Dekker et al. *EPJ Data Science* (2022) 11:44
https://doi.org/10.1140/epjds/s13688-022-00359-1

EPJ.org

EPJ Data Science
a SpringerOpen Journal

REGULAR ARTICLE Open Access

Modelling railway delay propagation as diffusion-like spreading

Mark M. Dekker[1,2,3*], Alexey N. Medvedev[4], Jan Rombouts[5], Grzegorz Siudem[6] and Liubov Tupikina[7,8]

*Correspondence:
m.m.dekker@uu.nl
[1]Department of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands
[2]Centre for Complex Systems Studies, Utrecht University, Utrecht, The Netherlands
Full list of author information is available at the end of the article

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, non-trivial optimal levels of spatial resolution are 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 [3–6].

The above shows the importance of better understanding 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 analysed 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. analysed the German and Italian railways and found a set of 'laws' that drive the spreading of delays [11] and Gurin et al. simulate train delay propagation using modified SIR models [12], both containing analogies to epidemic spreading models. Also beyond railways, models with contagion processes are used to simulate transportation and congestion [13].

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 simulations. However, the role of railway network topology is addressed by various scholars in relation to resilience properties [14–16]. 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 [2, 17–20], 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 [21]. 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 exactly are 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 [10, 22] or probability updating [23, 24]. 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 much more complex, due to potential long-range correlations.

In this paper we propose to treat delays not as bound to 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 Sect. 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. 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 [11, 16, 17]. 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 [25]. 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 [26]. Various clustering algorithms based on modularity optimisation exist, such as the Louvain method [27].

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 [28, 29], which defines centroids and groups nodes based on their respective distances to these cluster centroids (also known as Voronoi iteration [30]), 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) [31, 32]. 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. In particular, because of the heterogeneity of primary delays in the railway system, we focus on how this type of modelling can be used to predict the direction of delay propagation after a delay peak. We introduce the model that implements diffusion-like spreading in Sect. 2. We apply the model to the example case of the Belgian railways and discuss the data and methodology for this in Sect. 3. Section 4 discusses the model's performance, both overall and on different types of disrupted situations. In this section we also describe the results of a toy model. 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. In this section, we also outline several possible extensions of the model. We end with a summary and several conclusive remarks in Sect. 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 Additional file 1, Appendix A. Table 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

Table 1 Overview of the model variables and parameters

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)

is defined as the sum of the delays of all trains that are moving *towards* station i at time t:

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

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 at 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:

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

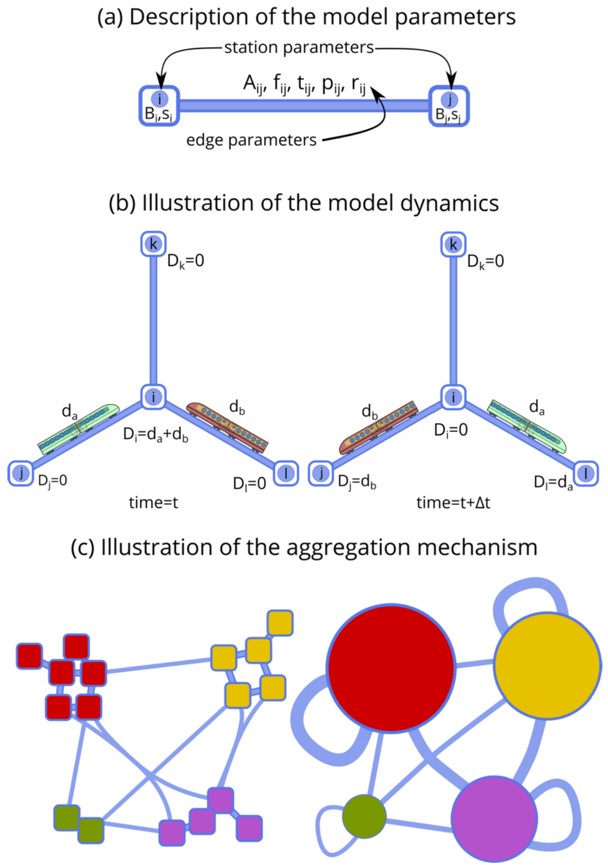
with $F_{1,i}$ describing how the delay at station i changes over a time step Δt by means of the first term above (the addition of delay), and $F_{2,i}$ likewise by the second term above (the removal of delay). In the next section we express both these functions $F_{1,i}$ and $F_{2,i}$ in terms of several 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


(a) Description of the model parameters

(b) Illustration of the model dynamics

(c) Illustration of the aggregation mechanism

Figure 1 Model visualization: (**a**) station and line (edge in the network) dependent parameters, (**b**) illustration of the two mechanisms behind the delay dynamics dynamics, i.e. the appearance of new trains with their delays and the departure of the already included ones, (**c**) an example of the network aggregation

frequency (a full derivation can be found in Additional file 1, 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:

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

Here $\mathcal{N}_{\text{in}}(i)$ 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:

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

where $\mathcal{N}_{\text{out}}(j)$ 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 the train does not end at j — s_j itself 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

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

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 Additional file 1, Appendix A):

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

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 Table 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 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 $\Delta t \to 0$. This leads to the expression

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

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

$$G_{ij} = A_{ji}p_{ji}B_j - \delta_{ij}B_j. \tag{8}$$

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

Let us also point out that the matrix G incorporates the topological structure of the network in combination with other parameters of the edges (see Fig. 1 and Table 1). The above allows us to describe the averaged dynamics of delays using averaged values of the parameters. While the analytical analysis of $\mathbf{G}$ is not the primary aim of this paper, as an example we have plotted this matrix in an aggregated form (see next section) in Additional file 1, Appendix C.

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 Sect. 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 [33–35].

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 Table 1). We 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

$$f_{IJ} = \sum_{i\in I}\sum_{j\in J} f_{ij}, \tag{9}$$

$$t_{IJ} = \frac{\sum_{i\in I}\sum_{j\in J} t_{ij}f_{ij}}{\sum_{i\in I}\sum_{j\in J} f_{ij}}. \tag{10}$$

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:

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

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:

$$s_I = \sum_{i\in I} s_i q_i. \tag{12}$$

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 Additional file 1, 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 analysing 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 **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 m of lines per km^2, being one of the world's densest national railway systems [36]. In contrast to, for example, the American or Chinese railways (both have about 10-25 m of railways per km^2). Additionally, freight and high-speed trains make up only a small fraction of the total railway transport in this country, which have a. 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 [37]. 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 if 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 besides passenger train stations also 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 Fig. 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. 2(a), 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. 2(b) 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

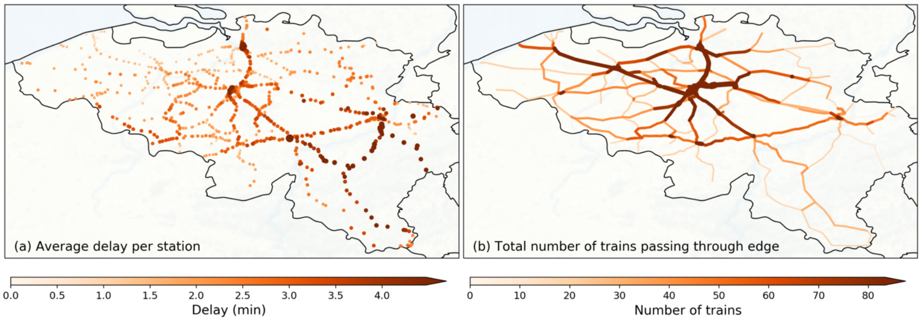


Figure 2 *Panel* (**a**): Average delay per train in November 2019, shown at every node. *Panel* (**b**): Average number of trains passed through the edge on April 11th, 2019 (taken as an example day). Only passenger trains are used when calculating these numbers

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 Δt = 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 — individual delays will be predicted erroneously due to the averaging assumptions in the model. 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 Additional file 1, Appendix B. For simulations of these disrupted days, we initialize our simulations at the peak in total delay, i.e. we determine the delay on each station at the time of peak delay and use this as initial vector of delays. We then use the model to describe the spread and dissipation of the delays present at the peak. We reason that after the moment of highest total delay, the relative importance of newly generated delays — which are not captured by the model — is small compared to existing delays.

3.3 Quantifying model performance

When assessing the model's ability 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 position of delays (rather than their exact value) 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 assesses 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 $\vec{D}_{\text{obs}}(t)$: a vector with delay entries per station. Likewise, we define a simulated delay vector $\vec{D}_{\text{sim}}(t)$. 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:

$$\begin{aligned}\rho(t) &= \text{Pearson}\big(r\big(\vec{D}_{\text{obs}}(t)\big), r\big(\vec{D}_{\text{sim}}(t)\big)\big) \\ &= \frac{\text{cov}[r(\vec{D}_{\text{obs}}(t)), r(\vec{D}_{\text{sim}}(t))]}{\sigma_{r(\vec{D}_{\text{obs}}(t))}\sigma_{r(\vec{D}_{\text{sim}}(t))}}.\end{aligned} \tag{13}$$

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 $\vec{D}_{\text{sim}}(t)$ 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, and moreover, we want to compare this for different K. We do this as follows: the observed delay vector is first aggregated onto K clusters, by summing the delays of the stations belonging to the same cluster. Next, these are redistributed to N stations by equidistributing a cluster's delay over its stations. The simulated delay is distributed over the N stations in the same way. In this way, 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. The highest 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 [28, 29] 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 van 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 closest 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.

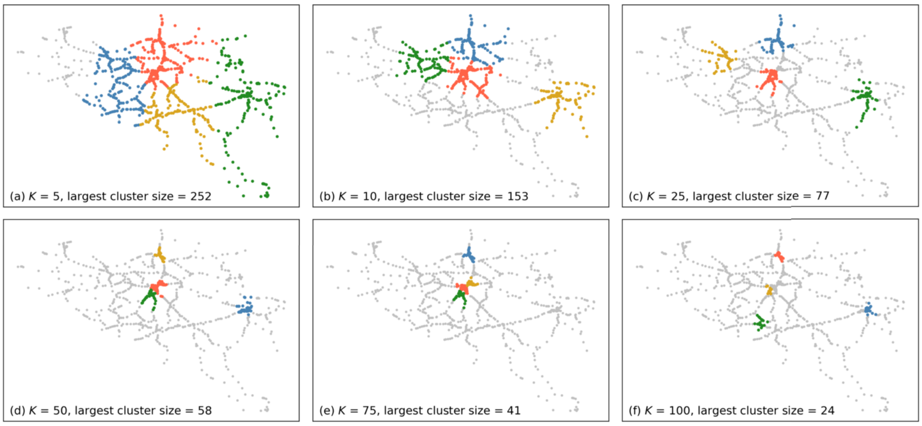

(a) K = 5, largest cluster size = 252
(b) K = 10, largest cluster size = 153
(c) K = 25, largest cluster size = 77
(d) K = 50, largest cluster size = 58
(e) K = 75, largest cluster size = 41
(f) K = 100, largest cluster size = 24

Figure 3 Clustering results for (**a**) K = 5, (**b**) K = 10, (**c**) K = 25, (**d**) K = 50, (**e**) K = 75 and (**f**) K = 100. For visibility purposes, only the fourth largest clusters are coloured (in the order of red, blue, green and yellow). Stations not belonging to any cluster are coloured grey. Cluster size is measured by amount of associated stations. Largest cluster sizes are denoted in panel labels

4 Results

In this section, we show the dynamics of the model and compare it to data. First we show example simulations. We then introduce toy networks to illustrate in which circumstances this model works well. Next, we discuss the overall performance on all 50 disrupted days. We end with a discussion of the model and possible extensions.

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. We initialised the non-clustered ('highest resolution') version of the model at this moment and 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. 4(a). 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 the highest resolution, we cluster the network into five clusters and redo the analysis. We observe that,

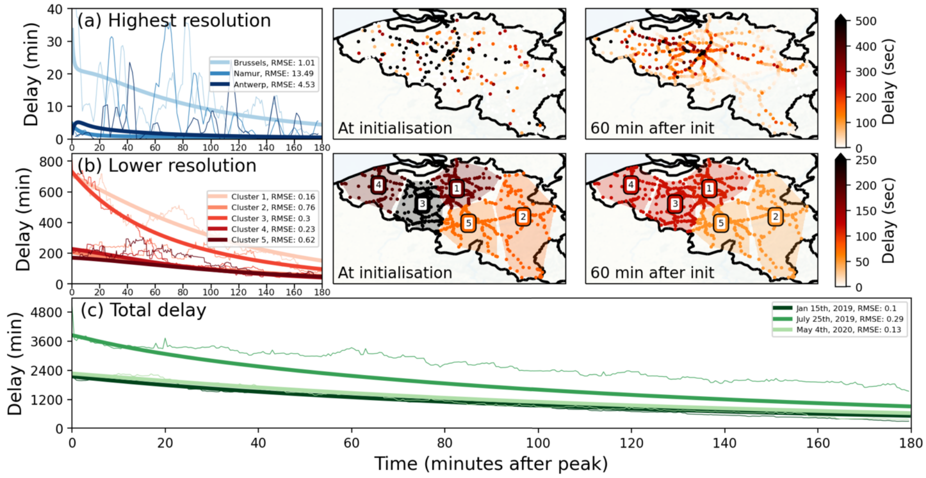


Figure 4 Example simulations at various resolutions. *Panel* (**a**): highest (non-clustered) resolution simulation of Jan 15th, 2019, initialised at the peak delay (18:11). The delay evolution over time of three example stations is displayed in blue lines (Brussels, Namur and Antwerp). The spatial situation of simulated delays at initialisation (middle) and 60 min after initialisation (right) are also shown. *Panel* (**b**): Simulation of the same day, but at an spatially aggregated level of five clusters. Red lines in the left panel show the temporal evolution of delays for each cluster. Again, the middle and right panels indicate spatial delay distribution at initialisation and 60 min after initialisation. For clarity, the clusters are shaded in the background. *Panel* (**c**): Total delay evolution of three example bad days: Jan 15th, 2019, July 25th, 2019 and May 4th 2020, all initialised at their peak delay moments. All maps show delays in seconds, with a cut-off at 500 and 250 seconds respectively, as higher delays were rare on these instances. Root mean square errors of each of the (normalized) time series are shown in the legends

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 resemble the simulation quite well, even though there are some deviations. 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) on 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 is 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

We now introduce two toy systems that allow us to study more fundamental properties of the model. 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.

The toy model was performed in the following way: we start with the network (random or start graph), which gives us an artificial structure of the railway network (we fix and save coordinates of every node). Knowing the configuration of stations (nodes) and edges, we draw "train lines": routes travelled by the trains in the toy model. These train lines are shortest paths between two randomly chosen stations. For the associated timetables, the starting time is drawn, and based on the edge weights, subsequent arrival times are determined. The simulations are initialised with a drawn distribution of delays and these delays are passed onto other trains by means of certain rules. The trains can not outstrip each other and later must wait until the earlier will pass. Delayed trains will block passage through stations for other stations. Such dynamics cause the natural piling up of trains at stations, and it is recreated in this toy models. For more information, see Additional file 1, Appendix D.

Figure 5(a) shows the performance of the introduced model for toy 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 in the beginning. This is 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

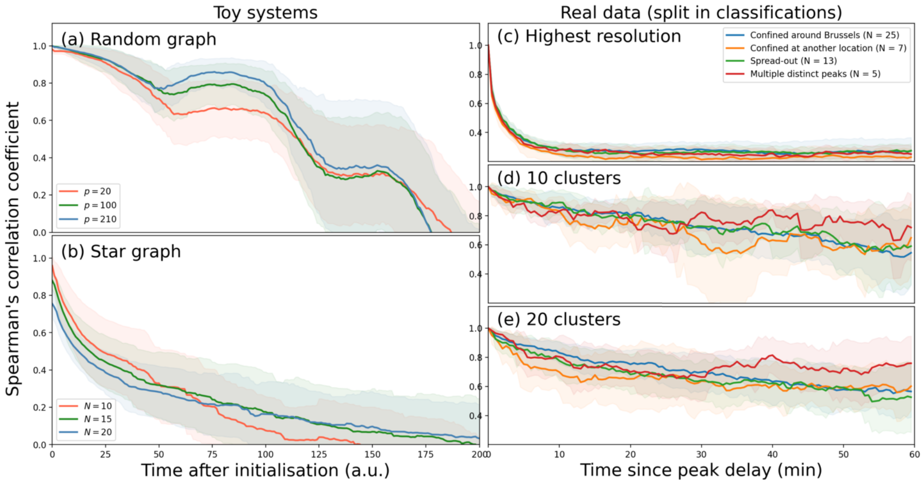


Figure 5 Model performance in toy systems and across classifications of disrupted days. *Panel* (**a**): Model performance of the random toy system for three different values of the amount of lines *p* (see Additional file 1, Appendix D for details). *Panel* (**b**): Model performance of the star graph toy system for different values of the amount of nodes *N* (keeping *p* fixed, see Additional file 1, Appendix D for more details). *Panel* (**c**)–(**e**): Model performance along time since peak delay of the disrupted days, averaged within each classification (see Additional file 1, Appendix B for details), for the model at (**a**) highest resolution, (**b**) using $K = 10$ and (**c**) using $K = 20$ clusters. Averages are shown in lines, shaded areas indicate the range of one standard deviation from the average

of lines (i.e., low p), the specific direction the train is going 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 brings the correlation down to much lower values.

For the star graphs (see panel (b) of Fig. 5) we fix the number of lines to $p = 50$ and take a look at the dependence on the number of nodes. This number does not seem to matter much, but it is clear that the star graph shows 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 (Additional file 1) 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. 5(c)–(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. 6(a). It is clear that in general, the higher t, the lower the correlation. This corresponds to intuition: at longer simulation times the model will start differing from reality more, for example due to new delays in the real data that are not captured by the model or model errors 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. On the other hand, this coarse graining corresponds to some of the model's assumptions, which averages the dynamics over many trains and ignores details. Our model's 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.

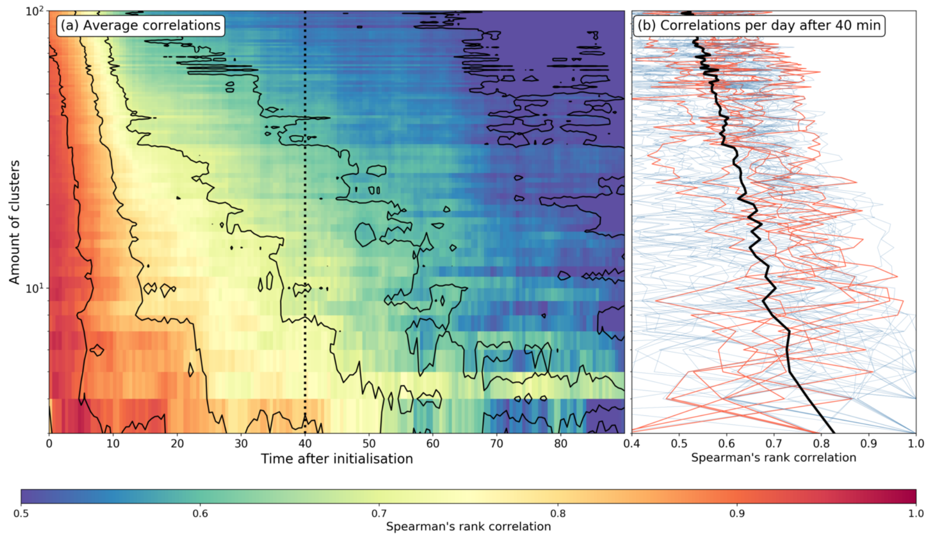


Figure 6 *Panel* (**a**): Average Spearman's rank correlation coefficients in colours averaged over the 50 disrupted days, for various values of K (vertical) and time points after model initialisation (horizontal, in minutes). The contours indicate the levels 0.5, 0.6, 0.7, 0.8 and 0.9. The vertical dashed line corresponds to 40 minutes after initialisation, which is used in the other panel. *Panel* (**b**): Spearman's rank correlation coefficients at 40 min after initialisation. Individual days are split into days that have their maximum at values of $K < 9$ (in blue) and those that have their optimum performance at values of $K \geq 9$ (in red). The black line indicates the 50-day average

Panel (b) in Fig. 6 shows the correlation 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. To illustrate the differences in levels of K where the disrupted days have performance optima, we show in red, thin lines the days in 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. (The level of $K = 9$ as boundary is chosen based on manual variation and does not have a physical importance other than serving this illustration.) On these seven days, the model performs best on a non-trivial level of K, between 9 and 11. An optimal intermediate level of coarse graining could indicate that our model has a spatial scale on which it performs best, at least in some situations. Intuitively, this could be explained as follows. We would expect the model's performance to increase as the number of clusters goes down, due to some of the model's assumptions as mentioned above. Yet, too few clusters means that all localization information is lost. Using a single cluster in the model, for example, would mean that delays in the west of the country would also dissipate due to trains stopping in the east of the country. This would lower the model's accuracy. Another explanation is based on the exact configuration of the clustering. For these red-coloured days, the 10-node network is discussed in more depth in Additional file 1, Appendix C. The blue lines indicate all other days (where optimal correlations are found with very low values of K).

4.5 Discussion and applicability

Our results show that modelling delay as a diffusion-like spreading phenomenon clearly has limitations: on the scale of individual stations (Fig. 4(a)), the discrete nature of delays is not simulated at all. The diffusion-like-spreading mechanism corresponds to the assumption that the delay propagation is based not on few trains carrying a larger delay, but on many tiny trains carrying smaller delays. Moreover, the model assumes these many trains randomly pick directions, weighted by parameters such as 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. 4(b) 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. Such delay generation is not accounted for by the model. This 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. If the network is dense, with many trains travelling on it, the delay spread is roughly described by 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. Hence, we propose to investigate such metrics further in future work.

The higher 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 **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 relatively simple parametrization. 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 simple differential equation (Eqn. (7)).

4.6 Potential model extensions

To fully understand the potential of diffusion-like modelling in studying railway delay propagation, a number of model extensions could be considered in future research, some of which were already shortly touched upon in previous sections.

In the derivation of the model equations, we have currently only taken very basic 'train actions' into account: trains carry delay, go from station to station, and have a certain stopping probability. From this we formally derived our ODE system, by introducing a number of assumptions. It is possible to perform the same derivation, but considering more complex train dynamics. One example would be to include the probability that a train loses some delay when it passes a station, but does not stop there. These probabilities could be inferred from the data for each station, or we could parametrize such a probability as function of the station's size or other features attributed to stations or edges. There are also simplifications in terms of track infrastructure or differences between stations (e.g., in terms of sizes): the current model treats parallel tracks as one single (broader) track and all stations as homogeneous 'nodes', which although common [2, 16], could be improved to enhance performance.

An important aspect that is currently not included is the generation of new delays. In our model, all the delay will disappear from the model because of the presence of sinks, i.e. trains stopping at stations. A first extension is thus to include a delay-generating mechanism in the model. This could be done by adding noise. The magnitude and distribution of stochastically generated delays at different stations could 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. A related extension would be to include a mechanism which might amplify, or mitigate, the stochastically generated small delays. Such feedbacks can be nonlinear, complicating the model but possibly generating new dynamics. This would make it possible to not only study the dissipation of delays, but also how small delays could be amplified into large disruptions due to so-called cascade effects.

Another aspect which could inspire other versions of the model is the fact that currently, delays are instantaneously transmitted from station to station. Even though the rate of this transmission depends on parameters such as the time it takes to cross tracks, a delay at one station will directly influence stations all over the network, which is nonphysical. One extension which could deal with this is a model with partial differential equations (PDEs). In such a model the variables of interest would be delay distributions on each edge. On an edge e_{ij} there would be a spatial delay distribution $d_{ij}(x,t)$, where x is a spatial coordinate on the edge. The evolution of this distribution, driven by trains moving on the edge, would be modeled by a PDE. This system of PDEs could then be linked through boundary conditions corresponding to the stations. Another way to deal with finite travel times would be to have the delay at station i depend on the delay of a neighbour j with a time lag equal to the traveling time on the edge from j to i. This would lead to delay differential equations (DDEs). These model extensions would be more realistic, but also more complicated from a computational point of view.

Finally, our model is 'first-order': the spread of delay is determined only by where it is, not where it came from. This could be improved as follows. Instead of considering probabilities p_{ij} that a train arriving at i will go to j, we would need to use probabilities $\tilde{p}_{ijk}$: the probability that a train coming from i and arriving at j will continue to k. Such an approach is analogous to second-order Markov models [38]. An approach which is conceptually similar would be to shift perspective and define total delay on edges instead of stations, e.g. analogous to [12]. Such a model would automatically include the inherent directionality of trains.

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 by grouping stations together into clusters. We conclude that this model is mainly of use when working on larger spatial scales. It can help to identify system properties related to delay dissipation and general directions of delay propagation, rather than provide accurate prediction of delays.

We believe that the model we described in this paper can be the starting point for many different studies. We have already described various extensions in the previous section. Here, we outline some ideas for further work related to the current model as it is.

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 southeast 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. Also, we see sudden jumps in the correlations in Fig. 6 around $K = 8$ and $K = 22$, possibly related to significant re-configurations of the clusters before and after these levels of K. A hierarchical clustering method prevents this, but has the disadvantage is that clustering boundaries set high up in the tree may affect the strength of the correlation significantly. A stochastic form of hierarchical clustering would probably be best.

When embedding these results in the broader scope of transport literature, it is important to acknowledge the relatively small fraction of railway delay studies with diffusion-like or epidemic models as compared to data-based [7–9, 22] or microscopic, discrete-event approaches [39, 40]. Several of the limitations of this type of modelling are stronger in the case of railways, such as the small amount of transport particles and how delays are transferred between them, which are less problematic in other transportation systems, e.g., when studying car traffic jams (e.g., [13]). Still, there are examples to be found. One example is Gurin et al. (2020), which show that classic Susceptible-Infected-Recovered (SIR) modelling applied to railways help understanding primary delay evolution, making use of 'transit coefficients' to account for spreading of delays to adjacent stations [12]. Another example is the paper by Monechi et al. (2018), which approaches the problem from a data perspective and infers 'laws' of railway delay propagation [11]. The niche of our current work can be found in the modelling of severe delays. More conceptually, our model fits in with other models that take a macroscopic, coarse-grained approach to delay spreading. It can be seen as an exercise to see to what extent diffusion-like spreading is applicable to railway delay spreading, and what we can learn about the real system from such a simplified description.

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 [41]. However, we argue that the increase in model performance when coarse graining is robust to these changes, for theoretical reasons: 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 application of our work would thus be to compare the network models for different countries and explore the properties in different spatial resolutions. Since the model's dynamics are all encoded in the matrix **G**, a comparison between different railway networks could be approached through a comparison of the properties of these matrices. It is possible that a few simple metrics, derived from this matrix, could be used for a quick international comparison of railway networks.

Exploring the spectral and topological properties of the weighted network that **G** describes and relating those properties to the dynamics of the railway system would of interest in itself. For such a study, it would be very interesting to expand the use of toy networks. One could generate a large database of artificial train networks and schedules, and computationally run the model on each of them. In this way, systematic scans over parameters can be done, and the accuracy of the model can be assessed as function of network-theoretical properties of the railway graph and properties of the schedule.

Diffusion-like spreading is researched in many fields besides transportation. In many systems — including social, epidemiological and engineered systems — the vulnerability to perturbations depends on the modularity of the system [26]. A more modular dynamical structure prevents large-scale spreading [2, 42]. This, and many other metrics can help quantify vulnerability to system-wide features (such as large-scale railway disruptions). Diffusion-like models like presented here can contribute to this, for example by determining the modularity from the **G**-matrix.

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 **G** matrix. For example, practitioners might be interested in how isolated regions are from each other: the off-diagonal elements of the **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. 6(b) (see also Additional file 1, Appendix C) can be used to categorise situations with large delays, 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.

Supplementary information

Supplementary information accompanies this paper at https://doi.org/10.1140/epjds/s13688-022-00359-1.

Additional file 1. 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 $K = 10$, 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.

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 Waldl 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 Nederlanse 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.

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 **G** matrix is available under https://github.com/an-medvedev/diffusion-like-delay-propagation. Both will be available upon publication.

Declarations

Competing interests

The authors declare that they have no competing interests.

Author contribution

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 done by JR. The model was solved and formally analysed 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. All authors read and approved the final manuscript.

Author details

[1]Department of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands. [2]Centre for Complex Systems Studies, Utrecht University, Utrecht, The Netherlands. [3]PBL Netherlands Environmental Assessment Agency, The Hague, The Netherlands. [4]ICTEAM, Université catholique de Louvain, Louvain-la-Neuve, Belgium. [5]EMBL, Heidelberg, Germany. [6]Faculty of Physics, Warsaw University of Technology, Warsaw, Poland. [7]Center for Research and Interdisciplinarity (CRI), Université de Paris, Paris, France. [8]Nokia Bell Labs, Paris, France.

Publisher's Note

Received: 3 April 2021 Accepted: 13 July 2022 Published online: 30 July 2022

References

1. Zieger S, Weik N, Nießen N (2018) The influence of buffer time distributions in delay propagation modelling of railway networks. J Rail Transp Plan Manag 8(3–4):220–232. https://doi.org/10.1016/j.jrtpm.2018.09.001
2. Dekker MM, Panja D (2021) Cascading dominates large-scale disruptions in transport over complex networks. PLoS ONE 16(1):1–17. https://doi.org/10.1371/journal.pone.0246077

3. Ludvigsen J, Klæboe R (2014) Extreme weather impacts on freight railways in Europe. Nat Hazards 70(1):767–787. https://doi.org/10.1007/s11069-013-0851-3
4. Tsuchiya S, Tatano H, Okada N (2007) Economic loss assessment due to railroad and highway disruptions. Econ Syst Res 19(2):147–162. https://doi.org/10.1080/09535310701328567
5. Büchel B, Spanninger T, Corman F (2020) Empirical dynamics of railway delay propagation identified during the large-scale rastatt disruption. Sci Rep 10(1):18584. https://doi.org/10.1038/s41598-020-75538-z
6. Dekker MM, van Lieshout RN, Ball RC, Bouman PC, Dekker SC, Dijkstra HA, Goverde RMP, Huisman D, Panja D, Schaafsma AM, van den Akker M (2018) 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. https://repub.eur.nl/pub/109054/
7. Kecman P, Goverde RMP (2015) Predictive modelling of running and dwell times in railway traffic. Public Transp 7(3):295–319. https://doi.org/10.1007/s12469-015-0106-7
8. Li D, Daamen W, Goverde RMP (2016) Estimation of train dwell time at short stops based on track occupation event data: a study at a Dutch railway station. J Adv Transp 50(5):877–896. https://doi.org/10.1002/atr.1380
9. Dekker MM, Panja D, Dijkstra HA, Dekker SC (2019) Predicting transitions across macroscopic states for railway systems. PLoS ONE 14(6):0217710. https://doi.org/10.1371/journal.pone.0217710
10. Oneto L, Fumeo E, Clerico G, Canepa R, Papa F, Dambra C, Mazzino N, Anguita D (2018) Train delay prediction systems: a big data analytics perspective. Big Data Res 11:54–64. https://doi.org/10.1016/j.bdr.2017.05.002
11. Monechi B, Gravino P, Di Clemente R, Servedio VDP (2018) Complex delay dynamics on railway networks from universal laws to realistic modelling. EPJ Data Sci 7:35. https://doi.org/10.1140/epjds/s13688-018-0160-x. arXiv:1707.08632
12. Gurin D, Prokhorchenko A, Kravchenko M, Shapoval G (2020) Development of a method for modelling delay propagation in railway networks using epidemiological sir models. East-Eur J Enterp Technol 6:6–13. https://doi.org/10.15587/1729-4061.2020.219285
13. Saberi M, Hamedmoghadam H, Ashfaq M, Hosseini SA, Gu Z, Shafiei S, Nair DJ, Dixit V, Gardner L, Waller ST, González MC (2020) A simple contagion process describes spreading of traffic jams in urban networks. Nat Commun 11(11):1–9. https://doi.org/10.1038/s41467-020-15353-2
14. Sen P, Dasgupta S, Chatterjee A, Sreeram PA, Mukherjee G, Manna SS (2003) Small-world properties of the Indian railway network. Phys Rev E, Stat Phys Plasmas Fluids Relat Interdiscip Topics 67(3):5. https://doi.org/10.1103/PhysRevE.67.036106. 0208535
15. Erath A, Löchl M, Axhausen KW (2009) Graph-theoretical analysis of the Swiss road and railway networks over time. Netw Spat Econ 9(3):379–400. https://doi.org/10.1007/s11067-008-9074-7
16. Bhatia U, Kumar D, Kodra E, Ganguly AR (2015) Network science based quantification of resilience demonstrated on the Indian railways network. PLoS ONE 10(11):0141890. https://doi.org/10.1371/journal.pone.0141890. arXiv:1508.03542
17. Goverde RMP (2010) A delay propagation algorithm for large-scale railway traffic networks. Transp Res, Part C, Emerg Technol 18(3):269–287. https://doi.org/10.1016/j.trc.2010.01.002
18. Gambardella LM, Rizzoli AE, Funk P (2002) Agent-based planning and simulation of combined rail/road transport. Simulation 78(5):293–303. https://doi.org/10.1177/0037549702078005551
19. Büker T, Seybold B (2012) Stochastic modelling of delay propagation in large networks. J Rail Transp Plan Manag 2(1–2):34–50. https://doi.org/10.1016/j.jrtpm.2012.10.001
20. Harrod S, Cerreto F, Nielsen OA (2019) A closed form railway line delay propagation model. Transp Res, Part C, Emerg Technol 102:189–209. https://doi.org/10.1016/j.trc.2019.02.022
21. Dekker MM, Panja D (2019) A reduced phase-space approach to analyse railway dynamics. IFAC-PapersOnLine 52(3):1–6. 15th IFAC Symposium on Large Scale Complex Systems LSS 2019
22. Oneto L, Fumeo E, Clerico G, Canepa R, Papa F, Dambra C, Mazzino N, Anguita D (2017) Dynamic delay predictions for large-scale railway networks: deep and shallow extreme learning machines tuned via thresholdout. IEEE Trans Syst Man Cybern Syst 47(10):2754–2767. https://doi.org/10.1109/TSMC.2017.2693209
23. Corman F, Kecman P (2018) Stochastic prediction of train delays in real-time using Bayesian networks. Transp Res, Part C, Emerg Technol 95:599–615. https://doi.org/10.1016/j.trc.2018.08.003
24. Berger A, Gebhardt A (2011) Stochastic delay prediction in large train networks. In: 11th workshop on, pp 100–111. https://doi.org/10.4230/OASIcs.ATMOS.2011.100
25. Fortunato S, Hric D (2016) Community detection in networks: a user guide. Phys Rep 659:1–44. https://doi.org/10.1016/j.physrep.2016.09.002. Community detection in networks: a user guide
26. Newman MEJ (2006) Modularity and community structure in networks. Proc Natl Acad Sci USA 103(23):8577–8582. https://doi.org/10.1073/pnas.0601602103
27. Blondel VD, Guillaume JL, Lambiotte R, Lefebvre E (2008) Fast unfolding of communities in large networks. J Stat Mech Theory Exp 2008(10). https://doi.org/10.1088/1742-5468/2008/10/P10008. arXiv:0803.0476
28. Steinhaus H (1956) Sur la division des corps matériels en parties. Bull Acad Pol Sci, Cl Trois IV(12):801–804
29. MacQueen J (1967) 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. https://projecteuclid.org/euclid.bsmsp/1200512992
30. Lloyd S (1982) Least squares quantization in pcm. IEEE Trans Inf Theory 28(2):129–137. https://doi.org/10.1109/TIT.1982.1056489
31. Kadir RA, Shima Y, Sulaiman R, Ali F (2018) Clustering of public transport operation using K-means. In: 2018 IEEE 3rd international conference on big data analysis, ICBDA 2018. IEEE Press, New York, pp 427–432. https://doi.org/10.1109/ICBDA.2018.8367721
32. Cerreto F, Nielsen BF, Nielsen OA, Harrod SS (2018) Application of data clustering to railway delay pattern recognition. J Adv Transp 2018:6164534. https://doi.org/10.1155/2018/6164534
33. Gao J, Buldyrev SV, Havlin S, Stanley HE (2011) Robustness of a network of networks. Phys Rev Lett 107:195701. https://doi.org/10.1103/PhysRevLett.107.195701
34. Kivelä M, Arenas A, Barthelemy M, Gleeson JP, Moreno Y, Porter MA (2014) Multilayer networks. J Complex Netw 2(3):203–271

35. Siudem G, Hołyst JA (2019) Diffusion on hierarchical systems of weakly-coupled networks. Phys A, Stat Mech Appl 513:675–686. https://doi.org/10.1016/j.physa.2018.08.078
36. International Union of Railways (2020) Synopsis 2020. https://uic.org/support-activities/statistics/. Accessed 25 Mar 2021
37. Infrabel Open Data. https://infrabel.opendatasoft.com/. Accessed 25 Mar 2021
38. Rosvall M, Esquivel AV, Lancichinetti A, West JD, Lambiotte R (2014) Memory in network flows and its effects on spreading dynamics and community detection. Nat Commun 5(1):4630. https://doi.org/10.1038/ncomms5630
39. Middelkoop AD, Loeve L (2006) Simulation of traffic management with FRISO. In: WIT transactions on the built environment. WIT transactions on the built environment, vol 88. WIT Press, Southampton, pp 501–509. https://doi.org/10.2495/CR060501. http://library.witpress.com/viewpaper.asp?pcode=CR06-050-1
40. Corman F, D'Ariano A, Hansen IA (2014) Evaluating disturbance robustness of railway schedules. In: Journal of intelligent transportation systems: technology, planning, and operations, vol 18. Taylor & Francis, London, pp 106–120. https://doi.org/10.1080/15472450.2013.801714
41. Schipper D, Gerrits L (2018) Differences and similarities in European railway disruption management practices. J Rail Transp Plan Manag 8(1):42–55. https://doi.org/10.1016/j.jrtpm.2017.12.003
42. Balcan D, Colizza V, Gonçalves B, Hud H, Ramasco JJ, Vespignani A (2009) Multiscale mobility networks and the spatial spreading of infectious diseases. Proc Natl Acad Sci USA 106(51):21484–21489. https://doi.org/10.1073/pnas.0906910106. arXiv:0907.3304