the time series of $D_i(t)$



in reality has a lot of jumps, namely every time a train reaches this station or starts towards it. In the model, $D_i(t)$ is smoothly varying. Another important consideration is that the model only propagates delay and removes delay from the system — it does not add any new delays. Moreover, the only mechanism by which delays are removed from the station is when a train ends its trajectory, which is encoded in the parameters s_i . An underlying assumption is thus that each train keeps its initial delay until it has reached its final stop. In practice, of course, delays are constantly generated, often due to small noise-like incidents or other (delayed) trains blocking platforms or tracks, and in more exceptional cases due to new disruptions. Moreover, trains can lose some delay by traveling faster or because of scheduled buffer times at stations, which is not included in our model. For these reasons, we will only compare the results of our model to data of days with a large disruption: by focusing on a time point with a large amount of delay and analyzing its dissipation through the network, we minimize the effects of smaller stochastic delays, which are expected to contribute less to the dynamics in these situations. A final limitation we would like to mention is that in our model, the finite travel time of trains and their location on an edge is lost: in our assumptions, a train's delay counts fully towards to the next station's delay, wherever the train is on an edge towards that station. For small time steps, this means the train's delay also counts immediately to the propagated delay further on in the network, even if in reality the train would still need more time to cross the edge.

Next to the limitations mentioned here, our model also has clear benefits: a compact description (the matrix \mathbf{G}), the fact that it is linear and thus amenable

to analytical study and the straightforward generalization to lower spatial resolution. We discuss advantages of the model throughout, and at the end of this paper.

Some of the limitations mentioned above directly stem from our choice for a network-based, diffusion-like model. It is one of the aims of this paper to investigate whether our model, and its built-in potential for spatial aggregation, can reproduce the dynamics of delay propagation observed in a real railway system.

3 Data and Methods

We apply the model to the Belgian railway system as an example. We chose the Belgian railway system for multiple reasons. Being a West-European country, Belgium has a rather dense and strongly utilised railway system with over 100 km of lines per 1000 m^2 , being one of the world's densest national railway systems [34]. In contrast to, for example, the American or Chinese railways (both have about 10-25 km of railways per 1000 m^2). Additionally, freight and high-speed trains make up only a small fraction of the total railway transport in this country. These aspects require more complex scheduling in the Belgian case, and it implies a more interesting delay evolution to use as an example. Another reason for analysing the Belgian railways is the availability of data, which is discussed below. A discussion on the international relevance of the results is given in the conclusions.

3.1 Data and Pre-processing

We use the open data provided by Infrabel, the service company of the Belgian railway network [35]. The data contains geographical information on railway stations and the physical railway lines, recorded tracks of passenger trains with details on scheduled and realised

departure and arrival times of their activities on each station or junction, as well as associated delays. The time stamps and delay data are in seconds. We use data from all Belgian passenger railway activities between January 2019 until May 2020. The data covers an average number of 3600 daily unique trains on business days, and 2200 on weekends or holidays.

The first step is to reconstruct the graph of the Belgian railway network. First we add all stations as nodes in our graph. We get the edges by mapping the geographical locations of railway stations onto geographical shapes of railway lines and every two stations are connected together iff there is a line connecting them without intermediate stations. The geometry of railway lines is more intricate than simple edges between stations, since there exist places of splits and merges of multiple lines. We implement these by adding so-called “junction” nodes along the lines.

The dataset contains all railway stations, which except of passenger train stations include merchandise platforms, technical depots, carwashes, etc. Passenger trains tend to skip those intermediate platforms and the passage information is not recorded. In order to bypass this limitation, in case when there is no edge between two consecutive stations in the track record, we assume that the train follows the shortest path between them. Delay accumulation or reduction is then evenly spread across the intermediate stations along that path.

There are two kinds of passenger trains available in the data that can be characterised by the proportion of skipped stations along the track: 1) local trains, which usually circulate at shorter distances and stop at every station along the path and 2) intercity trains, which circulate at larger distances and skip some intermediate stations. We exclude from the analysis the intercity trains that skip a significant portion of stations along the track (usually these are international trains) and extra trains that run ad hoc on a specific day. The amount of disregarded trains is less than 3-5% of the total data. We further use the notion of a railway graph and a railway network interchangeably.

The reconstructed network and two important delay statistics are shown in Figure 2. The graph contains 822 stations and 972 edges. Because the network has mostly a line-like structure, 78% of all stations have degree 2 and the average degree is 2.19. In panel Fig. 2a, we show the average delay of trains travelling towards stations in November 2019. A general trend from small average delays in the north-west to larger average delays in the south-east is visible, with the cities of Antwerp (north) and Brussels (centre, the capital) also having rather high average delays. Panel

Fig. 2b colours the edges of the network with the average amount of trains per day that crosses them. Several lines between the large cities of Bruges, Ghent, Brussels and Antwerp stand out.

We use the recorded tracks to estimate the model parameters. In particular, we calculate the edge parameters f_{ij} and t_{ij} and the station parameters s_i (see Table 1) for each month separately. Within a month we aggregate all frequency and temporal counts for each day of the week. Moreover, for each day we keep separate counts for six 4-hour periods of the day. For each station j this leads to the estimation of parameters s_j as the average fraction of arriving trains that end their trajectory at station j , and f_{ij} and t_{ij} , the average frequency and average passage time of trains going from station i to j . For simulations of disrupted situations, we use the parameters obtained for the month, day of the week and period of the day corresponding to the timing of the peak delay on the disrupted days. If not mentioned otherwise, we use $\Delta t = 30$ seconds in all results in this paper. Simulations of the model were coded in Python. For the clustering discussed in 3.4, we used the KMeans function of scikit-learn. The data and code is publicly available and we refer the reader for this to the appropriate section at the very end of the paper.

3.2 Disrupted situations

As discussed in the introduction, we expect diffusion-like models to be of most interest to study large-scale delay propagation: e.g., general directions of delay evolutions — as individual delays will be predicted erroneously due to lack of trajectory information of delays by abstracting away from individual trains. Therefore, we focus our model analysis on days in which such large-scale delay propagation can be assumed important, namely where the delays were severe. In contrast, when delays are small, they dissipate quickly and will not spread much — making identifying large-scale spread of delay of less interest. Another reason why we focus on days with severe delays is that understanding such days is of great importance to railway companies to be able to handle such situations well. We refer to days with severe delays as ‘disrupted days’. A list of disrupted days is obtained by looking at the peak in the total delays (i.e., delay summed over all nodes at any given moment in time) of every day in the dataset, and taking the 50 days with the highest peaks. The exact dates in this list are given in Appendix B. Throughout the rest of the paper, we initialize our simulations at the peak in total delay of these disrupted days, i.e. we determine the delay on each station at the time of peak delay and use this as initial vector of delays. The model captures the spread

and dissipation of existing delays, so these simulations will capture the propagation of the delays present at the peak — making such an initial point most interesting. Also, we reason that after the moment of highest total delay, the relative importance of newly generated (i.e., non-captured) delays is small as compared to existing delays.

3.3 Quantifying model performance

When assessing the model's performance to reproduce reality, we focus on whether the model reproduces the correct direction of delay evolution, rather than simulating exact values well. There are a number of reasons for this. First, when aiming to understand large-scale propagation of severe delays — which is the aim of this model — accurately tracking the positioning of delays (rather than the exact values) through space is already very important information to practitioners. Analysing such directional trends of delays provides us with information on how the system works, absolute values of delays are not always necessary for that. Second, in severely delayed circumstances, numeric performance comparison can quickly become biased by several high spikes in delays: particular trains being up to one hour delayed, compared to an average delay of a couple of minutes in the rest of the network. And third, our model was not designed to capture small, stochastic variations in the delays. However, such delays are always present in the data, which means that one will never get a good quantitative fit, even if the model would be a perfect representation of the propagation of existing delays. For these reasons, we use Spearman's correlation coefficient ρ to measure the model's performance. This metric is based on the rank of the variables, i.e., it assess monotonic relationships rather than linear relationships (which is the case, for example, for Pearson's correlation coefficient). We denote the observed delays at all stations at time t by $\vec{D}_{obs}(t)$: a vector with delay entries per station. Likewise, we define a simulated delay vector $\vec{D}_{sim}(t)$. We denote the vector containing the ranks of the stations based on their delays by $r(\vec{D}(t))$, using either observed or simulated delays. Then, Spearman's correlation coefficient at time t is given by:

$$\begin{aligned} \rho(t) &= \text{Pearson}\left(r(\vec{D}_{obs}(t)), r(\vec{D}_{sim}(t))\right) \\ &= \frac{\text{cov}[r(\vec{D}_{obs}(t)), r(\vec{D}_{sim}(t))]}{\sigma_{r(\vec{D}_{obs}(t))}\sigma_{r(\vec{D}_{sim}(t))}} \end{aligned}$$

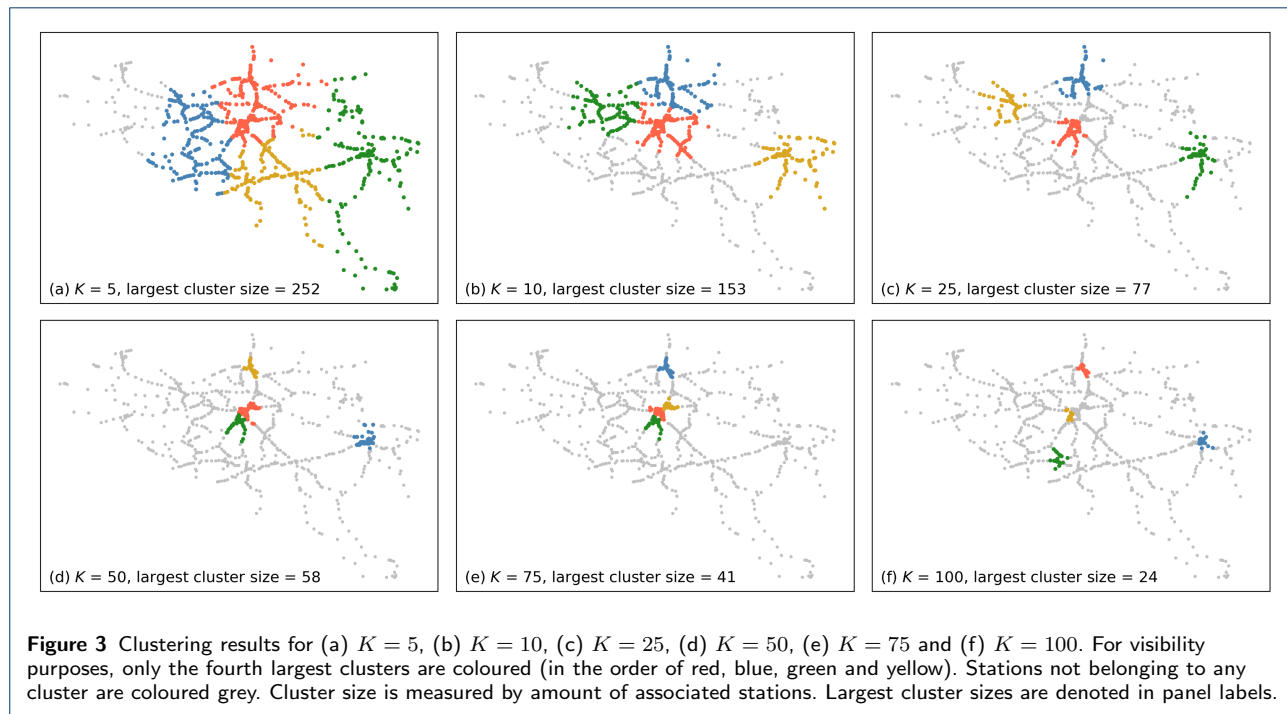
In the next section, we use this metric to compare the model performance on different levels of spatial aggregation. In the clustered case, the vector $\vec{D}_{sim}(t)$

will not have N elements (the number of stations), but $K < N$, the number of clusters. Each component of the vector is the total delay in one cluster. We want to compare this with the observed data on N stations. This observed delay vector therefore also has to be aggregated on the K clusters, by summing the delays of the stations belonging to the same cluster. To be able to compare the Spearman correlations across simulations with different K , the observed and simulated delay vectors of dimension K are de-aggregated towards dimension N (i.e., equally distributed across each cluster's stations), such that we always compute Spearman's correlation on vectors of length N , even if the model was simulated on the network of clusters.

3.4 Clustering

When referring to 'level of aggregation', we mean the spatial resolution of the model. Full resolution would mean using all stations as entities in the model (i.e., no clustering), and lower resolutions involve clusters or groups of stations as entities in the model. Section 2.3 describes how we translate node and edge parameters towards a lower resolution. Here we discuss the means of clustering itself: the process of grouping stations in an appropriate manner. Many of such clustering methods exist, and we have chosen to use K-means [26, 27] on the spatial coordinates of the nodes in the Belgian railway network (longitude, latitude). We do this for the following reasons. First of all, we aim to create groups of stations that are adjacent to each other. Although the way we use K-means does not explicitly incorporate network topology, it does make sure that the groups of stations are convex (i.e., there is no station from cluster A in the middle of cluster B), since the railway network is an inherently spatial network. This geographic basis for the groups also makes them easier to interpret. Another important reason for using K-means is that we can choose K — the desired number of clusters — which we can vary to get different levels of aggregation to assess the model performance with.

We vary K between a minimum amount K_{min} and maximum amount K_{max} of clusters, which in this case we set to be 3 and 100, respectively. Note that values of K_{min} lower than 3 are excluded because of the resulting coarseness of the resulting model, and values of K_{max} higher than 100 are excluded because they result in many single-station clusters. The K-means algorithm starts with an initial set of K points ('centroids') and assigns all stations to the closest centroid. Each centroid now corresponds to a cluster of stations. Next, the centroid coordinates are redefined as the average of all the stations in its cluster. This process is then



iterated (reassigning stations to closet centroid, updating centroid coordinates) until it converges to a point where the centroids do not change anymore.

The resulting clusters for several values K are shown in Fig. 3. In each plot, the four largest clusters in the network are shown in colours. Observe the small size of clusters in the $K = 100$ case, motivating the $K_{max} = 100$ threshold. We can also see that the largest clusters (in terms of number of stations) for high values of K are situated around the major cities of Brussels, Antwerp and Liège. This can be explained by the fact that these cities contain numerous smaller railway stations that are geographically close together, while in more rural areas like the south and west, the station density is much smaller. Urban areas are thus expected to contain larger clusters for relatively large values of K .

4 Results

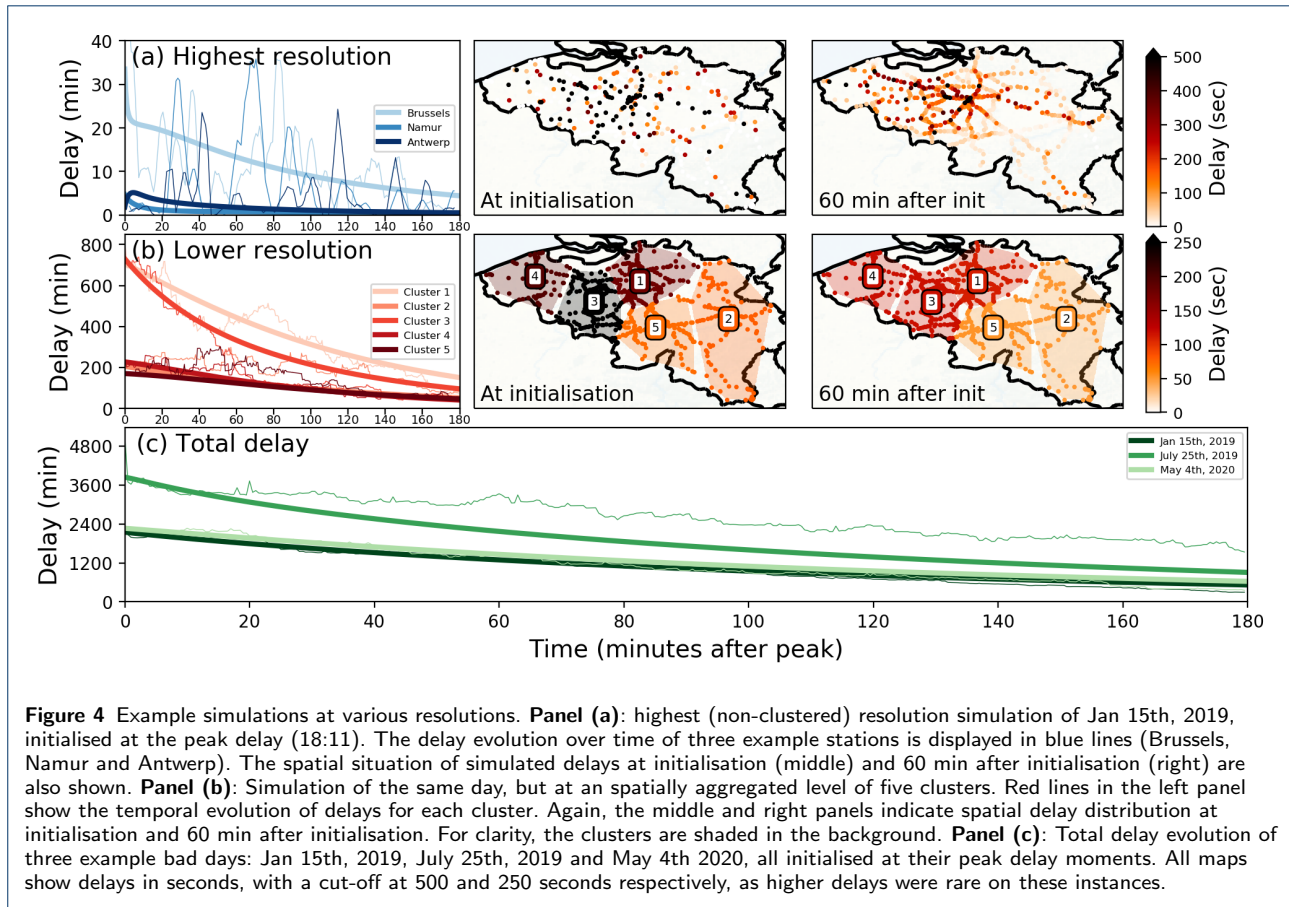
In this section, we show the dynamics of the model and compare it with the data for a number of disrupted days as example. Then, we apply the model to toy examples to illustrate in which circumstances this model works well and in which it does not. We end by discussing the overall performance on all 50 disrupted days.

4.1 Example simulation

We start by looking at a few example disrupted days. We start with Jan 15th, 2019, which had a peak delay

at 18:11. Initialising the non-clustered ('highest resolution') version of the model at this moment, we simulated the delays up to three hours after the peak. The delay evolutions at three major stations (Brussels, Namur and Antwerp) are displayed in Fig. 4a. It is clearly visible that the simulated delay time series is much smoother than the real time series, which has strong jumps as a consequence of the discrete nature of trains: either delayed trains are going to those stations (i.e. delay > 0), or not (i.e., delay $= 0$). This is also visible in the maps in the upper row of this figure: at initialisation time, the delays are distributed very discretely across the network (center-top panel). The model diffuses the delay across the network after 60 min (right-top panel). In this figure, we can clearly see one assumption on which the diffusion-like model is based: it assumes that delay is spread by a very large amount of trains, and that it travels to all other adjacent stations instantly (albeit weighted into small fractions). Of course, in reality this assumption does not hold.

In panel (b) of Fig. 4 we take the exact same day, but instead of modelling at full resolution, we cluster the network into five clusters and redo the analysis. We observe that, by aggregating over the many trains present in each cluster, the jumps in delays visible in panel (a) become less pronounced: the real delay evolution curves per cluster in panel (b) are more smooth. The general trends of the real delay curves in each respective cluster resembles the simulation quite well,



even though some undulations are visible. For example, increases in delays in cluster 1 around 70 minutes and in cluster 5 around 40 minutes after initialisation are visible. This is the result of newly generated delays. Snapshots of the spatial distributions of the delay are shown in the maps on the right. They show how delay dissipates and is transported across the five clusters. A quick comparison by eye between the highest resolution maps (top panels) and these lower resolution maps indicates the resemblance.

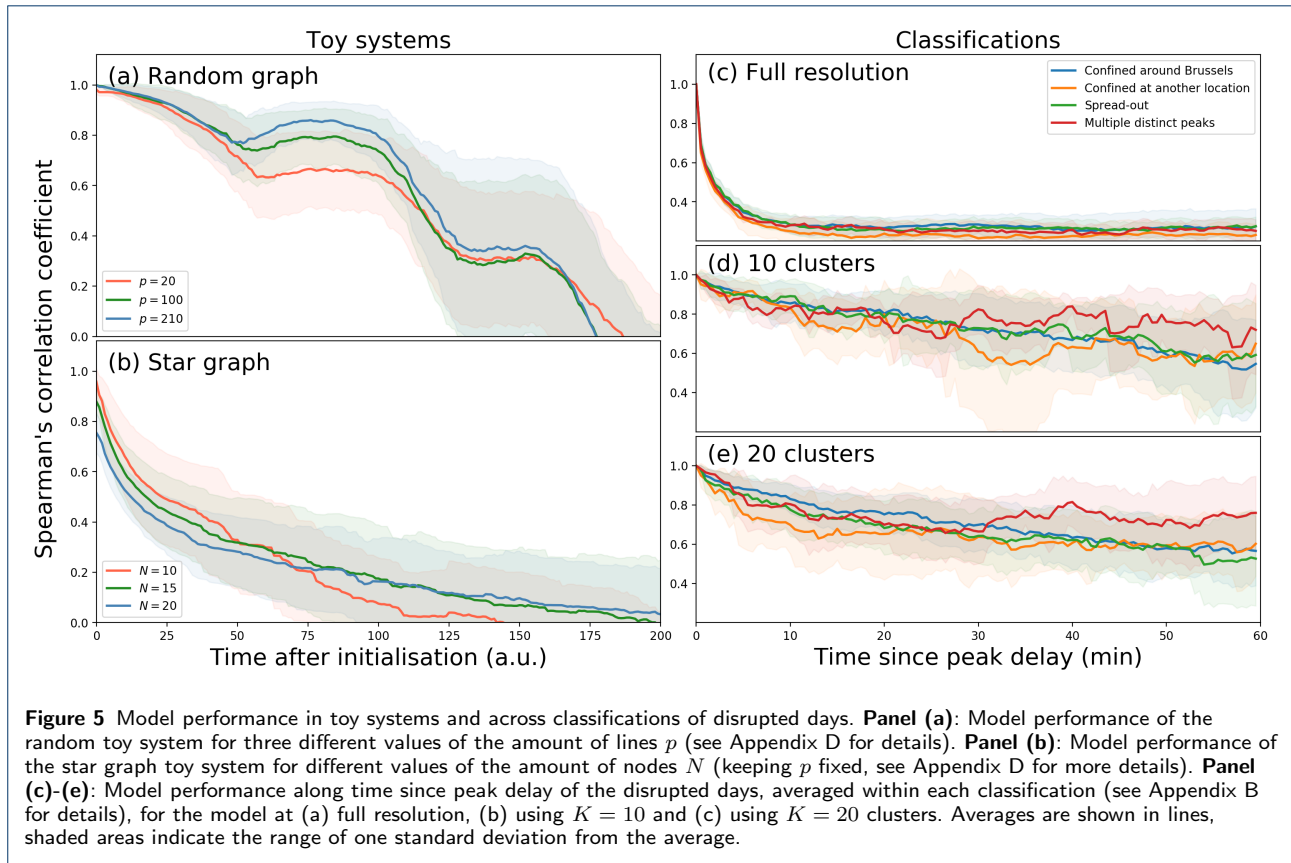
Panel (c) of Fig. 4 shows the evolution of the total delay in the system (both simulated and real) in three example disrupted days: Jan 15th, 2019, July 25th, 2019 and May 4th 2020. One can see that the total delay on Jan 15th and May 4th are simulated quite well over the whole three hours, but the real total delay on July 25th quickly overshoots the simulated curve — pointing towards the effect of newly generated delays.

4.2 Toy model

In this subsection we introduce two toy systems that allow us to study more fundamental properties of the model from section 2. They are explained in more detail in Appendix D. The toy systems represent implementations of the model for networks with very simple

topologies: random networks and star networks. Numerous other toy systems can be thought of, but we specifically compared these because they can test the model performance under different levels of the density of lines and connectivity of nodes. As for the real data we measure the model performance using the Spearman's rank correlation coefficient.

Fig. 5a shows the performance of the model on a random graph-topology (with 15 nodes and 20 edges). We vary the amount of lines p from 20, to 100, up to 210 (which is the maximum number of unique pairs in a 15-node connected graph). It is clear that the model performance decreases over time. At first, the different values of p do not matter: the model performance decreases slightly due to the fact that in the model, delays are instantly spreading to various directions further in the system, while in these systems (and in reality), delays need to arrive at next stations first (carried by trains) before moving onto next stations. This leads to a discrepancy. As soon as the first trains arrive at next stations (around $t = 50$), their delays contribute to delays on new edges where the model already predicted a small part of it to be. For a small amount of lines (i.e., low p), the specific direction the train is go-



ing is very important. For large amounts of lines (i.e., high p), all combined ‘chosen’ directions of the trains approximate the frequency and other attributes put in the model. In other words, the model approximates reality better for densely used lines. And this seems to be visible: high p (blue line) starts deviating positively from the red line after $t = 50$. At much later points, the initial delays start arriving at their ending stations, which collapses the correlation down to much lower values.

For the star graphs (see panel (b) of Fig. 5), where we fix the number of lines $p = 50$ we take a look on the dependence on the number of nodes. The number of nodes do not seem to matter much, but it is clear that the star graph indicates much smaller correlations than the random graph. Although this is merely an example system, we intuitively expect that as soon as trains start driving towards the center, other delays (as a consequence of the diffusion-like nature of the model) are simulated to be at each of the connected nodes, quickly limiting the correlation.

The above toy systems reflect that our model works better for denser networks with the higher number of train lines.

4.3 Classification of disrupted days

We now investigate whether the initial geographical delay distribution has an effect on the accuracy of the diffusion-like model. For this, we distinguish four categories across the 50 disrupted days, classified by eye based on the delay patterns on the peak delay moments. Appendix A discusses this classification in more detail and also shows the delay maps. The first and largest group (25 days) contains the situations where almost all of the delays are localized near Brussels, the capital city of Belgium and important railway hub. In Belgium, train lines between east and west and north and south respectively all pass through Brussels, which makes it an important factor in the delay dynamics in the railway network. The second group (7 days) contains situations where the delay is also localized, but on a different location than Brussels. The third group (5 days) contains those situations with multiple locations with high delays. Finally, we consider the group of stations (13 days) where the delay is not localized but instead spread out over a large region.

As before, we perform simulations with as initial condition the peak delay distribution. In Fig. 5c-e we show the evolution of the Spearman correlation over time, averaged per group. We show this for different spatial

resolutions. We find that there are no clear differences between the groups. The situations with delays localized near a city which is not Brussels (shown in orange) seem to perform a bit worse than the others, but we should be very cautious in interpreting this: the variation within a group is very large, as shown in the shaded areas.

The fact that there is no clear difference in model performance between groups could indicate that the spatial localization of the disruption is not a good determinant of the accuracy of our diffusion-like model. The obvious question then is: is there a better measure, or characteristic, which can distinguish between different disrupted situations and indicates whether a diffusion-like delay spread is warranted? We plan to explore this in future work.

4.4 Overall performance

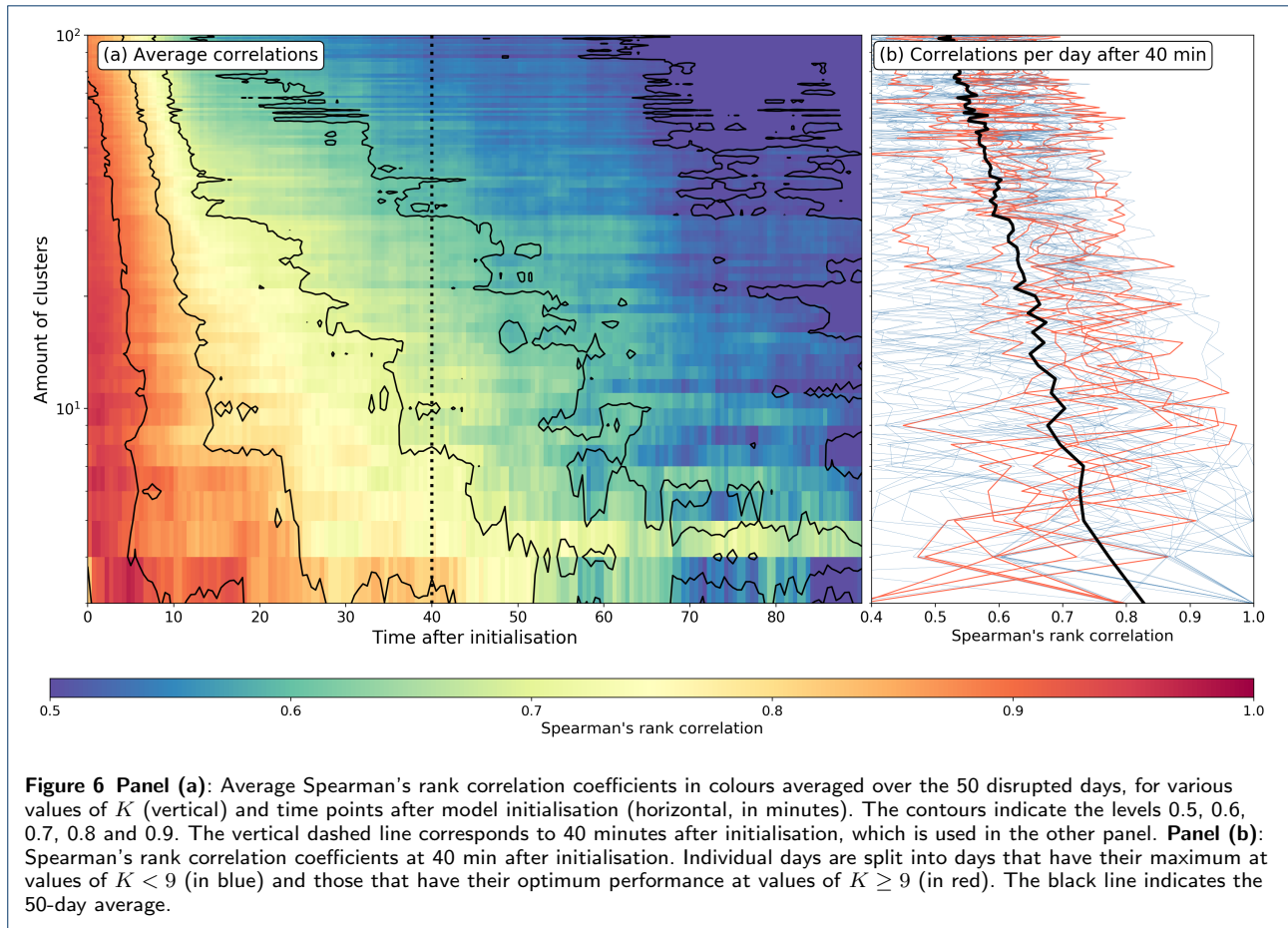
We now turn to the overall performance of the model over the 50 disrupted days. On each of these days, we determine the peak in the total delay and simulate the delays up to two hours after this peak. We then compare what really happened throughout these two hours to what we simulated by computing the Spearman's correlation coefficient ρ at each time point (see Eqn. 13). We do this for each number of clusters K ($3 \leq K \leq 100$). The average correlations per K and t over the 50 disrupted days is shown in Fig. 6a. It is clear that in general, the higher t , the lower the correlation. This is intuitively correct: the higher the time after initialisation, the more the model will start differing from reality, for example due to new incoming delays in the real data that are not captured by the model or errors the model that grow with time. In the same panel, we see that higher amount of clusters K also decreases the correlation, which is less obvious. On the one hand, information is lost when coarse graining: for lower values of K detailed information on the positioning of the trains is put together into larger clusters, which may reduce their simulation quality. On the other hand, the diffusion-like spreading is presumably more accurate when looking at a larger scale (lower K), since on these scales the discreteness of delays is averaged out in the data, too. Interestingly, panel (a) also shows bands of K values with near-equal correlations: up to $K = 8$, the correlations seem to be more or less the same (very high), at least up to $t = 45$ min. The second band of near-equal correlations is between $8 \leq K \leq 17$, followed by a more gradual decay of correlations with K , but a sharp decrease in those correlations at $K \approx 27$. One reason for these sudden correlation decreases could be a strong rearranging in the clustering at those K values: e.g., in Fig. 3, panels (a) and (b), one can see that for $K = 5$, Brussels is at

the border of the red cluster, while at $K = 10$, it is in fact in the middle of a cluster. Such rearranging can be quite sudden from one value of K to another. In contrast, the slow decrease in correlation *within* those K -bands can be related to a slow change in the clustering structure.

Panel (b) in Fig. 6 shows the correlation ρ 40 minutes after model initialisation, as function of K , for each individual disrupted day. Clearly, these curves seem less gradual as the average displayed in panel (a). In fact, changing K by 1 may impact the correlation up to 0.5 in some exceptional cases — specifically when K is small (which makes sense as the clustering structure changes rapidly around these values). The average is displayed in black, and the gradual decrease with K is visible. Red, thin lines indicate days on which the maximum correlation is at values $K \geq 9$, which is counteracting the overall pattern we see that the correlation keeps increasing with decreasing K — on these seven days, the model performs best on a non-trivial level of K , between 9 and 11. Appendix C discusses this non-trivial level (i.e., $K = 10$) more in-depth and potential reasons why this clustering configuration may be optimal in some circumstances. The blue lines indicate all other days (where optimal correlations are found with very low values of K).

4.5 Model discussion

Our results show that modelling delay as a diffusion-like spreading phenomenon clearly has limitations: on the scale of individual stations (Fig. 4a), the discrete nature of delays, accompanied with periodic undulations and sharp changes in delay is not simulated at all. Also, delays bound to trains usually travel in a specified direction, which is not captured by our model (which is a weighted form of unidirectional spreading). The diffusion-like-spreading assumption corresponds to the view that the delay propagation is based not on single trains carrying a delay, but on many tiny trains, all going in various possible directions with their portion of the delay, which are randomly chosen, weighted by the variables in the model (e.g., frequency and travel times). Thinking about the diffusion-like model in this way motivates the use of coarse graining to improve the model. Qualitatively we show this in Fig. 4b and c and quantitatively this is discussed in Fig. 6: a clear increase in performance is visible when comparing results from the clustered version of the model to clustered data. Still, there is a loss of correlation with simulation time: the further from the initialisation time, the more new trains and delays enter the system and change the real delay time series, unaccounted for by the model. It is important to note that



the latter will always be a caveat for delay propagation models due to the inherent stochasticity of delay generation.

The toy systems we tested are meant as illustrations to indicate what the model benefits from: average patterns. As soon as the network is dense and there are many trains travelling on it, the real delays spread roughly along simple statistics like train frequencies, which are the basis of this model. But as soon as the network becomes more sparse, especially when it becomes tree-like, the correlation drops.

Our discussion on the classification of disrupted situations showed no clear differences in model performance based on the initial condition, at least as far as its localization (which is the basis of the classification) is concerned. However, our approach was naive: we classified the disrupted days by eye into four groups. We cannot conclude that there are no other, better metrics that do distinguish situations in which the delays spread in a more diffusion-like manner than in other situations — something we do find in the toy examples. Hence, we propose to investigate such metrics further in future work.

The high performance at low values of K implies that a coarse resolution is better suited for these type of models. The disadvantage of that is the loss of detail. Also, as shown in the toy examples, there are cases we can think of that are not suitable to be modelled well by the model: high sparsity of trains increases the discrete nature of delays and decreases the applicability of the mean-field approximation. Another example where these models have low accuracy is when the delays are mainly governed by stochasticity, and not by propagation dynamics. This is the case in situations where the overall delay level in the network is low. Such situations are difficult to capture well in many delay propagation models, in fact.

We propose therefore that the model presented here finds its niche in the problem of simulating the propagation of severe delays on a large scale. In such circumstances, the exact magnitude of delays at fixed positions is not always of most interest, while the general trend, speed of delay decay and direction of the overall bulk of delay are of high importance. Such information is well retrievable from the clustered model. In fact, this model is arguably very suited to analyse these

large-scale dynamics and how they depend on network topology and high-level parameters such as train frequencies. All information of the system's dynamics is embedded in the \mathbf{G} -matrix (Eqn. 8) — a single matrix that can be analysed using spectral methods to investigate its eigenproperties, for example. Another advantage of this model is its simplicity. Using only a small set of parameters that are easily retrievable from the schedules (which can usually be found online for any European railway system), one can model the whole railways with a single simple differential equation (Eqn. 7).

5 Conclusions

In summary, we devised a model that simulates delays as a diffusion-like spreading phenomenon. The intuition is that, on average, the direction and dissipation of delay relates to aspects of the schedule such as train frequencies and travel times. We apply the model to the Belgian railways and investigate its strengths and weaknesses. In particular, we find that the model performance increases sharply when coarse graining by grouping stations together into clusters. We conclude that this model is mainly of use when working on larger scales and aiming to identify system properties related to delay dissipation and general directions of delay propagation, rather than accurate individual prediction.

We have illustrated the workings and performance of the model using the Belgian railway network as an example. Our framework, however, is general and could be applied to any country. Nevertheless, there are some international differences that should be taken into account. One aspect which may influence the performance of the model in different countries, is the position of important railway hubs. Hubs, such as large cities, are expected to play an important role in the delay propagation dynamics. In Belgium, these hubs are well distributed across the country, apart from the more rural areas in the south-east and far-west. This means that, in our geographic clustering, the hubs usually fall into different clusters. The Netherlands, in contrast, has its most important cities concentrated in the west of the country. In a model with a small number of clusters, it is possible that many of these hubs end up in the same cluster, which may have an unwanted effect on the model's performance and usefulness. For this reason, it might be important to consider other clustering methods.

Not only geographic differences should be considered when trying to extrapolate these results internationally. Various factors impacting delay propagation vary from country to country, like policy, protocols, infrastructure details and delay statistics in general

[36]. However, we argue that the increase in model performance when coarse graining is robust to these changes, because the reason for it is of a more theoretical nature: diffusion-like spreading captures average delay fluxes, which are more prominent in clustered systems.

Applying the model to other countries is straightforward, since its ingredients are general and easily obtained: train turnover rates, frequencies, travel times and adjacency matrices are readily derived from network architecture and railway schedules. A natural extension of our work would thus be to compare the network models for different countries and explore the properties in different spatial resolutions.

All of the model's dynamics are essentially derived from the matrix \mathbf{G} , also for the clustered versions. Exploring the spectral and topological properties of the weighted network that \mathbf{G} describes and relating those properties to the dynamics of the railway system are of interest in our future work. It is possible that a few simple metrics, derived from this matrix, could be used for a quick international comparison of railway networks.

Despite its simplicity, this model already gives us a tool to better understand railway network dynamics. However, there are a number of improvements to be made in future work. For example, it is possible to add noise to the model, to account for the generation of new delays. The magnitude and distribution of stochastically generated delays at different stations can possibly be derived from the data. It would be interesting to see how noise may lead to an 'equilibrium' delay distribution, in contrast to the highly disrupted situations we considered in this paper.

Once delay is generated in a real railway system, there are mechanisms and feedback loops that can amplify it or mitigate it. Such mechanisms are not present in the model, but would be a valuable addition. Such feedbacks can be nonlinear, complicating the model but possibly generating new dynamics. Finally, our model is 'first-order': the spread of delay is determined only by where it is, not where it came from. Including a memory mechanism will probably increase the accuracy of the model, but may also make it more complex. One way to do this is to not consider delay on stations, but instead on edges. In such a version of the model, the directional information of trains is partially kept, contrary to in our station-based model.

Diffusion-like spreading is researched in many other fields other than transport literature. In particular, it is well established that the vulnerability versus perturbations of networked systems including social, epidemiological and engineering systems depends on (any quantification of) the modularity [24]: a more modular dynamical structure prevents large-scale spreading

[37, 2]. Determining the modularity from the \mathbf{G} -matrix can be an interesting next step, as the \mathbf{G} -matrix does not only incorporate topology, but weights each edge by features from the schedules.

The results in this paper are not only of interest for modellers, but also for railway practitioners. First, the model output can provide insights into system-wide properties, like delay decay and general directions of delay propagation. Second, it is easy to use and all information is embedded in the \mathbf{G} matrix. For example, practitioners might be interested in how isolated regions are from each other: the off-diagonal elements of the \mathbf{G} matrix at the appropriate level of coarse graining reflects how strongly regions are connected, i.e. how much delay flows from one region to another. From an operational point of view, optimal levels of clustering like those seen for the red curves in Fig. 6b (see also Appendix C) can be used to categorise situations, issue protocols or form threat assessments in terms of delays.

We hope that the model itself and the results of the application to Belgium motivates researchers and practitioners to vary the spatial aggregation level to non-trivial levels. We believe these diffusion-like models can offer useful insights on how aspects such as network structure, basic schedule parameters and spatial resolution affect the delay propagation through a railway network. Ultimately, such models can lead to a better understanding of railway delay dynamics.

Availability of data and materials

The Belgian railway delay data for the period January 2019 to May 2020 is available under XYZ. The code for the model and associated \mathbf{G} matrix is available under XYZ. Both will be available upon publication.

Competing interests

The authors declare that they have no competing interests.

Author's contributions

All authors designed research and investigated the topic. The whole work was coordinated by MMD. ANM acquired, processed and managed data. All authors discussed the theoretical part of the model. Formal derivation of the model was made by JR and solved and formally analyzed by LT, JR and GS. ANM and JR wrote the model implementation and then the whole team worked on the simulations. MMD wrote first draft of the manuscript and all authors reviewed and edited the final text.

Acknowledgements

The authors gratefully thank the organisers of the Winter Workshop on Complex Systems 2019, where this project started. They also thank Debabrata Panja for his remarks on the manuscript and thank Matthew Garrod and Maria Walldl for their input at the start of this project. This work is part of the research programme 'Improving the resilience of railway systems' with project number 439.16.111. MMD was supported by this research project, which is financed by the Dutch Research Council (NWO) and co-financed by Nederlandse Spoorwegen (NS) and ProRail. JR was supported by an individual fellowship from the Research Foundation Flanders (FWO). ANM was supported by F.R.S-FNRS grant PDR T.0065.19 Collective Footprints and the grant 19-01-00682 of the Russian Foundation for Basic Research.

Author details

¹m.m.dekker@uu.nl, Department of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands. ²Centre for Complex Systems Studies, Utrecht University, Utrecht, The Netherlands. ³alexey.medvedev@unamur.be, ICTEAM, Université catholique de Louvain, Louvain-la-Neuve, Belgium. ⁴jan.rombouts@kuleuven.be, Laboratory of Dynamics in Biological Systems, KU Leuven, Leuven, Belgium. ⁵grzegorz.siuDEM@pw.edu.pl, Warsaw University of Technology, Faculty of Physics, Warsaw, Poland. ⁶liubov.tupikina@cri-paris.org, Center for Research and Interdisciplinarity (CRI), Université de Paris, Paris, France. ⁷Nokia Bell labs, Paris, France.

References

- Zieger, S., Weik, N., Nießen, N.: The influence of buffer time distributions in delay propagation modelling of railway networks. *Journal of Rail Transport Planning and Management* **8**(3-4), 220–232 (2018). doi:[10.1016/j.jrtpm.2018.09.001](https://doi.org/10.1016/j.jrtpm.2018.09.001)
- Dekker, M.M., Panja, D.: Cascading dominates large-scale disruptions in transport over complex networks. *PLOS ONE* **16**(1), 1–17 (2021). doi:[10.1371/journal.pone.0246077](https://doi.org/10.1371/journal.pone.0246077)
- Ludvigsen, J., Klæboe, R.: Extreme weather impacts on freight railways in Europe. *Natural Hazards* **70**(1), 767–787 (2014). doi:[10.1007/s11069-013-0851-3](https://doi.org/10.1007/s11069-013-0851-3)
- Tsuchiya, S., Tatano, H., Okada, N.: Economic loss assessment due to railroad and highway disruptions. *Economic Systems Research* **19**(2), 147–162 (2007). doi:[10.1080/09535310701328567](https://doi.org/10.1080/09535310701328567)
- Büchel, B., Spaninger, T., Corman, F.: Empirical dynamics of railway delay propagation identified during the large-scale rastatt disruption. *Scientific Reports* **10**(1), 18584 (2020). doi:[10.1038/s41598-020-75538-z](https://doi.org/10.1038/s41598-020-75538-z)
- Dekker, M.M., van Lieshout, R.N., Ball, R.C., Bouman, P.C., Dekker, S.C., Dijkstra, H.A., Goverde, R.M.P., Huisman, D., Panja, D., Schaafsma, A.M., van den Akker, M.: A Next Step in Disruption Management: Combining Operations Research and Complexity Science. In: *Conference on Advanced Systems in Public Transport (CASPT) 2018*, pp. 1–19 (2018). <https://repub.eur.nl/pub/109054/>
- Kecman, P., Goverde, R.M.P.: Predictive modelling of running and dwell times in railway traffic. *Public Transport* **7**(3), 295–319 (2015). doi:[10.1007/s12469-015-0106-7](https://doi.org/10.1007/s12469-015-0106-7)
- Li, D., Daamen, W., Goverde, R.M.P.: Estimation of train dwell time at short stops based on track occupation event data: A study at a Dutch railway station. *Journal of Advanced Transportation* **50**(5), 877–896 (2016). doi:[10.1002/atr.1380](https://doi.org/10.1002/atr.1380)
- Dekker, M.M., Panja, D., Dijkstra, H.A., Dekker, S.C.: Predicting transitions across macroscopic states for railway systems. *PLOS ONE* **14**(6), 0217710 (2019). doi:[10.1371/journal.pone.0217710](https://doi.org/10.1371/journal.pone.0217710)
- Oneto, L., Fumeo, E., Clerico, G., Canepa, R., Papa, F., Dambra, C., Mazzino, N., Anguita, D.: Train Delay Prediction Systems: A Big Data Analytics Perspective. *Big Data Research* **11**, 54–64 (2018). doi:[10.1016/j.bdr.2017.05.002](https://doi.org/10.1016/j.bdr.2017.05.002)
- Monechi, B., Gravino, P., Di Clemente, R., Servedio, V.D.P.: Complex delay dynamics on railway networks from universal laws to realistic modelling. *EPJ Data Science* **7**(1) (2018). doi:[10.1140/epjds/s13688-018-0160-x](https://doi.org/10.1140/epjds/s13688-018-0160-x). [1707.08632](https://arxiv.org/abs/1707.08632)
- Sen, P., Dasgupta, S., Chatterjee, A., Sreeram, P.A., Mukherjee, G., Manna, S.S.: Small-world properties of the Indian railway network. *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics* **67**(3), 5 (2003). doi:[10.1103/PhysRevE.67.036106](https://doi.org/10.1103/PhysRevE.67.036106). [0208535](https://arxiv.org/abs/0208535)
- Erath, A., Löchl, M., Axhausen, K.W.: Graph-theoretical analysis of the swiss road and railway networks over time. *Networks and Spatial Economics* **9**(3), 379–400 (2009). doi:[10.1007/s11067-008-9074-7](https://doi.org/10.1007/s11067-008-9074-7)
- Bhatia, U., Kumar, D., Kodra, E., Ganguly, A.R.: Network science based quantification of resilience demonstrated on the indian railways network. *PLoS ONE* **10**(11), 0141890 (2015). doi:[10.1371/journal.pone.0141890](https://doi.org/10.1371/journal.pone.0141890). [1508.03542](https://arxiv.org/abs/1508.03542)
- Goverde, R.M.P.: A delay propagation algorithm for large-scale railway traffic networks. *Transportation Research Part C: Emerging Technologies* **18**(3), 269–287 (2010). doi:[10.1016/j.trc.2010.01.002](https://doi.org/10.1016/j.trc.2010.01.002)
- Gambardella, L.M., Rizzoli, A.E., Funk, P.: Agent-based Planning and Simulation of Combined Rail/Road Transport. *SIMULATION* **78**(5), 293–303 (2002). doi:[10.1177/0037549702078005551](https://doi.org/10.1177/0037549702078005551)

17. Büker, T., Seybold, B.: Stochastic modelling of delay propagation in large networks. *Journal of Rail Transport Planning and Management* **2**(1-2), 34–50 (2012). doi:[10.1016/j.jrtpm.2012.10.001](https://doi.org/10.1016/j.jrtpm.2012.10.001)
18. Harrod, S., Cerreto, F., Nielsen, O.A.: A closed form railway line delay propagation model. *Transportation Research Part C: Emerging Technologies* **102**, 189–209 (2019). doi:[10.1016/j.trc.2019.02.022](https://doi.org/10.1016/j.trc.2019.02.022)
19. Dekker, M.M., Panja, D.: A reduced phase-space approach to analyse railway dynamics. *IFAC-PapersOnLine* **52**(3), 1–6 (2019). doi:[10.1016/j.ifacol.2019.06.001](https://doi.org/10.1016/j.ifacol.2019.06.001). 15th IFAC Symposium on Large Scale Complex Systems LSS 2019
20. Oneto, L., Fumeo, E., Clerico, G., Canepa, R., Papa, F., Dambra, C., Mazzino, N., Anguita, D.: Dynamic delay predictions for large-scale railway networks: Deep and shallow extreme learning machines tuned via thresholdout. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **47**(10), 2754–2767 (2017). doi:[10.1109/TSMC.2017.2693209](https://doi.org/10.1109/TSMC.2017.2693209)
21. Corman, F., Kecman, P.: Stochastic prediction of train delays in real-time using Bayesian networks. *Transportation Research Part C: Emerging Technologies* **95**, 599–615 (2018). doi:[10.1016/j.trc.2018.08.003](https://doi.org/10.1016/j.trc.2018.08.003)
22. Berger, A., Gebhardt, A.: Stochastic delay prediction in large train networks. 11th Workshop on, 100–111 (2011). doi:[10.4230/OASlcs.ATMOS.2011.100](https://doi.org/10.4230/OASlcs.ATMOS.2011.100)
23. Fortunato, S., Hric, D.: Community detection in networks: A user guide. *Physics Reports* **659**, 1–44 (2016). doi:[10.1016/j.physrep.2016.09.002](https://doi.org/10.1016/j.physrep.2016.09.002). Community detection in networks: A user guide
24. Newman, M.E.J.: Modularity and community structure in networks. *Proceedings of the National Academy of Sciences of the United States of America* **103**(23), 8577–82 (2006). doi:[10.1073/pnas.0601602103](https://doi.org/10.1073/pnas.0601602103)
25. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* **2008**(10) (2008). doi:[10.1088/1742-5468/2008/10/P10008](https://doi.org/10.1088/1742-5468/2008/10/P10008). 0803.0476
26. Steinhaus: Sur la division des corps matériels en parties. *Bulletin de l'Académie Polonaise des Sciences, Classe III* **IV**(12), 801–804 (1956)
27. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, pp. 281–296. The Regents of the University of California, ??? (1967). <https://projecteuclid.org/euclid.bsm/1200512992>
28. Lloyd, S.: Least squares quantization in pcm. *IEEE Transactions on Information Theory* **28**(2), 129–137 (1982). doi:[10.1109/TIT.1982.1056489](https://doi.org/10.1109/TIT.1982.1056489)
29. Kadir, R.A., Shima, Y., Sulaiman, R., Ali, F.: Clustering of public transport operation using K-means. In: *2018 IEEE 3rd International Conference on Big Data Analysis, ICBDA 2018*, pp. 427–432. Institute of Electrical and Electronics Engineers Inc., ??? (2018). doi:[10.1109/ICBDA.2018.8367721](https://doi.org/10.1109/ICBDA.2018.8367721)
30. Cerreto, F., Nielsen, B.F., Nielsen, O.A., Harrod, S.S.: Application of Data Clustering to Railway Delay Pattern Recognition. *Journal of Advanced Transportation* **2018** (2018). doi:[10.1155/2018/6164534](https://doi.org/10.1155/2018/6164534)
31. Gao, J., Buldyrev, S.V., Havlin, S., Stanley, H.E.: Robustness of a network of networks. *Phys. Rev. Lett.* **107**, 195701 (2011). doi:[10.1103/PhysRevLett.107.195701](https://doi.org/10.1103/PhysRevLett.107.195701)
32. Kivela, M., Arenas, A., Barthelemy, M., Gleeson, J.P., Moreno, Y., Porter, M.A.: Multilayer networks. *Journal of complex networks* **2**(3), 203–271 (2014)
33. Siudem, G., Hofyst, J.A.: Diffusion on hierarchical systems of weakly-coupled networks. *Physica A: Statistical Mechanics and its Applications* **513**, 675–686 (2019). doi:[10.1016/j.physa.2018.08.078](https://doi.org/10.1016/j.physa.2018.08.078)
34. International Union of railways: *Synopsis 2020* (2020). <https://uic.org/support-activities/statistics/> Accessed 2021-03-25
35. Infrabel: Open Data. <https://infrabel.opendatasoft.com/> Accessed 2021-03-25
36. Schipper, D., Gerrits, L.: Differences and similarities in European railway disruption management practices. *Journal of Rail Transport Planning and Management* **8**(1), 42–55 (2018). doi:[10.1016/j.jrtpm.2017.12.003](https://doi.org/10.1016/j.jrtpm.2017.12.003)
37. Balcan, D., Colizza, V., Gonçalves, B., Hud, H., Ramasco, J.J., Vespignani, A.: Multiscale mobility networks and the spatial spreading of infectious diseases. *Proceedings of the National Academy of Sciences of the United States of America* **106**(51), 21484–21489 (2009). doi:[10.1073/pnas.0906910106](https://doi.org/10.1073/pnas.0906910106). 0907.3304

Supplementary Information

Appendix A — Model derivations

Contains important model considerations, like the definition of delays, the derivations of the terms $F_{1,i}$ and $F_{2,i}$ in Eqn. 1 and analytical solutions of the model.

Appendix B — Disrupted days

Contains a list of the 50 disrupted days and plots of the geographic distribution of delays at the moment of peak delay.

Appendix C — Non-trivial optimal level of K

Discusses the case of $K = 9$, which turns out to be an optimal level of spatial aggregation for a subset of disrupted days.

Appendix D — Toy models

Describes the algorithm used to generate toy examples in Fig. 5 and provides example topologies.