

Functional connectivity networks for transportation delay analysis: From theory to software

Carlson Moses Büth ^a,* , Massimiliano Zanin ^b

^a Institute for Cross-Disciplinary Physics and Complex Systems (IFISC), CSIC-UIB, 07122, Palma de Mallorca, Spain

^b Institute for Cross-Disciplinary Physics and Complex Systems (IFISC), CSIC-UIB, Edifici Complex de Recerca de les Illes Balears, Parc Bit, Palma de Mallorca, 07120, Spain

ARTICLE INFO

Dataset link: <https://doi.org/10.5281/zenodo.16875272>, <https://delaynet.readthedocs.io/>, <https://github.com/cbueth/delaynet>

Keywords:

Transportation networks
Delays
Functional networks
Causality

ABSTRACT

Within the endeavour of modelling and understanding the propagation of delays in transportation networks, an approach that has attracted increasing interest in the last decade is the creation of functional network representations. These graphs map elements of interest (e.g. airports or stations) as nodes, and derive pairwise dependency patterns from their dynamics through correlation and causality tests. In spite of multiple notable results, this approach still lacks a coherent framework, with decisions related to many fundamental steps being left to the judgement of the researcher. We here provide an introduction to the theory behind functional networks for transportation systems, detailing the main steps and the associated pitfalls. We further introduce a Python package, *delaynet*, designed to support the researcher in the reconstruction and analysis of such networks. We finally present an analysis of the propagation of delays in the Swiss train system; and discuss future research steps.

1. Introduction

Transportation networks are complex systems composed of a large number of interconnected components that facilitate the movement of people and goods, and that have become the backbone of current societies (Cascetta, 2009; Button, 2010). These networks, whether air, rail, road, or maritime, share a common challenge: delays. Delays in transportation systems can propagate through the network, affecting multiple components and leading to cascading failures that significantly impact operational efficiency, passenger experience, and economic outcomes (Carlier et al., 2007; Peterson et al., 2013). Understanding how these delays propagate is crucial for developing effective strategies aimed at their mitigation.

A large body of literature has been devoted to this task, mainly through three complementary approaches. Firstly, one finds the statistical description and modelling of delays (Rebollo and Balakrishnan, 2014; Wang and Vaze, 2016; Kim et al., 2017; Yang et al., 2019; Mitsokapas et al., 2021), in which the shape of the probability distributions (e.g. the presence of long tails and asymmetries) is used to infer potential underlying mechanisms, both for delay generation and propagation. Secondly, the use of regression, data-based, or more recently machine learning and deep learning models (Cannas et al., 2013; Yu et al., 2019; Truong, 2021; Sajan and Kumar, 2021; Lapamonpinyo et al., 2022). These present the advantage of being able to connect causes to

effects, hence guiding towards strategies for improving the system; and to further yield tools for forecasting (and hence, hopefully, prevent) delays. Thirdly, one can resort to (usually microscale) simulations, to try to model the relevant aspects of the system under study (Higgins and Kozan, 1998; Barnhart et al., 2014; Wei and Vaze, 2018; Özlem et al., 2021; Zhao et al., 2025b; Xu et al., 2025; Zhao et al., 2025a); in this case, the user can develop what-if scenarios, and thus numerically test the implementation of new mitigation procedures — albeit at a high computational cost. Note that such categorisation is necessarily artificial, and that many studies cross these boundaries — as for instance, any numerical model must be based on real data and statistics. Still, and while these methods have provided valuable insights, they are challenged when the aim is to fully capture the complex dynamics of delay propagation in highly interconnected systems.

In recent years, concepts from statistical physics (Huang, 2009; Reichl, 2016) and complex network theory (Strogatz, 2001; Newman, 2003) have increasingly been applied to transportation systems (Woolley-Meza et al., 2011; Riccardo et al., 2012; Cook et al., 2015; Barthelemy, 2019; Sun and Wandelt, 2021; Olivares and Zanin, 2025). The leitmotif of statistical physics is the study of complex systems, i.e. systems composed of a large number of interacting elements (Anderson, 1972), by inferring microscale laws when only the macroscale

* Corresponding author.

E-mail address: carlson@cbueth.de (C.M. Büth).

dynamics can be observed. The prototypical example is the study of ideal gases: while one cannot observe the position and dynamics of all particles, global laws can still be extracted when observing macroscale properties like pressure and temperature. Similarly, complex networks aim at describing the interactions between the elements of a complex system, usually deriving them from macroscale observables. When applied to transportation systems, these approaches view them as dynamic systems where local perturbations can propagate and amplify through various mechanisms. It is then possible to identify critical nodes, understand systemic vulnerabilities, and finally develop more robust transportation systems.

The statistical physics approach has found extensive application in the study of delays in air transport through the so-called “functional networks”. These were initially developed in neuroscience to model the dynamics of information in the brain (Eguiluz et al., 2005; Power et al., 2010; Park and Friston, 2013), and involve representing air transport as a network. Individual airports are mapped into nodes, and links between pairs of them are created whenever a functional dependency between the corresponding delays is observed, by applying correlation and causality tests (Zanin, 2015; Zanin et al., 2017; Pastorino and Zanin, 2021; Jia et al., 2022; Wang et al., 2022; Feng et al., 2024; Gil-Rodrigo and Zanin, 2024). These dependencies can then be understood as instances of delay propagations, even though confounding effects may also be present. The main advantage of this approach is that only macroscale information is required, e.g. time series representing the average hourly delay at airports. At the same time, it also yields information about how delays may propagate, which airports have a major role in such propagation, or the appearance of constant dependency patterns.

The practitioner wanting to use the functional network approach to study delays, both in air transport and other modes, will face several methodological and computational challenges. To illustrate, while obtaining delay time series is usually simple, these are highly non-stationary (i.e. delays tend to appear with stronger intensity at peak hours), and this may generate spurious correlations and causalities; the solution involves applying one of the main detrending techniques that have been proposed (Olivares et al., 2025). Similarly, many tests can be used to assess dependencies, from correlation-based to causal ones, each yielding different views on the underlying process. Finally, networks have to be processed (by e.g. deleting irrelevant links) and analysed, using one of the many available topological metrics (Costa et al., 2007). Throughout all these steps, choices have to be made, for instance regarding the parameters of a given method, in many cases without theoretical guidelines and subordinate to the researcher’s experience. These problems finally compound due to the lack of unified software libraries, hindering subsequent validations and reproducibility.

In this contribution we review these steps and their pitfalls, explore solutions that have been presented in the literature, and provide the researcher with basic guidelines. This is complemented by presenting *delaynet*, an open-source Python package that aims to solve the aforementioned challenges by combining and systematising time series analysis techniques, network reconstruction methods, and complex network metrics. It offers a unified framework that encompasses the complete analytical pipeline — from data preprocessing to network reconstruction — reducing the risk of methodological inconsistencies that often arise when combining disparate analysis tools, thereby improving the reliability and reproducibility of delay studies. In addition, *delaynet* comes as an off-the-shelf package, flexible to be used for various types of data; thus, while existing research has focused on air transportation (Zanin, 2015; Zanin et al., 2017; Pastorino and Zanin, 2021; Jia et al., 2022; Wang et al., 2022; Feng et al., 2024; Gil-Rodrigo and Zanin, 2024), the implemented methods are applicable to any transportation mode where delay data can be collected as time series.

The software package presented here, and the associated theoretical discussion, are aimed at filling a gap in the literature. Many software

pipelines are available to support the reconstruction of functional networks in other fields, mostly in neuroscience (e.g. MNE-Python (Gramfort et al., 2013), NeuroPycon (Meunier et al., 2020), bctpy (LaPlante et al., 2014), dyconnmap (Marimpis et al., 2021), nilearn (Abraham et al., 2014)) and genomics (pySCENIC (Kumar et al., 2021), GReNaDine (Schmitt et al., 2023), Augusta (Musilova et al., 2024)). The interested researcher can also resort to software packages for general causal inference (Runge et al., 2023; Zhao and Liu, 2023), and for network analysis (Al-Taie and Kadry, 2017). Yet, to the best of our knowledge, none specifically focusing on transportation data has hitherto been proposed, combining all necessary ingredients in a single place. In addition, this approach should not be considered as a complete solution, but rather as one of the many tools available to the practitioner to unveil the underlying dynamics of the system. In other words, functional networks can (and should) be used in conjunction with other techniques, including microscale models and classical statistical analyses, as these are not mutually exclusive. Still, in order for this to be a synergistic process, the researcher should be aware of the underlying hypotheses and limitations of functional networks.

In the remainder of the paper we firstly present a theoretical overview of the main steps in the reconstruction and analysis of delay functional networks, see Section 2. While its content mimics the structure of the *delaynet* package, it is also designed to be an introduction to the topic for the interested researcher, by discussing both fundamental tools and common challenges. Next, Section 3 describes the structure of the package itself, and how the previously mentioned main steps are organised from a software perspective. We then demonstrate the complete workflow on a real dataset in Section 4, a case study of Swiss long-distance rail delays (2022–2025) built from SBB realised timetable data. We finally draw some conclusions and discuss future steps in Section 5.

2. Theoretical foundations

The analysis of delay dynamics in transportation systems using functional networks follows a systematic approach that combines various analytical techniques, integrating contributions from different fields including transportation, information theory and complex network theory. This approach can generally be divided into five fundamental steps: data preparation, detrending methods, connectivity analysis, network reconstruction, and network analysis — see Fig. 1 for a graphical representation. Each component addresses specific challenges in analysing delay data and builds upon the previous steps to provide a complete picture of the delay dynamics.

The following subsections explain each step in detail, focusing on the theoretical foundations, practical considerations and challenges. This structure is also reflected in the internal organisation of *delaynet*, as discussed in Section 3; consequently, what is presented here is both a primer for the interested researcher, and a starting point for the use of the package.

2.1. Data preparation

The availability of high-quality and reliable data is the basic requirement behind any data-driven analysis; and, not surprisingly, it also applies to the problem at hand. Functional networks use as starting point a set of time series, each representing the delay observed at the elements of interest (e.g., airports, stations, or route segments) of the system under study. Note that, while previous studies in air transport have focused on airports (Zanin, 2015; Zanin et al., 2017; Pastorino and Zanin, 2021; Jia et al., 2022; Wang et al., 2022; Feng et al., 2024; Gil-Rodrigo and Zanin, 2024), nothing prevents different viewpoints, as e.g. the study of the propagation of delays between sectors. In addition, these time series must be regularly sampled, time aligned, and of uniform length. Crucially, they must have a time resolution consistent with the aim of the analysis — to illustrate, the propagation of delays

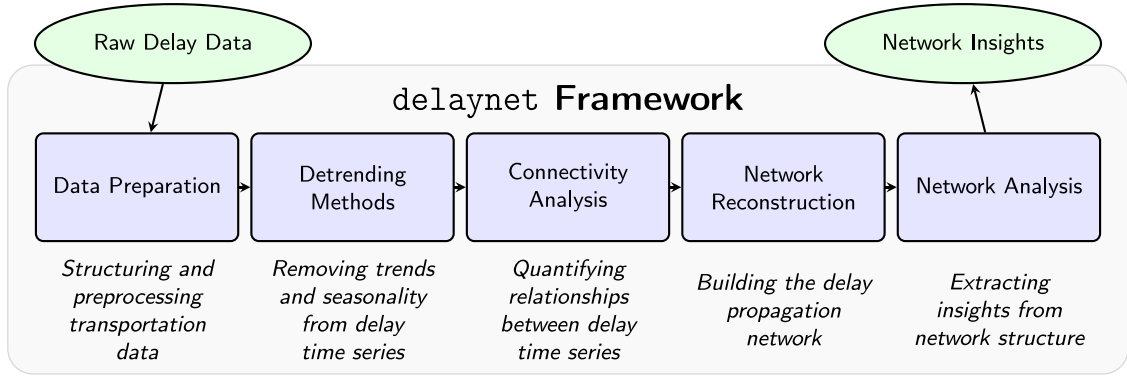


Fig. 1. Overview of the main steps in delay functional network analysis, and their implementation in *delaynet*. The framework transforms raw delay data through five sequential stages: (1) data preparation and preprocessing; (2) detrending methods, to remove systematic patterns and seasonality that could mask genuine propagation signals; (3) connectivity analysis, quantifying statistical relationships between delay time series; (4) network reconstruction, which applies thresholding strategies to build functional networks from pairwise connectivity values; and (5) network analysis, to extract structural insights about delay patterns, critical nodes, and system vulnerabilities.

between airports in the same country cannot be assessed using daily time series, while such low resolution may be suitable in a maritime context. We here also assume that missing values have already been corrected by the user, e.g. through interpolations or forward-filling.

2.2. Detrending methods

Raw delay data often contains trends, seasonality, and other systematic patterns that can mask the underlying relationships between components of the transportation network; or, even worse, that may result in the appearance of spurious relationships. Such trend may arise from daily, weekly, or seasonal patterns in transportation demand; long-term growth or decline in usage; or even biases in delay reporting. If not properly addressed, these trends can lead to spurious correlations and misleading results in connectivity analyses.

To illustrate this point, imagine the case of two independent airports, whose traffic peaks respectively at 13:00 and 14:00 UTC. It is reasonable to assume that delays will also peak at those times, as the resources of the airports may be overexerted. If one then calculates a correlation between both airports' time series of delays, this will yield a propagation between the former and the latter, even though such propagation does not exist; and similar results may be obtained with other causality metrics. In mathematical terms, any measure averaged over a non-stationary time series will not be accurate; stationarity is a prerequisite for standard tests (including the t - and F -tests) and models (like the Auto Regression Moving Average, ARMA).

Multiple detrending methods that can be applied to transportation delay time series have hitherto been proposed, each with its own strengths and limitations. Comparisons using synthetic and real-world time series are also available, see for instance Olivares et al. (2025). For the sake of completeness, we here describe the main ones:

- **Identity transformation.** Simple copy of the original time series, i.e. no detrending is performed. Not to be used in real analyses, it is nevertheless an important method for testing purposes, e.g. as a baseline against which other methods can be compared.
- **Delta Detrending.** This method performs local mean subtraction by removing the average of a window of values centred around each point:

$$x'(t) = x(t) - \frac{1}{2w+1} \sum_{k=t-w}^{t+w} x(k), \quad (1)$$

where w is the window size. Delta detrending is particularly effective at removing local trends while preserving short-term fluctuations that might indicate delay propagation. The choice of window size is critical: too small a window might not capture

the trend effectively, while too large a window might remove important signals.

- **Second-Order Differentiation.** This method applies the difference operator twice to remove both constant and linear trends (Kirchgässner et al., 2013):

$$x'(t) = [x(t) - x(t-1)] - [x(t-1) - x(t-2)] = x(t) - 2x(t-1) + x(t-2). \quad (2)$$

Second difference detrending is particularly useful for transportation delay data that exhibits linear growth or decline over time.

- **Z-Score.** This method normalises the data by subtracting the mean and dividing by the standard deviation, often calculated within specific time windows or periods:

$$x'(t) = \frac{x(t) - \mu_p}{\sigma_p}, \quad (3)$$

where μ_p and σ_p are the mean and standard deviation for the period p to which time point t belongs. Z-Score detrending is effective for removing periodic patterns, such as daily or weekly cycles in transportation delays.

2.3. Connectivity analysis

After detrending, the next step in reconstructing functional networks is to quantify the relationships between delay time series of the different components of the transportation network. In other words, we aim at detecting and measuring how delays in one part of the system influence delays in another one, accounting for the time it takes for these influences to propagate. Given two time series, such analysis is done by applying different metrics and tests to them; this process is eventually repeated for all pairs of time series, to obtain a network view.

In the literature, many alternative metrics have been used, mostly adapted from the solutions common in brain functional network reconstruction. The output in all cases is the same, and comprises both the p -value of the metric and the best lag — i.e. the statistical significance and the time difference of the strongest observed relationship. Starting from the linear ones, these include:

- **Linear Correlation (LC).** The Pearson's correlation coefficient measures the linear relationship between two time series at a lag τ :

$$r_\tau = \frac{\text{cov}_{xy}}{\sigma_x \sigma_y} = \frac{\sum_{t=\tau+1}^T [x(t-\tau) - \bar{x}][y(t) - \bar{y}]}{\sqrt{\sum_{t=\tau+1}^T [x(t-\tau) - \bar{x}]^2} \sqrt{\sum_{t=\tau+1}^T [y(t) - \bar{y}]^2}}, \quad (4)$$

where $\bar{x}$ and $\bar{y}$ are the sample means, and σ_x and σ_y are the sample standard deviations of x and y , respectively. cov_{xy} is the sample covariance of x and y .

- **Rank Correlation (RC).** The Spearman's rank correlation uses the ranks of the values rather than the values themselves, making it more robust to outliers and nonlinear monotonic relationships:

$$\rho_r = \frac{\text{cov}(R_x, R_y)}{\sigma_{R_x} \sigma_{R_y}}, \quad (5)$$

where R_x and R_y are respectively the ranks of $x(t)$ and $y(t)$.

- **Granger Causality (GC).** This approach tests whether past values of one time series provide statistically significant information about future values of another time series, beyond what is explained by the past values of the latter (Granger, 1969; Diebold, 1997; Kirchgässner et al., 2013). In the context of transportation networks, GC examines whether historical delays at a source node help predict future delays at a destination node, after accounting for the destination's own delay history.

The method works by comparing two linear models, usually Ordinary Least Squares ones: a restricted model that predicts future delays at the destination using only its own past, and an unrestricted model that incorporates past delays from both the destination and potential source nodes. If including the source node's delay history significantly improves prediction accuracy, we can infer that delays propagate from the source to the destination.

In practice, the implementation tests multiple time lags to identify the optimal propagation time between nodes. For each potential lag, the algorithm constructs appropriate lag matrices, fits both restricted and unrestricted regression models, and compares their predictive performance. The lag that yields the strongest statistical evidence for delay propagation is selected as the optimal propagation time between the two components.

Since its inception, the GC test has found wide-ranging applications in various fields such as economics, engineering, sociology, biology, and neuroscience (Freeman, 1983; Reuveny and Kang, 1996; Ding et al., 2006; Shojaie and Fox, 2022). Limitations have also been highlighted, like the sensitivity to confounding effects (Grassmann, 2020), the failure to detect non-linear relationships (Maziarz, 2015), or the inability to handle missing or irregular data (Zanin, 2021).

These linear metrics present the advantage of being simple, both in the computation and the interpretation of the results. On the other hand, their linear nature implies that only the linear part of the propagation is described, and they may therefore understate, or completely miss, more complex dependencies. The practitioner may then want to resort to non-linear tests, including:

- **Mutual Information (MI).** This information-theoretic measure quantifies the amount of information shared between two time series X and Y (Büth et al., 2025):

$$I(X; Y) = \sum_{x \in X} \sum_{y \in Y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}. \quad (6)$$

Mutual information can capture both linear and nonlinear relationships, but crucially does not indicate directionality, i.e. $I(X; Y) = I(Y; X)$.

- **Transfer Entropy (TE).** This measure quantifies the directed transfer of information from one time series X to a second one Y , by measuring how the uncertainty in the latter is reduced by the former (Schreiber, 2000; Büth et al., 2025):

$$T_{X \rightarrow Y} = \sum p(y_{t+1}, y_t^{(k)}, x_t^{(l)}) \log \frac{p(y_{t+1} | y_t^{(k)}, x_t^{(l)})}{p(y_{t+1} | y_t^{(k)})}, \quad (7)$$

where $y_t^{(k)}$ and $x_t^{(l)}$ represent the past k and l values of Y and X respectively. Transfer entropy can detect nonlinear causal relationships, being a generalisation of the GC (Barnett et al., 2009); and is thus particularly useful in the context at hand.

- **Continuous Ordinal Patterns (COP).** This approach generalises traditional ordinal pattern analysis (Bandt and Pompe, 2002) by evaluating time series in terms of their distance to continuously defined patterns (Zanin, 2023):

$$\phi_X(t) = \frac{1}{2D} \sum_{i=0}^{D-1} |\pi_i - \text{norm}(x_{t+i})|. \quad (8)$$

In the previous equation, π represents a predefined continuous pattern of length D , the latter also known as the embedding dimension; and $\text{norm}(\cdot)$ is the operator normalising a segment of the time series to values between -1 and 1 . $\phi(t)$ can then be seen as a transformed version of the original time series, representing the evolution through time of the relevance of the pattern π . Given two input time series X and Y , the causality between them can be evaluated by firstly applying the same pattern π to both of them, thus obtaining two time series $\phi_X(t)$ and $\phi_Y(t)$; for then calculating the GC test between them (Zanin, 2024). In other words, π is used to extract what makes the two time series unique, and the GC test is used to detect whether the unique dynamics of the two elements are causally connected. Due to the non-linear nature of the transformation induced by π , this approach allows to efficiently detect non-linear causality relations without the need of a priori assumptions.

To identify the most informative pattern for connectivity analysis, COP generates multiple random patterns and evaluates their discriminative power. For each candidate pattern, the method calculates the statistical difference between the pattern-transformed original time series and a shuffled version, selecting the pattern that maximises this difference using the Kolmogorov-Smirnov test. This pattern selection process ensures that the chosen transformation captures genuine temporal structures rather than random fluctuations.

2.3.1. Statistical significance and interpretation

To determine whether a detected connection is statistically significant, it is customary to rely on the p -values yielded by the connectivity measures; these are obtained by comparing the obtained metric to what is expected under a null hypothesis of no connection. Some measures (e.g. correlations, and the GC and COP tests) naturally yield such p -value; in all other cases, an approximation is obtained either by random permutations or bootstrapping, as selected by the user.

All measures further include a lag, representing the expected time of the functional relationship. This is usually optimised by testing different lag values, to identify the optimal delay that produces the smallest p -value. This is illustrated in Fig. 2, which shows the evolution of the p -values calculated for each lag from 1 to 50 time steps, as obtained by the GC on two synthetic time series created with a delayed causal network (see Section 2.1). The minimum p -value (at lag 7) is interpreted as the optimal delay, indicating the most significant causal connection. Note how p -values increase for high values of the lag; as their calculation requires more complex models, the corresponding degrees of freedom also increase, resulting in less statistically significant results.

As a final note, it is important to highlight the meaning of the patterns identified through the aforementioned tests. These may prima facie be understood as propagation instances, i.e. flows of delays between two airports, in a way similar to how these links are commonly described as flow of information in functional brain networks. Yet, this is only true if no confounding effects are present, i.e. sources of delays not considered in the evaluation of the link and that can affect both nodes at the same time. These confounding effects may also stem from the presence of non-stationarities in the data, which could remain after the detrending process described above. Even when these are not present, it is important to note that measures called “causal”, as e.g. the Granger Causality, only measure predictive causality, i.e. how the delay of one node helps predict the future evolution at a second one; in

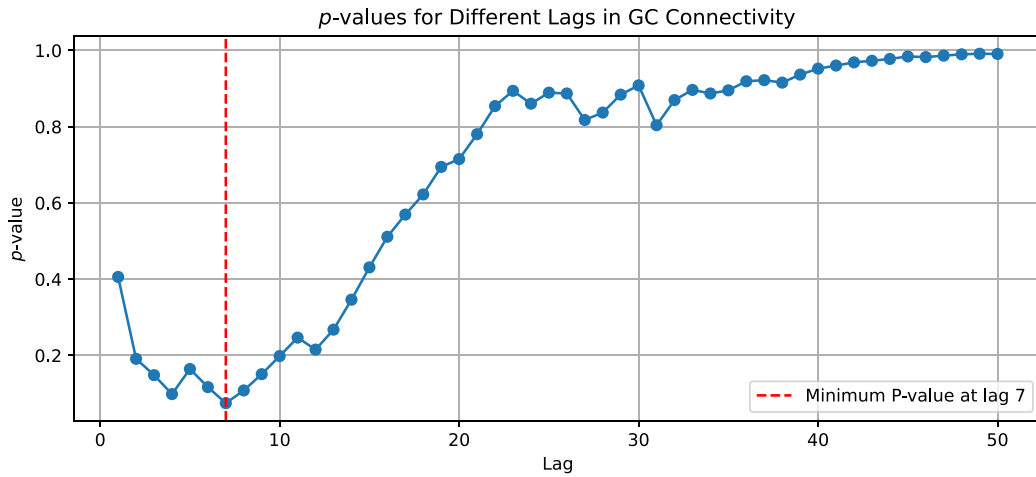


Fig. 2. Determination of optimal lag using p -values in a Granger Causality analysis between two synthetic time series. The vertical red dashed line indicates the minimum p -value obtained. See main text for details.

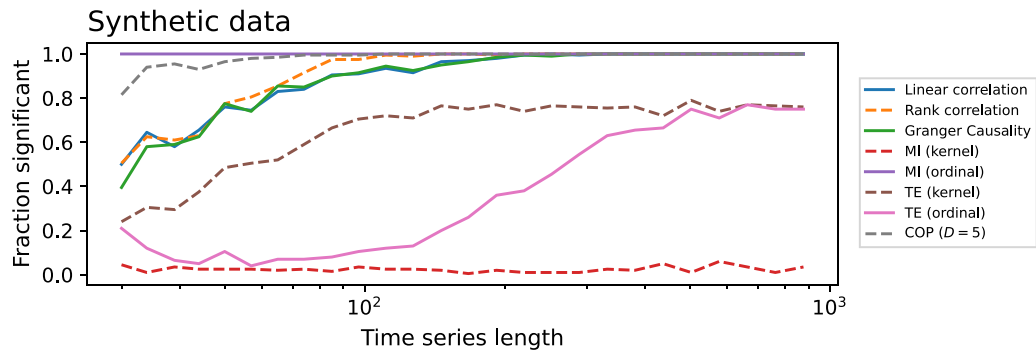


Fig. 3. Comparison of different connectivity measures as a function of the time series length, on synthetic data generated using the delayed causal networks (see Section 2.1). See the legend for a list of measures, with the corresponding estimator within parentheses. Results correspond to 200 randomly generated time series.

contrast, a true causality also requires the possibility of making changes in the system through intervention and counterfactual checks (Granger, 1988; Bareinboim et al., 2022). In short, the researcher should be aware of the limitations of the approach, and consider them in the interpretation of the results.

2.3.2. Comparing connectivity measures

One may intuitively expect the existence of a hierarchy between the connectivity measures described above; for instance, the Transfer Entropy is a much more sophisticated metric than a simple linear correlation, and hence should generally be preferred. Reality is nevertheless more complex. Firstly, non-linear approaches usually require larger quantities of data (i.e. longer time series) to detect relationships; hence they may not be suitable when fast dynamics ought to be analysed, e.g. what happened in a specific day or week. Secondly, when the propagation under analysis is mainly linear, non-linear approaches may be ill-suited to describe them. Thirdly, one has to consider that linear approaches do not necessarily disregard non-linear relationships; more correctly, they are only able to describe their linear part, or to generate a linear approximation of them.

To illustrate this point, we considered the delayed causal network model described in Section 2.1, and generated time series of variable length with two nodes randomly connected between them; Fig. 3 then represents the fraction of times each measure was able to detect the propagation (using a significance level $\alpha = 0.01$), as a function of the time series length. It can be appreciated that linear approaches (linear correlation, rank correlation and Granger Causality) generally require

shorter time series to achieve good results. Non-linear approaches can also work well, but their performance is highly dependent on the underlying estimator. To illustrate, MI detects the relationship when using an ordinal estimator, but not with a kernel one; yet, the opposite occurs for the TE.

We repeated the same analysis using real delay data for the Hartsfield-Jackson Atlanta International Airport (ATL) and the Denver International Airport (DEN). Delays of landing flights between years 2015 and 2019 have been obtained from the Reporting Carrier On-Time Performance database of the Bureau of Transportation Statistics, U.S. Department of Transportation, freely accessible at <https://www.transtats.bts.gov>. Time series have been detrended using a Z-Score approach with a period of 24 h; and sub-windows of different lengths have randomly been extracted. Fig. 4 then reports the results in the same format as Fig. 3 above.

As ought to be expected, the results with real data are more complex to interpret. On the one hand, linear approaches underperform when compared to what obtained in Fig. 3; for instance, both GC and LC barely reach 50% of significant fractions. This is possibly due to the non-linear nature of the underlying relationships, which in turn requires longer time series to be correctly estimated. It is also noteworthy the behaviour of the TE with ordinal estimator, which yields good results only for short time series. This may be due to the presence of short-term and non-linear patterns, which are nevertheless lost (or diluted) when considering multiple weeks of data.

In short, the practitioner must be aware that “one size does not fit all”: different connectivity measures can reveal different patterns

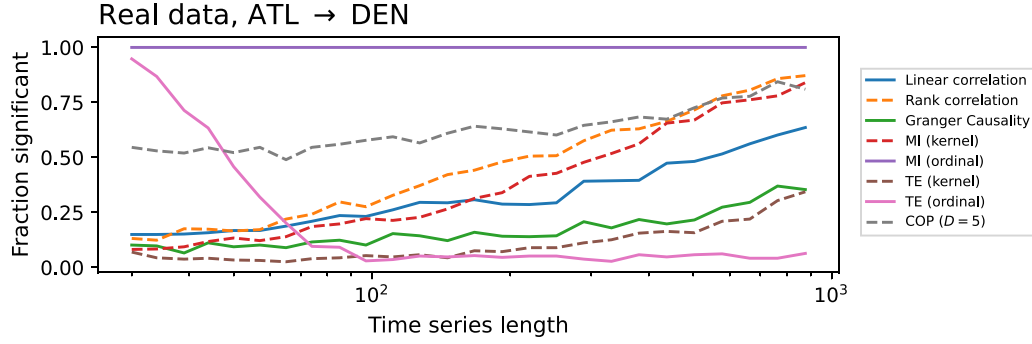


Fig. 4. Comparison of different connectivity measures as a function of the time series length, on real data for the Atlanta (ATL) and Denver (DEN) airports. See the legend for a list of measures, with the corresponding estimator within parentheses. Results correspond to 500 randomly extracted time series.

of delays, at the same time having different requirements in terms of data volumes. This highlights the importance of using multiple complementary approaches to get a complete understanding of the system's dynamics.

2.4. Network reconstruction

Once connectivity measures have been calculated between all pairs of components in the transportation system, the next step is to reconstruct the functional delay network. This network represents how delays are connected throughout the system, with nodes representing transportation components, e.g. airports or train stations, and edges representing the relationships between them. In a simplified interpretation, these edges would be interpreted as propagation patterns, although the limitations described above apply.

The output is twofold. On the one hand, it includes a weight matrix W , where each element w_{ij} reports the p -value between components i and j . Note that self-connections are usually discarded, i.e. $w_{ij} = 1.0$ for $i = j$. On the other hand, it also includes a lag matrix L , where each element l_{ij} represents the time lag at which the strongest connection was found, as illustrated in Section 2.3.1. This last matrix can be used to describe the time scale of the underlying relationship or propagation, and hence to discriminate between direct and potentially indirect ones — see for instance Wang et al. (2022), Pastorino and Zanin (2023).

The framework reconstructs networks that are directed by design, i.e. $w_{ij} \neq w_{ji}$, reflecting the inherently directional nature of delay propagation. This is naturally obtained in the case of causality tests (including GC and TE), where such directionality is inherent to their definition. In contrast, similar results are obtained by shifting time series in time for correlation metrics (LC, RC and MI). To illustrate, delays at node i affecting node j after τ hours represent a fundamentally different relationship than delays at node j affecting node i after a similar number of hours.

In order to discard those relationships that are not statistically significant, two alternatives can be employed:

- **Statistical Thresholding.** This approach uses a significance level (e.g., $\alpha = 0.05$ or $\alpha = 0.01$) to determine which connections to include:

$$A_{ij} = \begin{cases} 1 & \text{if } w_{ij} < \alpha \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

Where A is the resulting binary adjacency matrix of the network. This approach ensures that only statistically significant connections are included, but it does not account for multiple testing issues.

- **False Discovery Rate (FDR) Control.** This method adjusts the threshold to control the expected proportion of false positives among all detected connections:

$$A_{ij} = \begin{cases} 1 & \text{if } w_{ij} < \alpha_{\text{FDR}} \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

α_{FDR} is determined using procedures like the Benjamini–Hochberg method. FDR control is particularly important when analysing large transportation networks with many potential connections.

2.4.1. Example of reconstructed networks

As an example of the reconstruction of adjacency matrices, we consider again the data of the Bureau of Transportation Statistics, this time for the top-15 airports (in terms of the total number of operations) and for year 2019. From it, we extracted time series of average hourly landing delays, and detrended them using the Z-Score approach (with a periodicity of 24 h). We finally reconstructed the adjacency matrices using three different connectivity measures, and filtered the network using a Bonferroni correction for $\alpha = 0.01$. The resulting networks and adjacency matrices are depicted in Fig. 5, for Granger Causality (left), COP (centre), and Transfer Entropy (right).

Several interesting facts can be observed. First of all, all three networks are highly connected. This is to be expected, as we only considered very large airports that are highly interconnected, and that can easily spread delays between them — see Li and Jing (2021) for a discussion on the role of large airports in the US, and especially for the correlation between size of an airport and its importance for the propagation. Secondly, the networks reconstructed with the COP and TE measures are denser than the one obtained by the GC; this may indicate that many relationships have a non-linear nature, which the latter test is ill-designed to detect. Thirdly, the GC and TE seem to mostly detect relationships in short time scales; larger lags (represented by dark links) are more frequent in the COP network.

A word of caution should be raised regarding the high number of propagation instances with lags equal to one hour, i.e. on time scales shorter than the duration of the corresponding flight. Several reasons may be behind this. On the one hand, airlines may adapt their operations using information about the delay of flights that have yet to land, for instance deciding not to recover a delay when a connecting flight is also known to be delayed — something known as Dynamic Cost Indexing (Cook et al., 2009; Delgado et al., 2016). Notably, this implies that what is here measured is the propagation of information about delays, and not delays themselves; a fact that may even create paradoxical situations, like propagations going backwards in time (Wang et al., 2022; Pastorino and Zanin, 2023). On the other hand, such short propagation instances may be due to disruptions happening concurrently at several airports, as e.g. global adverse weather conditions; or, in other words, to the presence of confounding effects. This can be confirmed by repeating the analysis over shorter time windows,

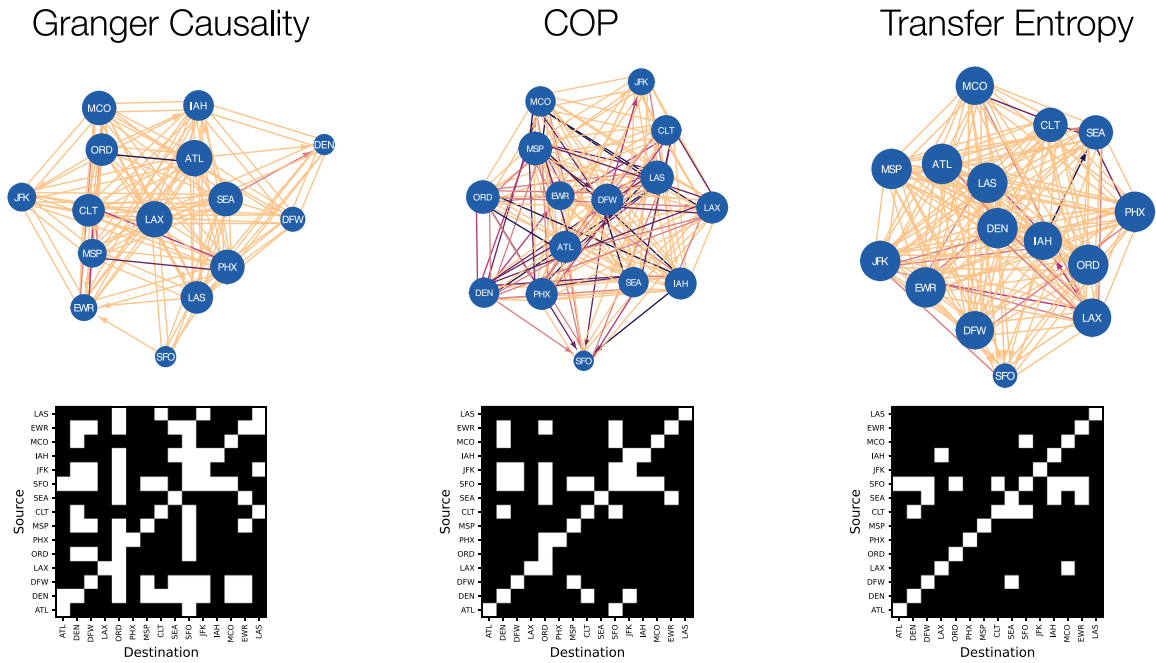


Fig. 5. Functional networks corresponding to the top-15 US airports, using data for year 2019. Left, centre and right networks have been reconstructed using respectively Granger Causality, COP, and Transfer Entropy (with a box kernel). The top pictures depict the network, with node sizes proportional to the out-degree, and the colour of link indicating the optimal lag (from 1 h, light shade, to 6 h, dark shade). The bottom graphs report the corresponding adjacency matrices; black squares indicate the presence of a causal connection between the corresponding airports, white squares the absence thereof.

ideally in periods in which no such events occurred; and can be solved by deleting propagation links that are physically unfeasible. In short, this simple example illustrates the importance of a correct measure selection; the need of carefully inspecting the results for soundness; and that caution should be exercised in interpreting relationships as propagation instances.

2.4.2. On the interpretation of p -values

As previously detailed and further illustrated in Fig. 5, the result of the analysis in an unweighted network, i.e. each link either exists or does not — in other words, each element A_{ij} of the adjacency matrix can only be zero or one. While alternatives are available, these have to be evaluated with caution under the light of the output of the analysis.

It is firstly tempting to consider the obtained p -value as a metric of the strength of the relationship. One should nevertheless recall that the p -value of a test only represents the statistical significance of results, i.e. how frequently a relation of the same (or larger) magnitude is observed under the null hypothesis; to illustrate, in the case of the GC, it is assessing whether the causing time series is reducing the error in the forecast of the caused one in a significant way. It is therefore only a way to accept or reject the presence of a relationship (here, of a delay propagation). Again in the case of the GC, a smaller p -value indicates that the prediction is better, but this is also affected by aspects like the data size and their probability distribution. For instance, consider two time series of length 1000, $x(t) = \mathcal{N}(0, 1)$ and $y(t) = x(t - 1) + \mathcal{N}(0, 1)$, with $\mathcal{N}$ representing numbers drawn from a normal distribution. Due to the delayed coupling, a GC test correctly detects the presence of a causality $x \rightarrow y$ with a median p -value of $\approx 10^{-150}$. Now let us increase the standard deviation of the second time series, i.e. $y(t) = x(t - 1) + \mathcal{N}(0, 2)$; the resulting median p -value drops to $\approx 10^{-50}$, even though the intensity of the coupling is exactly the same. In short, interpreting a p -value as a proxy of the intensity of the relationship can only be done *caeteris paribus*, i.e. everything else being equal.

Secondly, it is customary to apply a small threshold for the statistical significance, e.g. $\alpha = 0.05$ or 0.01 , as this ensures a minimal number of

false positives (i.e. instances in which the measure detects a relationship that is actually not there). Yet, this also implies that some true propagation links are deleted by the filtering — i.e. false negatives may appear. A more truthful representation of the system may be obtained by balancing the number of true and false positives, for instance by increasing the value of α . To the best of our knowledge, this has only been studied once (Acharya et al., 2024), and no strategies for setting the best α in this context have hitherto been proposed.

2.5. Network analysis

Once the functional networks have been reconstructed, the literature provides a large number of topological metrics to characterise them, i.e. measures describing specific properties or structural features (Costa et al., 2007). These metrics help to identify critical components, understand global connectivity patterns, and inform strategies for improving system resilience. For the sake of clarity, we here describe some foundational ones, organised in families in what follows.

2.5.1. Macroscale properties

- **Link Density.** The proportion of possible connections that are actually present in the network:

$$D = \frac{|E|}{|V|(|V| - 1)}, \quad (11)$$

where $|E|$ is the number of edges and $|V|$ is the number of nodes. For directed networks, this represents the ratio of existing connections to the maximum possible number of connections, i.e. $n(n-1)$. A high density indicates extensive relationships throughout the system, hence more chances for interdependencies.

- **Global Efficiency.** This metric quantifies how efficiently information can propagate through the network, calculated as the average of the inverse of the shortest path length between all pairs of nodes (Latora and Marchiori, 2001):

$$E_{glob} = \frac{1}{|V|(|V| - 1)} \sum_{i \neq j} \frac{1}{d_{ij}}, \quad (12)$$

where d_{ij} is the shortest path length between nodes i and j . Values range from 0, completely disconnected, to 1, perfectly efficient. Higher values imply that a delay generated in one part of the network can easily affect other regions.

- **Transitivity.** Also known as the global clustering coefficient, transitivity measures the tendency of the network to form clusters, calculated as the fraction of all possible triangles present in the graph:

$$T = 3 \frac{\text{number of triangles}}{\text{number of triads}}. \quad (13)$$

Triangles here refer to groups of three nodes fully connected between them, while triads are sets of three nodes with at least two edges between them. Hence, high values of the transitivity (i.e. close to one) indicate that the neighbours of a node tend to be connected to each other. In the context of delay networks, high transitivity may indicate the presence of regional clusters, or of groups of elements that tightly share delays. Note that for directed networks, the direction of edges is usually ignored when calculating transitivity, as the concept of triangles is not uniquely defined in these networks.

- **Reciprocity.** This metric measures the tendency of pairs of nodes to form mutual connections in a directed network; or, in other words, whether a delay pattern $a \rightarrow b$ tends to be associated to an opposite pattern $b \rightarrow a$. It is defined as the fraction of edges that are reciprocated:

$$R = \frac{1}{|E|} \sum_{i,j} A_{i,j} A_{j,i} = \frac{1}{|E|} \text{Tr } A^2, \quad (14)$$

with A being the adjacency matrix of the graph. Note that $A_{i,j} A_{j,i} = 1$ if and only if i connects to j and vice versa. Reciprocity is only defined for directed networks, as in undirected networks all connections are reciprocal by definition.

2.5.2. Centrality measures

Centrality measures aim at identifying the most important nodes in the network, or, in other words, to assign a relative importance to each node, which can then support their ranking. Note that this is an ill-defined concept, as importance can have different interpretations in different contexts; as a consequence, a large number of complementary centrality measures have been proposed in the past (Batool and Niazi, 2014; Grando et al., 2016; Saxena and Iyengar, 2020). *delaynet* includes the following ones:

- **Betweenness Centrality.** This measure quantifies the extent to which a node lies within the shortest paths between other pairs of nodes (Freeman, 1977; Brandes, 2008):

$$C_B(i) = \sum_{s \neq i \neq t} \frac{\sigma_{st}(i)}{\sigma_{st}}. \quad (15)$$

Here, σ_{st} is the number of shortest paths from node s to node t , and $\sigma_{st}(i)$ is the number of those paths that pass through node i . Nodes with high betweenness centrality act as bridges or bottlenecks in the functional network; when links are interpreted as propagation instances, their disruption, for instance by increasing the resources available to that portion of the system, may result in widespread benefits.

- **Eigenvector Centrality.** This measure assigns a relative score to each node that is proportional to the score of its neighbours:

$$C_E(i) = \frac{1}{\lambda} \sum_j A_{i,j} C_E(j), \quad (16)$$

where λ is the largest eigenvalue of the adjacency matrix A . In other words, a node is as central as its neighbours are. The previous equation indicates that the centrality of nodes is given by the eigenvector of the largest eigenvalue of the adjacency matrix A , hence the name. In a delay network, nodes with high eigenvector centrality derive their importance from the fact of being connected with other central nodes, thus potentially forming a core.

2.5.3. Isolated nodes analysis

In the case of delay functional networks, isolated nodes, i.e. nodes whose delays are not functionally dependent on other nodes, are a special and very interesting case. For instance, whenever such nodes exist, it may be relevant to understand which properties (in terms of e.g. availability of resources) is supporting their independence, for potentially replicating this across the network. This can be measured by a two-fold metric: the number of **Isolated Nodes (Inbound/Outbound)**:

$$I_{in} = \sum_{i=1}^{|V|} \mathbf{1} \left(\sum_{j=1}^{|V|} A_{ji} = 0 \right), \quad (17)$$

and

$$I_{out} = \sum_{i=1}^{|V|} \mathbf{1} \left(\sum_{j=1}^{|V|} A_{ij} = 0 \right). \quad (18)$$

In both cases, $\mathbf{1}(\cdot)$ is the indicator function that equals one when the condition is true, and zero otherwise.

2.5.4. Temporal evolution of topological metrics

Transportation networks and their delay patterns can change over time due to seasonal variations in demand, changes in scheduling or operations, infrastructure developments, or external disruptions like weather events and industrial actions. It is therefore interesting not only to analyse general (or stable) patterns; but also how those evolve with time, to identify common trends and recurring structures. The previously described topological metrics can be calculated over any functional network; it is thus trivial to evaluate their evolution through time, by changing the corresponding time window of the data used in their reconstruction.

To illustrate this point, Fig. 6 depicts the evolution over time of eight topological metrics previously described, for the data of the top-15 US airports from 2015 to 2019. The corresponding functional networks have been extracted using a rolling window of 60 days of length, shifted of one week for each point; time series have further been detrended using the Z-Score method with a periodicity of 24 h. Several interesting trends can be observed. As also obtained in Figs. 4 and 5, the COP (orange lines) seems to detect more functional links than the GC (blue); the latter is also more sensitive to changes over time, with larger variations in metrics like the efficiency and number of isolated nodes.

2.5.5. Normalisation of topological metrics

The analysis of the topological metrics of Fig. 6 has one important limitation: the values are not normalised according to the characteristics of the network under study. Let us explain this point by using the efficiency as an example. This topological metric strongly depends on the number of nodes and links in the network — as by definition denser networks will be able to propagate information in a more efficient way. Even assuming a constant number of nodes, this implies that the efficiency of two networks cannot directly be compared. To solve this, it is customary to use a normalisation procedure, in which the observed value of the topological metric is transformed using a Z-Score, calculated against the values of the same metric obtained in ensembles of random equivalent (i.e. same number of nodes and links) graphs (Zanin et al., 2018).

To illustrate this point, Fig. 7 depicts the evolution of three metrics previously reported in Fig. 6, this time normalised using a Z-Score. Let us consider the transitivity as an illustrative example. When considering the raw metrics, i.e. Fig. 6, the transitivity for GC and TE were essentially the same — average and standard deviation of respectively 0.554 ± 0.144 and 0.594 ± 0.143 . On the other hand, the Z-Score normalised counterparts have substantially different means: 2.027 ± 1.711 for GC, and 0.009 ± 1.376 for TE. The metric normalisation can thus help in making explicit topological differences that arise when using multiple connectivity metrics.

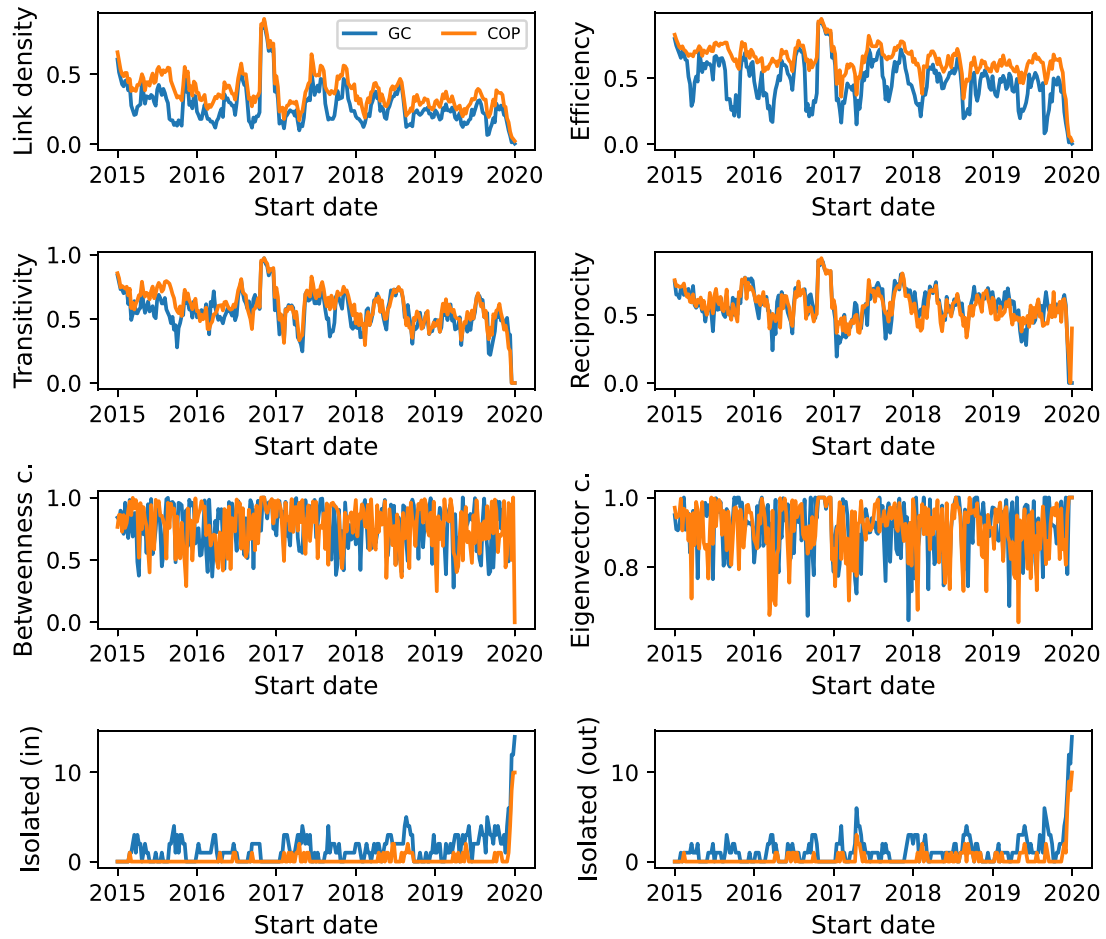


Fig. 6. Example of the evolution of eight topological metrics described in Section 2.5, for the top-15 US airports, calculated using a rolling window of 60 days. Blue and orange lines respectively correspond to networks reconstructed using GC and COP.

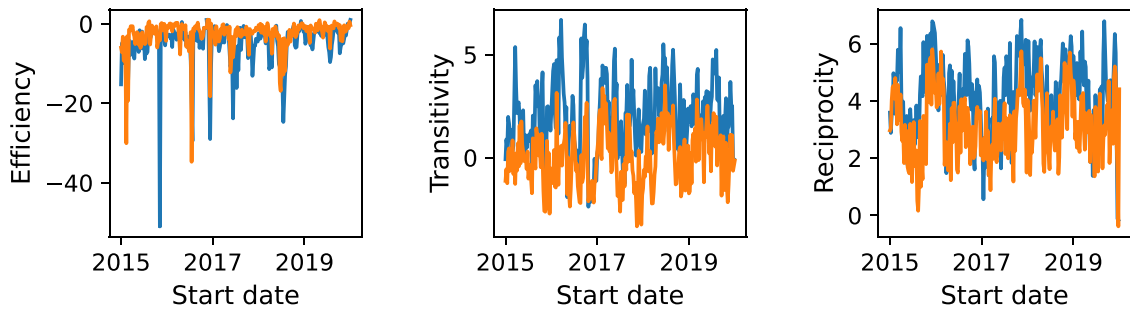


Fig. 7. Temporal evolution of three metrics of Fig. 6, normalised using a Z-Score and ensembles of 1000 random equivalent networks. Blue and orange lines respectively correspond to networks reconstructed using GC and COP.

In short, the practitioner reaching this stage has to take into account the kind of information that they want to obtain from the analysis. Whenever the aim is to simply describe the structure, e.g. to assess how many triangles are present, the raw (non-normalised) metrics are the correct option. On the other hand, if the goal is to compare networks, and furthermore understand if their structure is fundamentally different beyond what is consequence of a different link density, then the normalised version is a better solution.

3. The *delaynet* package: Implementation and usage

The *delaynet* package provides a complete implementation of the delay analysis framework described in Section 2, offering researchers

and practitioners a powerful set of tools for analysing delay relationships in transportation networks. The package is available on [PyPI](#) and [conda-forge](#), making it easily accessible to the research community. The package is developed following best coding practices, adhering to open-source principles, and includes a comprehensive Code of Conduct (https://github.com/cbueh/delaynet/blob/main/CODE_OF_CONDUCT.md) and Contributing Guidelines (<https://github.com/cbueh/delaynet/blob/main/CONTRIBUTING.md>), fostering a welcoming and collaborative community.

3.1. Package overview

The *delaynet* package is designed with a modular architecture that closely follows the framework illustrated in Fig. 1. This modular design

Listing 1: Importing the delaynet package and printing its version.

```

1 import delaynet as dn
2 print(dn.__version__) # Check version

```

allows users to select appropriate methods for each stage of the analysis process, tailoring the approach to their specific data characteristics and research objectives; and, whenever needed, use their own implementation of these methods. The package is organised into several core modules, each corresponding to a key component of the delay analysis framework:

- `delaynet.preparation`: Tools for data preparation and synthetic data generation.
- `delaynet.detrrending_methods`: Implementation of various detrending techniques.
- `delaynet.connectivities`: Collection of connectivity measures for detecting relationships between time series.
- `delaynet.network_reconstruction`: Functions for reconstructing networks from time series data.
- `delaynet.network_analysis`: Tools for analysing and interpreting reconstructed networks.
- `delaynet.utils`: Utility functions supporting the core functionality.

The package can be imported, and its version checked using the code shown in listing 1.

3.2. Data generation and preparation

The starting point of the analysis is a set of time series for each node in the network, which must be complete and of sufficient quality standards, as discussed in Section 2.1. For validation and testing of the analytical pipeline, the package supports synthetic data generation models, which are described below and are illustrated in listing 2.

Firstly, synthetic random time series with explicit causal relations between them can be generated through delayed causal networks. This method constructs networks through the systematic generation of adjacency matrices defining the network topology, weight matrices specifying connection strengths, and lag matrices determining the propagation of information. Data are then synthesised by drawing random values, and further adding delayed contributions from connected time series. The resulting time series exhibit controlled functional delay patterns where connections are activated probabilistically, simulating the sporadic nature of real transportation delay influences.

Secondly, the package incorporates the generation of transportation-specific synthetic time series through integration with the specialised simulation tool *SynthATDelays* (Büth and Zanin, 2025). Unlike comprehensive air transport simulators that aim for maximum realism at high computational cost, this method focuses on generating highly tuneable scenarios to test specific conditions and hypotheses. The *delaynet* package provides direct access to two predefined *SynthATDelays* scenarios:

- **Random connectivity scenario.** Simulates a set of airports randomly connected by independent flights with random and homogeneous en-route delays. This scenario allows customisation of the number of airports, number of aircraft, and buffer time between operations.
- **Independent operations with trends.** Creates two groups of two airports, in which flights connect airports within the same group but not across them. When trends are activated, delays are added at specific hours, generating spurious causality relations between airports despite no actual propagation pathways between groups.

The functions used in listing 2 are:

- **Delayed Causal Network (DCN):** The `gen_delayed_causal_network` function creates time series with explicit causal relationships and temporal delays between nodes, returning the adjacency matrix, weight matrix, and generated time series.
- **SynthATDelays:** Two specialised functions provide access to aviation-specific synthetic delay generation:

- `gen_synthatdelays_random_connectivity`: Implements the Random Connectivity scenario with customisable parameters for airports, aircraft, and buffer times.
- `gen_synthatdelays_independent_operations_with_trends`: Implements the Independent Operations with Trends scenario that can generate spurious causality relations.

These synthetic data generation capabilities enable researchers to validate their analytical pipelines against known ground truth, compare the performance of different connectivity measures under controlled conditions, and explore the impact of various parameters on delay propagation detection. The *SynthATDelays* integration is particularly valuable for transportation research, as it provides realistic delay patterns based on a simplified but accurate model of air transportation operations (Büth and Zanin, 2025).

3.3. Detrending algorithms

Time series detrending is an essential preparatory step that helps standardise data, remove trends, and make different time series comparable. The *delaynet* package provides a consistent interface for applying various detrending methods to time series data, including delta, second difference, and Z-Score detrending — see Section 2.2 for definitions. Examples of the corresponding code are reported in listing 3, and the list of available algorithms and parameters is presented in Table 1.

The unified interface allows users to easily experiment with different detrending methods and select the most appropriate one for the specific data characteristics and research questions. Switching detrending approaches only requires selecting the wanted method and passing its parameters, if needed. The input and output formats stay the same, meaning any following analysis — calculations and plotting — can be reused without modification.

3.4. Connectivity analysis

After detrending, the next step is to quantify the relationships between delay time series of different components of the transportation network. The *delaynet* package implements various connectivity measures that can detect and measure how delays in one part of the network influence delays in other parts, accounting for the time it takes for these influences to propagate.

The package provides a simple interface for applying connectivity measures, as illustrated in listing 4. The implemented measures include Linear Correlation (LC), Rank Correlation (RC), Granger Causality (GC), Mutual Information (MI), Transfer Entropy (TE), and Continuous Ordinal Patterns (COP) — see Section 2.3 for definitions and details. Note that each measure has some specific parameters that can be defined by the user, and which are listed in Table 2.

For each connectivity measure, the package determines the optimal lag by systematically testing different time lags and selecting the one that produces the strongest relationship, which is always determined by the lowest *p*-value. This process is handled by the `find_optimal_lag` utility function, which ensures consistent behaviour across all connectivity measures.

Listing 2: Generation of synthetic time series data using both the DCN method and the SynthATDelays package.

```

1  from delaynet.preparation import (
2      gen_delayed_causal_network, gen_synthatdelays_random_connectivity,
3      gen_synthatdelays_independent_operations_with_trends
4  )
5
6  # Generate synthetic data using Delayed Causal Network
7  adjacency_matrix, weight_matrix, dcn_time_series = gen_delayed_causal_network(
8      ts_len=1000,                # Length of time series
9      n_nodes=5,                  # Number of nodes
10     l_dens=0.3,                  # Density of the adjacency matrix
11     wm_min_max=(0.5, 1.5),       # Min and max of the weight matrix
12     rng=default_rng(1249687)    # Fixed seed
13 )
14
15 # Generate synthetic data using SynthATDelays Random Connectivity scenario
16 random_conn_results = gen_synthatdelays_random_connectivity(
17     sim_time=5,                  # Simulation time in days
18     num_airports=6,              # Number of airports
19     num_aircraft=40,             # Number of aircraft
20     buffer_time=0.25,            # Buffer time between operations in hours
21     seed=0                       # Random seed for reproducibility
22 )
23
24 # Extract the average arrival delay matrix
25 arrival_delays = random_conn_results.avgArrivalDelay
26
27 # Generate synthetic data using SynthATDelays Independent Operations with Trends scenario
28 trends_results = gen_synthatdelays_independent_operations_with_trends(
29     sim_time=5,                  # Simulation time in days
30     activate_trend=True,         # Activate trends at specific hours
31     seed=0                      # Random seed for reproducibility
32 )
33
34 # Extract the average departure delay matrix
35 departure_delays = trends_results.avgDepartureDelay

```

Listing 3: Examples of time series detrending using different methods and parameters.

```

1  # Detrend using the delta method
2  detrended_node = dn.detrend(single_time_series, method="delta", window_size=10)
3
4  # Detrend using the Z-Score method, multiple series at once
5  detrended_nodes = dn.detrend(many_time_series, method="zs", periodicity=24, axis=1)

```

Listing 4: Example of the calculation of the Granger Causality between two time series.

```

1  # Calculate Granger causality between two time series
2  p_value, optimal_lag = dn.connectivity(
3      ts1, ts2, connectivity_measure="gc", lag_steps=10
4  )

```

Table 1

List of detrending algorithms included in *delaynet*. The second column reports the primary identifier to be used in the *method* parameter of the *dn.detrend* function, while the third column lists all additional parameters that can be defined. Additional method aliases are supported; see [detrending documentation](#) for the complete identifier list.

Algorithm	Identifier	Parameters
Identity	identity	–
Delta	delta	<i>window_size</i> : window size to use for calculating the mean.
Second Difference	2dt	–
Z-Score	zs	<i>periodicity</i> : expected periodicity of the time series, i.e. reoccurrence of the same pattern. <i>max_periods</i> : maximum number of periods to consider before and after the current value.

Table 2

List of connectivity measures included in *delaynet*. The second column reports the primary identifier to be used in the *connectivity_measure* parameter of the *dn.connectivity* function, while the third column lists all additional parameters that can be defined. Additional method aliases are supported; see [measure documentation](#) for the complete identifier list.

Measure name	Identifier	Parameters
Linear correlation	lc	<i>lag_steps</i> : time lags to consider.
Rank correlation	rc	<i>lag_steps</i> : time lags to consider.
Granger Causality	gc	<i>lag_steps</i> : time lags to consider.
Mutual Information	mi	<i>lag_steps</i> : time lags to consider. <i>approach</i> : approach to use. <i>hypothesis_type</i> : type of hypothesis test to use, either “permutation_test” or “bootstrap”. <i>n_tests</i> : number of iterations or resamples to perform within the hypothesis test. <i>**mi_kwargs</i> : additional estimator parameters.
Transfer Entropy	te	<i>lag_steps</i> : time lags to consider. <i>approach</i> : approach to use. <i>hypothesis_type</i> : type of hypothesis test to use, either “permutation_test” or “bootstrap”. <i>n_tests</i> : number of iterations or resamples to perform within the hypothesis test. <i>**te_kwargs</i> : additional estimator parameters.
Continuous Ordinal Patterns	cop	<i>lag_steps</i> : time lags to consider. <i>p_size</i> : size of the ordinal pattern. <i>num_rand_patterns</i> : number of random patterns to consider. <i>linear</i> : start with the identity pattern.

Listing 5: Network reconstruction from time series, with a statistical pruning of connections.

```

1 # Reconstruct network from time series
2 weights, lags = dn.reconstruct_network(
3     many_time_series, connectivity_measure="gc", lag_steps=10
4 )
5
6 # Apply statistical pruning
7 significant_connections = dn.network_analysis.statistical_pruning(
8     weights, alpha=0.05, correction='fdr_bh'
9 )

```

3.5. Network reconstruction

The network reconstruction involves applying connectivity measures to each pair of time series in the dataset, resulting in a matrix of p -values and a matrix of optimal lags. The *delaynet* package provides the *reconstruct_network* function to streamline this process, as demonstrated in listing 5.

Note that the *reconstruct_network* function internally includes three steps, namely: the pairwise analysis, which, for each node pair (i, j) , computes the p -values $p(X_i, X_j, \tau)$ for the connectivity measure across lags $\tau = 1, \dots, \tau_{\max}$ or a specified list of lags; the lag optimisation, which selects the lag yielding the most significant connection $\tau_{ij}^* = \arg \min_{\tau} p(X_i, X_j, \tau)$; and finally the matrix reconstruction, which yields the weight matrix $W_{ij} = p(X_i, X_j, \tau_{ij}^*)$ and the lag matrix $L_{ij} = \tau_{ij}^*$.

After the reconstruction, the package provides several thresholding strategies to convert the continuous p -value matrix into a binary adjacency matrix. These include a statistical-thresholding, which retains connections with p -values below a specified significance level; and a False Discovery Rate control, which adjusts the threshold to control the expected proportion of false positives among all detected connections, with support for various correction methods from *statsmodels.stats.multitest.multipletests* (Seabold and Perktold, 2010).

3.6. Network analysis

Once the delay functional network has been reconstructed, the *delaynet* package provides a comprehensive set of network analysis

tools to extract meaningful insights about the structure and dynamics of delays, as shown in listing 6.

The full list of network topological metrics, which have been selected for their relevance in transportation delay networks, includes: link density, global efficiency, transitivity, reciprocity, betweenness centrality, eigenvector centrality, and number of isolated nodes. Beyond these, the output matrices W and L can be analysed by any other network analysis tool, providing flexibility for researchers with specific analytical needs. Additionally, each topological metric can be called with a parameter *normalise*, which normalises the result according to what observed in an ensemble of random equivalent networks — see Section 2.5.5.

3.7. Package availability and documentation

The *delaynet* package is readily available through standard Python package repositories and can be installed using the commands shown in listing 7.

The documentation is available online, see <https://delaynet.readthedocs.io>, and includes up-to-date references and examples. The documentation demonstrates the complete workflow from data preparation to network analysis, providing researchers with a solid foundation for applying the package to their own transportation delay data. The package is versioned with Zenodo, ensuring reproducibility and proper citation in scientific research. Finally, *delaynet* is continuously tested for Python ≥ 3.10 . For the latest updates and detailed usage instructions, users are encouraged to visit the package documentation.

Listing 6: Example of the calculation of topological metrics to analyze the structure of the obtained networks.

```

1 # Calculate network metrics
2 density = dn.network_analysis.link_density(binary_adjacency)
3 efficiency = dn.network_analysis.global_efficiency(binary_adjacency, normalise=True)
4 transitivity = dn.network_analysis.transitivity(binary_adjacency, normalise=True)
5 reciprocity = dn.network_analysis.reciprocity(binary_adjacency, normalise=True)
6 betweenness = dn.network_analysis.betweenness_centrality(binary_adjacency, normalise=True)
7 eigenvector = dn.network_analysis.eigenvector_centrality(binary_adjacency, normalise=True)
8 isolated_in = dn.network_analysis.isolated_nodes_inbound(binary_adjacency)
9 isolated_out = dn.network_analysis.isolated_nodes_outbound(binary_adjacency)

```

Listing 7: Installation commands for the delaynet package using pip or conda package managers.

```

1 pip install delaynet # From PyPI
2 conda install -c conda-forge delaynet # From conda-forge

```

4. Case study: Swiss long-distance rail delays (2022–2025)

We illustrate the *delaynet* workflow on Swiss Federal Railways (SBB) realised timetable data (*istdaten*) accessed via <https://opentransportdat.a.swiss/>. The feed provides, for each stop event, scheduled and realised arrival/departure times with reliability statuses. We exclude cancelled services (FAELLT_AUS_TF) and pass-throughs (DURCHFART_TF) and prioritise REAL over PROGNOSE and GESCHÄTZT (i.e. predicted) when deriving executed times. Delays are computed as realised minus planned times in seconds. We consider data from January 2022 to August 2025, and focus on long-distance SBB-operated services, specifically line prefixes IC, IR, EC, ICE, and TGV, operated by SBB (BETREIBER_ABK = 'SBB') with PRODUKT_ID = Zug. Single-month views presented below refer to August 2025.

From these events we construct station-level arrival-delay time series in fixed 15-minute bins. The 15-minute resolution balances temporal detail against data sparsity: it is of the same order as passenger transfer intervals at major hubs, while retaining more observations per bin than a finer resolution. Five-minute bins are more susceptible to sparsity during low-traffic periods, whereas 30- and 60-minute bins may smooth short-timescale delay dependencies. Appendix C shows that the qualitative evolution of the network characteristics considered here is consistent across these resolutions. For each station and bin we aggregate the delay by summation, yielding a delay matrix $D_{i,s}$ with columns ordered as the top-50 busiest long-distance stations by event volume in July 2025. To reduce the non-stationarity due to daily cycles, we apply the Z-Score detrending with daily periodicity, i.e. periodicity of 96 for 15-minute bins; for a full discussion about the non-stationarity of these time series, and the rationale for using a daily detrending period, see App. A.

We reconstruct a directed functional network of delay associations using *delaynet*'s unified interface. For every ordered station pair we evaluate the Granger Causality across a set of candidate lags $\{1, 2, \dots, 16, 18, 20, 24, 28, 32\}$ (time steps of 15 min, i.e., up to eight hours with intermediate values), and retain the minimum p -value together with its optimal lag. This ensures that the best lag is retained for each pair of stations, thus accounting for the presence of different relationships between time series. Note that increasing this maximum lag has a minimal impact on the results, as discussed in App. B. The resulting p -value matrix is statistically pruned at significance level $\alpha = 0.01$ using Benjamini–Hochberg FDR control. We then compute standard metrics (link density, reciprocity, global efficiency, transitivity) and node centralities (eigenvector, betweenness; all reported as normalised scores in the figures). Community structure is obtained with the Louvain algorithm applied to the undirected (symmetrised) projection of the pruned network. Using 15-minute arrival-delay time series, the August 2025 functional network exhibits a link density of 0.5143; there are 0 nodes without incoming connections and 0 nodes without outgoing connections; the transitivity equals -1.8201 and the reciprocity equals 12.2859. Note that both the transitivity and the

reciprocity are here reported as Z-Scores relative to a random null model, and not as raw ratios (see Section 2.5.5). A high reciprocity Z-Score should be interpreted with care: it only indicates that the number of bidirectional connections is significantly higher than expected in the random baseline network (here, about 12 standard deviations above the ensemble mean), but does not imply that the vast majority of connections in the network are bidirectional. Note further that this significance is much lower than what observed in the underlying physical network, see App. D. Geographic maps are rendered in the Swiss LV95 projection (EPSG:2056) with an OpenRailwayMap overlay for context. In terms of the *delaynet* API, this corresponds to the concise sequence `detrend → reconstruct_network → network_analysis.statistical_pruning → network_analysis.metrics` (link density, reciprocity, global efficiency, transitivity, eigenvector, and betweenness centralities).

The left panel of Fig. 8 presents, for August 2025, the empirical delay distribution at Zürich HB, while the right panel depicts the out-degree distribution of the reconstructed arrival-delay network across the top-50 stations. The former summarises local variability in operational performance, while the latter characterises the heterogeneity of inferred functional associations by node out-degree. For Zürich HB, the arrival-delay distribution is markedly heavy-tailed with strong positive skewness (4.886) and leptokurtosis (excess kurtosis, Fisher of 39.282); after the daily Z-Score detrending, both the skewness and the excess kurtosis increase to respectively 5.947 and 52.106. Across the most skew stations, tail-heaviness varies considerably, e.g.: Basel SBB skewness = 7.325, excess kurtosis = 92.465; Lausanne skewness = 5.950, excess kurtosis = 64.637; Olten skewness = 4.990, excess kurtosis = 44.589; Genève skewness = 16.514, excess kurtosis = 424.648. This indicates that, globally, the majority of trains arrive on time or with small delays, while rare but severe incidents (e.g. accidents, adverse weather, technical failures) generate the pronounced right tails. Genève stands out as particularly volatile (skewness ≈ 16.5 , excess kurtosis ≈ 425), indicating a higher propensity for extreme outliers than at other major hubs. Notably, detrending increases both skewness and kurtosis at Zürich HB, suggesting that once daily cycles are removed, the residual dynamics are even more dominated by rare, high-impact events.

Fig. 9 displays the pruned arrival-delay network on the map, with eigenvector centrality encoded by node colour and betweenness centrality by node size. Eigenvector centrality highlights stations embedded in globally central structures, whereas betweenness emphasises bridge nodes that represent key points of functional association. In August 2025, betweenness is dominated by junctions in the Aargau/Mittelland such as Brugg AG and Olten, with notable bridge roles also at Vevey and Zürich HB; by contrast, eigenvector centrality peaks at Bern and Lausanne, with Olten and Brugg AG also ranking highly. For station abbreviations, full names, community membership, and exact centrality values, see Table 3.

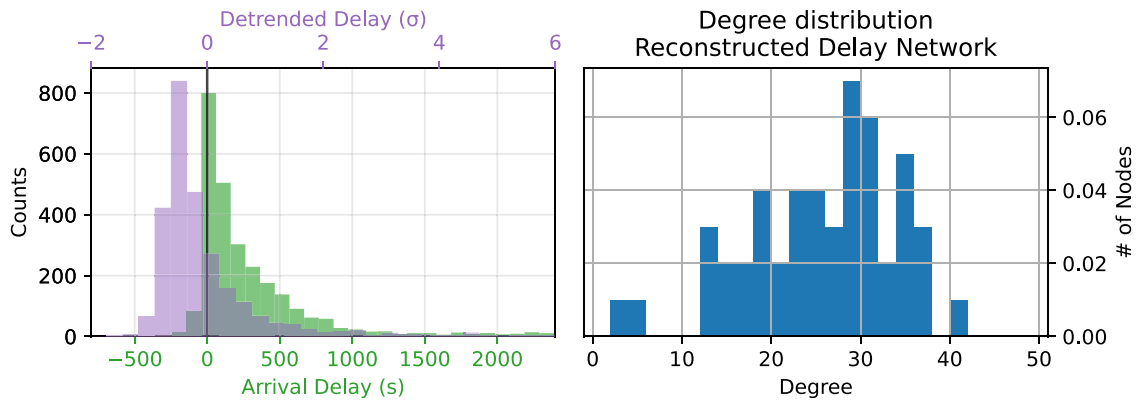


Fig. 8. Analysis of delay data. (Left) Distribution of arrival delays at Zürich HB in August 2025 (15-minute bins). Green and magenta bars respectively correspond to raw (bottom X axis) and detrended (top X axis) delays. (Right) Out-degree distribution of the reconstructed arrival-delay functional network for August 2025 across the top-50 stations, using bars centred between left-inclusive bin edges.

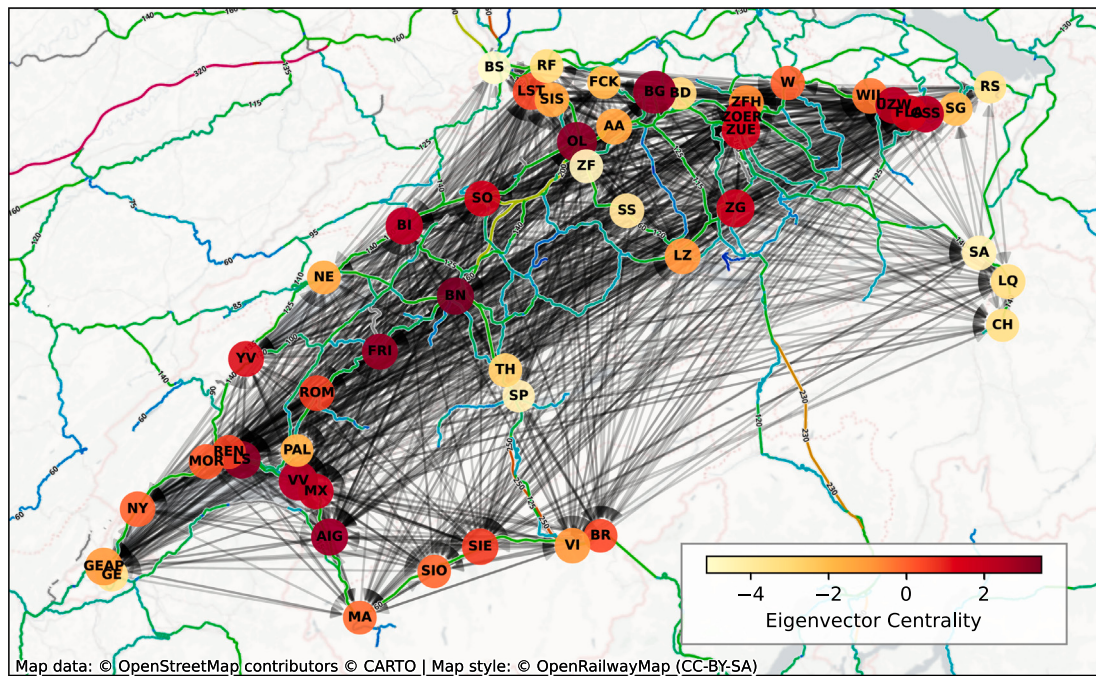


Fig. 9. Pruned arrival-delay functional network for August 2025. Node colour encodes eigenvector centrality (normalised), while node size encodes betweenness centrality (normalised). Map in LV95 (EPSG:2056) with OpenRailwayMap overlay, with colour representing the maximum speed on the rail segment. Basemap data courtesy of OpenStreetMap contributors, CARTO, and OpenRailwayMap (CC-BY-SA). See Table 3 for station names and exact centrality values.

Fig. 10 further shows the community partition of the same network using the Louvain algorithm, computed on the symmetrised graph where the undirected edge weight equals the number of directed links between a pair (0, 1, or 2). The August 2025 partition yields five communities that align with corridor-like structures; see Table 3 for the station composition (abbreviations and full names) sorted by betweenness within each community. These clusters seem to indicate that delays are preferentially associated along main operating corridors; at the same time, the presence of nodes like ZOER (Zürich Oerlikon) in the Lémanic-Fribourg community highlights non-local ties created by through-services and long-range dependencies. The practitioner should nevertheless exercise caution when interpreting these results. It is known that stochastic modularity maximisation algorithms, like the Louvain one, can yield inconsistent results, which are further dependent on the FDR pruning procedure — see App. C for a detailed discussion. A full evaluation of the community structure would therefore require the comparison of the solution yielded by

multiple algorithms, to achieve a consensus viewpoint; and eventually the integration of external information about operations.

Finally, Fig. 11 depicts the evolution of some topological metrics through time. These have been estimated using a rolling-window approach: networks are reconstructed on 28-day windows sliding weekly (Monday–Sunday), for then calculating the corresponding metrics — see also App. C for an analysis of the sensitivity to the sliding window size. While *prima facie* the depicted evolutions seem random, some patterns emerge. To illustrate, the link density and the reciprocity for the arrival delay functional networks present spikes during the last weeks of each year; while, on the other hand, the transitivity drops during those same dates. Note that the transitivity and the reciprocity here reported correspond to the normalised versions of both metrics (see Section 2.5.5), hence their variations are not a direct consequence of the change in the number of links. On the contrary, it may be concluded that these changes are the result of an increase in traffic during the holiday season; the higher level of stress imposed on the

Table 3

Network centrality measures by community in the Swiss railway delay network (August 2025). Communities are identified using the Louvain algorithm on the symmetrised adjacency matrix. Station abbreviations correspond to those shown in Figs. 9 and 10; full station names are provided for reference. Centrality values correspond to normalised scores, with colour coding indicating relative magnitudes within each measure.

Abbr.	Station	Betw.	Eigenv.
Community 1 (15 stations)			
BG	Brugg AG	12.643	3.088
OL	Olten	7.315	3.198
ZG	Zug	4.180	1.679
BI	Biel/Bienne	3.019	2.144
BN	Bern	2.567	3.464
SO	Solothurn	0.798	1.571
YV	Yverdon-les-Bains	0.528	1.426
ROM	Romont FR	0.105	0.568
REN	Renens VD	-2.131	0.444
NE	Neuchâtel	-2.418	-1.490
RF	Rheinfelden	-2.655	-3.235
TH	Thun	-2.677	-2.862
FKK	Frick	-3.051	-2.090
BD	Baden	-4.180	-3.302
SP	Spiez	-4.386	-4.588
Community 3 (11 stations)			
ZUE	Zürich HB	4.871	1.511
LST	Liestal	1.712	0.447
AA	Aarau	0.839	-1.479
LZ	Luzern	0.656	-1.031
SS	Sursee	-1.001	-3.964
LQ	Landquart	-1.034	-3.330
SIS	Sissach	-1.269	-1.479
SA	Sargans	-1.454	-4.540
ZF	Zofingen	-2.420	-4.882
CH	Chur	-3.354	-3.396
BS	Basel SBB	-3.694	-5.135
Community 0 (8 stations)			
VV	Vevey	5.684	2.394
AIG	Aigle	2.499	2.906
SIE	Sierre/Siders	1.625	0.683
MX	Montreux	0.100	1.835
VI	Visp	0.098	-0.989
MA	Martigny	-2.552	-0.143
BR	Brig	-2.585	0.448
SIO	Sion	-2.831	-0.063
Community 2 (8 stations)			
GEAP	Genève-Aéroport	3.607	-1.005
LS	Lausanne	3.100	3.227
FRI	Fribourg/Freiburg	0.872	2.964
NY	Nyon	0.144	-0.013
MOR	Morges	-0.583	0.089
ZOER	Zürich Oerlikon	-1.001	1.514
PAL	Palézieux	-3.788	-1.814
GE	Genève	-3.854	-3.561
Community 4 (8 stations)			
GSS	Gossau SG	2.038	2.275
UZW	Uzwil	1.706	2.231
FLA	Flawil	0.554	1.577
WIL	Wil SG	0.487	-0.432
ZFH	Zürich Flughafen	-0.776	-0.805
W	Winterthur	-1.793	0.011
SG	St. Gallen	-2.610	-2.137
RS	Rorschach	-4.390	-3.613

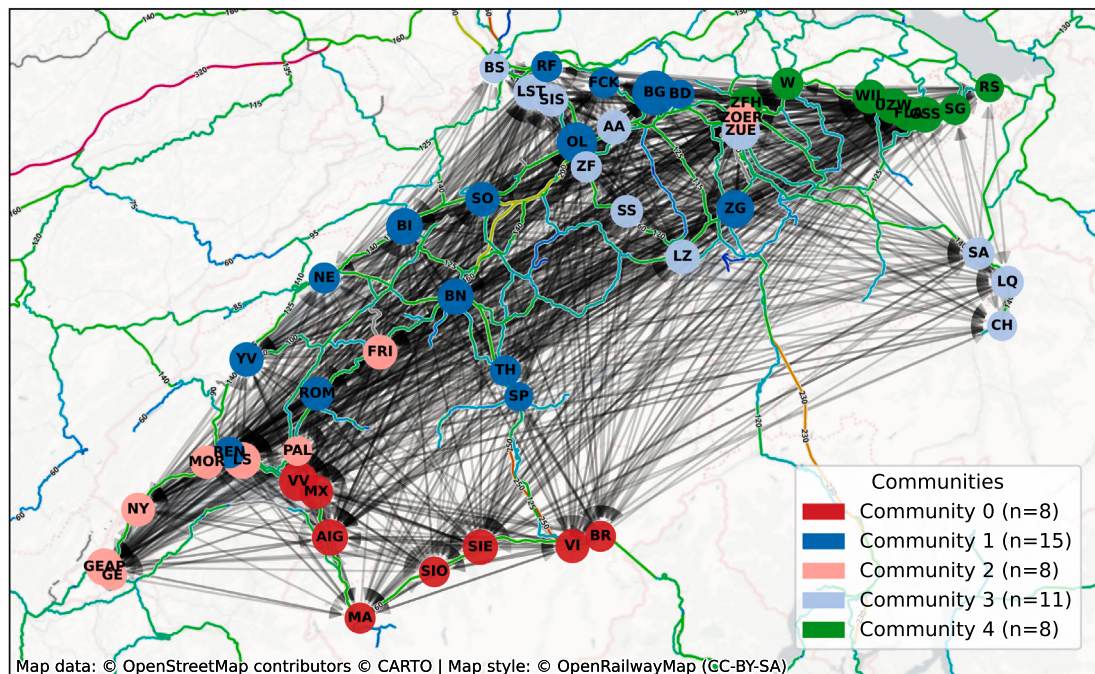


Fig. 10. Louvain communities of the August 2025 network, computed on the symmetrised adjacency with undirected weights equal to the number of directed links (0/1/2). Communities reflect corridor-like structures (abbreviations shown). Projection and basemap as in Fig. 9. Station names provided in Table 3.

available resources implies that delays become more interconnected, resulting in a denser, yet less structured, functional network. Still, as in the previous case, the complete interpretation of these results would require operational information beyond what here presented. It

is finally worth noting some abnormal values, specifically for the Z-Score of the Global Efficiency (see top right panel of Fig. 11). These are due to the high link density of the functional network, which results in a limited variability in the metric value for the random ensemble, and

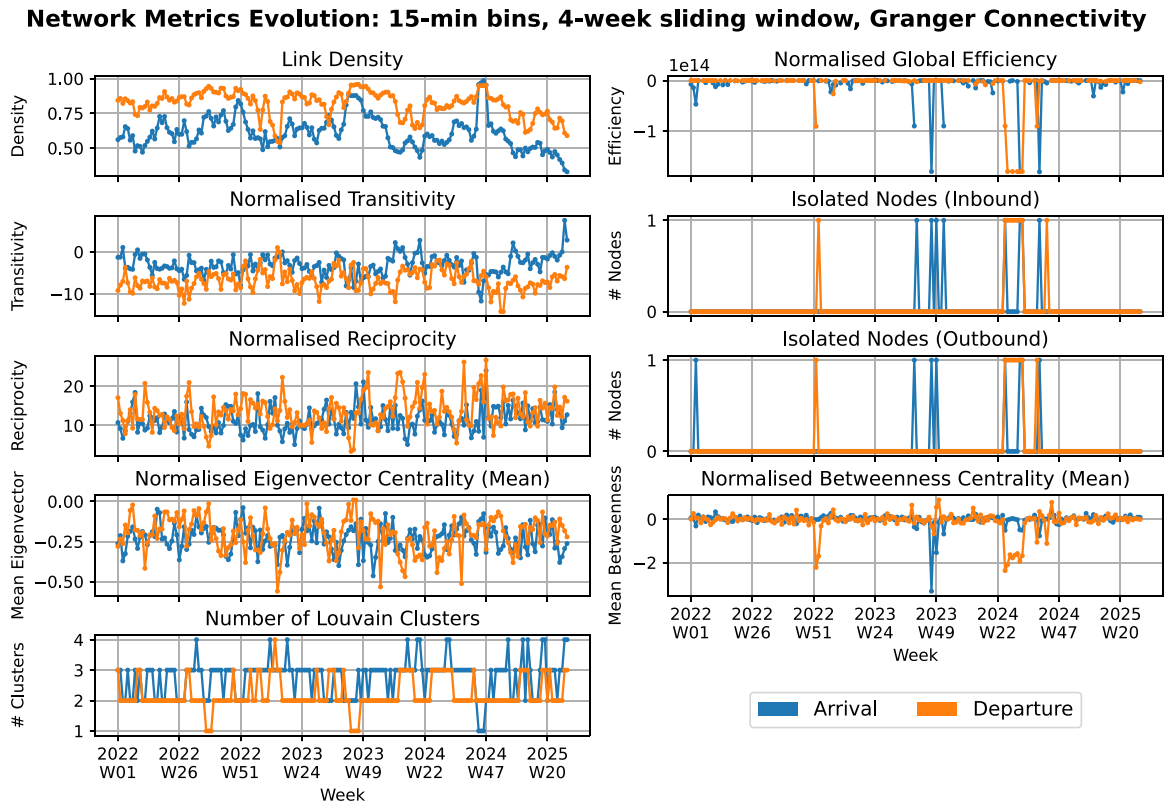


Fig. 11. Evolution of topological metrics for 15-minute arrival (blue) and departure (orange) delays in the Swiss rail system, calculated using a 28-day sliding-window. The normalised metrics, including the reciprocity and the global efficiency, are reported as Z-Scores relative to a random null model, and not as raw ratios (see Section 2.5.5). See panel titles for metric names.

therefore in extremely large (in absolute terms) Z-Scores. Beyond the specific values, these have to be interpreted as instances of networks whose structure is clearly not compatible with the null model.

5. Discussion and conclusions

In this contribution we reviewed the main elements and challenges of the use of functional network representations to study delay relationships in transportation systems. In spite of many important results that have been obtained by the research community using this approach, the interested practitioner has also to face a large number of options, e.g. in terms of how to preprocess the data, which connectivity metric to apply, or how to evaluate the resulting structures. We here discussed these options, reviewing existing alternatives, highlighting limitations, and showing their impact in the final results through a series of examples drawn from air transport.

As already discussed in the introduction, the reconstruction of functional networks should not be understood as the complete solution to the analysis of transportation systems, but rather as a tool complementary to other approaches. More in detail, functional representations are helpful in describing the interactions between the elements of the system, but they are also hindered by three limitations. Firstly, relationships mostly describe correlations; even when using causality metrics, as the Granger Causality test, these may not correspond to genuine causality due to the presence of confounding effects. Secondly, while functional networks allow one to establish that a relationship between the delays at two stations exists, they do not explain the underlying reason - e.g. which elements or policies are responsible for such relation. Thirdly, they do not allow to simulate what-if scenarios, and thus cannot directly guide interventions in the system. These limitations notwithstanding, functional networks can be used as a first step in a more complete analysis, to identify relevant elements and relations to be later studied using, e.g., micro-scale simulations.

We complemented this theoretical discussion with *delaynet*, a holistic framework for analysing delay relationships in transportation networks. It integrates time series analysis techniques, network reconstruction methodologies and complex network metrics to enable a nuanced understanding of the structures created by (and supporting) the potential propagation of delays. The *delaynet* framework consists of five key components that together form a systematic approach: data preparation, detrending methods, connectivity analysis, network reconstruction, and network analysis. Each of these components is designed to address a specific aspect of delay analysis. By providing these in a unified package, it simplifies the analysis of real-world data and ensures that no essential steps are skipped, which in turn especially benefits practitioners not familiar with these techniques. At the same time, the package is also highly flexible; this allows for the application to different transportation modes and the accommodation of various data types and time scales, but also the integration within larger data analysis pipelines. The methods employed are founded upon well-established theories from statistical physics, information theory and network science; and, whenever available, leverage existing open-source libraries to foster validation and robustness.

The use of *delaynet* has been illustrated through an analysis of Swiss long-distance rail operations between 2022 and 2025. Note that, while functional network analyses have frequently been performed on air transport data (Zanin, 2015; Zanin et al., 2017; Pastorino and Zanin, 2021; Jia et al., 2022; Wang et al., 2022; Feng et al., 2024; Gil-Rodrigo and Zanin, 2024), to the best of our knowledge this is the first instance of such type of study on rail data. We have shown how detrended station-level delay series, lag-aware connectivity, and statistically-pruned networks yield interpretable spatial structures and temporal dynamics — see Section 4. These structures are further different from the topology of the underlying physical network, i.e. of how stations are physically connected by trains visiting them in a sequential

fashion — see App. D for full results. While the physical backbone is mainly static, the functional network's evolution (see Fig. 11) highlights time-local dependencies that deviate from the underlying infrastructure. In addition, the stations most central in both representations are highly different (see Fig. 20). This application underscores both the practical usefulness of the pipeline and avenues for further refinement (e.g., alternative estimators, robustness checks, and operational validation).

Notwithstanding the advantages of the *delaynet* framework, it is important to acknowledge its limitations and associated challenges. First of all, the framework requires high-quality time series data with sufficient temporal resolution and coverage. While not being something specific of the proposed package, the practitioner has to be aware of the data limitations, as well as of their specific properties — to illustrate, air transport delays can be calculated against originally filed or last filed flight plans, or using gate arrival times as opposed to touch-down times. Secondly, in spite of several optimisation strategies, some methods can be computationally intensive for large transportation networks, i.e. for a large number of nodes; but also when using sophisticated connectivity measures, or multiple parameters have to be compared. Thirdly, most methods included in the framework involve several parameters that need to be selected appropriately, which may require domain expertise or extensive experimentation to optimise for specific transportation systems. This also intermingles with the problem of interpreting the results; to illustrate, inasmuch as different connectivity measures describe different aspects of the propagation of delays, results yielded by them have to be carefully interpreted and compared. In synthesis, it would be ill-advised to apply *delaynet* as a black-box. To transition from such descriptive monitoring to a diagnostic understanding, it is necessary to resort to external event logs (e.g., weather data, infrastructure failures, or specific holiday schedules). Linking the detected topological fluctuations to these real-world events is a crucial step in moving from 'when' and 'where' a structural change occurred to 'why' it happened.

As a final limitation, while *delaynet* includes all fundamental steps and analysis tools, we also acknowledge that many more have been proposed in the literature — as illustrated by the huge variety of topological metrics that have been published in different contexts (Costa et al., 2007). It is our intention to keep evolving the package, also by accepting suggestions and comments from users. Therefore, the interested reader should also check the online documentation at <https://delaynet.readthedocs.io> for the most up-to-date list of available options.

CRedit authorship contribution statement

Carlson Moses Büth: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Massimiliano Zanin:** Writing – review & editing, Writing – original draft, Validation, Supervision, Project administration, Methodology, Investigation, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 851255). This work was partially supported by the María de Maeztu project CEX2021-001164-M funded by the MICIU/AEI/10.13039/501100011033 and FEDER, EU.

Appendices. Supplementary data

Supplementary material related to this article can be found online at <https://doi.org/10.1016/j.trip.2026.102187>.

Data availability

Accession codes: The *delaynet* package (version 0.3.3) is archived on Zenodo, at <https://doi.org/10.5281/zenodo.16875272>.

Documentation: The full documentation is available at <https://delaynet.readthedocs.io/>.

Source Code: The source code is publicly available on GitHub: <https://github.com/cbueth/delaynet>.

References

- Abraham, A., Pedregosa, F., Eickenberg, M., Gervais, P., Mueller, A., Kossaifi, J., Gramfort, A., Thirion, B., Varoquaux, G., 2014. Machine learning for neuroimaging with scikit-learn. *Front. Neuroinformatics* 8, 14. <http://dx.doi.org/10.3389/fninf.2014.00014>.
- Acharya, K., Olivares, F., Zanin, M., 2024. How representative are air transport functional complex networks? a quantitative validation. *Chaos* 34, <http://dx.doi.org/10.1063/5.0189642>.
- Al-Taie, M.Z., Kadry, S., 2017. Python for graph and network analysis. Springer, <http://dx.doi.org/10.1007/978-3-319-53004-8>.
- Anderson, P.W., 1972. More is different: broken symmetry and the nature of the hierarchical structure of science. *Science* 177, 393–396. <http://dx.doi.org/10.1126/science.177.4047.393>.
- Bandt, C., Pompe, B., 2002. Permutation entropy: a natural complexity measure for time series. *Phys. Rev. Lett.* 88, 174102. <http://dx.doi.org/10.1103/physrevlett.88.174102>.
- Bareinboim, E., Correa, J.D., Ibeling, D., Icard, T., 2022. On pearl's hierarchy and the foundations of causal inference. *Probabilistic and Causal Inference: The Works of Judea Pearl*. ACM Books, pp. 507–556. <http://dx.doi.org/10.1145/3501714.3501743>.
- Barnett, L., Barrett, A.B., Seth, A.K., 2009. Granger causality and transfer entropy are equivalent for gaussian variables. *Phys. Rev. Lett.* 103, 238701. <http://dx.doi.org/10.1103/physrevlett.103.238701>.
- Barnhart, C., Fearing, D., Vaze, V., 2014. Modeling passenger travel and delays in the national air transportation system. *Oper. Res.* 62, 580–601. <http://dx.doi.org/10.1287/opre.2014.1268>.
- Barthelemy, M., 2019. The statistical physics of cities. *Nat. Rev. Phys.* 1, 406–415. <http://dx.doi.org/10.1038/s42254-019-0054-2>.
- Batool, K., Niazi, M.A., 2014. Towards a methodology for validation of centrality measures in complex networks. *PLoS One* 9, e90283. <http://dx.doi.org/10.1371/journal.pone.0090283>.
- Brandes, U., 2008. On variants of shortest-path betweenness centrality and their generic computation. *Soc. Netw.* 30, 136–145. <http://dx.doi.org/10.1016/j.socnet.2007.11.001>.
- Büth, C.M., Acharya, K., Zanin, M., 2025. Infomeasure: A comprehensive Python package for information theory measures and estimators. *Sci. Rep.* 15, 29323. <http://dx.doi.org/10.1038/s41598-025-14053-5>.
- Büth, C.M., Zanin, M., 2025. SynthATDelays: A Minimalist Python Package for the Generation of Synthetic Air Transport Delay Data. <http://dx.doi.org/10.20944/preprints202508.1091.v1>.
- Button, K., 2010. *Transport economics*. Edward Elgar Publishing.
- Cannas, M., Fadda, P., Fancello, G., Frigau, L., Mola, F., 2013. Delay prediction in container terminals: A comparison of machine learning methods. In: *13th World Conference on Transportation Research*. pp. 1–6.
- Carlier, S., De Lépinay, I., Hustache, J.C., Jelinek, F., 2007. Environmental impact of air traffic flow management delays. In: *7th USA/Europe Air Traffic Management Research and Development Seminar (ATM2007)*. Citeseer, p. 16.
- Cascetta, E., 2009. *Transportation systems analysis: models and applications*. vol. 29, Springer Science & Business Media, <http://dx.doi.org/10.1007/978-0-387-75857-2>.
- Cook, A., Blom, H.A., Lillo, F., Mantegna, R.N., Miccichè, S., Rivas, D., Vázquez, R., Zanin, M., 2015. Applying complexity science to air traffic management. *J. Air Transp. Manag.* 42, 149–158. <http://dx.doi.org/10.1016/j.jairtraman.2014.09.011>.
- Cook, A., Tanner, G., Williams, V., Meise, G., 2009. Dynamic cost indexing—managing airline delay costs. *J. Air Transp. Manag.* 15, 26–35. <http://dx.doi.org/10.1016/j.jairtraman.2008.07.001>.
- Costa, L.d.F., Rodrigues, F.A., Travieso, G., Villas Boas, P.R., 2007. Characterization of complex networks: A survey of measurements. *Adv. Phys.* 56, 167–242. <http://dx.doi.org/10.1080/00018730601170527>.
- Delgado, L., Martín, J., Blanch, A., Cristóbal, S., 2016. Hub operations delay recovery based on cost optimisation-dynamic cost indexing and waiting for passengers strategies. *Sixth SESAR Innov. Days*.

- Diebold, F.X., 1997. *Elements of Forecasting*, 4th edition ed. Cengage Learning, Cincinnati, Ohio.
- Ding, M., Chen, Y., Bressler, S.L., 2006. Granger causality: basic theory and application to neuroscience. *Handb. Time Ser. Anal.: Recent. Theor. Dev. Appl.* 437–460. <http://dx.doi.org/10.1002/9783527609970.ch17>.
- Eguiluz, V.M., Chialvo, D.R., Cecchi, G.A., Baliki, M., Apkarian, A.V., 2005. Scale-free brain functional networks. *Phys. Rev. Lett.* 94, 018102. <http://dx.doi.org/10.1103/physrevlett.94.018102>.
- Feng, D., Hao, B., Lai, J., 2024. Tracing delay network in air transportation combining causal propagation and complex network. *Int. J. Intell. Netw.* 5, 63–76. <http://dx.doi.org/10.1016/j.ijin.2024.01.006>.
- Freeman, L.C., 1977. A set of measures of centrality based on betweenness. *Sociometry* 35–41. <http://dx.doi.org/10.2307/3033543>.
- Freeman, J.R., 1983. Granger causality and the times series analysis of political relationships. *Am. J. Polit. Sci.* 327–358. <http://dx.doi.org/10.2307/2111021>.
- Gil-Rodrigo, S., Zanin, M., 2024. Low cost carriers induce specific and identifiable delay propagation patterns: an analysis of the eu and us systems. *IEEE Access* 12, 75323–75336. <http://dx.doi.org/10.1109/access.2024.3406392>.
- Gramfort, A., Luessi, M., Larson, E., Engemann, D.A., Strohmeier, D., Brodbeck, C., Goj, R., Jas, M., Brooks, T., Parkkonen, L., et al., 2013. Meg and eeg data analysis with mne-python. *Front. Neuroinform.* 7, 267. <http://dx.doi.org/10.3389/fnins.2013.00267>.
- Grando, F., Noble, D., Lamb, L.C., 2016. An analysis of centrality measures for complex and social networks. In: 2016 IEEE Global Communications Conference. IEEE, pp. 1–6. <http://dx.doi.org/10.1109/glocom.2016.7841580>.
- Granger, C.W.J., 1969. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica* 37, 424. <http://dx.doi.org/10.2307/1912791>.
- Granger, C.W., 1988. Some recent development in a concept of causality. *J. Econ.* 39, 199–211. [http://dx.doi.org/10.1016/0304-4076\(88\)90045-0](http://dx.doi.org/10.1016/0304-4076(88)90045-0).
- Grassmann, G., 2020. New considerations on the validity of the wiener-granger causality test. *Heliyon* 6, <http://dx.doi.org/10.1016/j.heliyon.2020.e05208>.
- Higgins, A., Kozan, E., 1998. Modeling train delays in urban networks. *Transp. Sci.* 32, 346–357. <http://dx.doi.org/10.1287/trsc.32.4.346>.
- Huang, K., 2009. *Introduction to statistical physics*. Chapman and Hall/CRC.
- Jia, Z., Cai, X., Hu, Y., Ji, J., Jiao, Z., 2022. Delay propagation network in air transport systems based on refined nonlinear granger causality. *Transp. Metr. B* 10, 586–598. <http://dx.doi.org/10.1080/21680566.2021.2024102>.
- Kim, S., Kim, H., Park, Y., 2017. Early detection of vessel delays using combined historical and real-time information. *J. Oper. Res. Soc.* 68, 182–191. <http://dx.doi.org/10.1057/s41274-016-0104-4>.
- Kirchgässner, G., Wolters, J., Hassler, U., 2013. *Introduction to modern time series analysis*. In: *springer texts in business and economics*. Springer Berlin Heidelberg, Berlin, Heidelberg. <http://dx.doi.org/10.1007/978-3-642-33436-8>.
- Kumar, N., Mishra, B., Athar, M., Mukhtar, S., 2021. Inference of gene regulatory network from single-cell transcriptomic data using pyscenic. In: *Modeling Transcriptional Regulation: Methods and Protocols*. Springer, pp. 171–182. http://dx.doi.org/10.1007/978-1-0716-1534-8_10.
- Lapamonpinyo, P., Derrible, S., Corman, F., 2022. Real-time passenger train delay prediction using machine learning: A case study with amtrak passenger train routes. *IEEE Open J. Intell. Transp. Syst.* 3, 539–550. <http://dx.doi.org/10.1109/ojits.2022.3194879>.
- LaPlante, R.A., Douw, L., Tang, W., Stufflebeam, S.M., 2014. The connectome visualization utility: Software for visualization of human brain networks. *PLoS One* 9, e113838. <http://dx.doi.org/10.1371/journal.pone.0113838>.
- Latora, V., Marchiori, M., 2001. Efficient behavior of small-world networks. *Phys. Rev. Lett.* 87, 198701. <http://dx.doi.org/10.1103/PhysRevLett.87.198701>.
- Li, Q., Jing, R., 2021. Characterization of delay propagation in the air traffic network. *J. Air Transp. Manag.* 94, 102075. <http://dx.doi.org/10.1016/j.jairtraman.2021.102075>.
- Marimpis, A.D., Dimitriadis, S.I., Goebel, R., 2021. Dyconnmap: Dynamic connectome mapping—a neuroimaging python module. *Hum. Brain Mapp.* 42, 4909–4939. <http://dx.doi.org/10.1002/hbm.25589>.
- Maziarz, M., 2015. A review of the granger-causality fallacy. *J. Philos. Econ.* 8, 86–105. <http://dx.doi.org/10.46298/jpe.10676>.
- Meunier, D., Pascarella, A., Altukhov, D., Jas, M., Combrisson, E., Lajnef, T., Bertrand-Dubois, D., Hadid, V., Alamian, G., Alves, J., et al., 2020. Neuropycon: An open-source python toolbox for fast multi-modal and reproducible brain connectivity pipelines. *NeuroImage* 219, 117020. <http://dx.doi.org/10.1016/j.neuroimage.2020.117020>.
- Mitsokapas, E., Schäfer, B., Harris, R.J., Beck, C., 2021. Statistical characterization of airplane delays. *Sci. Rep.* 11, 7855. <http://dx.doi.org/10.1038/s41598-021-87279-8>.
- Musilova, J., Vafek, Z., Puniya, B.L., Zimmer, R., Helikar, T., Sedlar, K., 2024. Augusta: From rna-seq to gene regulatory networks and boolean models. *Comput. Struct. Biotechnol. J.* 23, 783–790. <http://dx.doi.org/10.1016/j.csbj.2024.01.013>.
- Newman, M.E., 2003. The structure and function of complex networks. *SIAM Rev.* 45, 167–256. <http://dx.doi.org/10.1137/s003614450342480>.
- Olivares, F., Marín-Rodríguez, F.J., Acharya, K., Zanin, M., 2025. Evaluating methods for detecting time series using ordinal patterns, with an application to air transport delays. *Entropy* 27, 230. <http://dx.doi.org/10.3390/e27030230>.
- Olivares, F., Zanin, M., 2025. Beyond classical models: Statistical physics tools for the analysis of time series in modern air transport. *arXiv preprint arXiv:2507.20927*.
- Özlem, Ş., Or, İ., Altan, Y.C., 2021. Scheduling and simulation of maritime traffic in congested waterways: An application to the strait of istanbul. *J. Navig.* 74, 656–672. <http://dx.doi.org/10.1017/s0373463320000715>.
- Park, H.J., Friston, K., 2013. Structural and functional brain networks: from connections to cognition. *Science* 342, 1238411. <http://dx.doi.org/10.1126/science.1238411>.
- Pastorino, L., Zanin, M., 2021. Air delay propagation patterns in Europe from 2015 to 2018: An information processing perspective. *J. Phys. Complex.* 3, 015001. <http://dx.doi.org/10.1088/2632-072x/ac4003>.
- Pastorino, L., Zanin, M., 2023. Local and network-wide time scales of delay propagation in air transport: A granger causality approach. *Aerospace* 10, 36. <http://dx.doi.org/10.3390/aerospace10010036>.
- Peterson, E.B., Neels, K., Barci, N., Graham, T., 2013. The economic cost of airline flight delay. *J. Transp. Econ. Policy* 47, 107–121. <http://dx.doi.org/10.3828/jtep.2013.47.1.107>.
- Power, J.D., Fair, D.A., Schlaggar, B.L., Petersen, S.E., 2010. The development of human functional brain networks. *Neuron* 67, 735–748. <http://dx.doi.org/10.1016/j.neuron.2010.08.017>.
- Rebollo, J.J., Balakrishnan, H., 2014. Characterization and prediction of air traffic delays. *Transp. Res. Part C: Emerg. Technol.* 44, 231–241. <http://dx.doi.org/10.1016/j.trc.2014.04.007>.
- Reichl, L.E., 2016. *A modern course in statistical physics*. John Wiley & Sons, <http://dx.doi.org/10.1002/9783527690497>.
- Reuveny, R., Kang, H., 1996. International trade, political conflict/cooperation, and granger causality. *Am. J. Polit. Sci.* 943–970. <http://dx.doi.org/10.2307/2111801>.
- Riccardo, G., Armando, B., Sandro, R., 2012. Towards a statistical physics of human mobility. *Internat. J. Modern Phys. C* 23, 1250061. <http://dx.doi.org/10.1142/s012918312500611>.
- Runge, J., Gerhardus, A., Varando, G., Eyring, V., Camps-Valls, G., 2023. Causal inference for time series. *Nat. Rev. Earth Env.* 4, 487–505. <http://dx.doi.org/10.1038/s43017-023-00431-y>.
- Sajan, G.V., Kumar, P., 2021. Forecasting and analysis of train delays and impact of weather data using machine learning. In: 2021 12th International Conference on Computing Communication and Networking Technologies. ICCCNT, IEEE, pp. 1–8. <http://dx.doi.org/10.1109/icccnt51525.2021.9580176>.
- Saxena, A., Iyengar, S., 2020. Centrality measures in complex networks: A survey. *arXiv preprint arXiv:2011.07190*.
- Schmitt, P., Sorin, B., Frouté, T., Parisot, N., Calevro, F., Peignier, S., 2023. Grenadine: a data-driven python library to infer gene regulatory networks from gene expression data. *Genes* 14, 269. <http://dx.doi.org/10.3390/genes14020269>.
- Schreiber, T., 2000. Measuring Information Transfer. *Phys. Rev. Lett.* 85, 461–464. <http://dx.doi.org/10.1103/PhysRevLett.85.461>.
- Seabold, S., Perktold, J., 2010. Statsmodels: Econometric and statistical modeling with Python. In: van der Walt, S., Millman, J. (Eds.), *Proceedings of the 9th Python in Science Conference*. pp. 92–96. <http://dx.doi.org/10.25080/Majora-92bf1922-011>.
- Shojaie, A., Fox, E.B., 2022. Granger causality: A review and recent advances. *Annu. Rev. Stat. Appl.* 9, 289–319. <http://dx.doi.org/10.1146/annurev-statistics-040120-010930>.
- Strogatz, S.H., 2001. Exploring complex networks. *Nature* 410, 268–276. <http://dx.doi.org/10.1038/35065725>.
- Sun, X., Wandelt, S., 2021. Robustness of air transportation as complex networks: Systematic review of 15 years of research and outlook into the future. *Sustainability* 13, 6446. <http://dx.doi.org/10.3390/su13116446>.
- Truong, D., 2021. Using causal machine learning for predicting the risk of flight delays in air transportation. *J. Air Transp. Manag.* 91, 101993. <http://dx.doi.org/10.1016/j.jairtraman.2020.101993>.
- Wang, Y., Li, M.Z., Gopalakrishnan, K., Liu, T., 2022. Timescales of delay propagation in airport networks. *Transp. Res. Part E: Logist. Transp. Rev.* 161, 102687. <http://dx.doi.org/10.1016/j.jtre.2022.102687>.
- Wang, S., Vaze, V., 2016. Modeling probability distributions of primary delays in the national air transportation system. *Transp. Res. Rec.* 2569, 42–52. <http://dx.doi.org/10.3141/2569-05>.
- Wei, K., Vaze, V., 2018. Modeling crew itineraries and delays in the national air transportation system. *Transp. Sci.* 52, 1276–1296. <http://dx.doi.org/10.1287/trsc.2018.0834>.
- Woolley-Meza, O., Thiemann, C., Grady, D., Lee, J.J., Seebens, H., Blasius, B., Brockmann, D., 2011. Complexity in human transportation networks: a comparative analysis of worldwide air transportation and global cargo-ship movements. *Eur. Phys. J. B* 84, 589–600. <http://dx.doi.org/10.1140/epjb/e2011-20208-9>.
- Xu, Q., Huang, X., Zhang, W., Zhao, H., Zhang, H., Jin, Z., 2025. Eurasian container intermodal transportation network: a robust optimization with uncertainty and carbon emission constraints. *Front. Mar. Sci.* 12, 1576006. <http://dx.doi.org/10.3389/fmars.2025.1576006>.
- Yang, Y., Huang, P., Peng, Q., Li, J., Wen, C., 2019. Statistical delay distribution analysis on high-speed railway trains. *J. Mod. Transp.* 27, 188–197. <http://dx.doi.org/10.1007/s40534-019-0188-z>.
- Yu, B., Guo, Z., Asian, S., Wang, H., Chen, G., 2019. Flight delay prediction for commercial air transport: A deep learning approach. *Transp. Res. Part E: Logist. Transp. Rev.* 125, 203–221. <http://dx.doi.org/10.1016/j.jtre.2019.03.013>.

- Zanin, M., 2015. Can we neglect the multi-layer structure of functional networks? *Phys. A* 430, 184–192. <http://dx.doi.org/10.1016/j.physa.2015.02.099>.
- Zanin, M., 2021. Assessing granger causality on irregular missing and extreme data. *IEEE Access* 9, 75362–75374. <http://dx.doi.org/10.1109/ACCESS.2021.3082014>.
- Zanin, M., 2023. Continuous ordinal patterns: Creating a bridge between ordinal analysis and deep learning. *Chaos* 33, 033114. <http://dx.doi.org/10.1063/5.0136492>.
- Zanin, M., 2024. Augmenting granger causality through continuous ordinal patterns. *Commun. Nonlinear Sci. Numer. Simul.* 128, 107606. <http://dx.doi.org/10.1016/j.cnsns.2023.107606>.
- Zanin, M., Belkoura, S., Zhu, Y., 2017. Network analysis of chinese air transport delay propagation. *Chin. J. Aeronaut.* 30, 491–499. <http://dx.doi.org/10.1016/j.cja.2017.01.012>.
- Zanin, M., Sun, X., Wandelt, S., 2018. Studying the topology of transportation systems through complex networks: Handle with care. *J. Adv. Transp.* 2018, 3156137. <http://dx.doi.org/10.1155/2018/3156137>.
- Zhao, Y., Liu, Q., 2023. Causal ml: Python package for causal inference machine learning. *SoftwareX* 21, 101294. <http://dx.doi.org/10.1016/j.softx.2022.101294>.
- Zhao, H., Tang, Y., Tian, R., Yan, Q., Yuan, Q., Wang, T., Zhou, D., Zhang, Y., Yang, L., Xu, Q., 2025a. A lattice hydrodynamic model for on-ramp and off-ramp traffic flow considering non-equilibrium characteristics and heterogeneous mixed-flow speed delay. *Chaos Solitons Fractals* 198, 116548. <http://dx.doi.org/10.1016/j.chaos.2025.116548>.
- Zhao, H., Tang, Y., Wang, Y., Yuan, Q., Wang, T., Zhou, D., Zhang, Y., Lu, Y., Liu, Y., Nong, L., 2025b. Lattice modeling and simulation of traffic flow in speed-limited and free-flowing regions considering speed limit coefficients. In: Tenth International Conference on Electromechanical Control Technology and Transportation (ICECTT 2025). SPIE, pp. 577–581. <http://dx.doi.org/10.1117/12.3078893>.