A Model-Based Characterisation of Delay Propagation in Metro Networks

Developing a Data-Driven Framework for Discovering and Quantifying Delay Behaviour

Master Thesis

M.T.S. (Max) Lange

Delft University of Technology

TU Delft

A Model-Based Characterisation of Delay Propagation in Metro Networks

Developing a Data-Driven Framework for Discovering and Quantifying Delay Behaviour

by

M.T.S. (Max) Lange

to obtain the degree of Master of Science
in Civil Engineering: Traffic & Transport Engineering
at the Delft University of Technology,
to be defended publicly on October 30th, 2025.

Student number: 5169402
Project duration: January 2025 – October 2025
Thesis committee: Prof. dr. O. Cats, TU Delft, chair
Dr. Y. Xin, TU Delft, supervisor

Cover: 'Metro Train in Underground Station' by Chris Duan on Pexels
Style: TU Delft Report Style, with modifications by Daan Zwaneveld

An electronic version of this thesis is available at http://repository.tudelft.nl/.
The codebase of this thesis is available at https://github.com/Max-Lange/Delay-Propagation.

TU Delft

Preface

This thesis was conducted to achieve my Master's degree in Civil Engineering: Traffic & Transport Engineering. The successful completion of this thesis is a double-edged sword, as while a passing grade acknowledges my capabilities as a full-fledged civil engineer, it also marks the end of my student life. The last six years have shaped me into who I am as a person. First academically, with my minor in Computer Science and the modelling during my studies leading me to discover my interest in programming, leading to my first proper job at a civil engineering company, co-writing my first academic paper, as well as the subject of this thesis itself. Secondly, I have learned a great deal about myself as a person: what interests me, how I prefer to work and live my life, and it has given me direction for what I want to do with the rest of it. With the conclusion of this thesis, I am off to the next chapter of my life, and I am excited to see what lies ahead.

I want to offer my gratitude to my suporvisory committee members, Oded Cats and Yanan Xin. Oded, thank you not only for your advice during this thesis, but also for your help with the paper with Francisco. Your clear and objective feedback helped me improve my writing style and ability to focus on the critical parts of my work. Yanan, thank you for advising me throughout our multitude of meetings, being a clear voice of direction when I found it difficult to properly oversee the results or next steps of my thesis. Thank you both for helping me complete my study.

M.T.S. (Max) Lange
Delft, October 2025

Abstract

The reliability of a metro system is an important factor in building user trust and ridership, as it helps maintain efficient and sustainable urban mobility. Within the high-frequency environment of metro networks, even minor disturbances can propagate across the network, resulting in secondary delays that reduce schedule stability and passenger satisfaction. With the need for analysis approaches that balance predictive flexibility with analytical interpretability, this thesis develops a data-driven framework for modelling and quantifying delay propagation in metro networks.

The proposed framework reduces the modelling of delay propagation to the statistical fitting of relationship functions between network elements using subsets of available variable data. The methodology follows five general steps: defining relationship structures, setting analysis dimensions and their granularity, selecting the method for fitting the relationship functions, fitting the functions using available data, and quantifying the residual variability. This framework is applied to the Washington D.C. Metro using schedule and operational train movement data to analyse and characterise delay propagation behaviour.

Two model implementations are developed. The first explores the full breadth of variables available in the delay data to identify those offering statistically significant delay relationships, revealing that propagation occurs predominantly between directly connected stations and along shared lines. The second, more targeted model quantifies these relationships, demonstrating that propagation strength is independent of delay magnitude and mostly consistent across different time periods. Further cross-examination revealed minimal sensitivity to operational and network design variables, such as headways and connectivity, suggesting that other variables like scheduling regimes might be the cause of the different observed delay propagation behaviours.

The results highlight that localised delay effects dominate over network-wide influences, suggesting that metro operators can improve service reliability by focusing mitigation efforts on key inter-station connections rather than entire lines. The framework offers a versatile and multiplicative foundation for future applications in delay propagation prediction and analysis.

Contents

Nomenclature

Terminology

Term	Definition
Berth	Stopping place for a train at a train station or platform
Connection	Directed link between two consecutive stations
Station-direction pair	ID for the part of a station facilitating the travel of trains in one direction
Stop	Station-direction pair along a train's trip
Trip	All stops a single train makes when travelling along its route (in one direction)

Abbreviations

Abbreviation	Definition
GTFS	General Transit Feed Specification
ID	Identification
SSE	Sum of squared errors
WMATA	Washington Metropolitan Area Transit Authority

Symbols

Symbol	Definition
i	Time step instance
j	Station-direction pair
k	Train in sequence of all trains arriving at a certain station-direction pair
$n_{relationships}$	Number of relationship functions
$n_{connections}$	Number of directed network connections
$n_{timesteps}$	Number of time steps
ϵ	Distribution
μ	Mean

List of Figures

List of Tables

List of Algorithms

1

Introduction

The reliability and efficiency of metro systems are important areas of focus in advancing sustainable urban transport. Their ability to provide fast and predictable services in densely populated metropolitan areas can significantly improve a city's productivity. For passengers, punctuality and reliability are two significant factors when considering ridership. When trains arrive on time and delays are clearly communicated, travellers feel more confident about using the metro system as part of their journey, thereby increasing trust. Conversely, unpredictability due to poor delay management can cause greater uncertainty and missed connections, leading to reduced public confidence in the system and lost productivity for the city.

From an operational standpoint, delay prediction and management are key components in maintaining a network's stability. For busy and high-frequency systems, even minor disturbances in the operating schedule can lead to delays, reduce efficiency, increase costs, and degrade the passenger experience. Metro operators aim to minimise the initial occurrence of any delay and their subsequent spread by using thoughtful timetable construction, setting network and vehicle constraints, and implementing delay mitigation strategies. Developing methods to analyse and predict delays is essential for effective metro operation and planning, as they can offer direct insights into the underlying patterns and behaviours of the delays. This, in turn, can lead to better utilisation of crew and rolling stock, lowering operational costs.

Delays in rail networks rarely remain localised to the location where they occur. Suppose a train's delay is not mitigated immediately. In that case, subsequent vehicle travel can create a knock-on effect, where the delay is carried across other parts of its route, possibly affecting other trains or lines and causing secondary delays. This phenomenon, known as delay propagation, is a defining characteristic of urban rail systems.

Understanding delay propagation is crucial for proper network management. It can provide insight into system vulnerabilities, showing connections or network elements that are susceptible to or have a propagating effect on secondary delays. This, in turn, can pinpoint where delays are most likely to spread, allowing network operators to focus on those locations when employing delay-mitigating measures, making more effective use of resources. Studying the existing propagation patterns can also support predictive modelling, improving information on delay impacts, allowing for better real-time informing of passengers and delay mitigation decision-making.

As the propagation behaviour can be affected by many different variables, various methods have been employed to model delay propagation. Analytical models provide transparent and interpretable mathematical definitions, but rely on simplification that limits their scalability and complexity. Simulation-based frameworks offer in-depth analysis of the effect of individual elements on the propagation behaviour. However, their reliance on detailed inputs and behavioural rules restricts their transferability across systems. Probabilistic graphical and Bayesian models offer more flexibility, yet face computational challenges when applied to larger systems or when trying to incorporate many different variables.

More recently, the increase in readily available computational power and large-scale data gathering has proliferated the use of machine learning and data-driven methods. These models identify complex patterns within delay data, leading to stronger predictive power, while being easy to scale. However, as these models become more complex, they often become more ”black-box” in nature, leading to a loss in interpretability, making it difficult for researchers of network operators to understand and analyse why specific patterns occur. Consequently, propagation modelling approaches follow one of two major paradigms, focusing methods on either the prediction accuracy of their models or the transparency and analysis capabilities, but rarely both.

This divide highlights an important gap in the current research landscape that this thesis will try to address: the need for a flexible, data-driven framework that balances interpretability with predictive power. Such a framework should be capable of quantifying how and where delay propagation varies across any desired dimensions, like temporal, spatial, and operational, while maintaining adaptability across data sources, network structures, and modelling objectives. It will do so while trying to answer the following research question:

How do operational and network design characteristics influence the propagation of delays in a metro network?

To answer this main question, the following sub-questions are formulated to support it:

- *How can a modelling methodology be constructed to allow for free exploration of delay propagation modelling implementations?*
- *What relationships between delay propagation behaviour and operational variables can be found?*
- *How can the discovered propagation relationships be modelled and quantified?*
- *To what extent do temporal variables affect the magnitude or variability of delay propagation?*
- *To what extent do operational and network design characteristics affect the magnitude or variability of delay propagation?*

To answer these questions, this thesis develops a modelling framework and applies it to the Washington D.C. Metro system. This well-documented network serves as an adequate representative case for metro systems worldwide. The study aims to construct two model implementations:

- A generalised model, designed to explore the complete set of available variables and identify statistically significant relationships between them and delay propagation behaviour.
- A focused model, aimed at capturing and quantifying the discovered relationships with greater detail.

The structure of this thesis is as follows: Chapter 2 contains a literature review of the current state of delay propagation modelling, highlighting methodologies, their strengths and limitations, and the operational variables they examine. Chapter 3 presents the modelling framework, outlining its methodology and steps. Chapter 4 describes the case study network, the available data’s contents, and its processing in preparation for the models. In Chapter 5, a generalised implementation is constructed to identify significant relationships and correlations between delay propagation and other variables. Chapter 6 then focuses on these found correlations, implementing a more focused model to capture the delay propagation behaviour and quantify its variability. Chapter 7 examines the influence of operational and design characteristics on the observed propagation behaviour from the second model. Lastly, Chapter 8 contains an overall discussion of key findings, application limitations and possibilities for further study.

2

Literature Review

Research on delay propagation in urban rail networks has expanded significantly in recent years. With greater computational power and more advanced machine learning methods being more generally available, many new methods have been employed to model and analyse delay behaviours. Across these different models and results, we identify two main modelling goals: The first is to analyse which variables, internal or external, influence the delay behaviour and to determine their relative significance. The second goal is to construct delay propagation functions to predict delay spread, either by formulating and fitting mathematical equations or by directly using the discovered variable influences.

To position this thesis, the following literature review identifies the primary methods used in delay propagation modelling and highlights significant contributions from each. As we aim to find a general modelling methodology, we also extract their goals and the commonly examined variables to understand what our methodology should include at a minimum.

Analytical approaches were among the first attempts to characterise delay propagation behaviours in rail systems, using max-plus algebra or queuing terms to provide closed-form measures of expected delay propagation and network robustness in combination with simplified operational rules (Higgins & Kozan, 1998). They abstract away complex interactions in order to yield clear, traceable relationships between timetable variables (headway, buffer times, etc.) and delays. These have been extended to improve tractability in obtaining propagation behaviours for single lines or portions of the network by using polynomial functions (Harrod et al., 2019) or clustering methods (Szymanski et al., 2018); however, these prevent easy analysis of individual elements of the network, such as specific stations or connections, in more detail.

Simulation-based frameworks are effective at capturing the complex, real-world interactions that analytical models often oversimplify. Wang et al. (2019) tunes desires and behavioural rules for passengers of an existing network by taking human behaviour into account to showcase how passenger behaviour itself can induce or exacerbate delays. Most often, simulation approaches are used for optimising timetables or for deriving rescheduling strategies (Shakibayifar et al., 2018). While detailed, simulation approaches require extensive knowledge about existing operational rules, timetables, and passenger behaviours. They may also overlook other patterns that are present in the data if these cannot be included in the formulated behavioural rules.

As delays are related to discrete events, the arrival and departure of trains, they can be represented using a graphical or Bayesian method. They can be used to express conditional dependencies and allow for stochastic inference. The nodes and links of the graph structure can represent the stations and connections of the network, allowing for easy analysis of dependencies between network elements and probabilistic forecasting of delay propagation (Büker & Seybold, 2012; Corman & Kecman, 2018). Another approach is to make network characteristics and operational or external variables the nodes of the graph. The training of the model can then reveal the magnitude by which each variable affects the delay states within the network (Srivastava et al., 2024). It can also be used

to group the prorogation behaviour of network elements like stations or connections into similar groups, providing easy overviews of problematic network areas, as in Ulak et al. (2020).

Empirical analysis of delay datasets with respect to their distribution across the network has revealed multiple propagation phenomena, such as localised, train-to-train transfers along a line, and in other regimes, they show broader, network-scale cascading behaviour (Büchel et al., 2020). Other studies use empirical data to construct diffusion models that can simulate the spread and fizzling out of significant delays, such as in Dekker et al. (2022).

Over the last five years, machine learning methods have been increasingly used, providing more data-driven options for modelling delays. Many of them are used for prediction and delay forecasting, including traditional machine learning, Long Short-Term Memory networks, attention models and graph-based neural networks, summarised well in Spanninger et al. (2022). Due to their data-driven nature, they allow for more freedom when examining dimensions or heterogeneous network elements (for example, examining both trains and stations at the same time), like in Li et al. (2023). However, these models and their black-box nature raise concerns about interpretability. Huang et al. (2024) improves upon this by fusing different machine learning models. Nevertheless, the complexity of these models often makes it difficult to trace their conclusions back to the original data, thereby reducing the transparency of their results.

The propagation analysis and delay prediction landscape shows many different approaches and end-goals, for which each model has its trade-offs: analytical models give interpretability and computational efficiency but are limited in their fidelity and their handling of datasets with many dimensions; simulation or optimisation approaches give operational realism but limit direct inference from data by their need to either discover or pre-define operational rules to which the simulation must comply; Graphical and machine learning methods provide predictive power and can easily include stochastic behaviours, but either scale poorly to large networks (for graphical models) or lack easy interpretability (for black-box deep learning models).

Across all the examined papers, four significant variables/dimensions to examine delay propagation behaviour by emerge: Temporally, such as the time of day or day of the week; The magnitude of the delay itself; Spatially, either per area or branch of the network, or per specific network element, like stations or connections; Design or operational variables or the design or operational properties of the network (i.e. headways, dwell times, centrality, etc). While some methods do examine all of these major dimensions, the primary trend across all is that it is difficult to extend the models to include additional variables while also achieving both the main goals of variable analysis and propagation prediction. This limit arises because the model focuses solely on analysing specific subsets of variables, leaving no consideration for others, or due to the manually formulated nature of their model function, which requires significant reworking of the entire model to include additional variables. This mixed landscape motivates the need for a middle path: a data-driven method that enables analysis and inclusion of any desired variable, utilises these results for delay propagation prediction, and allows for customisation of methods based on the use case and desired level of interpretability.

Paper	Goal	Temporally	Delay magnitude	Spatially	Design or Operational variables
Higgins and Kozan (1998)	Prediction	No	Yes	No	Yes
Harrod et al. (2019)	Prediction	No	Yes	Yes	Yes
Szymanski et al. (2018)	Behaviour analysis	Yes	Yes	No	No
Wang et al. (2019)	Prediction	Partial	No	Yes	Yes
Corman and Kecman (2018)	Behaviour analysis	No	Yes	Yes	Yes
Büker and Seybold (2012)	Prediction	No	Yes	Yes	Yes
Srivastava et al. (2024)	Prediction	Yes	Yes	Yes	Yes
Ulak et al. (2020)	Behaviour analysis	Partial	Yes	Yes	No
Büchel et al. (2020)	Behaviour analysis	No	Yes	Yes	No
Dekker et al. (2022)	Prediction	No	Yes	Partial	Yes
Li et al. (2023)	Behaviour analysis	Yes	Yes	Yes	No
Huang et al. (2024)	Behaviour analysis	No	Yes	Yes	Yes
This study	**Both**	**Yes**	**Yes**	**Yes**	**Yes**

Table 2.1: Research goals and examined variable dimensions of the examined literature.

3

Methodological Framework

From the literature review in Chapter 2, we conclude that a general-purpose modelling methodology should be able to focus both on delay behaviour analysis and delay propagation prediction, while allowing for the inclusion of any variable to differentiate different delay behaviours by, and leaving room for any relationship-fitting method. Constructing a multiplicative methodology that standardises its definitions and steps will allow for easier comparison of results between models. This is because, even though they might be constructed for different purposes and using different fitting methods, their results will be formatted the same. It should also allow for easy additions or changes to any model, as each step will be separated and applicable to any fitting method.

We propose a general five-step framework for defining and constructing a delay propagation model. Each of the five steps addresses a specific set of choices required for model definition: 1. Define between which network elements or variables we want to find relationships between; 2. Define additional variables for which we want to discern differences in delay behaviour between states; 3. Define the method by which to extract relationship functions from the data; 4. Fit the relationship functions using the method from step 3, keeping data limitations in mind; 5. Quantify the remaining variability to gain information on the confidence of the relationship.

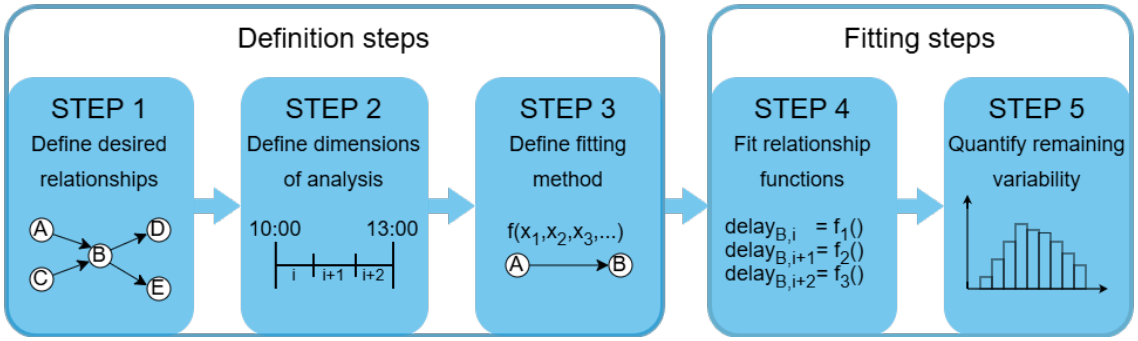


Figure 3.1: Methodology steps.

3.1. Define Relationship Structure

With the data-driven focus of our modelling method, the modelling of delay propagation behaviour turns into finding relationships in different subsets of data. If it is desired to analyse differences in behaviour between any variable states within this dataset, we need to decide which variables the subset division is performed along. The most important of these choices is defining which network elements or variables we want to relate delays between, as this will denote the primary goal of the model.

A metro network has many different elements (stations, connections, routes, branches, etc.) to which we can assign delays, and subsequently, for which we can explore the relationships between. With a data-driven approach, the only limiting factor in making this choice is in how we can assign the delay data to the network elements. The type of relationships we want to model ultimately depends on the use case. For most current literature, this involves finding relationships between different parts of the network, such as physical network components like stations or connections. This step thereby often involves splitting up the delay data spatially.

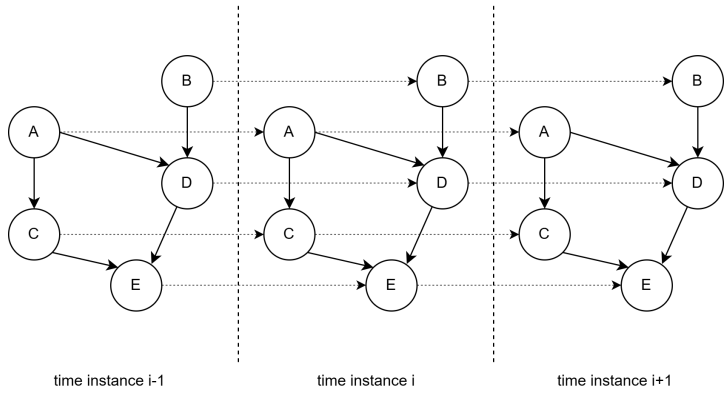


Figure 3.2: Example of a possible relationship structure, in which a station is related to all others with which it shares a physical connection, as well as future instances of itself.

3.2. Define Dimensions and Granularity

While the desired relationships to fit are defined in Section 3.1, the delay data can have other associated variables for which we want to see if different states can affect the delay propagation behaviour. Including these additional variables for analysis requires splitting up each of the relationships (and their associated delay datasets) from Section 3.1 into multiple relationships for each unique combination of states of the additional variables.

Again, with the data-driven workflow, this only requires each delay instance to have an associated state of each of these additional variables. However, the definition of what counts as a unique state for the variables can also be manually defined. Nominal variables such as the day of the week or the train route already have distinctly separate states, but continuous variables such as the time of day or the delay magnitude do not, and so if the fitting method chosen in Section 3.3 cannot handle continuos variables, its values need allocating to manually set ranges (such as the hour of the day for the time of day). Again, as long as each delay instance has an associated state for each variable, it can be included in the analysis. Each variable inclusion divides the existing relationships further, providing an additional dimension for analysis.

Examples of additional variables for analysis that are commonly included in existing literature are:

- **Time of day**: The time of the day at which the delay has occurred. This can be done either by manually defining different time periods (i.e., peak and off-peak) or by splitting the day into similarly sized segments, such as per hour.
- **Magnitude of delay**: The amount of delay at one or more network elements to which the delay values are associated.
- **Days of interest**: This can be (a) specific day(s) of the week, like Monday, a specific set of days, like all the holidays or days with a soccer game for example, or all days in the given dataset.
- **Number of passengers**: Depending on the type of relationship the model is after, either the number of people on a train, travelling between two points in the network, or getting on or off at a station.
- **External factors**: Weather at the time of the delay occurrence, distributions of job types around stations, etc.

3.3. Define a Fitting Method

As described in Section 3.1, the actual quantification of the relationship is a matter of finding relationships between different datasets. Exploring different relationship-fitting methods, we find many different options, each with its own advantages and disadvantages. While not an exhaustive list, Figure 3.3 shows some possible options. Analytical approaches allow for more interpretability, making it easier to analyse the behaviour of individual elements and propagation paths. At the same time, machine learning methods can find more complex relationships, leading to greater predictive power. However, their more ”black-box” nature can limit the interpretability and traceability of results. Ultimately, the method choice depends on considering these pros and cons with respect to the use case.

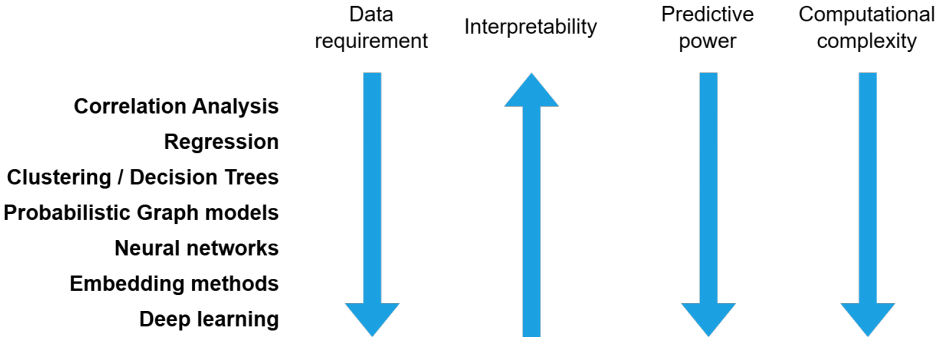


Figure 3.3: Relationship finding methods.

3.4. Fit Relationship Functions

With all model parameters defined, we can fit our functions per the defined relationship and dimension intervals using the method chosen in Section 3.4. Depending on the minimum data requirement for a proper fit with the chosen method and the amount of available data, checks should be put in place to ensure that enough data is available for each function attempted to be fitted. Suppose the minimum data requirement for a particular relationship is insufficient. In that case, we can either consolidate relationship functions along one or multiple dimension intervals, continue fitting the function (which might lead to volatile, inconsistent, or erroneous results), or consider that specific relationship void.

3.5. Quantify Remaining Variability

Regardless of the method used for relationship fitting, the resulting fitted functions will exhibit some variance or residuals when compared to the data. To include this data, alongside the fitting of the propagation function at each time step, we also want to fit the remaining error to a distribution. Methods that incorporate stochastic behaviour in fitting the relationship already include this process in the step of Section 3.4. These distributions can be used to provide confidence margins for our found relationships and can give information about the general variability of the propagation behaviours.

4

Data and Case Study: The Washington D.C. Metro

To both explore the primary goal of this thesis and to test the applicability of the methodology proposed in Chapter 3, we will attempt to discover and quantify the characteristics of delay propagation in the Washington D.C. metro network. Also known as the MetroRail, it is owned and operated by the Washington Metropolitan Area Transit Authority (WMATA), and is the second busiest metro network in the United States. It services 98 stations in Virginia, Maryland, and the District of Columbia, transporting more than 600.000 passengers daily (WMATA, 2024) (see Figure 4). As one of the largest passenger transit systems in the United States, and having an average ridership by worldwide standards, it serves as an adequate use case for answering our research questions.

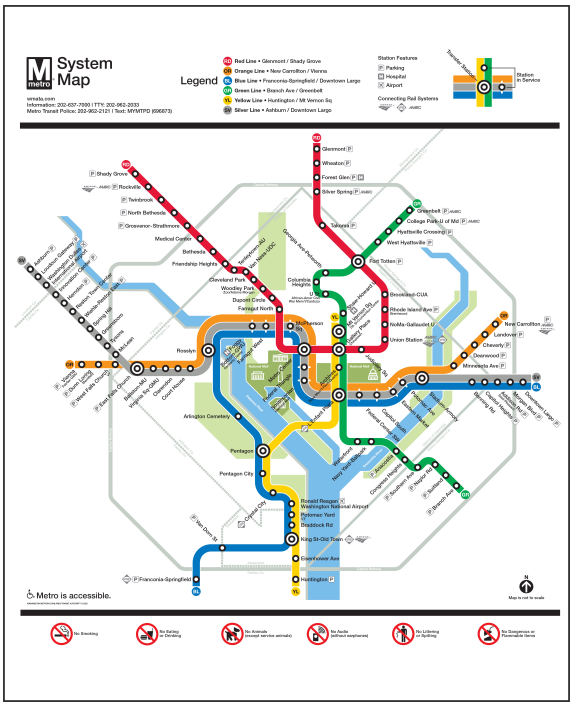


Figure 4.1: Washington D.C. Metro map (WMATA, 2024).

Comprehensive datasets of train and passenger movements across the network have been provided by the WMATA. Of these, the passenger movement dataset has already been explored extensively in other papers: Krishnakumari et al. (2020) uses smartcard data to extract passenger trajectories, from which they obtain passenger delays. These are then used to obtain delay informativity indicators in the study by Cats and Hijner (2021), which finds that the strongest relationships between delays at different stations occur between stations that share a metro line. The impact that different causes for delay have on these passenger delays is explored in Yap and Cats (2021), using them to predict impacts for future years. With the existing coverage of passenger-related delays, this thesis will examine train delays.

The delay of a train at a stop is defined as the difference between its actual arrival time and the scheduled arrival time. To determine the delays, we need to find the actual and scheduled arrival times for all stops along a train's route. Where the WMATA train movement datasets provide extensive and detailed data for the actual arrival times, the scheduled movements are not included. To obtain the train delay values, a schedule dataset must be identified, and the arrival times between it and the actual dataset must be linked.

4.1. Actual Arrival Time Data

The WMATA-provided dataset of train movements spans from August 1st, 2019, to December 31st, 2022. However, due to COVID-19 restrictions causing irregular arrival times compared to the schedule data, only data up to March 12th reflects regular operational schedules. Thus, only this data is considered viable for use. The data contains the arrival and departure times of all trains at each stop, along with many identifying keys and IDs for stops, stations, trips, vehicles, berths, and other entities. Due to data privacy, only the variables from the data that are used for this thesis are listed below:

- The name of the route the train's trip belongs to, i.e. Red, Blue, Silver, etc.
- The ID of the specific berth of a station that a train stops at.
- The ID of the berth at the next station a train is heading to after its current stop.
- The arrival date and time of a train at a station/berth.
- A unique identifier for the train trip that the stop belongs to.

In preparation for matching to the schedule data, the berth-specific IDs are converted to stop IDs by removing their berth-specific affix (D04-2, becomes D04 for example), as well as by standardising the names of stations which have multiple names in the data (for example, stations A01 and C01 are the same physical station, and so becomes A01/C01) (see Figure 4.2). These station IDs will be used to refer to the stations from this point forward.

4.2. Scheduled Arrival Time Data

For the scheduled arrival times, we can turn to the General Transit Feed Specification (GTFS). The GTFS is a data format standard that provides a standardised file structure for public transit agencies to describe the details of their services, such as schedules, stops, and fares, among other information (MobilityData, 2025). The Rail GTFS for the Washington Metro is managed and updated by WMATA itself, making it a reliable source for train schedules.

Versions of the GTFS prior to September 2019 are missing the southernmost stations. The first complete version, dated September 14th, 2019, contains schedule data up to March 31st, 2020. The GTFS data is spread across multiple files, which are linked via matching keys or IDs. Appendix A shows the internal structure and relations of the GTFS files and which attributes were extracted. The GTFS dataset contains the coordinates of all stations and track segments, making it the source for the geographical layout of the rail network (see Figure 4.2). The schedule data is restructured, and variables are renamed to mirror the data structure and naming conventions of the actual arrival times WMATA dataset in Section 4.1.

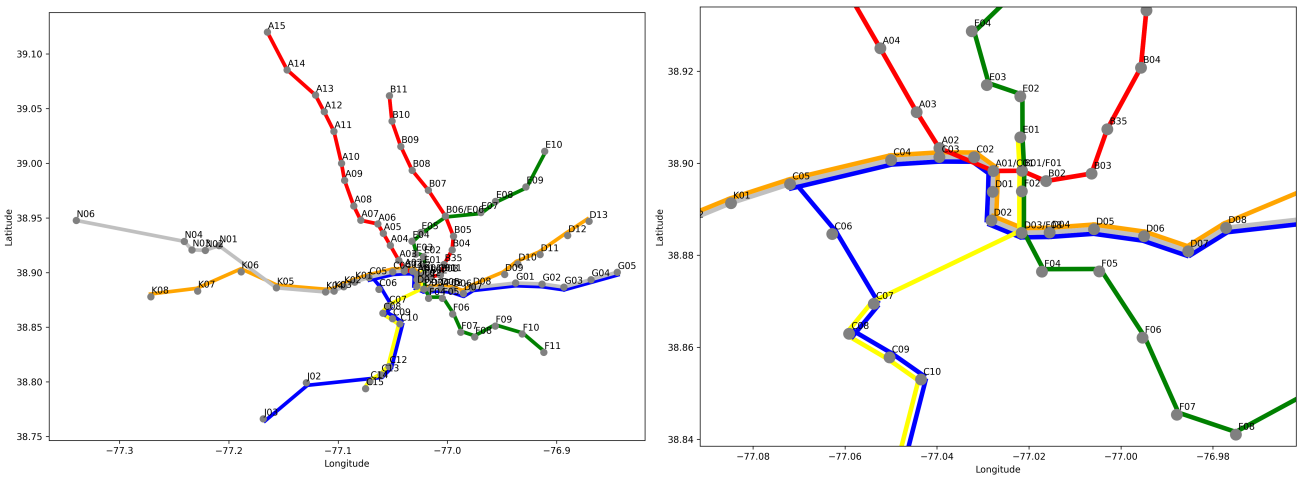


Figure 4.2: Geographical network layout (left), and zoomed in on the centre (right).

4.3. Obtaining Delay Values

We want to find the delay of each train at each stop. With the delay defined as the difference between the actual arrival time and the scheduled arrival time, we must match individual data points between the arrival time datasets to obtain our delay values. Using the novel approach of matching each scheduled arrival time to its closest actual arrival time by time, we encounter the issue of the order in which trains appears at each stop changing partway through the route. This is due to the changes in the difference between actual and scheduled arrival times between stops, leading to different matchings at each stop (see Figure 4.3). As the sequential nature of the metro routes does not allow this (except with manual intervention), a more complex method is needed to match the datasets properly.

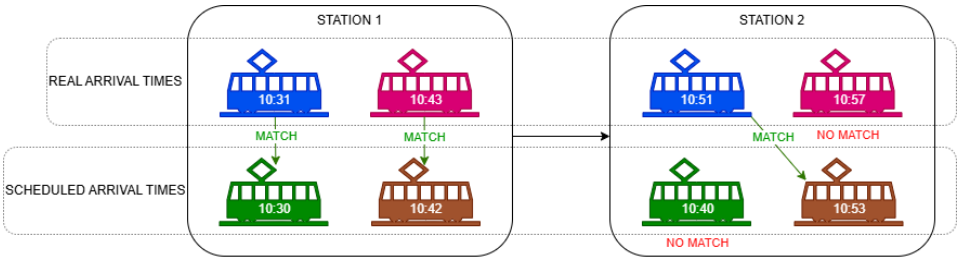


Figure 4.3: Example of train match order changing when trying to match per individual arrival time.

Instead of matching each arrival time separately, we match entire train trips to one another. A trip is defined as the entire sequence of stops a train makes while travelling a specific route. We first create lists of all unique trips for both the actual and scheduled trains, containing the trip ID, the first stop, the last stop, and the arrival time at the first stop. As shown in Algorithm 1, we then find each unique first-stop-last-stop pair. We iterate through all scheduled trips with that stop pair, identifying which actual trips started at the first stop within a two-hour window around the scheduled trip's start time. For all found actual trains, we then review the stops of both the scheduled and actual trips and calculate the total difference in arrival times between them. The actual train with the lowest total arrival time difference then becomes the match for the scheduled train. With the trips matched, individual stops can then also be matched by iterating over each stop in the trip. Delay values are then calculated in the same fashion as during the trip-matching algorithm.

Algorithm 1 Trip matching based on schedule and actual data.

Require: Scheduled trips $\mathcal{S}$, actual trips $\mathcal{R}$, scheduled stop times $\mathcal{T}^{scheduled}$, actual stop times $\mathcal{T}^{actual}$
Ensure: Mapping of actual trip keys to scheduled trip IDs

$TripMatches \leftarrow \{\}$ ▷ dictionary of matches
for all $(\mathcal{S}_i, \mathcal{R}_i)$ grouped by $(first\_stop, last\_stop)$ **do**
  for all $s \in \mathcal{S}_i$ **do**
    $\mathcal{R}_{possible} \leftarrow \{r \in \mathcal{R}_i \mid |r.first\_stop\_arrival - s.first\_stop\_arrival| < 1h\}$
    $\mathcal{T}_s^{scheduled} \leftarrow \mathcal{T}^{scheduled}[trip\_id = s.id]$
    $TotalDelays \leftarrow \{\}$
    for all $r \in \mathcal{R}_{possible}$ **do**
      $\mathcal{T}_r^{actual} \leftarrow \mathcal{T}^{actual}[trip\_key = r.key]$
      $TotalDelays[r.key] \leftarrow 0$
      for all $(t^{scheduled}, t^{actual}) \in (\mathcal{T}_s^{scheduled}, \mathcal{T}_r^{actual})$ **do**
        $delay \leftarrow t^{actual} - t^{scheduled}$
        $TotalDelays[r.key] \leftarrow TotalDelays[r.key] + delay$
      end for
    end for
    $Match \leftarrow \arg\min_{r.key}(TotalDelays[r.key])$
    $TripMatches[s.id] \leftarrow Match$
  end for
end for

Within the temporal overlap between the schedule and actual data described in sections 4.1 and 4.2, 193.930 scheduled trips and 256.181 actual trips occurred. 155.359 of the scheduled train trips were able to be matched to actual trips, resulting in 3.9 million delay values. The delays are noted in seconds and are spatially associated with the station where they occur, as well as the previous and next stops in their associated train's trip. They are also temporally associated with the scheduled arrival time of the train. Plotting all delay values, we notice a normal-type distribution, with characteristics also found in other papers, such as in Corman and Kecman (2018), showing a centre just below zero and a slightly heavier right tail. Some example values of the delay dataset are shown in Table 4.1.

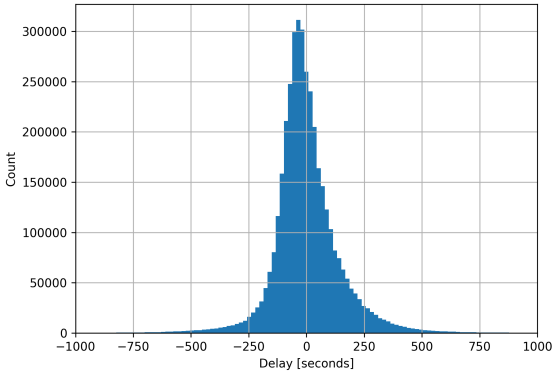


Figure 4.4: Delay values distribution.

Line	Trip Id	Stop ID	Previous stop ID	Next stop ID	Scheduled arrival time	Actual arrival time	Delay
Silver	3104406_18174_2019-11-08	G05	G04		2019-11-08 18:45:00	2019-11-08 18:45:56	56
Red	3124537_18167_2019-11-08	A05	A04	A06	2019-11-08 18:45:00	2019-11-08 18:44:21	-39
SILVER	3104502_18174_2019-11-08	C02	C03	A01/C01	2019-11-08 18:45:00	2019-11-08 18:46:23	83
Red	3124272_18167_2019-11-08	A03	A02	A04	2019-11-08 18:45:00	2019-11-08 18:43:08	-112
Yellow	3104872_18174_2019-11-08	E10	E09		2019-11-08 18:45:00	2019-11-08 18:47:46	166
Blue	3103763_18174_2019-11-08	C12	C13	C10	2019-11-08 18:45:00	2019-11-08 18:43:45	-75
Silver	3104449_18174_2019-11-08	G02	G01	G03	2019-11-08 18:45:00	2019-11-08 18:44:59	-1

Table 4.1: Example delay data.

5

Model I: Discovering Delay Relationships

As we do not yet know which variables influence the delay behaviour, our first model aims to identify which variables are significant. For this, we aim to capture the delay behaviour in a general way, while taking all available variables of our delay data into account. To keep the complexity of this model within the scope of this thesis, we simplify our model by only looking at the relationships between two individual components of the network (i.e. only using the state of one element to predict the state of one other element). The main goal and other assumptions are as follows:

***Goal**: Construct a model that can quantify the delay behaviour at one point and time in the network, given a delay with a certain magnitude at another point and time.*

- We will use only the variables present in the delay data itself; no external datasets will be included.
- We assume no delay mitigation strategies to be present when filtering our data for the fitting of the propagation functions.
- We assume a train station has one berth per possible direction of travel.
- We only examine the delay behaviour in response to actual delays (ignoring early arrivals / below-zero delay values).

5.1. Defining the Model

As we assume one berth per possible direction of travel at a station, each station has several simultaneous delay states, equal to the number of possible directions of travel. We say that 'a station has a delay for each station-direction pair'. As we do not yet know if there is a relationship between the delay states of station-direction pairs at the same station, separate relationships are fit for each station-direction pair. The number of station-direction pairs is equal to the number of directed connections in the network. As we aim to find the relationship between the delays of each combination of two station-direction pairs, between any two points in the network, the number of relationships equals the square of the number of connections, 34 thousand in the case of Washington D.C.

$$n_{relationships} = n_{connections}^2 = 185^2 = 34.225 \tag{5.1}$$

With our desired relationships covering the stop-related variables of our delay data, we can assign the remaining variables as additional dimensions for our analysis. This includes the scheduled arrival time, the magnitude of the delay value, and the line that the train belongs to. Since the goal of our model does not involve the specific line a train delay belongs to, the first model will not include this variable (This variable is explored in the cross-examination of the second model in Chapter 7), leaving us with two dimensions to be defined: the temporal and the delay magnitude states.

First, we look at splitting the delay temporally. Given the amount of data, splitting the arrivals by date does not yield informative results; instead, we split the temporal dimensions by time of day. As will become apparent in Section 6.4, the best time step size for our modelling goal turns out to be whole hours, with 18 hours of the day having enough data to fit relationships, leading to 324 relationships per relationship from Equation 5.1. However, we can reduce this by considering that a delay cannot affect the behaviour of delays that occurred before it. This means that a relationship at one time step can only relate to the same or later time steps. This turns out to be the nth triangle number rule (see Equation 5.2), reducing our number of time step relationships to 171. Including the amount of time step relationships into our total increases our number of relationships to 5.9 million (Equation 5.3).

$$T_n = \sum_{k}^{n} k = 1 + 2 + \ + n = \frac{n^2 + n}{2} = \frac{n(n+1)}{2} \tag{5.2}$$

$$n_{relationships} = n_{connections}^2 \frac{n_{time\ steps}(n_{time\ steps}+1)}{2} = 185^2 * \frac{18(18+1)}{2} = 5.852.475 \tag{5.3}$$

Second is the magnitude of the delay itself. Splitting this up into intervals allows us to analyse the differences in behaviour between different delay severities. Again, as is discovered in Section 6.4, to ensure intervals have enough data to fit a relationship, we split up the delays into segments of 60 seconds. The usable intervals also range from 0 to 420 seconds, with an additional '420 and above' state. These eight delay magnitude intervals increase our total number of relationships to 46.8 million (see Equation 5.4), around 240 thousand per station-direction pair.

$$n_{relationships} = n_{connections}^2 \frac{n_{time\ steps}(n_{time\ steps}+1)}{2} n_{delay\ ranges} = 185^2 * \frac{18(18+1)}{2} * 8 = 46.819.800 \tag{5.4}$$

As we do not yet know what possible relationships exist, we must also choose a method which can infer relationships without manual input. To keep the application within the scope of this thesis, we will fit a distribution to describe the behaviour of each relationship. Which distribution is chosen depends on the structure of the relationship-specific data, and is chosen in Section 5.3. Altogether, these definitions make it so that our model describes 'The behaviour of delays at a station-direction pair at one hour given a delay with a magnitude within a certain one-minute range at a certain station-direction pair at a different hour.' All of these definitions make each relationship identifiable by the state of the following five variables:

- The station-direction pair at which a delay occurs.
- The time step at which that delay occurs.
- The magnitude of the occurring delay.
- The station-direction pair for which we want to show the behaviour.
- The time step at the station-direction pair for which we want to show the behaviour.

5.2. Fitting Relationship Functions

To find related delay values between the station-direction pair with the delay and the station-direction pair we want to find the related behaviour for, we iterate over all delays at the former that match the desired occurrence time step and delay range, and find all delays at the latter station-direction pair at the desired time step that occurred on the same date as the delay. This means that for a single delay, an hour of delay values at our desired station-direction pair is found and added to the total delays list for that relationship. Once all delays have been iterated over, we fit a distribution for that relationship using the whole list (see Algorithm 2).

Algorithm 2 Fitting of the first model.

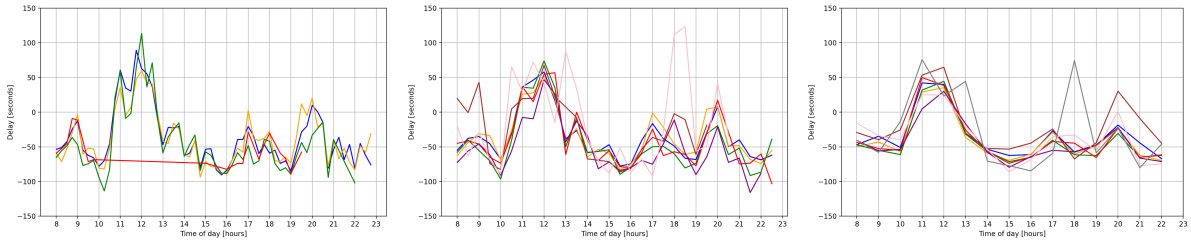
Require: Time steps T, all delays $delays$, positive delays $significant\_delays$
Ensure: Connection-wise fitted delay distributions

Split $delays$ by $(stop, next\_stop)$ into A_{conn}
Split $significant\_delays$ by $(stop, next\_stop)$ into S_{conn}
For each $connection \in A_{conn}$:
    Split into $(date, timestep) \mapsto$ list of delays
    Store in R_{conn}
for each $connection \in S_{conn}$ **do**
    Initialize lists: $L_{conn}, L_{delay}, L_{time}, L_{other}, L_{future}, L_{params}$
    Split $connection$ by delay range d_{range}
    for each d_{range} **do**
        Split by timestep $t \in T$
        for each t **do**
            Split by date d
            for each $other\ connection \in A_{conn}$ **do**
                Collect R_{other} such that $other\ connection \in R_{conn}$
                for each $t_{future} \in T, t_{future} \geq t$ **do**
                    Collect delays $\mathcal{D}$ where $(d, t_{future}) \in R_{other}$
                    Fit distribution $F = fit(\mathcal{D})$
                    Append $(connection, d_{range}, t, other\ connection, t_{future}, F)$
                end for
            end for
        end for
    end for
    Construct result table $\mathcal{T}$ from lists
end for

To determine when a relationship has sufficient data to be fit, we follow the reasoning in Heathcote et al. (2002), which states that the minimum adequate amount of data to fit a general distribution is 40 values. With this, we examine both additional analysis dimensions to determine if they are limited by the amount of data and to what extent.

Testing different time step sizes —15 minutes, 30 minutes, and whole hours —results in an example analysis plot as seen in Figure 5.1. Immediately it becomes clear that the 15 minute size is not viable, as many relationships do not have a singe time step with enough data, and so are missing entirely (note the lower amount of relationship lines in Figure 5.1a when compared to Figures 5.1b and 5.1c). While this problem is largely absent in the 30-minute step size version, the lower amount of data per relationship makes the differences between relationships of neighbouring time steps more volatile. This is not necessarily a problem, but it does make consistent differences in delay behaviour between variable states more difficult to see. As this is the primary goal of this model, we will use the 60-minute or whole-hour time step size. Although this loses delay behaviour detail, it makes the relationship transitions smoother. We also find that for all of the tested step sizes, too few values exist after 12 AM and before 5 AM, limiting us to 18 hours of the day.

Testing the magnitude of delay ranges similarly, whole-minute intervals contain enough data to fit most relationships. As we are only looking at delay values that are actual delays (i.e. above zero), our ranges start from 0 and go up by 60-second intervals. As less data is available for higher magnitude (see Figure 4.4), the limit for individual ranges is at 360 to 420 seconds, allocating all delay occurrences with magnitudes above 420 seconds to a '420+' range state.



(a) 15 minutes. **(b)** 30 minutes. **(c)** 60 minutes.

Figure 5.1: Differences in behaviour analysis plots for different time step sizes.

5.3. Quantifying Model Variance

For choosing a distribution, as we do not assume anything about how the relationship data is distributed, we implement a hands-off approach. We test a random sample of 50 relationship data lists on all 110 distributions used in the SciPy statistical functions packages, which represent almost all generally used distributions (The SciPy community, 2025) (see the example in Figure 5.2). From each, we calculate the sum of the squared errors (SSE). We take the top 10 performing distributions across these 50 tests, and test them on a larger sample set of 200, creating a boxplot of the SSEs for each in Figure 5.4. It shows that, in general, the performance of the first six distributions has only marginally different means and quantile ranges. To ease future comparison of results with the second model from Chapter 6, we opt for the exponentially modified Normal distribution.

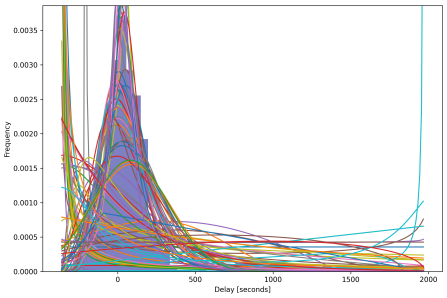


Figure 5.2: Example of test fitting of all distributions for a single relationship dataset.

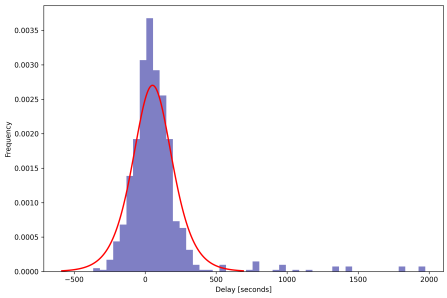


Figure 5.3: Chosen distribution (exponentially modified Normal) for the first model compared to a single relationship dataset.

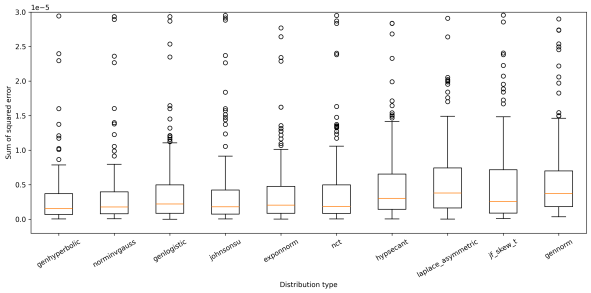


Figure 5.4: Boxplot of 200 SSE values for each of the 10 best performing distributions.

5.4. Analysis of Fitted Relationships

Before looking for delay propagation patterns, we first check if our fitted distribution properly encapsulates the data. As shown in the example in Figure 5.5, the fit distribution mirrors the actual delay behaviour well. As our time step size of one hour makes the model generalise the behaviour per hour, we lose some of the higher and lower peaks. The confidence range is noticeably expansive, while the mean seems to hover around zero. This is because, in each relationship, the average delay behaviour for the entire time step is modelled. The averaging of all delays within the hour causes the mean to hover around zero, but since the more extreme values remain in the dataset, the confidence interval remains large.

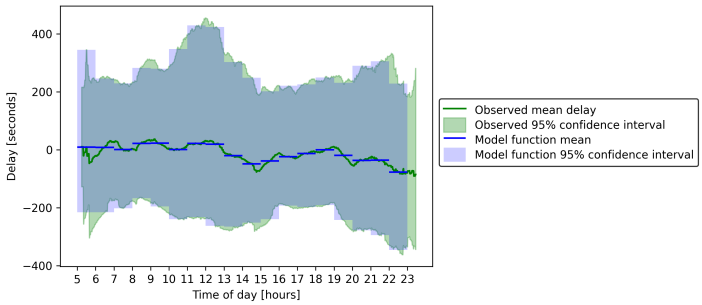


Figure 5.5: Example Distribution of fit model functions compared to the data. (For delay at station-direction D06-D07, given a delay of 0 to 60 seconds at station-direction D05-D06 at the same time step as the one analysed, for all time steps.

With the results of the model, we can already use it as a tool for analysing delay behaviour, such as showing the average delay of the entire network given a specific delay occurrence (Figure 5.6). However, the use of this model was to discover relationships between delay behaviour and our included variables, so further analysis is needed.

As mentioned at the end of Section 5.1, the model has five dimensions by which we can analyse our results: The station-direction pair at which a delay occurs (delayed station-direction); The magnitude of the occurring delay (delay magnitude range); The time step at which that delay occurs (delay occurrence time step); the station-direction pair for which we want to show the resulting delay behaviour (station-direction to analyse); The time step at the station-direction pair for which we want to show the resulting delay behaviour (time step at to analyse). Within a 2D graph, with the resulting delay behaviour as one dimension we always want to show, we can only display the full breadth of two other variables and must make example choices for the others.

When examining the behaviour for all delay ranges and time steps of occurrence, it shows that in general, one delay in the network does not have a significant effect on other parts, no matter its magnitude (Figure 5.7) or time of occurrence (Figure 5.8), with one major exception. Examining the delay behaviour of all station-direction pairs during a single high-magnitude delay (Figure 5.9a), we find that, first, the end-of-line station-direction pairs show relatively extreme behaviour compared to the rest of the network (Figure 5.9b). Tracing these results back to the original data reveals that inconsistencies in the arrival data are causing erroneous values. Disregarding these pairs highlights the second observation: station-direction pairs that share a line with the station-direction at which the delay occurred appear to have some correlation, as their delay values around the time of the delay occurrence are elevated compared to the rest of the network (Figure 5.9c).

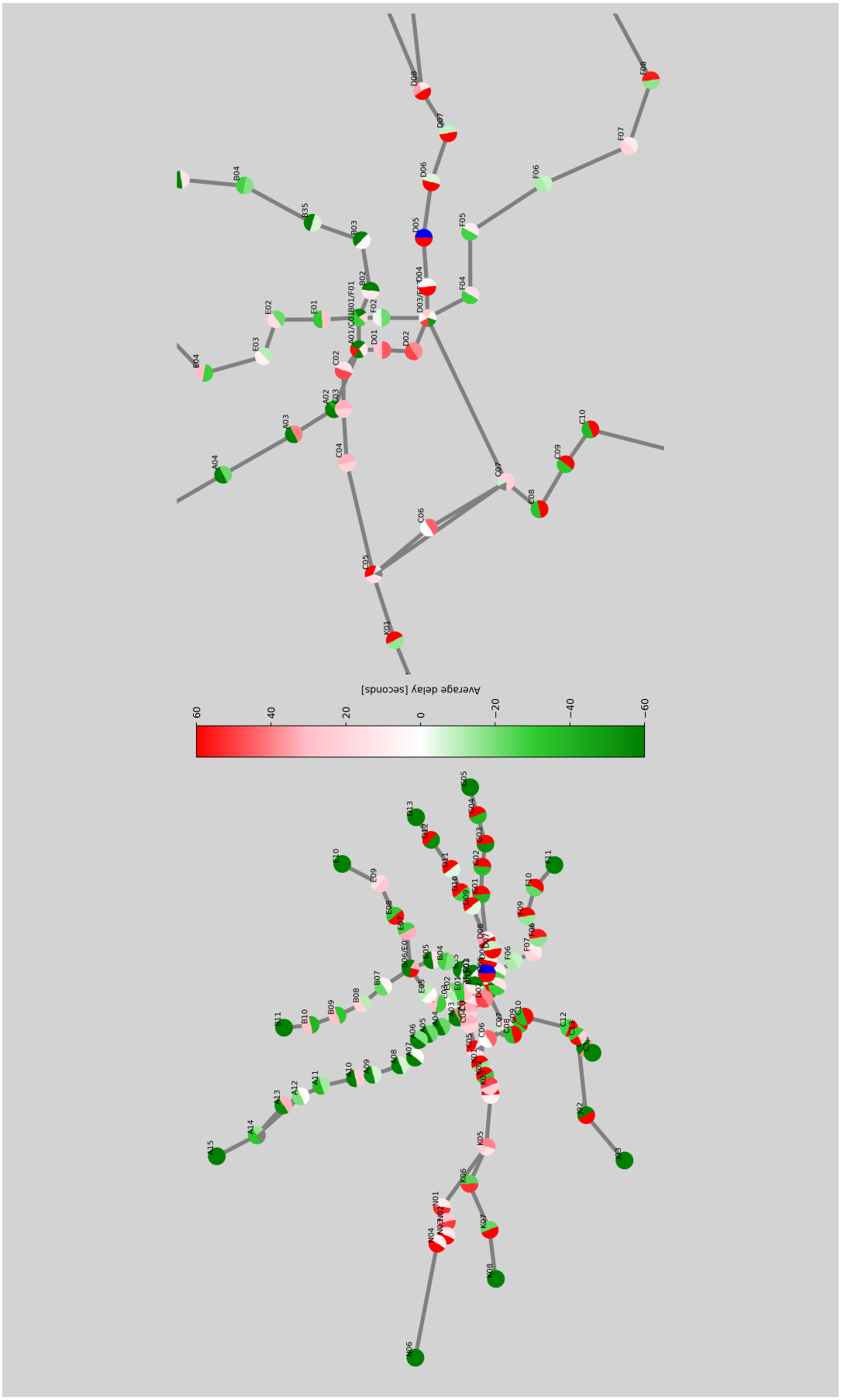


Figure 5.6: Mean delay at all station-direction pairs at 10 AM given a 240 to 300 second delay at station D05 in the direction of D06 (highlighted in blue) at 9 AM. The direction in which each segment of a station delays circle points represents the direction of travel of the station-direction pair.

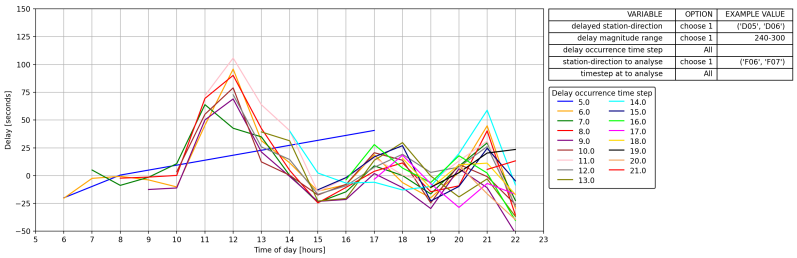


Figure 5.7: Delay behaviour per delay occurrence time step.

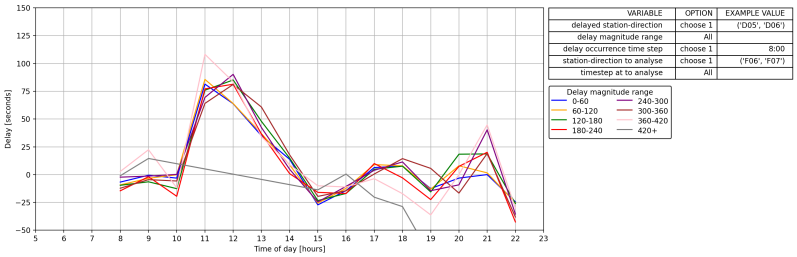
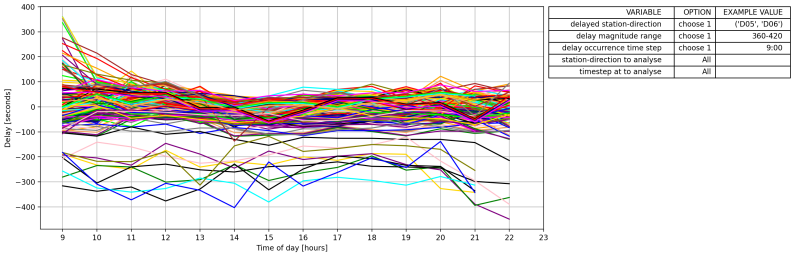
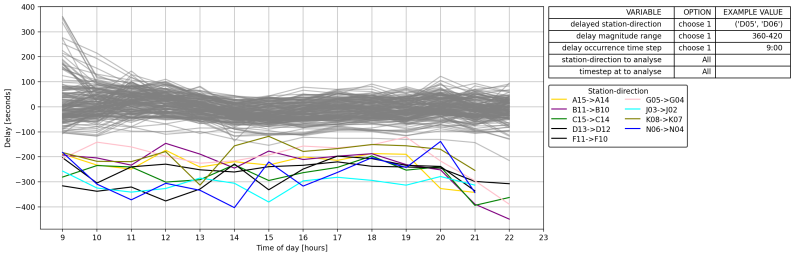


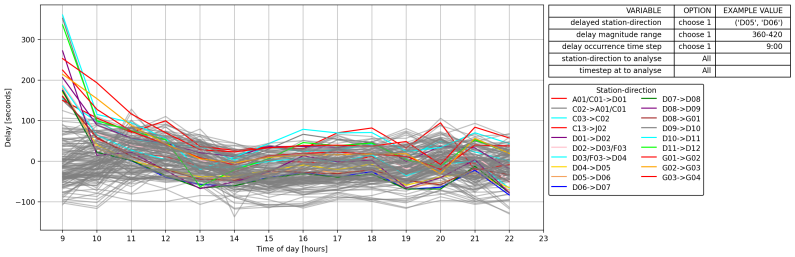
Figure 5.8: Delay behaviour per delay magnitude range.



(a) All behaviours.



(b) Out-of-ordinary lower values highlighted.

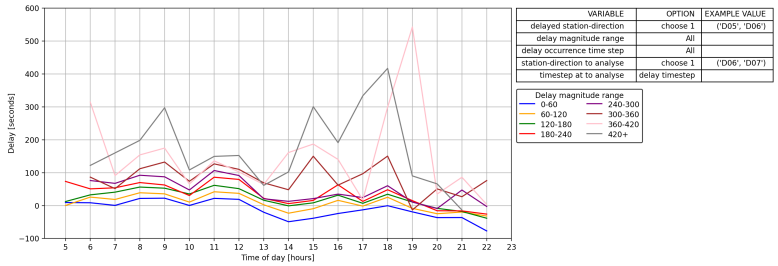


(c) Out-of-ordinary higher values highlighted (with lower out-of-ordinary removed).

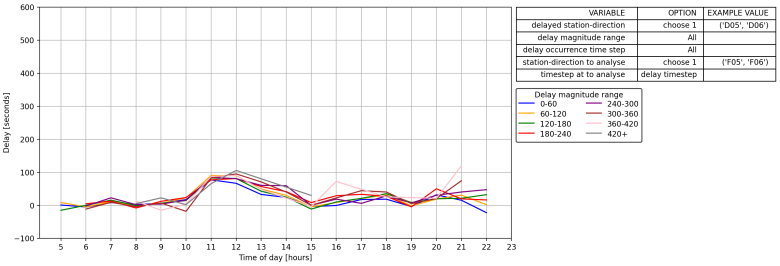
Figure 5.9: Delay behaviour at all station-direction pairs for a specific significant delay.

5.5. Exploring Significant Propagation Patterns

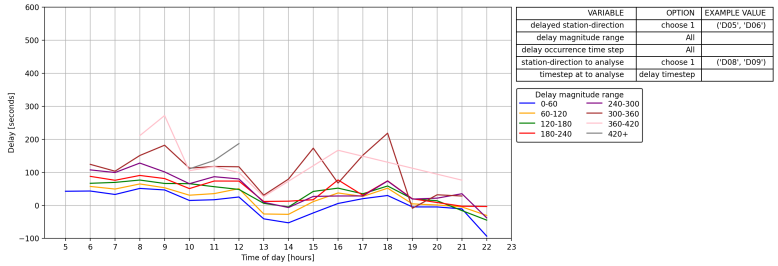
The model exploration reveals that the delay behaviour for route-sharing stations exhibits some correlation. As the correlation only seems to exist for time steps close to the time step of occurrence, we can examine the correlation closely by plotting the resulting behaviour of route-sharing stations for the same time step as the delay occurrence time step, and plot this for all hours of the day (Figure 5.10a). There is a clear higher mean behaviour for each higher delay magnitude. Looking at non-route-sharing station-direction pairs, this behaviour is not visible (Figure 5.10b). Lastly, the behaviour seems to be strongest between station-direction pairs that share a direct connection. However, it remains present even as we move further away from the location of delay occurrence (see Figure 5.10c). This agrees with the findings of Cats and Hijner (2021), which studies the informativity of specific stations and network attributes on passenger delays. They also discover that the station-direction pairs that can tell the most about delays at other station-direction pairs if they share a line.



(a) For directly connected station-direction pairs.



(b) For non-route sharing station-direction pairs.



(c) For route-sharing but non-connected station-direction pairs.

Figure 5.10: Examples of delay behaviour per station-direction per delay magnitude at the same time step as the delay occurrence.

This phenomenon becomes explainable when we take the logic of a metro network into account. If a delay does not increase due to external circumstances, a train with a significant delay at one stop will experience a similar or slightly reduced delay at the next stop, and even more reduced at the third. Not only does this mean that the propagation relationship exists mainly between directly connected stations, but single instances of delays between stations can also be directly correlated, as they belong to the same train. Using the train IDs from our data, we find correlated delays between two connected stations by shifting our delay values as shown in Algorithm 3. As this data links a delay value to a single other delay, rather than to all delay values in the time step window, we expect to see a tighter confidence interval. Indeed, when performing a similar confidence test as in Figure 5.5, but using just related delay values, it shows a significantly tighter range than our models are fitted for (Figure 5.11a), and also shows a strong correlation with the magnitude of the delay (Figure 5.11b). As the current model is designed to examine all possible relationships, it is too general to provide an in-depth analysis of the significant relationships discovered. A second model focused on this relationship can offer more insight.

Algorithm 3 Relating delays between directly connected stops.

Require: Table $delays$ with $(trip\_id, sched\_arrival\_time, stop, delay)$
Ensure: Table $related$ with paired delays between consecutive stops

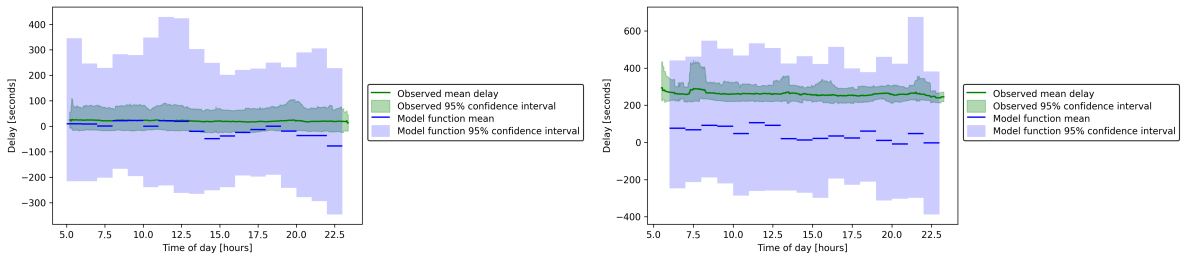
1: $delays\_sorted \leftarrow$ sort $delays$ by $(trip\_id, sched\_arrival\_time)$
2: Extract arrays:

$$stop\_values \leftarrow delays\_sorted.stop,$$
$$delay\_values \leftarrow delays\_sorted.delay,$$
$$trip\_ids \leftarrow delays\_sorted.trip\_id$$

3: Construct $related$ with columns:

$$stop \leftarrow stop\_values$$
$$next\_stop \leftarrow (stop\_values[2..end], stop\_values[end])$$
$$delay \leftarrow delay\_values$$
$$delay\_at\_next \leftarrow (delay\_values[2..end], delay\_values[end])$$
$$trip\_id \leftarrow trip\_ids$$
$$prev\_trip\_id \leftarrow (trip\_ids[end], trip\_ids[1..end-1])$$
$$next\_trip\_id \leftarrow (trip\_ids[2..end], trip\_ids[1])$$

4: Keep only rows where $trip\_id = prev\_trip\_id$ or $trip\_id = next\_trip\_id$
5: For rows with $trip\_id \neq next\_trip\_id$, set $next\_stop, delay\_at\_next \leftarrow \emptyset$
6: Filter $related$ to rows with non-null $delay$ and $delay\_at\_next$
7: **return** $related$



(a) Delay range of 0 to 60 seconds.

(b) Delay range of 240 to 300 seconds.

Figure 5.11: Confidence intervals of the first model result compared to related delay values of directly connected stops.

6

Model II: Quantifying Local Propagation

To further explore the relationship between notable delay behaviour and line-sharing stops discovered by the first model, we construct a second model focused solely on quantifying this relationship. To keep the complexity of the model within the scope of the thesis, we look only at the propagation behaviour between directly connected stations. The goals, simplifications and assumptions of this second model are as follows:

***Goal**: Construct a model that quantifies the delay propagation effects between directly connected station-direction pairs.*

- We will use only the variables present in the delay data itself; no external datasets will be included.
- Only the correlation between directly connected stations will be explored.
- All connections are assumed to have one track per direction, and all stations are assumed to have one berth per connection.
- We assume no delay mitigation strategies to be present when filtering our data for the fitting of the propagation functions.

6.1. Defining the Model Relationship Structure

By looking at relationships between directly connected stations, we can link individual delays between two station-direction pairs by seeing which delays have the same trip ID using Algorithm 3 from Chapter 5.5. Having the delay values linked individually allows us to establish mathematical dimensions that can aid in defining our delay propagation functions. As a train's route is linear and directed, and order of the trains travelling the same route cannot change (unless intervention measures are taken), we can set one dimensions as the sequence of stops of the trip, and the other as the sequence of trains travelling along the route (see Figure 6.1). As we are concerned with finding the propagation of delays between connected station-direction pairs, we thereby want to find the relationship between $Delay_k^{j-1}$ and $Delay_k^j$.

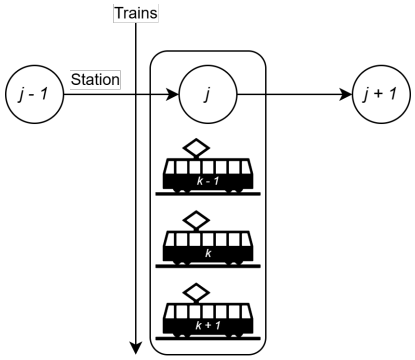


Figure 6.1: Second model propagation dimensions.

6.2. Identifying Propagation Types

Although we are concerned with the delay propagation between station-direction pairs, looking at our propagation dimensions, we can actually identify two propagation types, one along each axis: The first is a case that most, if not all, public transport users are familiar with: a train experiences a delay at a stop, and this delay is propagated to some extent at the next stop in its trip. This is the case we want our new model to quantify. The second situation involves a train being delayed significantly, causing the next train on the same route to wait for the first train to depart before it can reach the same station, thereby inheriting some of the delay. As the first type relates to the propagation between two stops along a train's route, we call it 'stop-to-stop propagation'. The second denotes the propagation of delay between two trains, which we call 'train-to-train' propagation. We are concerned with modelling the stop-to-stop propagation type. However, since the train-to-train propagation type can also be the cause for observed delays, we must first identify and filter out all instances of this propagation from our delay dataset.

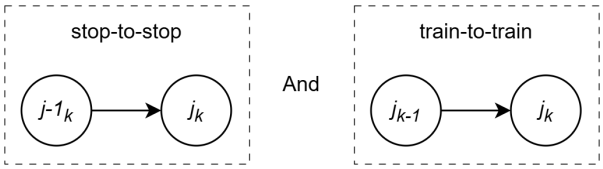


Figure 6.2: Propagation types for directly connected stations as defined using the axes from Section 6.1.

6.2.1. Train-to-train Delay Propagation

For our train-to-train propagation, we are concerned with finding all instances where $Delay^j_{k-1}$ actually affects $Delay^j_k$. We find all related $Delay^j_{k-1}$ and $Delay^j_k$ values in our data using Algorithm 4. Plotting the results for our entire dataset in Figure 6.3 shows some notable stripes in the dataset.

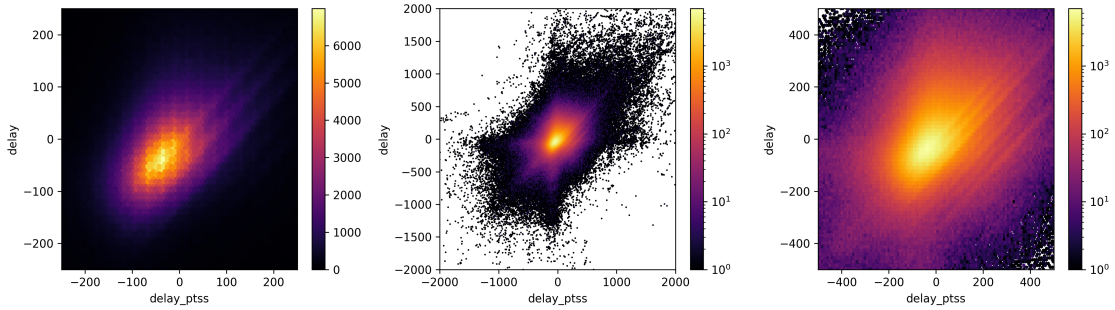


Figure 6.3: Heatmap of delay at current connection versus the delay at the current connection from the previous train (ptss). Normally plotted (left), plotted logarithmically (middle), zoomed in on the centre of the data (right).

To filter out the occurrence of train-to-train propagation, we turn to the general operational logic of a metro network. First, we check for a conflict between the two trains by filtering for all arrivals where the actual arrival time of the first train is later than the scheduled arrival time of the second train. The middle figure of Figure 6.4 shows the results of this filter for a single station-direction pair. We notice that although the arrivals of the two trains interfere, if the first train experiences some delay above zero, the second train's delay sometimes actually decreases, falling below zero, visible in the values on the bottom half of the middle graph. If this is due to a manual delay mitigating intervention by network operators, or due to errors in the data, we do not know. In either case, we do not want to count them towards our train-to-train propagation occurrences. A second filter, therefore, checks if the real arrival time of the second train is actually later than that of the first train. This then gives instances of train-to-train propagation (right figure of Figure 6.4), which can be filtered out when fitting our stop-to-stop propagation functions.

Algorithm 4 Finding all instances of delays that could be related via train-to-train propagation.

Require: Table $delays$ with $(trip\_id, sched\_arrival\_time, real\_arrival\_time, stop, delay, line\_name)$
Ensure: Table $related$ with paired delays across stops of the same trip and across trains at the same station

$delays\_sorted \leftarrow$ sort $delays$ by $(trip\_id, sched\_arrival\_time)$

Extract arrays:

$$stop\_values \leftarrow delays\_sorted.stop$$
$$delay\_values \leftarrow delays\_sorted.delay$$
$$trip\_ids \leftarrow delays\_sorted.trip\_id$$

Construct $related\_inplace$ with columns:

$$line \leftarrow delays\_sorted.line\_name$$
$$stop \leftarrow stop\_values$$
$$prev\_stop \leftarrow (stop\_values[end], stop\_values[1..end-1])$$
$$next\_stop \leftarrow (stop\_values[2..end], stop\_values[1])$$
$$delay \leftarrow delay\_values$$
$$sched\_arrival \leftarrow delays\_sorted.sched\_arrival\_time$$
$$real\_arrival \leftarrow delays\_sorted.real\_arrival\_time$$
$$trip\_id \leftarrow trip\_ids$$
$$prev\_trip\_id \leftarrow (trip\_ids[end], trip\_ids[1..end-1])$$
$$next\_trip\_id \leftarrow (trip\_ids[2..end], trip\_ids[1])$$

Keep only rows where $trip\_id = prev\_trip\_id$ or $trip\_id = next\_trip\_id$
For rows with $trip\_id \neq prev\_trip\_id$, set $(prev\_stop, real\_arrival\_prev\_stop) \leftarrow \emptyset$
$related \leftarrow$ sort $related$ by $sched\_arrival$
Partition $related$ into groups $(stop, next\_stop) \mapsto station\_stops$
for all $station\_stops$ in groups **do**
    Add columns:

$$delay\_ptss \leftarrow (\emptyset, station\_stops.delay[1..end-1])$$
$$real\_arrival\_prev\_train \leftarrow (\emptyset, station\_stops.real\_arrival[1..end-1])$$
$$sched\_arrival\_prev\_train \leftarrow (\emptyset, station\_stops.sched\_arrival[1..end-1])$$

    Append updated $station\_stops$ to results
end for
$related \leftarrow$ concatenate all updated groups
return $related$

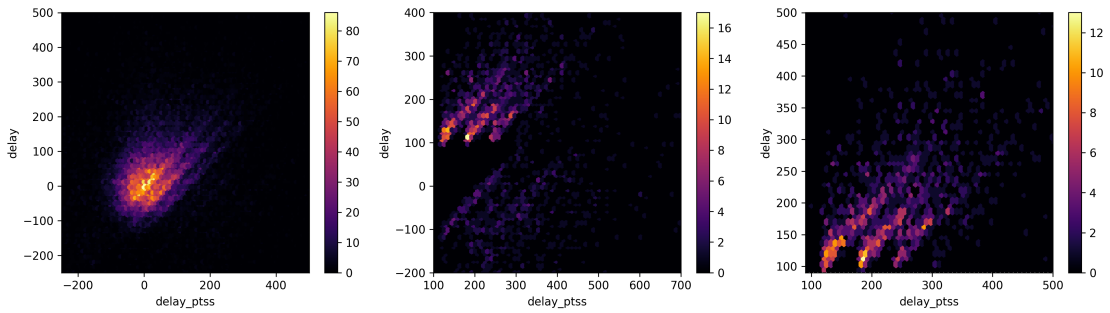


Figure 6.4: Heatmap of train-to-train correlated delay data for a single station-direction pair, plotting the total (left), only instances of train arrivals interfering with each other (middle), and the actual instances of train-to-train propagation (right).

6.2.2. Stop-to-stop Delay Propagation

For stop-to-stop propagation, we want to find a function to express the relationship of $Delay_k^{j-1}$ to $Delay_k^j$. We find the related delay values using Algorithm 3 from Chapter 5.5. First, we find and filter out all occurrences of train-to-train propagation, which accounts for 80 thousand of the 3.9 million delay values. Plotting the filtered delay data with respect to the delay at the previous station, we see a clear linear relationship between the two (left figure of Figure 6.5). Plotting the same data for a single station-direction pair (middle figure of Figure 6.5), this linear behaviour is still present. We extract this linear component by subtracting the delay at the previous station from the current delay, resulting in a delay difference. Plotting these residual differences (right figure of Figure 6.5) shows that, besides the linear component, no clear relationship is left, except for a general distribution for all delay values at the previous station. Although it might appear like the distribution varies across the x-axis, this is due to fewer values being available at the ends of the axis range. Notably, the mean of the residual data seems to hover around some non-zero value. This means that the distribution's mean can also be extracted, providing a single value that describes the delay propagation effect of travelling across that connection for that relationship's timeslot. This means that our stop-to-stop propagation function can be defined as equation 6.1, with ϵ being the residuals distribution.

$$Delay_k^j = Delay_k^{j-1} + \epsilon \approx Delay_k^{j-1} + \epsilon.\mu \tag{6.1}$$

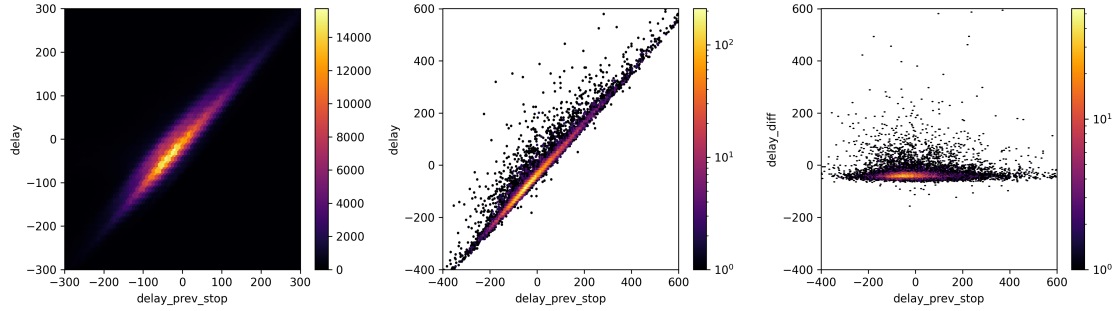


Figure 6.5: Heatmap of delays at a station-direction pair with respect to the delay at the previous trains travelling through that station-direction. Plotted is the total dataset (left), the data for one station-pair (middle), and the same data as the middle figure, but with the linear component removed (right).

6.3. Defining Analysis Dimensions

As we only look at the propagation between directly connected stations, we are actually modelling the effect on delays from travelling along one of the connections in the network. As we examine the difference between two delay values, the final residual distribution will describe the propagation effect of travelling across a particular connection. This gives us 185 connections for which we want to model the propagation behaviour.

With the function found in Section 6.2, how the magnitude of the initial delay translates to delays at the end of the relationship is already defined in the functions. Therefore, the magnitude of the delay does not need to be an additional dimension of analysis, as it was in the first model. This means we only need to separate our function temporally. Just as with the first model, we will separate the functions per time step; our time step size will be 1 hour. We do not need to select two time instances, as the affected time is always the scheduled arrival time of the train carrying the delay; therefore, we only need 18 time steps-split relationships per connection.

The dimensional complexity of our first model (3 analysis dimensions: delay range, time of delay occurrence, future time step next to the selection of origin and destination connection) made it more difficult to explore any single dimension in more detail. The more focused approach of this second model allows us to explore the time dimension in greater depth. As Ayana et al. (2023) shows that behaviour can vary by day of the week, in addition to fitting each relationship for that time step, we will also split the relationships by day to see if any patterns emerge. With this, we run the model fitting eight times (one for each day of the week, and one for all days combined), making the total number of functions to fit 26.6 thousand, as seen in Equation 6.2.

$$n_{relationships} = n_{connections} n_{timesteps} * 8 = 185 * 18 * 8 = 26.640 \tag{6.2}$$

6.4. Fitting Model Functions and Variance

With our relationship function defined and our dimensional granularity decided, we can start fitting our propagation functions. Again, we find our stop-to-stop related delays using Algorithm 3. We then find and filter out the instances of train-to-train propagation. The residuals for which the distributions need to be fitted are calculated in advance by subtracting the delay at the previous stop from the delay. Then we can fit our function distributions on these residuals (see Algorithm 5), including a minimum data check similar to the first model to assure a proper distribution fit can be achieved, again following the minimum data amount of 40 values advised in Heathcote et al. (2002). Performing the identical distributions test as in Chapter 5.3 (see Figure 6.6), we find that the exponentially modified Normal is the best distribution overall (Figure 6.7).

Algorithm 5 Fitting of the second model.

Require: Table $related\_delays$ with $(prev\_stop, stop, timestep, delay, delay\_prev\_stop)$

Partition $related\_delays$ by $(prev\_stop, stop, timestep) \mapsto connection\_split\_data$

Initialise empty lists:

$$timestep\_list,\ from\_list,\ to\_list,\ params\_list$$

for all $(from\_stop, to\_stop, t) \mapsto data$ in $connection\_split\_data$ **do**
    $data\_list \leftarrow \{delay\_diff \mid delay\_diff \in data\}$
    if $|data\_list| > 40$ **then**
        Append t to $timestep\_list$
        Append $from\_stop$ to $from\_list$
        Append to_stop to to_list
        $dist\_params \leftarrow fit(data\_list)$
        Append $dist\_params$ to $params\_list$
    end if
end for
Construct $function\_parameters$ with columns:

$$(timestep, from\_stop, to\_stop, dist\_params) \leftarrow (timestep\_list, from\_list, to\_list, params\_list)$$

return $function\_parameters$

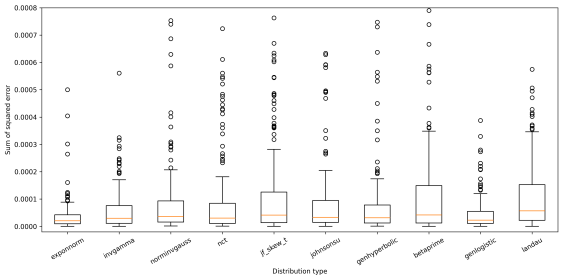


Figure 6.6: Boxplot of 200 SSE values for each of the 10 best performing distributions.

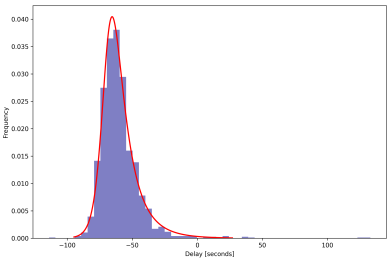


Figure 6.7: Chosen distribution (exponentially modified Normal) for the second model compared to a single relationship dataset.

6.5. Evaluating Model Performance

With our fitted distributions, we first check if our model matches our data. Plotting the means and 95th percentile confidence interval of both our model distribution and our delay-difference data in Figure 6.8, we can see the model properly encapsulates the delay propagation data. We can also use the model and the delay at the previous stop to predict what the delays at the next stop might be. By doing this for one connection and plotting the resulting mean and confidence interval (Figure 6.9), it shows that the model's prediction aligns well with the original delay data. Returning to the confidence test on directly connected stations for the first model, we can now incorporate our new model's prediction for the same dataset (Figure 6.10). Note that the second model aligns with the data significantly better than the first model, demonstrating its accuracy even with datasets on which it was not trained, albeit slightly wider confidence margins that the data.

As we have now fitted a function instead of just a distribution, we also perform a goodness-of-fit test. Since we are not testing the relative performance of different propagation functions, and tests like the lack-of-fit sum of squares are already partially covered by our SSE distribution test, we use a coefficient of determination test (Steel & Torrie, 1960) to test for goodness-of-fit. With zero being worst and one being best, the second model's relationships have a mean score of 0.931, a standard deviation of 0.085, and a median of 0.958, indicating that the model adequately captures the underlying relationships.

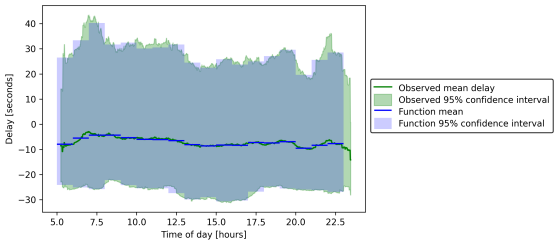


Figure 6.8: Comparing the means and confidence intervals between the model and the delay difference data.

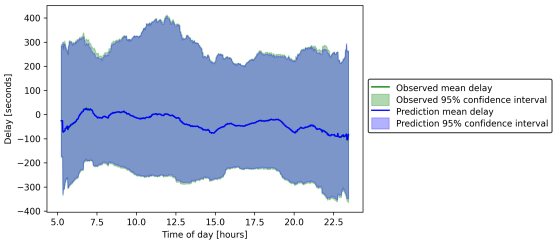


Figure 6.9: Comparing the means and confidence interval between the delay prediction of the model and the delay data.

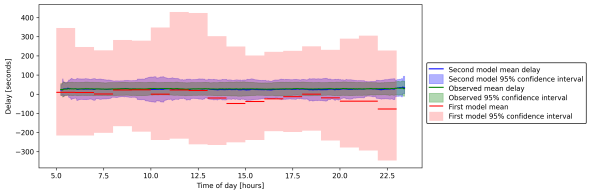


Figure 6.10: Comparing propagation behaviour prediction of first and second models for a directly connected station for delays within the range of 0-60 seconds.

6.6. Analysing Model Results

Displaying the mean delay change of each relationship function's distribution in a single table in Figure 6.11, it seems that while the mean behaviour might vary from connection to connection, the behaviour of a single connection remains relatively consistent across the hours of the day. Plotting the same table for the upper and lower bounds of the 95th percentile confidence range (Figure 6.12 and 6.13 respectively), we see similar behaviour. Just like in the first model, the end-of-the-line connections seem to have extreme values due to data inconsistencies. A notable exception to the general behaviour is a slight worsening of propagation behaviour between the hours of 10 and 12 AM. Although this could be due to slight inconsistencies in the data, the more relaxed timetables during these hours (when compared to peak periods) could lead to different behaviour. However, this needs to be investigated further. Based on the results in the three tables, each connection has its own magnitude of ability to mitigate the delays of the train travelling along it.

Another aspect we can analyse is the differences in behaviour with respect to the day of the week. We observe that, in most cases, the 'all days' behaviour is matched across all days of the week, as shown in Figures 6.14a, 6.14b, and 6.14c. A select few connections exhibit different behaviour when looking at different days (Figure 6.14d). Additionally, other connections further exhibit both day-of-the-week- and hour-specific differences in behaviour (Figure 6.14e) (mainly around the off-peak hours as discussed in the examination of the all-connection tables). Considering this to be the final results of this model, we list the tables for all connections in Appendix B. In general, we observe that the day of the week or hour of the day does not give significant variations in delay propagation behaviour.

With most delay propagation values per connection being consistent for all hours and days of the week, we calculate a single, average delay propagation amount per connection, and plot it on the metro network map (Figure 6.15) (Versions of the map for the upper and lower bound results can be found in Appendix C). Here we observe similar results as in contemporary papers like Büchel et al. (2020), with the magnitude of the propagation effect not seeming to be located in a specific part or centred around specific stations (notice the multiple different behaviours around station D03/F03 in the centre). It seems that the delay propagation behaviour of each connection is not related to the behaviour of neighbouring connections. Notably, the behaviour is similar for most connections between the same two stations, suggesting that it might be related to properties of the connection itself.

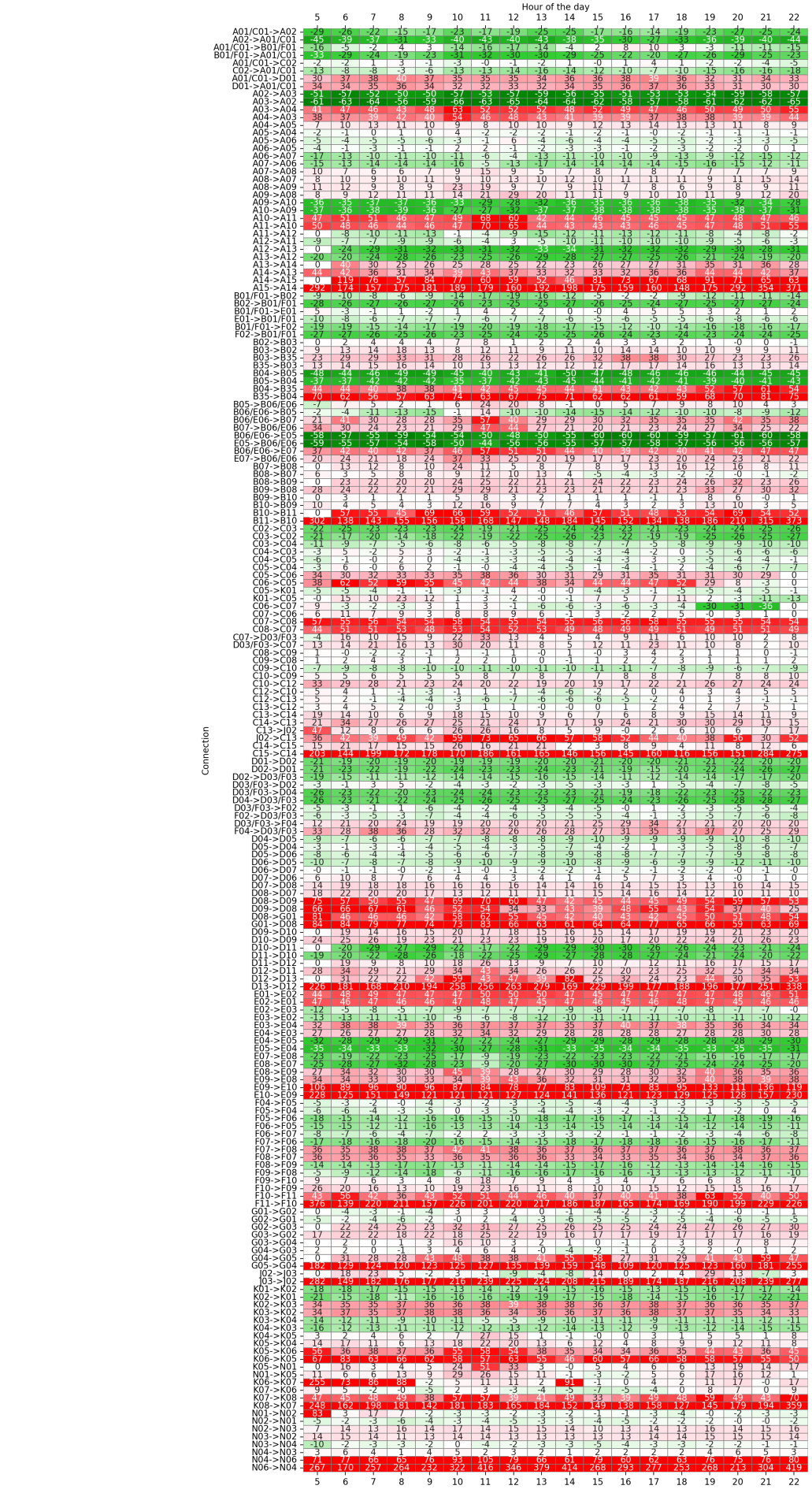


Figure 6.11: Mean delay propagation behaviour per connection.

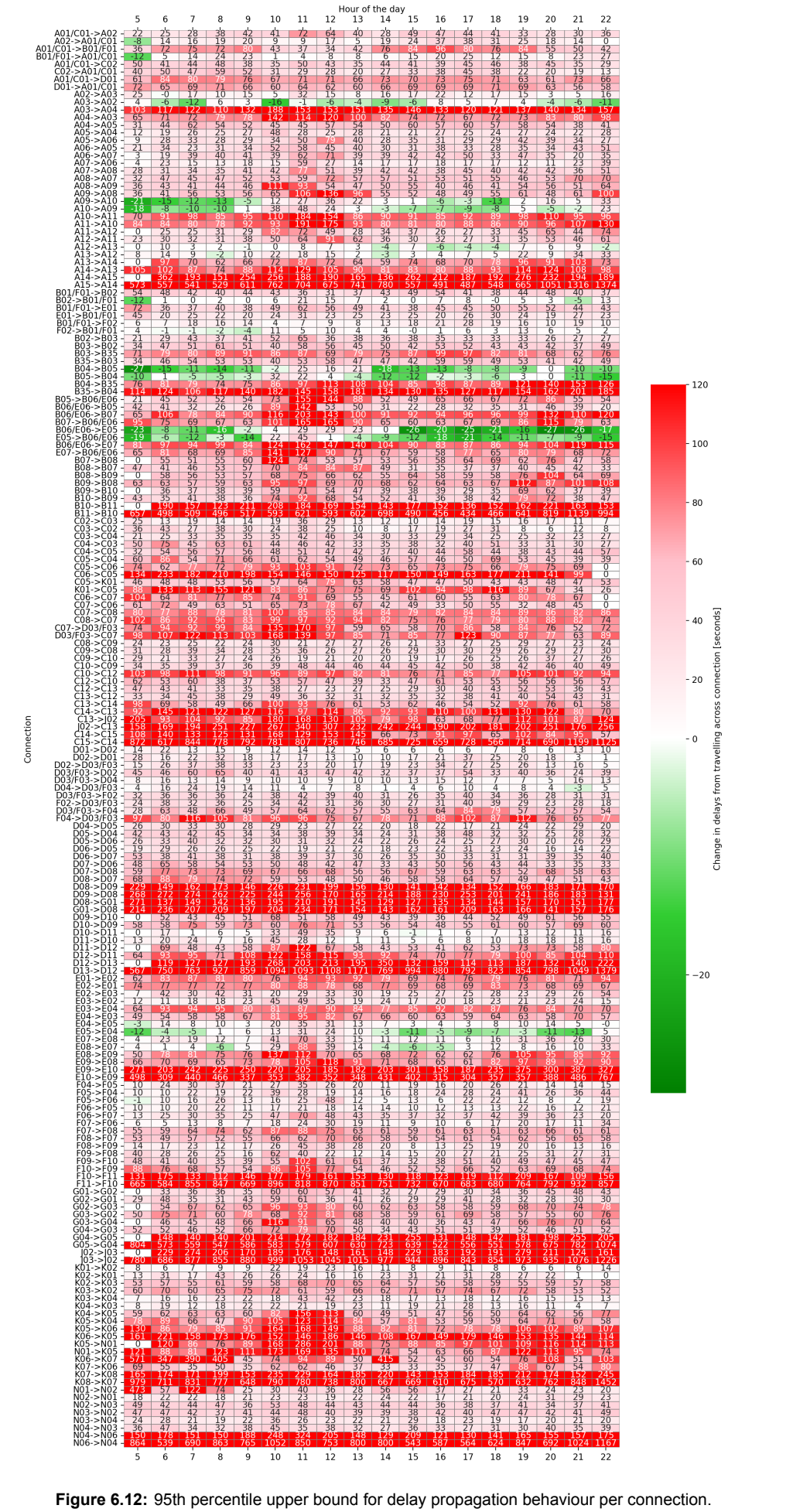


Figure 6.12: 95th percentile upper bound for delay propagation behaviour per connection.

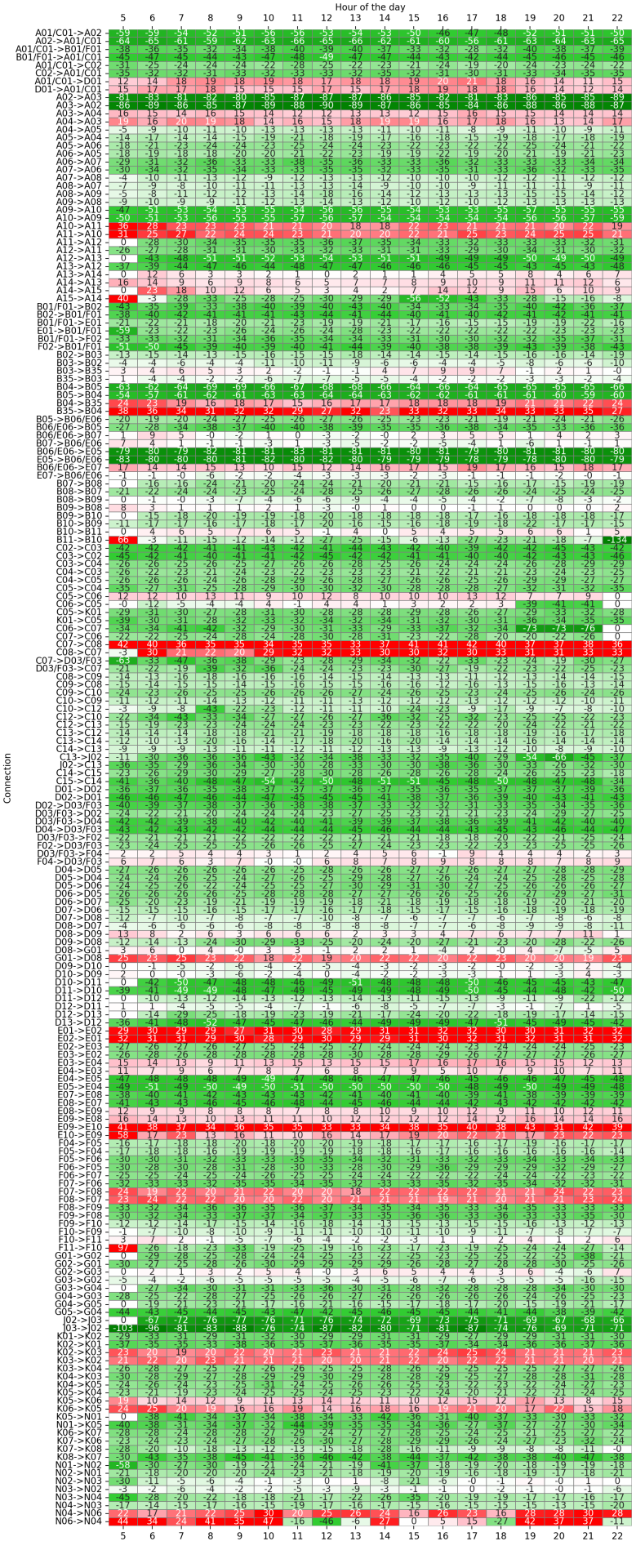
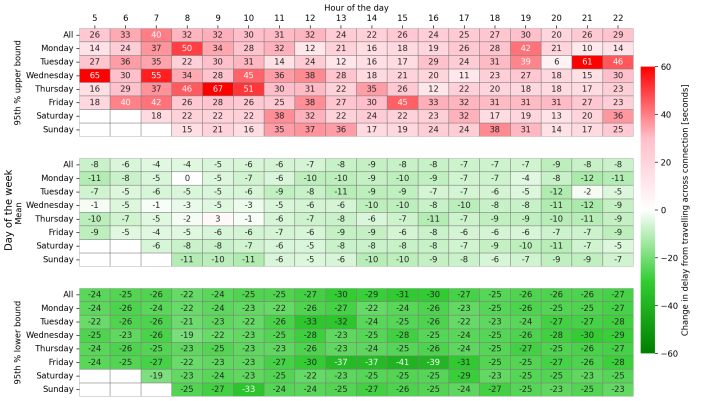
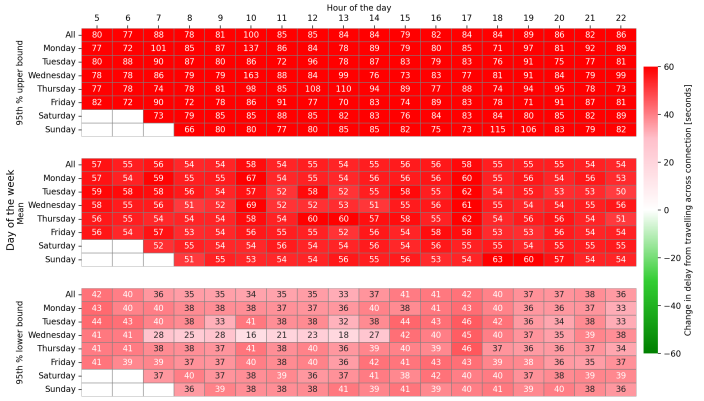


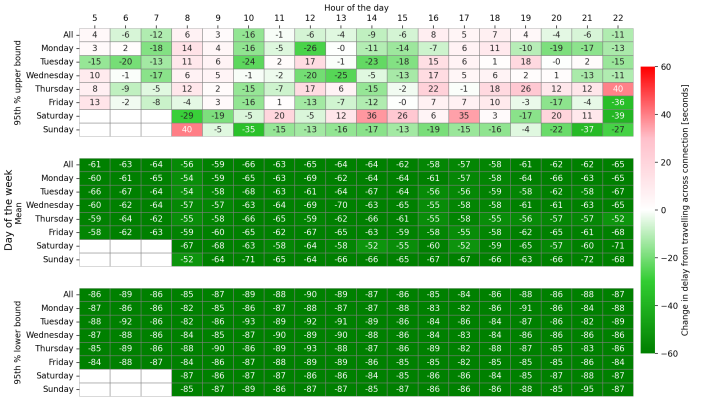
Figure 6.13: 95th percentile lower bound for delay propagation behaviour per connection.



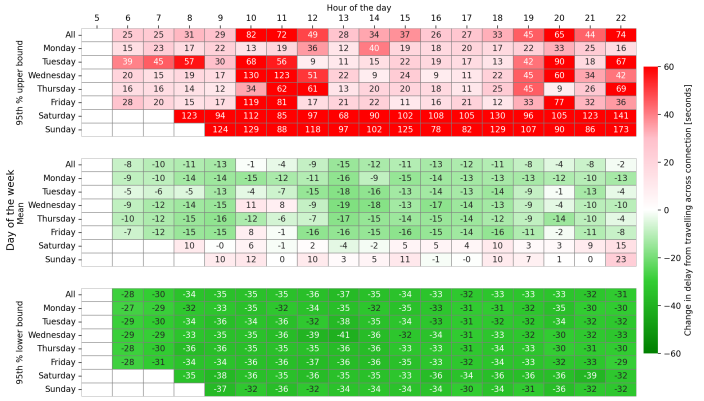
(a) Connection D05 to D06.



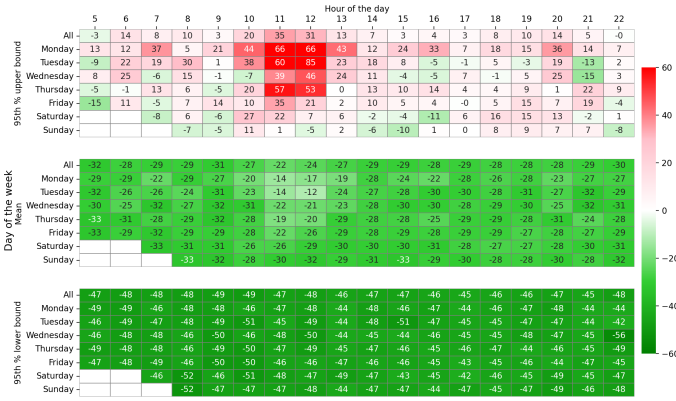
(b) Connection C07 to C08.



(c) Connection A03 to A02.



(d) Connection A11 to A12.



(e) Connection E04 to E05.

Figure 6.14: Notable examples of delay propagation behaviour of connections per day of the week.

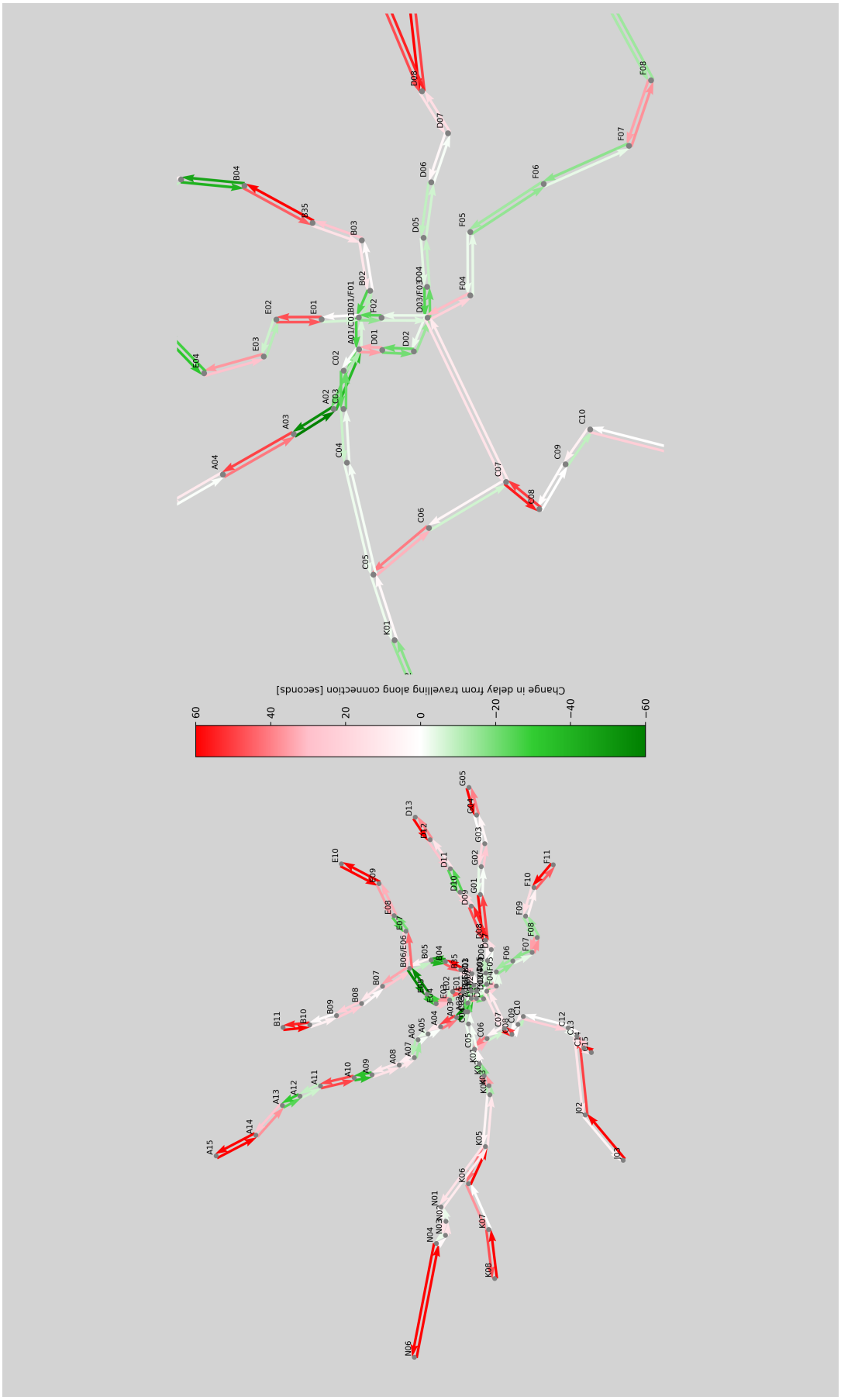


Figure 6.15: Mean delay propagation behaviour per connection, with the total metro network on the left, and zoomed in on the network's centre on the right. For two connected stations, the behaviour of the connections in both directions is similar. At the same time, there seems to be no correlation between neighbouring connections or those that share the same routes or lines.

7

Cross Examination: Propagation Behaviour Versus Network Characteristics

With most of the stop-to-stop propagation functions showing no significant variation over time or for different days of the week, we examine which other network design or operational variables can show secondary patterns in the propagation behaviour. As this project only uses the WMATA-provided data and the GTFS dataset, this chapter will examine the network design and operational variables available in this data and attempt to uncover possible existing patterns in propagation behaviour for each of them. The extreme values of the end-of-the-line connection are filtered out from these results, as they are due to data inconsistencies. The unfiltered graphs can be found in Appendix D. For each variable, we look at the mean, as well as the lower and upper bounds of the 95th percentile of the distributions.

7.1. Line Related

We first look at the metro lines. By gathering all stations that are part of a single line and creating a boxplots of their behaviours, we see that they show very similar behaviours (Figure 7.1). To check if the behaviour within one line may vary, we split them up per direction (Figure 7.2). The Red and Green lines show better mean and lower bound behaviour, with Red showing more extreme quantiles when split per direction. A possible reason for this might be that the Red line is the only one that does not share any part of its route with any other line (see Figure 4). However, separating the propagation behaviours by the number of lines that use the connection in Figure 7.3 shows that this is not the case, indicating that the different behaviour of the Red line is not due to its non-route-sharing characteristic.

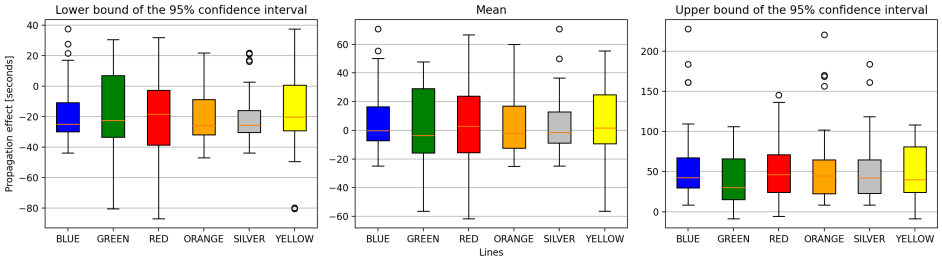


Figure 7.1: Propagation behaviour per line.

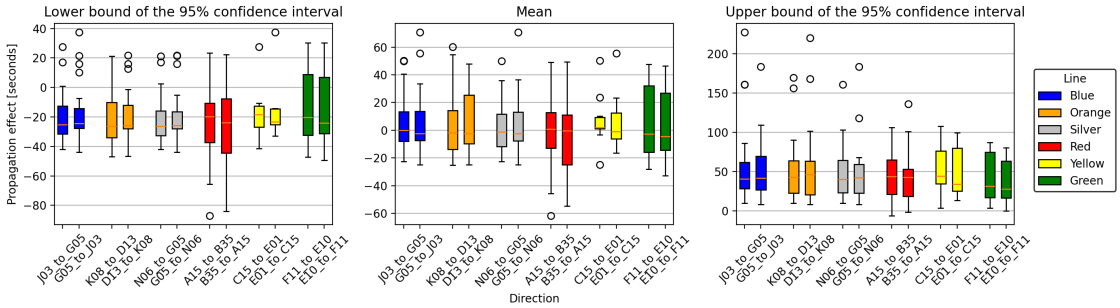


Figure 7.2: Propagation behaviour per line per direction.

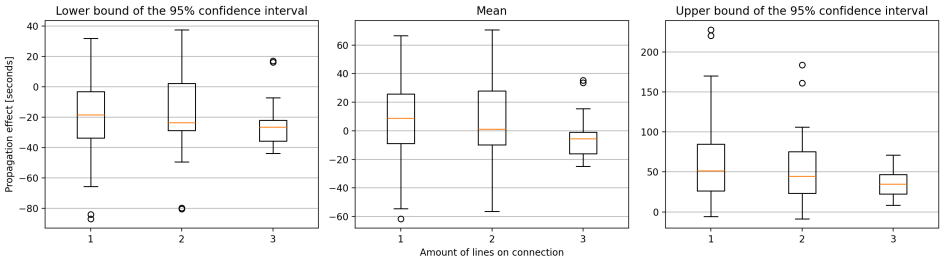


Figure 7.3: Propagation behaviour per number of lines on the connection.

7.2. Operational Related

Each connection also has some variables that can be examined. The propagation behaviour per average headway (Figure 7.4) shows a slight worsening trend in propagation behaviour, being stronger for the upper bound, but still low when compared to the overall range general upper bound behaviour shown in Figure 6.12. Comparing the propagation behaviour against the scheduled travel time reveals similar trends (with the 60 and 300 values having erroneous data due to the low number of data points). Both of these findings agree with those in Huang et al. (2024), which show the same slight trends for their models' residual distributions when comparing them against dwell times, headways and travel times.

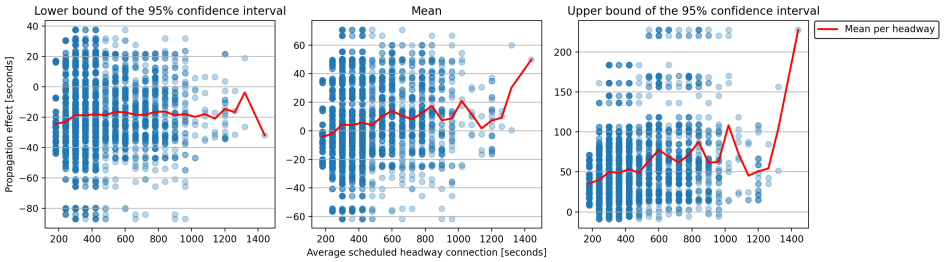


Figure 7.4: Propagation behaviour per average scheduled headway.

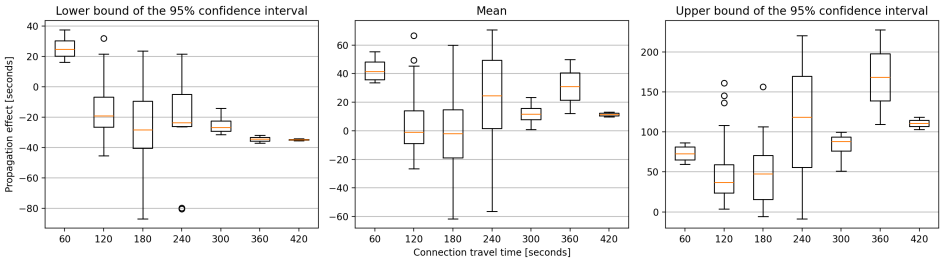


Figure 7.5: Propagation behaviour per amount of allocated travel.

7.3. Network Design Related

The physical design of the network shows minimal patterns. Examining the behaviour per number of connections per station (Figure 7.6), and per the direction of the connection with respect to the centre (Figure 7.7), we find no significant trends. However, when we separate the behaviour per branch of the network (Figure 7.8, see Figure 4.2 for the branch arrangement), we find that branches A, B and E have significantly lower lowest quantiles when compared to the other branches. The J branch exhibits very compact lower-bound behaviour and worse upper-bound behaviour compared to the rest of the network. However, this could be due to the limited number of J-branch-related connections, as it is associated with only two stations when the end-of-line connections are filtered out.

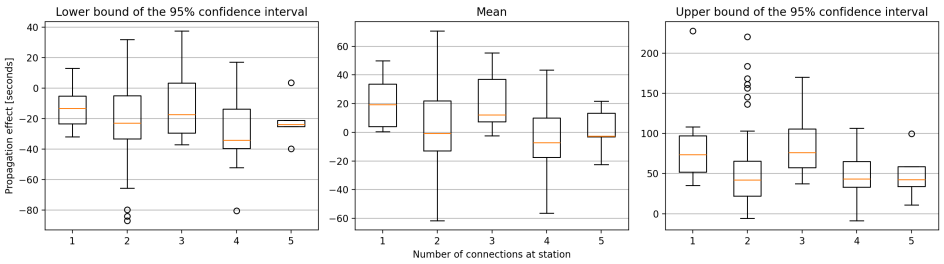


Figure 7.6: Propagation behaviour per number of connections at a station.

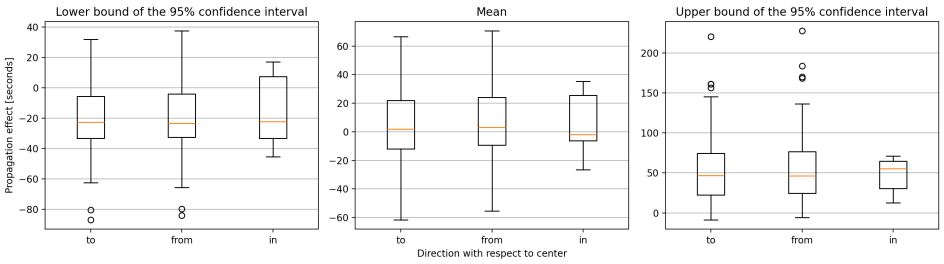


Figure 7.7: Propagation behaviour per direction with respect to the network centre.

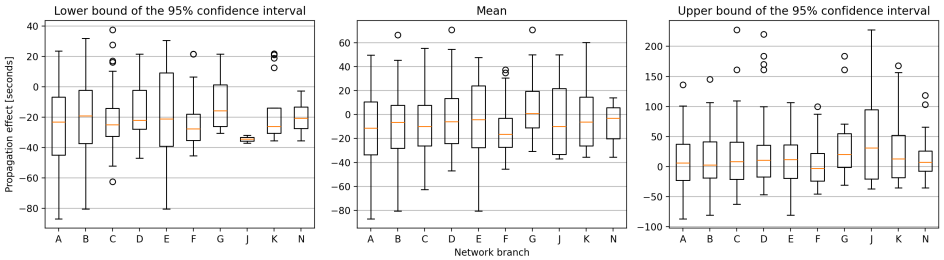


Figure 7.8: Propagation behaviour per network branch.

8

Conclusions

This thesis aimed to characterise which operational and network design variables influence delay propagation in metro networks, and by what magnitude. To bridge the gap in existing literature between analysis-focused and prediction-oriented paradigms, a data-driven framework was developed that balances interpretability with adaptability, allowing for flexible adjustments in model complexity, scope, and application.

The framework was applied to the Washington D.C. Metro network using arrival time data, enabling the discovery, modelling and analysis of delay propagation across the available temporal, spatial and operational variable dimensions. Two model implementations were developed: a generalised model for identifying significant relationships and a focused model that quantifies the found relationships in greater detail.

8.1. Key Findings

The main contributions of this thesis can be grouped into methodological, analytical, and practical findings.

8.1.1. Methodological Findings

A five-step data-driven framework was constructed, allowing for systematic exploration and quantification of delay propagation relationships. The framework's segmented and easily adjustable structure enables researchers to construct and compare propagation models under a consistent methodology, while allowing for more representative comparisons between different implementations. The framework can be configured to prioritise either interpretability or predictive performance, depending on the chosen relationship-fitting method. This modular structure aligns with recent trends in transport performance modelling, which emphasise reproducible and data-centric methodologies. It can also include as many variables in the analysis as long as the data is sufficient for fitting.

The applications in the Washington D.C. metro use case demonstrate this, as an hourly temporal resolution and a delay magnitude segmentation of 60 seconds are already possible with the limited amount of available data. The resulting approach can be readily adapted for other metro networks, offering an easily transferable tool for data-driven analysis across different urban rail systems.

8.1.2. Analytical Findings

The results of the first model revealed that significant propagation occurs almost exclusively along line-sharing stations, with the most potent effects around directly connected station-direction pairs. While wider network effects can still exist, their magnitude is minimal, confirming that delay propagation between elements is largely local rather than system-wide. This behaviour may stem from the high-frequency and sequential nature of the metro network, which inherently allows delays to flow only along specific routes. From these findings, networks with more independent lines may exhibit minimal inter-line propagation, but this has yet to be proven.

The focused model shows that the magnitude of the propagation effect on any delay is independent of the magnitude of the delay itself. In addition, propagation patterns are largely consistent across the hours of the day and days of the week, with only marginal variations for certain connections between peak and off-peak hours. Lastly, the propagation behaviour of any connection is independent of the behaviour of the connections around it, except for the connection between the same stations going in the opposite direction, with which it is almost entirely similar. These results hint that the delay propagation behaviour is not related to the severity of the initial disruption but rather to timetable construction, and it can be controlled using mechanisms such as headway regulation, dwell times, or real-time vehicle speed measures. The stability of propagation behaviours across the temporal dimension also indicates an inherent equilibrium that arises in the delay propagation given a particular operational regime.

When examining which secondary variables could influence propagation behaviour, shorter headways and higher line usage were associated with slightly increased propagation sensitivity, although their statistical significance could not be determined. Network design-related features, such as centrality or branch structures, showed localised impacts but did not significantly influence propagation behaviour overall. Overall, these results reinforce that propagation in metro networks is driven primarily by local operational interactions rather than global network structures.

8.1.3. Practical Findings

The developed methodology provides metro operators with an easily customisable framework for analysing and predicting delay propagation behaviour and delay spread, without requiring extensive preliminary research on system-specific behavioural rules. The results can be used to identify critical connections where delays are most likely to propagate and to quantify the magnitude of these secondary delays precisely. This can aid in timetable design, the development of new delay mitigation strategies, and the highlighting of specific locations for possible infrastructure improvements. Because the framework is data-driven yet interpretable, it can also be incorporated into real-time control and decision-support systems for operational monitoring and short-term delay prediction.

8.2. Limitations

While the proposed framework and its implementation achieve the goals set out during the introduction, several limitations should be noted:

- **Data limitations**: The analysis was constrained to periods of normal operations, leading to the exclusion of much of the available movement data, as this covered COVID-19 restricted operational periods. This significantly limited the amount of data available for analysis, leading to the need to employ relatively simple relationship-fitting methods. A larger amount of data could enable a more prediction-focused model implementation, leveraging the power of machine learning techniques.
- **Model simplifications**: More complex interactions and possible existing delay mitigation strategies were not incorporated into the data preparation or model construction. Incorporating these elements could improve the model results by either integrating them into the propagation functions or using them to filter the delay data further.
- **Undiscovered causality**: While the model can find and quantify the propagation relationships successfully, the results and subsequent cross-examination do not explicitly establish a causal characteristic or mechanisms that could explain the different delay propagation behaviours. Integrating causal inference methods or developing a dedicated framework implementation to focus on finding behavioural causes could provide strong insights.
- **Use case scope**: The framework was validated, and relationship characteristics were discovered using a single case study network. While the Washington D.C. metro system is an adequate general representation of a metro system, applying the model to other networks to confirm the found relationships would strengthen this thesis's findings.

8.3. Recommendations for Further Research

Based on the findings and limitations identified in this thesis, several avenues for future research are proposed. The inclusion of more variables, control strategies, or operational rules could yield better results and deepen the insights gained. Implementing the framework with a focus on causal discovery could reveal the origins of the obtained delay propagation behaviours. Similarly, applying the same two models to other metro systems worldwide could confirm that the propagation characteristics identified in this thesis are a general phenomenon of metro networks. Lastly, the framework or its specific implementations could be further extended into a tool for integration into real-time monitoring and decision-support systems. Implementing more complex online and continuous learning methods could allow the model to adapt dynamically to evolving network or operational conditions. Together, these extensions would enhance both the explanatory and predictive capacities of the developed framework, potentially leading to the discovery and establishment of more intrinsic characteristics of delay propagation in the metro networks as a whole.

8.4. Concluding Remarks

This thesis contributes to the characterisation and understanding of delay propagation in metro systems by employing a data-driven method that can yield both interpretable and operationally relevant insights into propagation behaviour. The developed framework can be adjusted to provide transparency and predictive capabilities, offering researchers and network operators a robust and adaptable tool suitable for most use cases.

The results of the model implementation suggest that the metro network exhibits some intrinsic locality in the delay propagation dynamics. Rather than widespread influence, delays at one station tend to affect only closely connected stations, governed primarily by local operational characteristics. The delay magnitude-independent nature and the behaviour's relative temporal invariability suggest a controllable equilibrium state for the propagation effect, which could be leveraged to improve network reliability.

By offering a structured, interpretable framework for analysing propagation dynamics, this thesis provides a foundation for future work in data-driven reliability analysis. It offers network operators an adjustable tool for delay propagation prediction and contributes to the broader goal of building more resilient and efficient urban transport systems.

References

Ayana, H., Mulangi, R. H., & Harsha, M. M. (2023). Analysis of bus stop delay variability using public transit gps data. *Lecture Notes in Civil Engineering*, *261*, 301–315. https://doi.org/10.1007/978-981-19-2273-2_21

Büchel, B., Spanninger, T., & Corman, F. (2020). Empirical dynamics of railway delay propagation identified during the large-scale rastatt disruption. *Scientific Reports*, *10*, 18584. https://doi.org/10.1038/s41598-020-75538-z

Büker, T., & Seybold, B. (2012). Stochastic modelling of delay propagation in large networks. *Journal of Rail Transport Planning and Management*, *2*, 34–50. https://doi.org/10.1016/j.jrtpm.2012.10.001

Cats, O., & Hijner, A. M. (2021). Quantifying the cascading effects of passenger delays. *Reliability Engineering & System Safety*, *212*, 107629. https://doi.org/https://doi.org/10.1016/j.ress.2021.107629

Corman, F., & Kecman, P. (2018). Stochastic prediction of train delays in real-time using bayesian networks. *Transportation Research Part C: Emerging Technologies*, *95*. https://doi.org/10.3929/ethz-b-000281218

Dekker, M. M., Medvedev, A. N., Rombouts, J., Siudem, G., & Tupikina, L. (2022). Modelling railway delay propagation as diffusion-like spreading. *EPJ Data Science*, *11*, 44. https://doi.org/10.1140/epjds/s13688-022-00359-1

Harrod, S., Cerreto, F., & Nielsen, O. A. (2019). A closed form railway line delay propagation model. *Transportation Research Part C: Emerging Technologies*, *102*, 189–209. https://doi.org/10.1016/j.trc.2019.02.022

Heathcote, A., Brown, S., & Mewhort, D. J. K. (2002). Quantile maximum likelihood estimation of response time distributions. *Psychonomic Bulletin & Review*, *9*, 394–401. https://doi.org/10.3758/BF03196299

Higgins, A., & Kozan, E. (1998). Modeling train delays in urban networks. *Transportation Science*, *32*, 346–357. https://doi.org/10.1287/trsc.32.4.346

Huang, P., Guo, J., Liu, S., & Corman, F. (2024). Explainable train delay propagation: A graph attention network approach. *Transportation Research Part E: Logistics and Transportation Review*, *184*, 103457. https://doi.org/https://doi.org/10.1016/j.tre.2024.103457

Krishnakumari, P., Cats, O., & van Lint, H. (2020). Estimation of metro network passenger delay from individual trajectories. *Transportation Research Part C: Emerging Technologies*, *117*, 102704. https://doi.org/https://doi.org/10.1016/j.trc.2020.102704

Li, Z., Huang, P., Wen, C., & Rodrigues, F. (2023). Railway network delay evolution: A heterogeneous graph neural network approach. https://arxiv.org/abs/2303.15489

MobilityData. (2025). What is gtfs? https://gtfs.org/getting-started/what-is-GTFS/

Shakibayifar, M., Sheikholeslami, A., & Corman, F. (2018). A simulation-based optimization approach to rescheduling train traffic in uncertain conditions during disruptions. *Scientia Iranica*, *25*, 646–662. https://doi.org/10.24200/sci.2017.4186

Spanninger, T., Trivella, A., Büchel, B., & Corman, F. (2022, June). A review of train delay prediction approaches. https://doi.org/10.1016/j.jrtpm.2022.100312

Srivastava, N., Gohil, B. N., & Ray, S. (2024). Rail transit delay forecasting with causal machine learning. *Proceedings of the 1st ACM SIGSPATIAL International Workshop on Spatiotemporal Causal Analysis*, 1–10. https://doi.org/10.1145/3681778.3698784

Steel, R. G. D., & Torrie, J. H. (1960). *Principles and procedures of statistics.* McGraw-Hill Book Company, Inc., New York, Toronto, London.

Szymanski, P., Zolnieruk, M., Oleszczyk, P., Gisterek, I., & Kajdanowicz, T. (2018). Spatio-temporal profiling of public transport delays based on large-scale vehicle positioning data from gps in wrocław. *IEEE Transactions on Intelligent Transportation Systems*, *19*, 3652–3661. https://doi.org/10.1109/TITS.2018.2852845

The SciPy community. (2025). Scipy statistical functions. https://docs.scipy.org/doc/scipy/reference/stats.html

Ulak, M. B., Yazici, A., & Zhang, Y. (2020). Analyzing network-wide patterns of rail transit delays using bayesian network learning. *Transportation Research Part C: Emerging Technologies*, *119*, 102749. https://doi.org/https://doi.org/10.1016/j.trc.2020.102749

Wang, X., Yao, E., & Liu, S. (2019). Simulation of metro congestion propagation based on route choice behaviors under emergency-caused delays. *Applied Sciences (Switzerland)*, *9*. https://doi.org/10.3390/app9204210

WMATA. (2024). System map. https://www.wmata.com/schedules/maps/wmata-system-map.cfm

Yap, M., & Cats, O. (2021). Predicting disruptions and their passenger delay impacts for public transport stops. *Transportation*, *48*, 1703–1731. https://doi.org/10.1007/s11116-020-10109-9

A

WMATA Rail GTFS File Structure

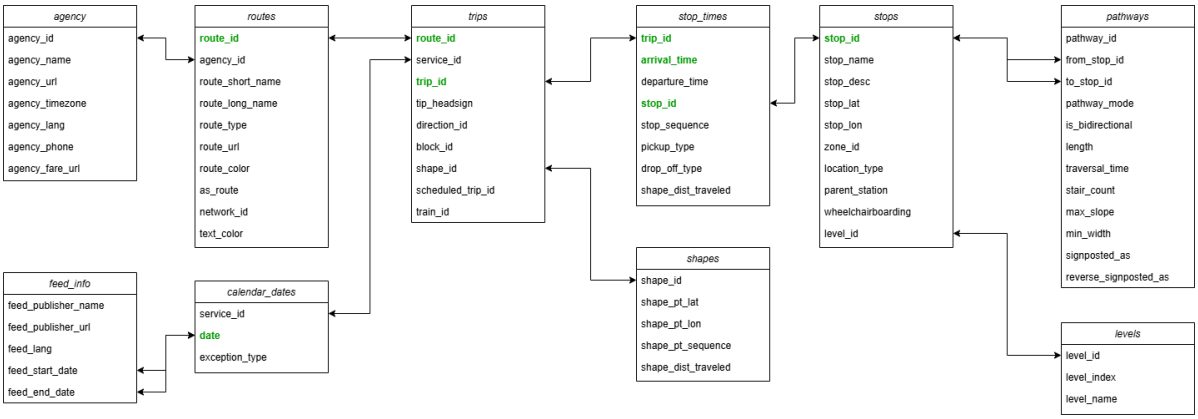
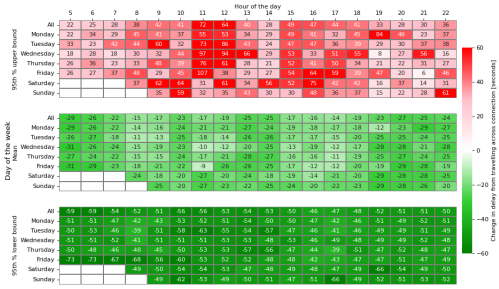


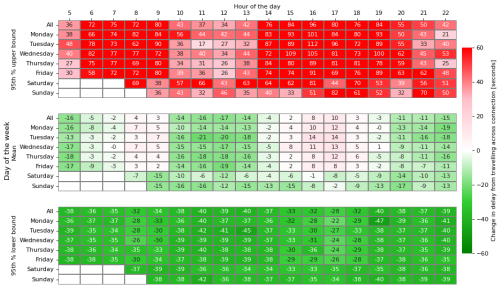
Figure A.1: WMATA GTFS Schedule files and relations, with the variables used for constructing the scheduled arrival time data highlighted in green.

B

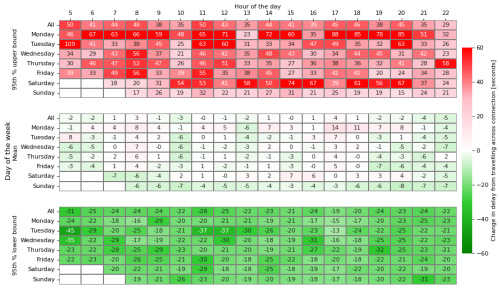
Model II All Results



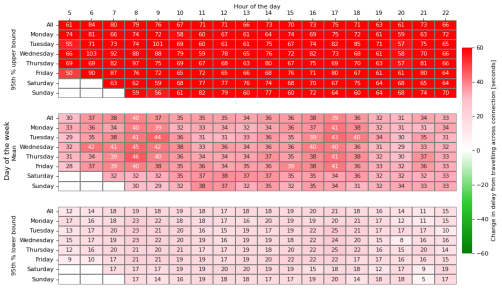
(a) Connection A01/C01 to A02.



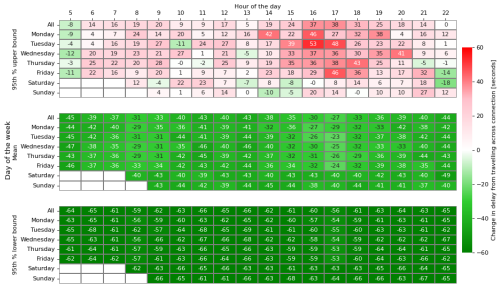
(b) Connection A01/C01 to B01/F01.



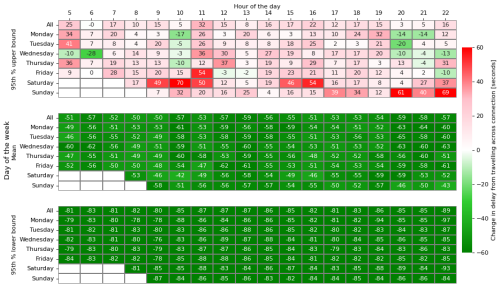
(c) Connection A01/C01 to C02.



(d) Connection A01/C01 to D01.

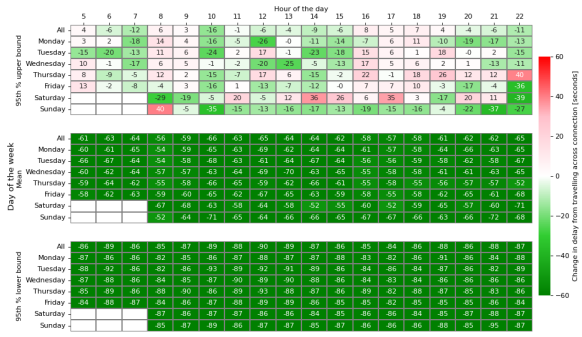


(e) Connection A02 to A01/C01.

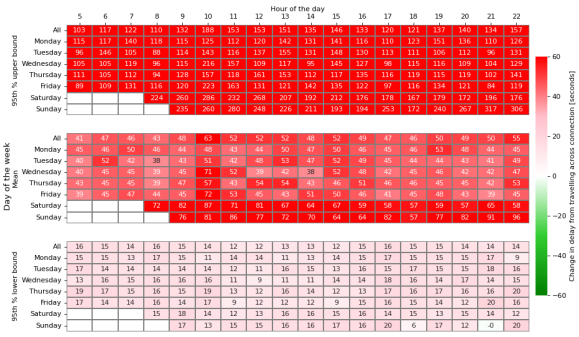


(f) Connection A02 to A03.

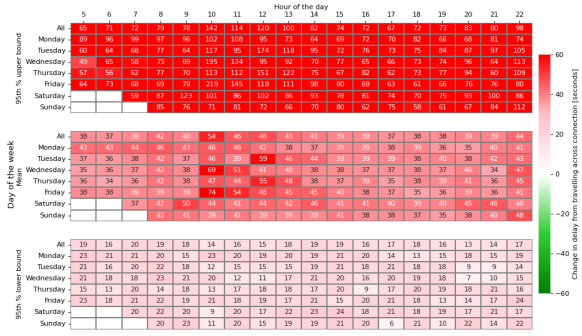
Figure B.1: Second model results from A01/C01 to A02 to A02 to A03.



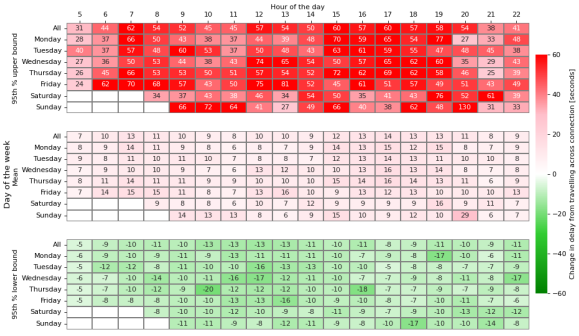
(a) Connection A03 to A02.



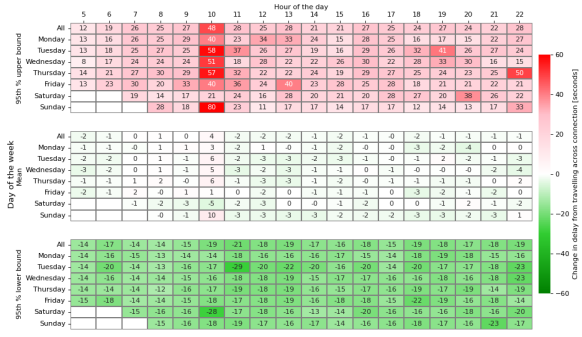
(b) Connection A03 to A04.



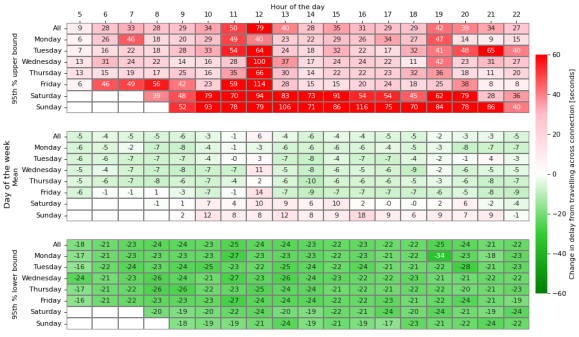
(c) Connection A04 to A03.



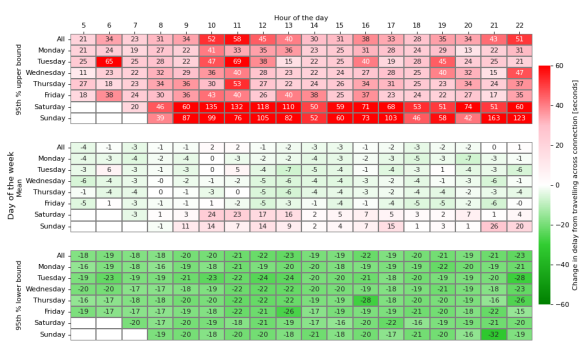
(d) Connection A04 to A05.



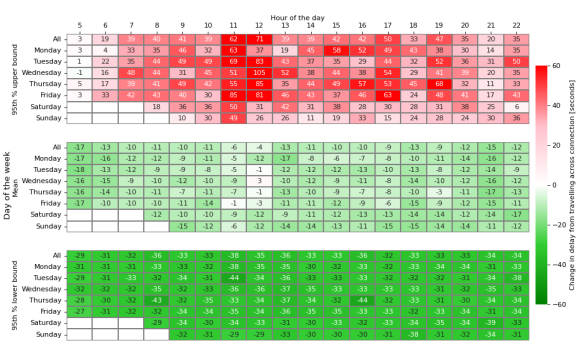
(e) Connection A05 to A04.



(f) Connection A05 to A06.

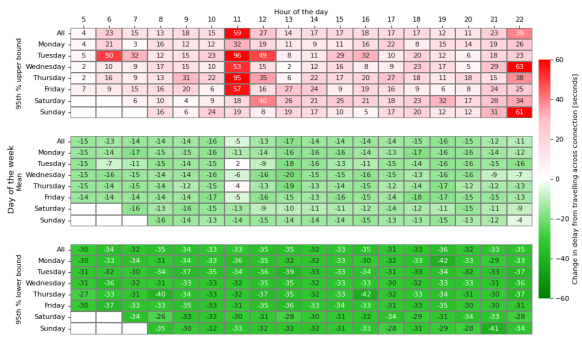


(g) Connection A06 to A05.

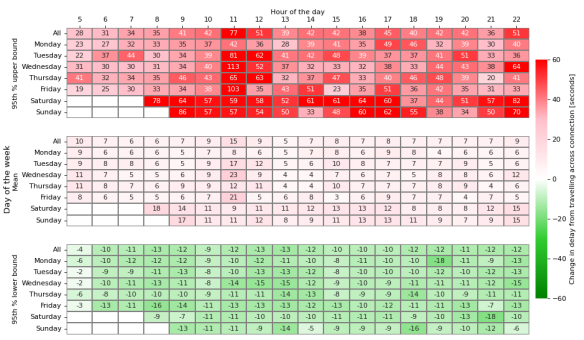


(h) Connection A06 to A07.

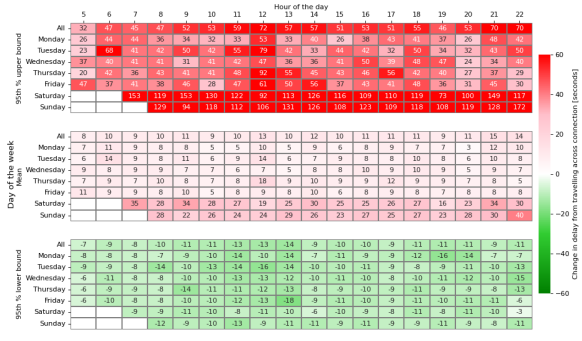
Figure B.2: Second model results from A03 to A02 to A06 to A07.



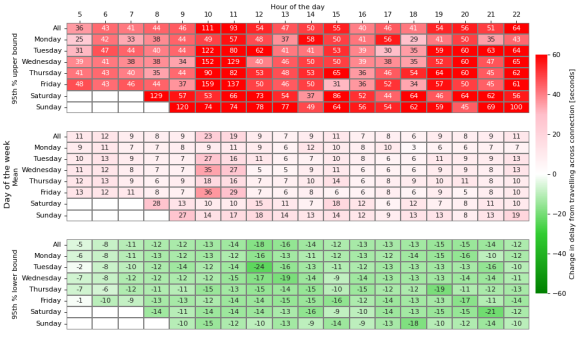
(a) Connection A07 to A06.



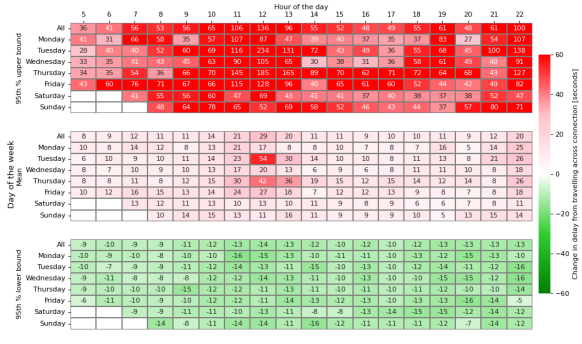
(b) Connection A07 to A08.



(c) Connection A08 to A07.

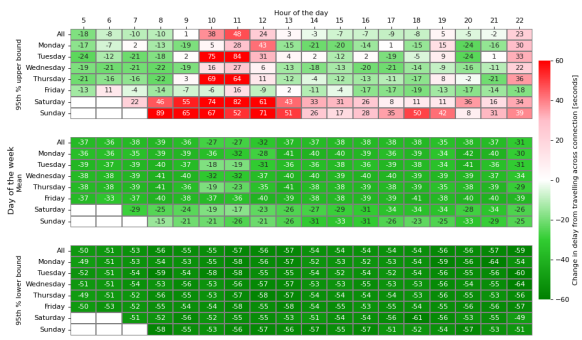


(d) Connection A08 to A09.

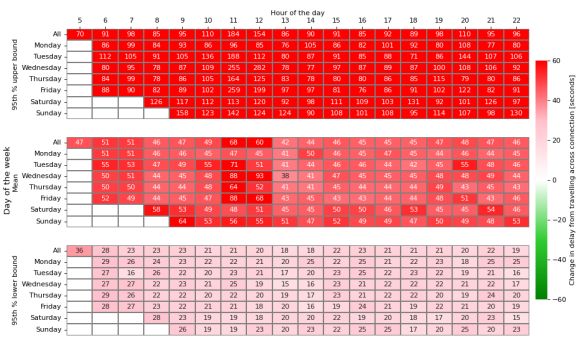


(e) Connection A09 to A08.

(f) Connection A09 to A10.

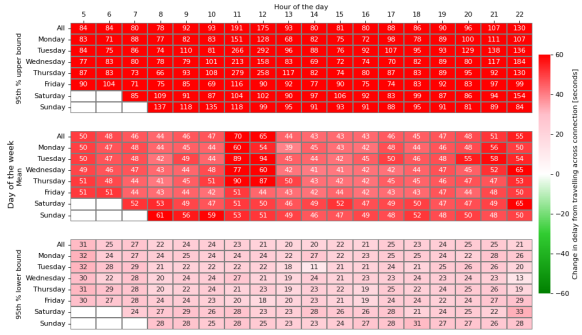


(g) Connection A10 to A09.

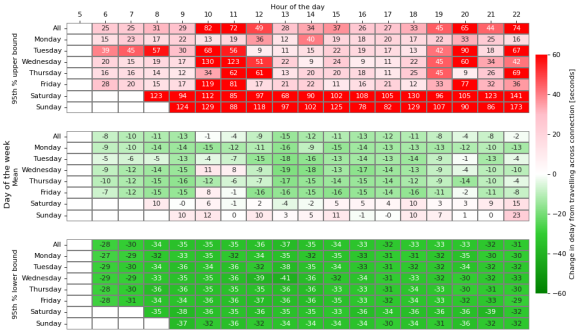


(h) Connection A10 to A11.

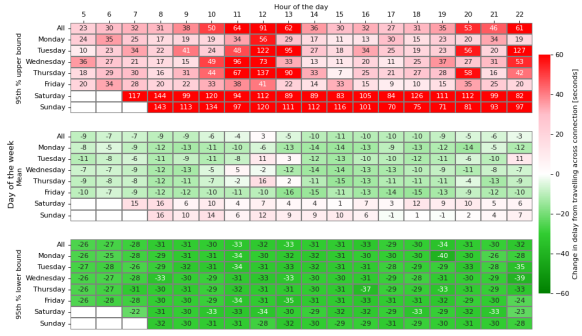
Figure B.3: Second model results from A07 to A06 to A10 to A11.



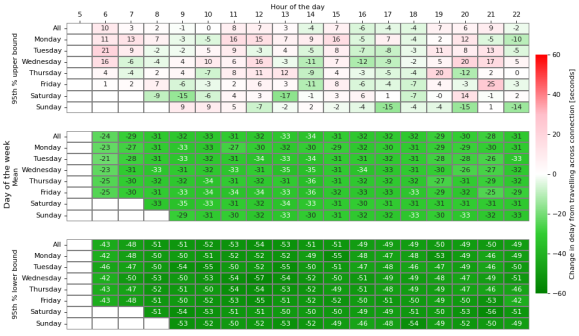
(a) Connection A11 to A10.



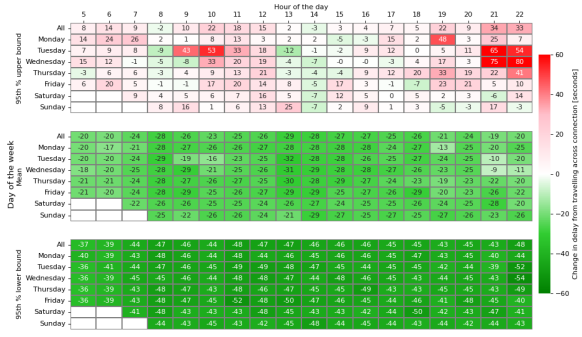
(b) Connection A11 to A12.



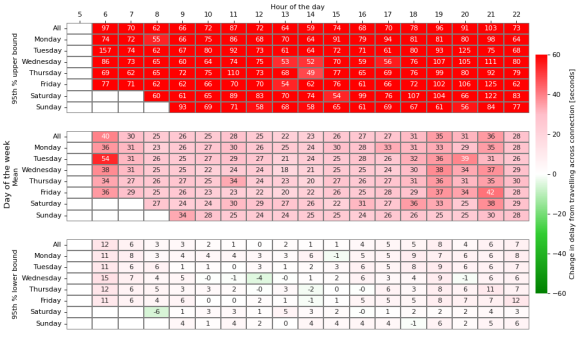
(c) Connection A12 to A11.



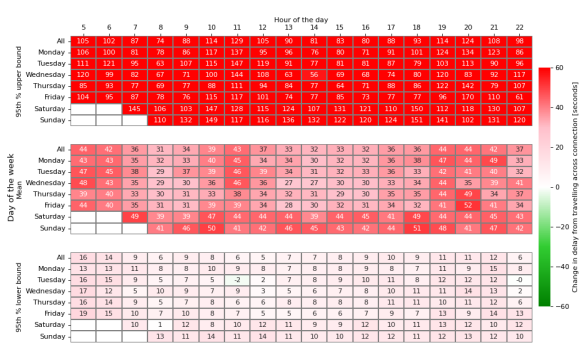
(d) Connection A12 to A13.



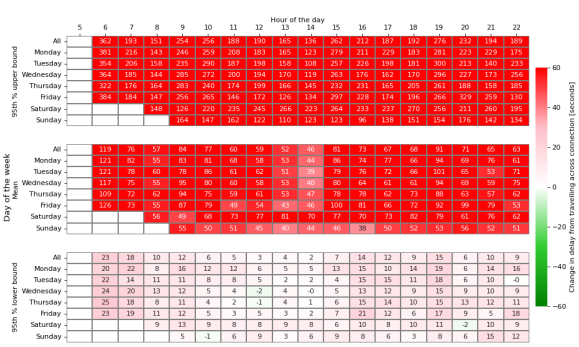
(e) Connection A13 to A12.



(f) Connection A13 to A14.

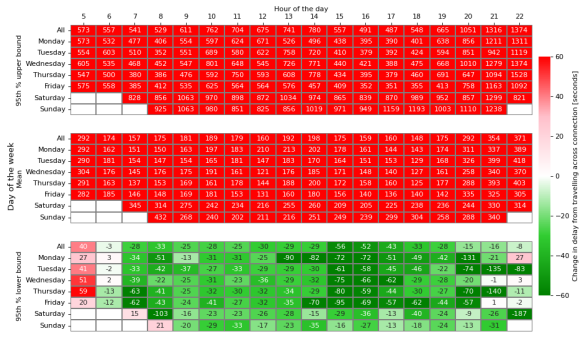


(g) Connection A14 to A13.

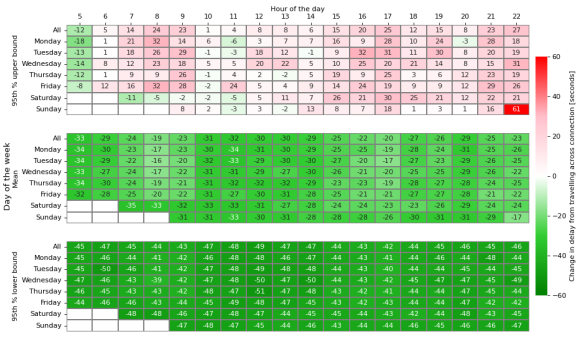


(h) Connection A14 to A15.

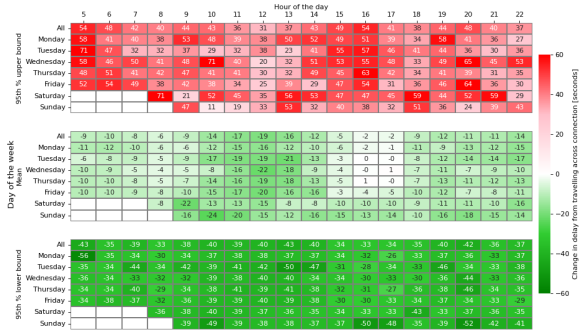
Figure B.4: Second model results from A11 to A10 to A14 to A15.



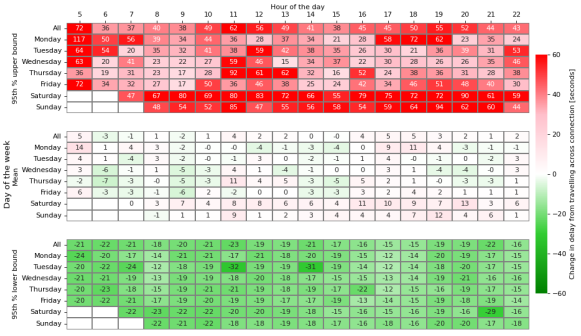
(a) Connection A15 to A14.



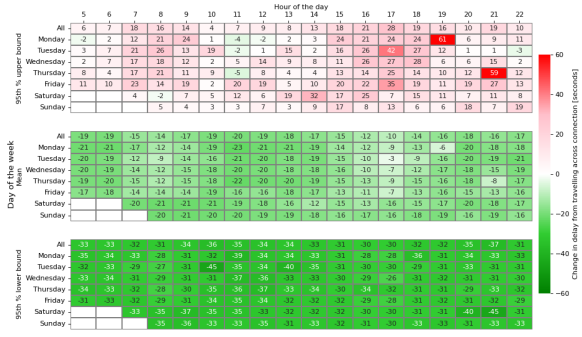
(b) Connection B01/F01 to A01/C01.



(c) Connection B01/F01 to B02.

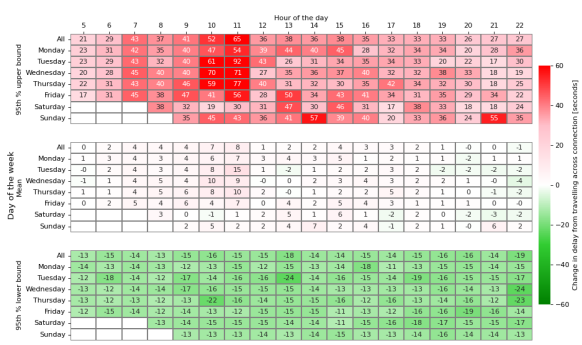


(d) Connection B01/F01 to E01.

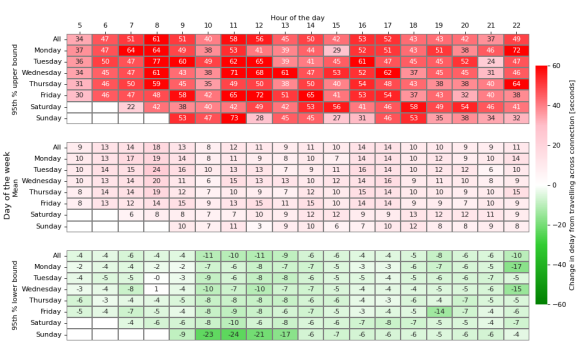


(e) Connection B01/F01 to F02.

(f) Connection B02 to B01/F01.

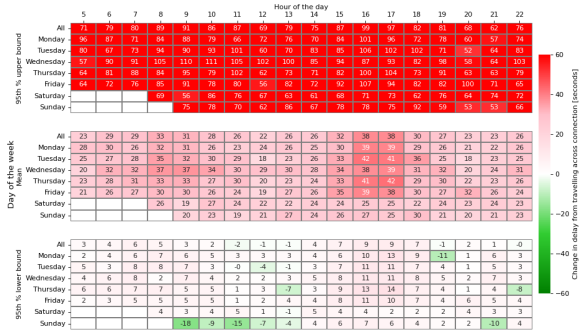


(g) Connection B02 to B03.



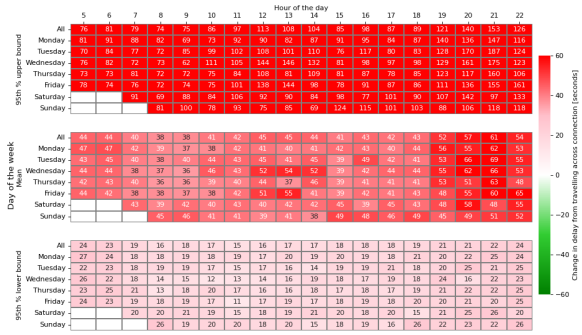
(h) Connection B03 to B02.

Figure B.5: Second model results from A15 to A14 to B03 to B02.

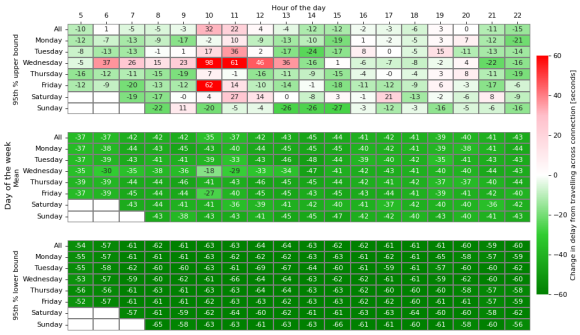


(a) Connection B03 to B35.

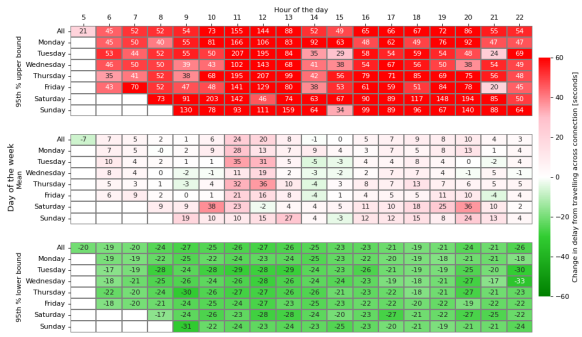
(b) Connection B04 to B05.



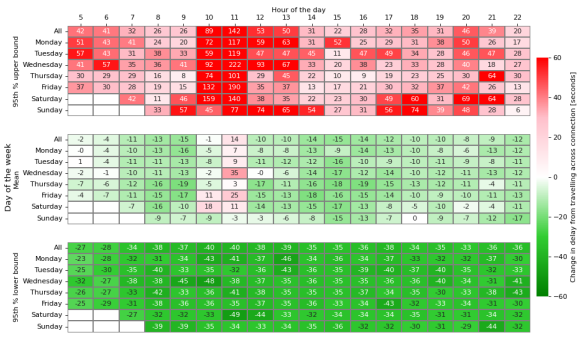
(c) Connection B04 to B35.



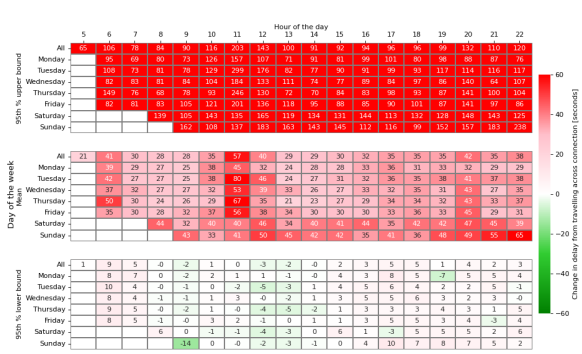
(d) Connection B05 to B04.



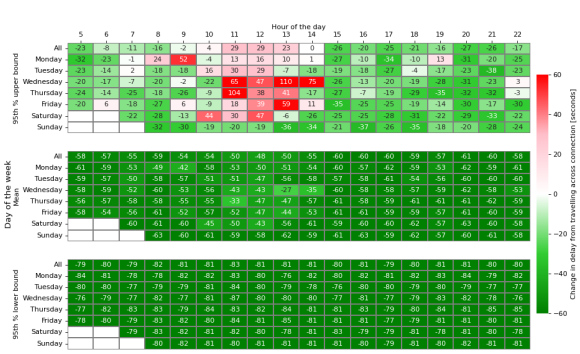
(e) Connection B05 to B06/E06.



(f) Connection B06/E06 to B05.

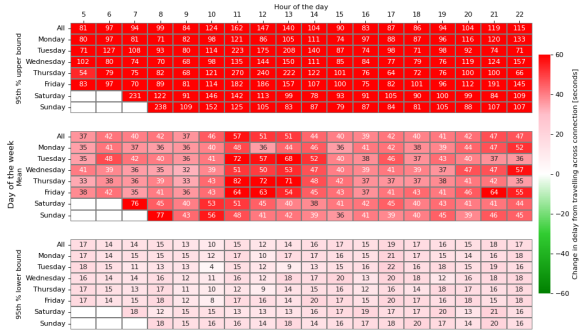


(g) Connection B06/E06 to B07.

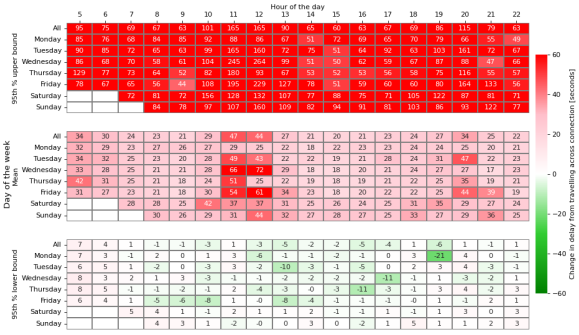


(h) Connection B06/E06 to E05.

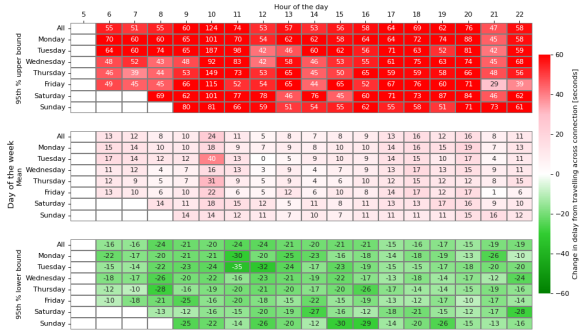
Figure B.6: Second model results from B03 to B35 to B06/E06 to E05.



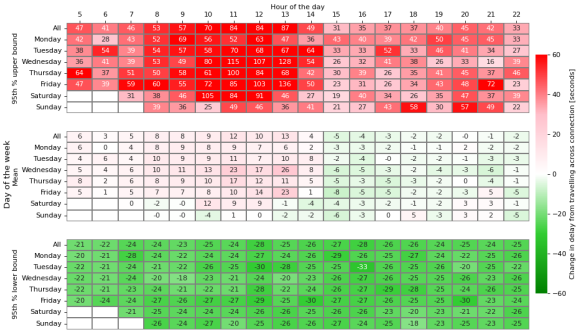
(a) Connection B06/E06 to E07.



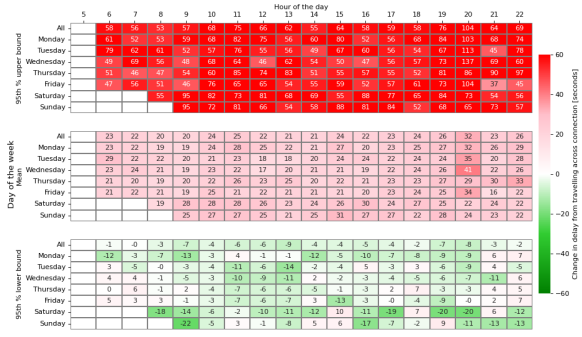
(b) Connection B07 to B06/E06.



(c) Connection B07 to B08.

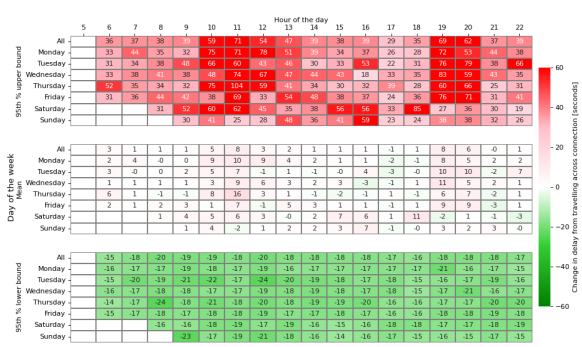


(d) Connection B08 to B07.

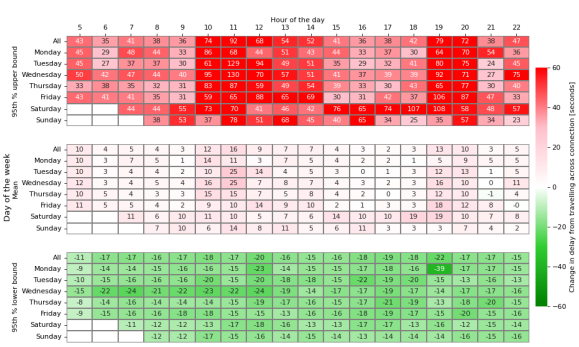


(e) Connection B08 to B09.

(f) Connection B09 to B08.



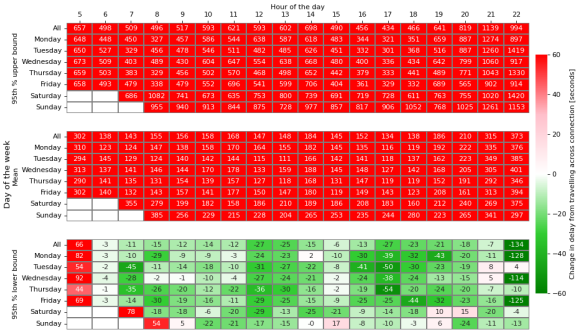
(g) Connection B09 to B10.



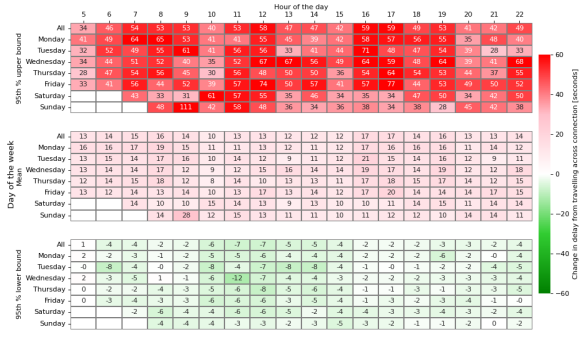
(h) Connection B10 to B09.

Figure B.7: Second model results from B06/E06 to E07 to B10 to B09.

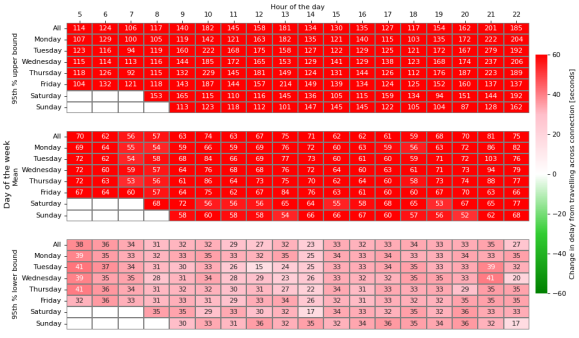
(a) Connection B10 to B11.



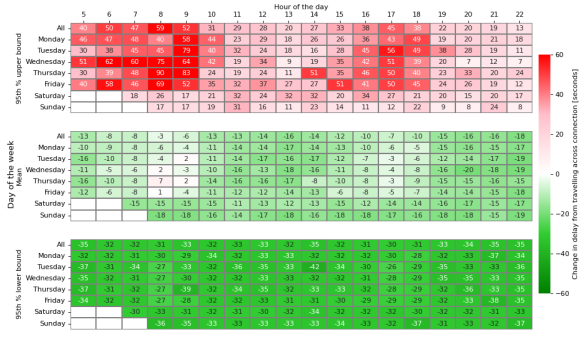
(b) Connection B11 to B10.



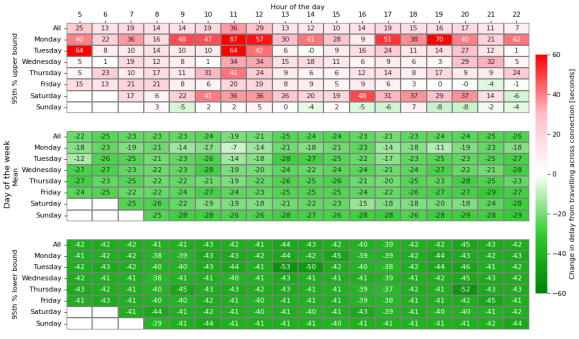
(c) Connection B35 to B03.



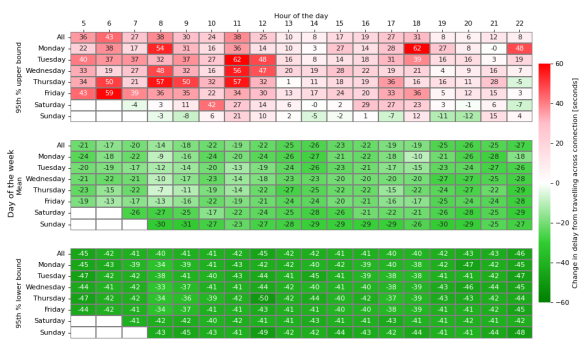
(d) Connection B35 to B04.



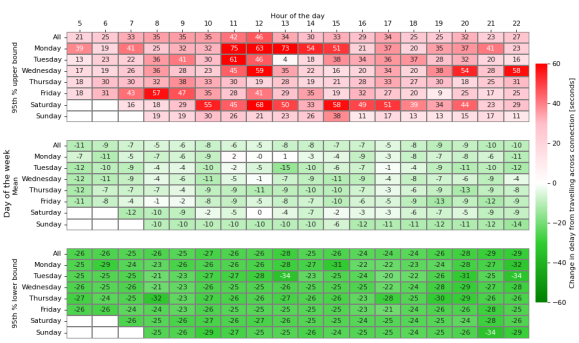
(e) Connection C02 to A01/C01.



(f) Connection C02 to C03.

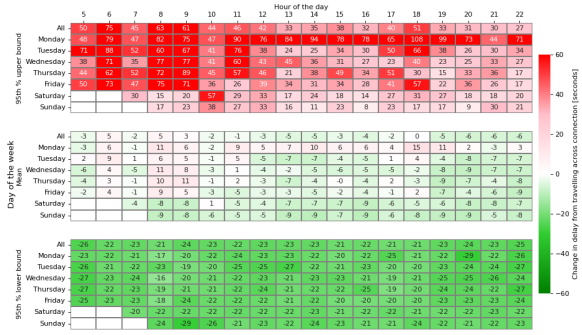


(g) Connection C03 to C02.

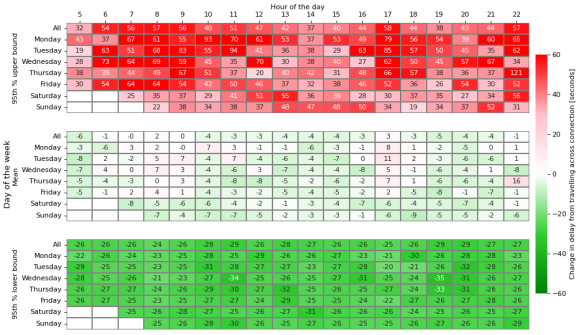


(h) Connection C03 to C04.

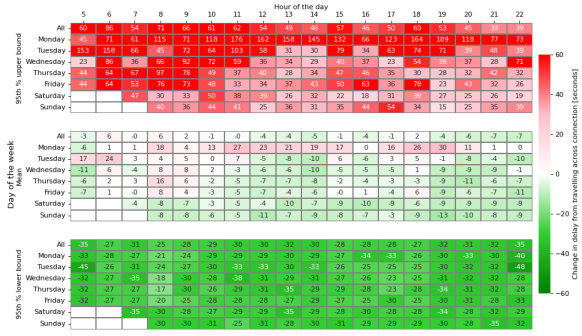
Figure B.8: Second model results from B10 to B11 to C03 to C04.



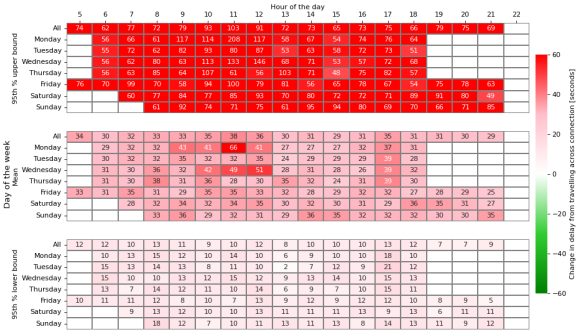
(a) Connection C04 to C03.



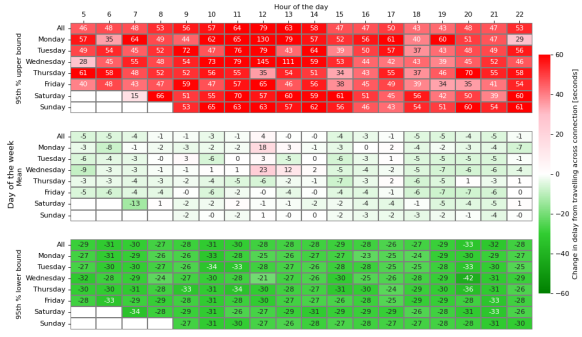
(b) Connection C04 to C05.



(c) Connection C05 to C04.

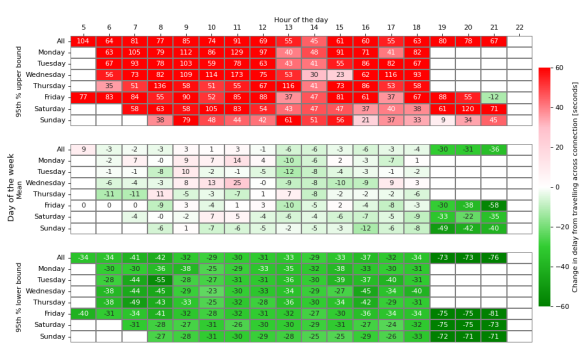


(d) Connection C05 to C06.

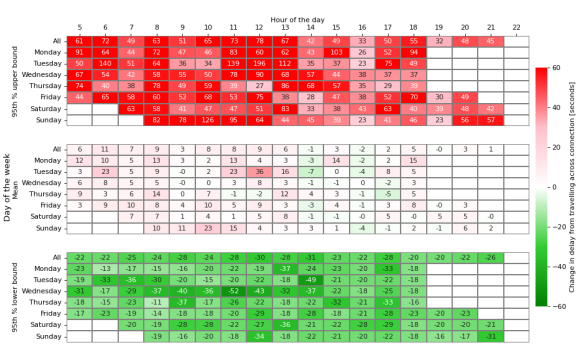


(e) Connection C05 to K01.

(f) Connection C06 to C05.

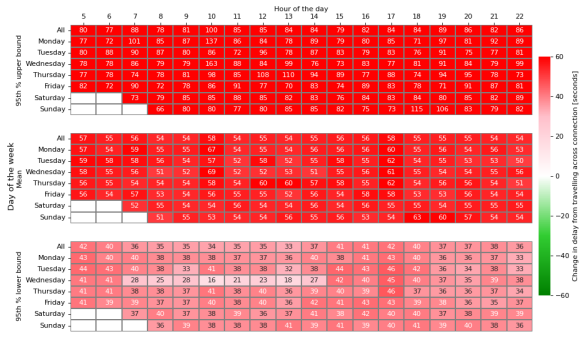


(g) Connection C06 to C07.

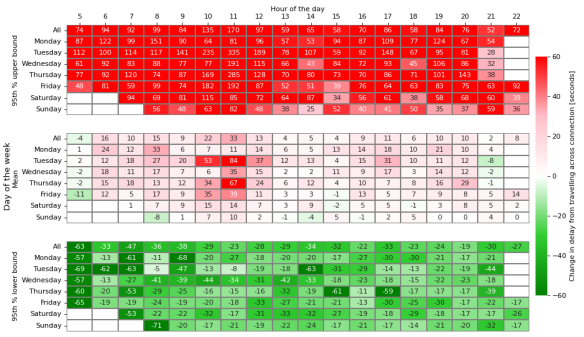


(h) Connection C07 to C06.

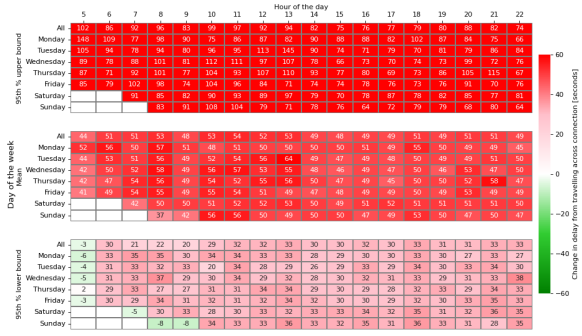
Figure B.9: Second model results from C04 to C03 to C07 to C06.



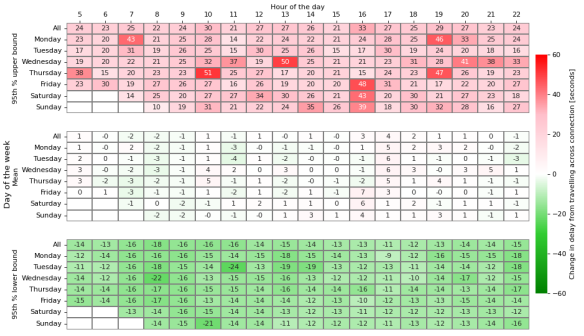
(a) Connection C07 to C08.



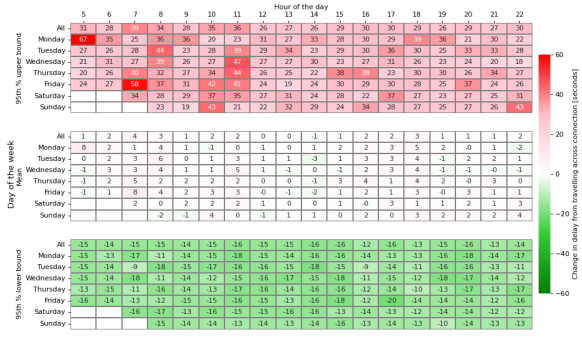
(b) Connection C07 to D03/F03.



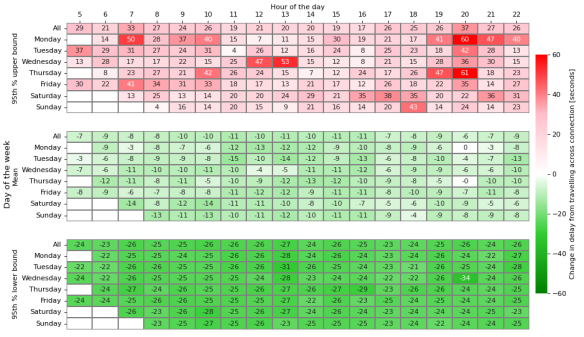
(c) Connection C08 to C07.



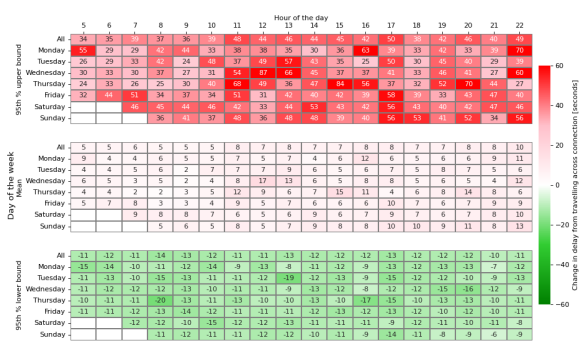
(d) Connection C08 to C09.



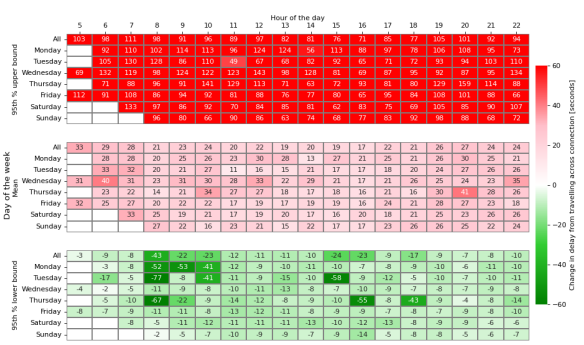
(e) Connection C09 to C08.



(f) Connection C09 to C10.

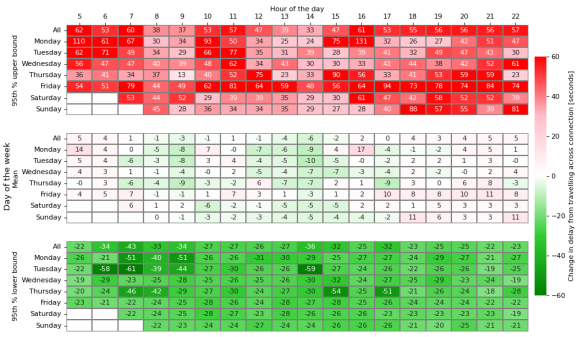


(g) Connection C10 to C09.

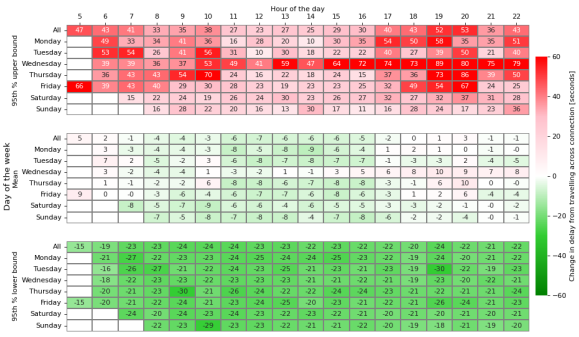


(h) Connection C10 to C12.

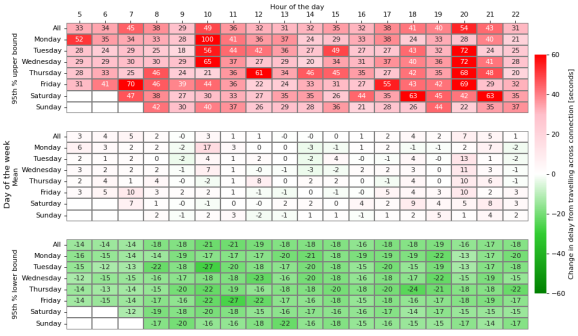
Figure B.10: Second model results from C07 to C08 to C10 to C12.



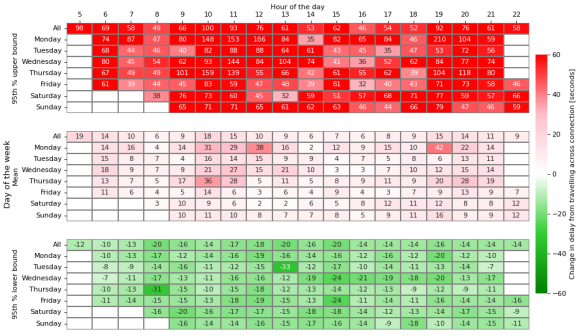
(a) Connection C12 to C10.



(b) Connection C12 to C13.

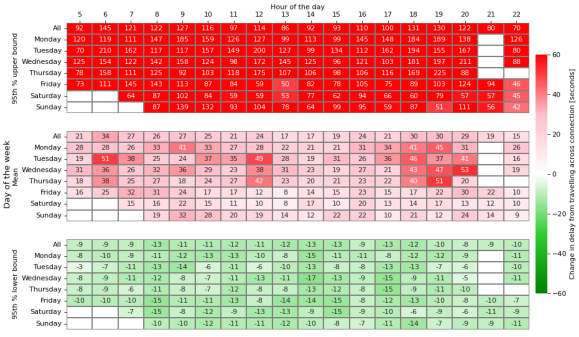


(c) Connection C13 to C12.



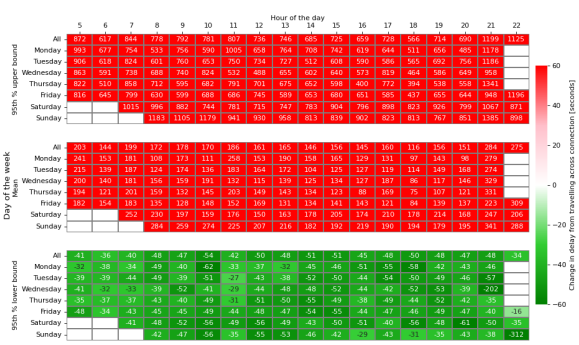
(d) Connection C13 to C14.

(e) Connection C13 to J02.



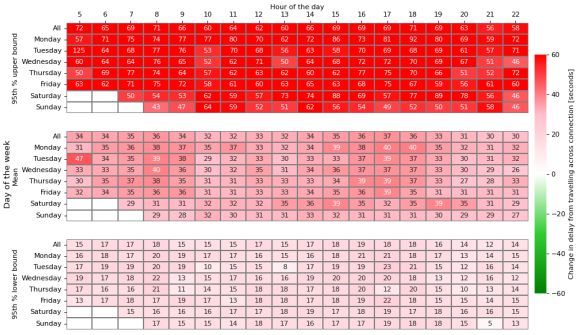
(f) Connection C14 to C13.

(g) Connection C14 to C15.

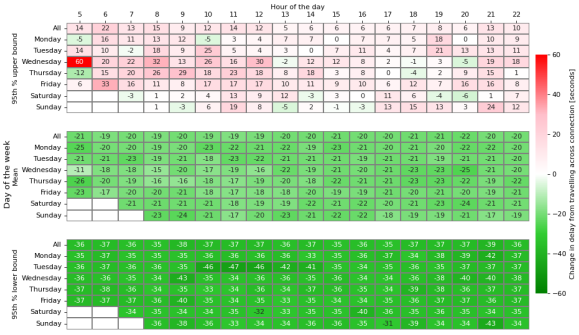


(h) Connection C15 to C14.

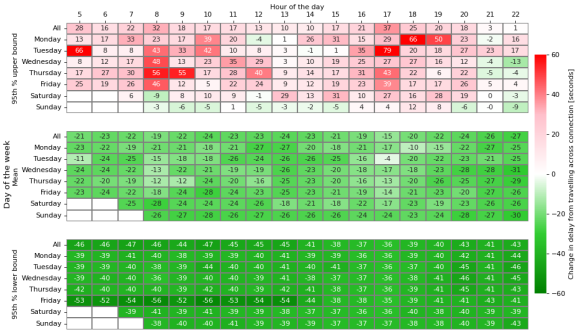
Figure B.11: Second model results from C12 to C10 to C15 to C14.



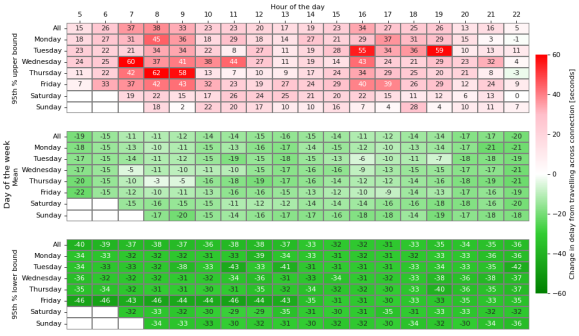
(a) Connection D01 to A01/C01.



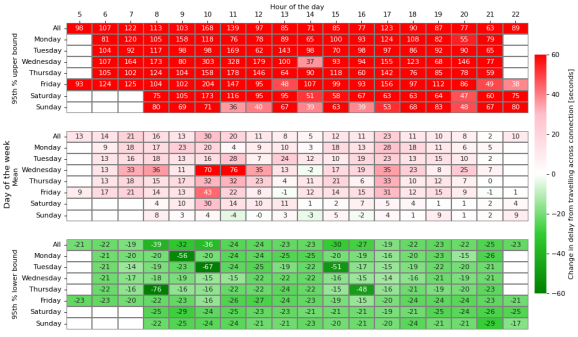
(b) Connection D01 to D02.



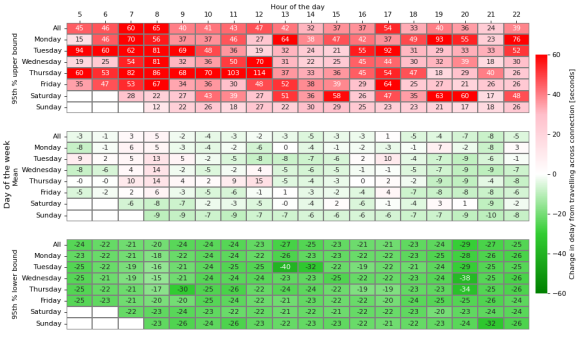
(c) Connection D02 to D01.



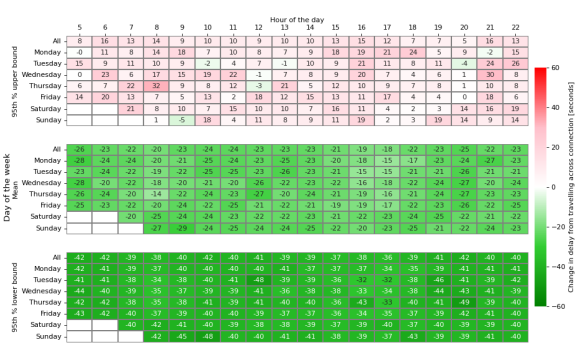
(d) Connection D02 to D03/F03.



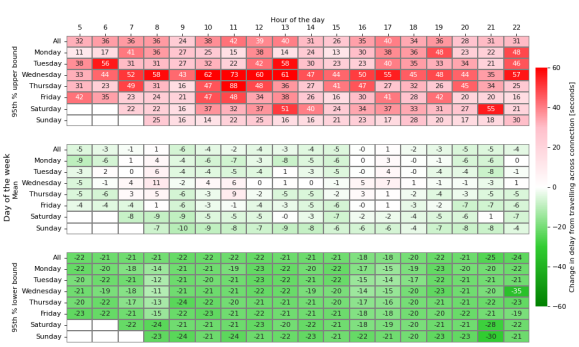
(e) Connection D03/F03 to C07.



(f) Connection D03/F03 to D02.

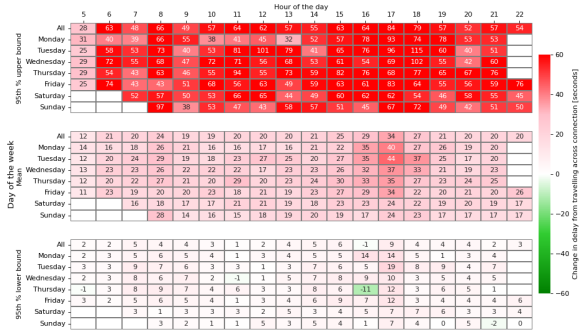


(g) Connection D03/F03 to D04.

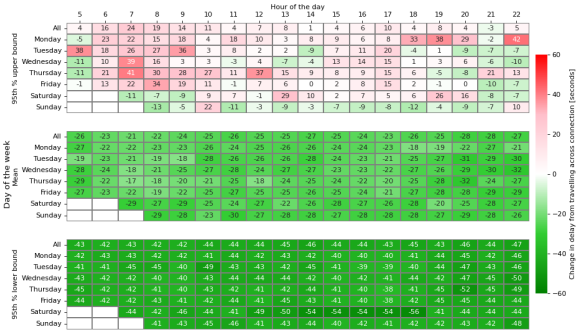


(h) Connection D03/F03 to F02.

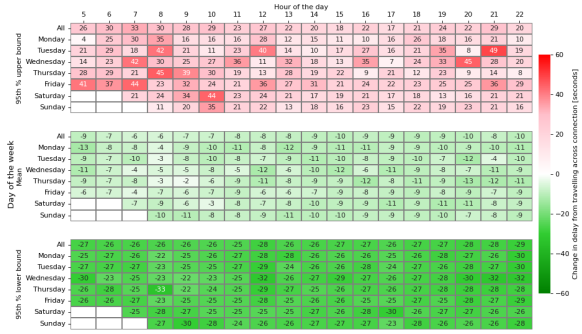
Figure B.12: Second model results from D01 to A01/C01 to D03/F03 to F02.



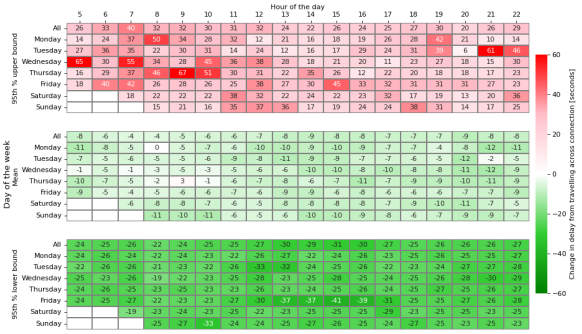
(a) Connection D03/F03 to F04.



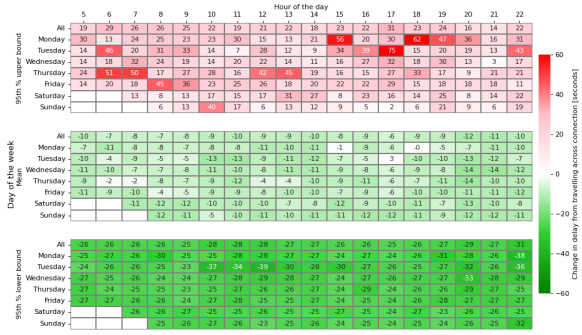
(b) Connection D04 to D03.



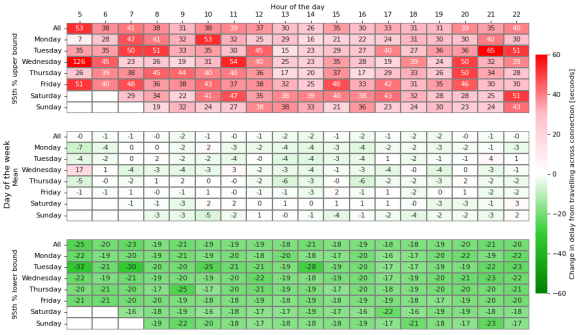
(c) Connection D04 to D05.



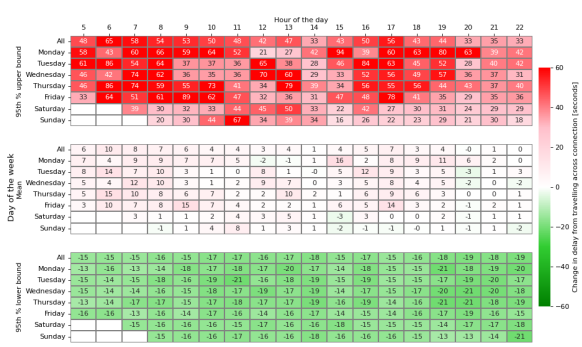
(d) Connection D05 to D06.



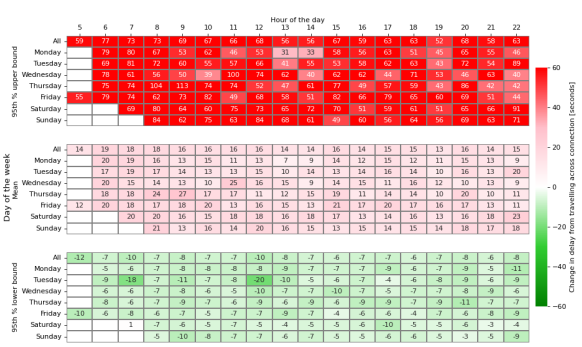
(e) Connection D06 to D05.



(f) Connection D06 to D07.

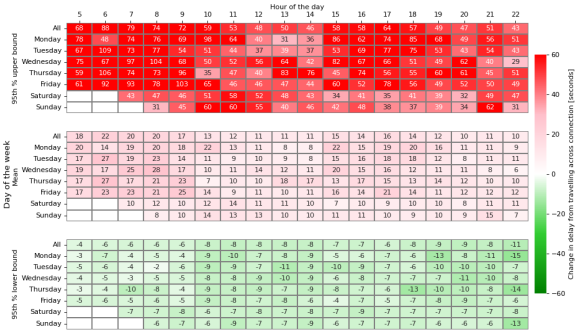


(g) Connection D07 to D06.

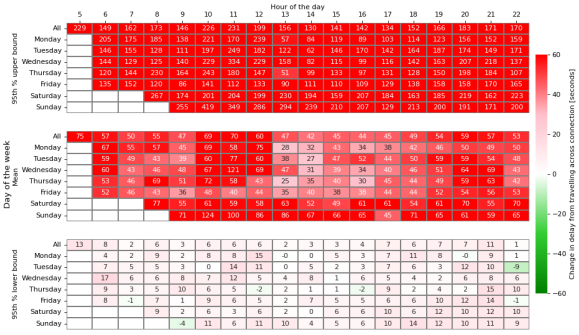


(h) Connection D07 to D08.

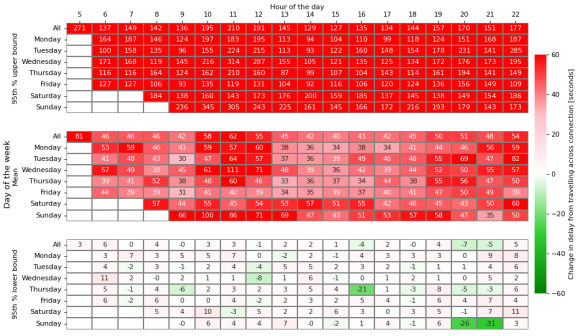
Figure B.13: Second model results from 03/F03 to F04 to D07 to D08.



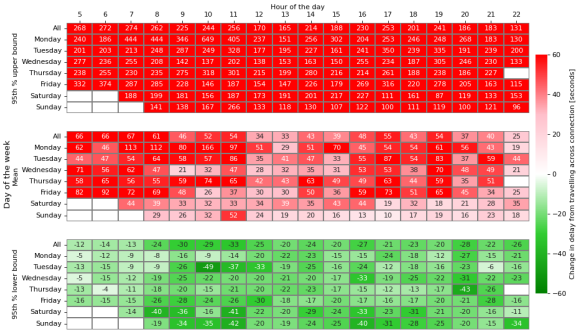
(a) Connection D08 to D07.



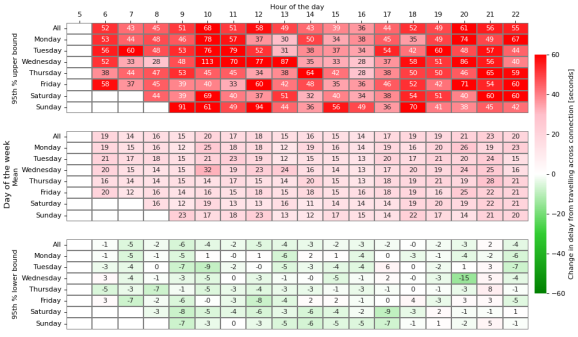
(b) Connection D08 to D09.



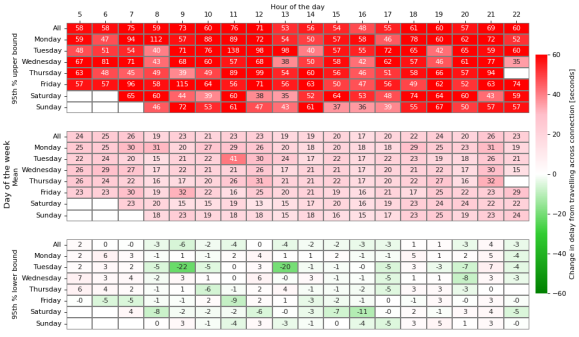
(c) Connection D08 to G01.



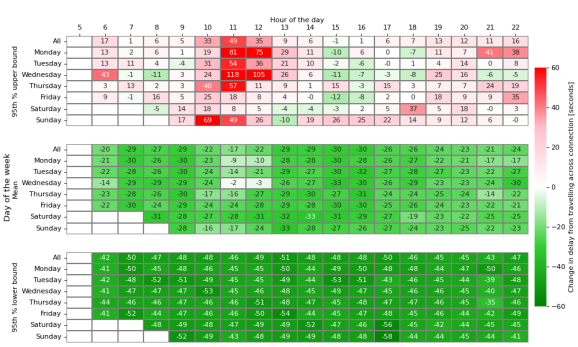
(d) Connection D09 to D08.



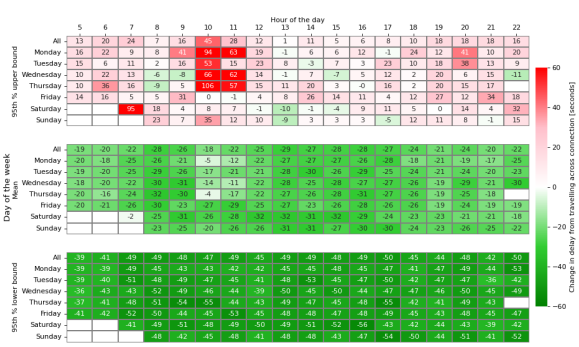
(e) Connection D09 to D10.



(f) Connection D10 to D09.

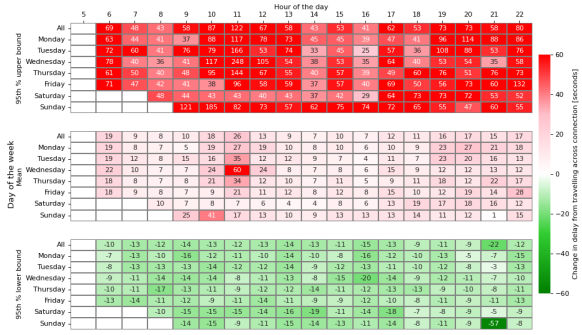


(g) Connection D10 to D11.

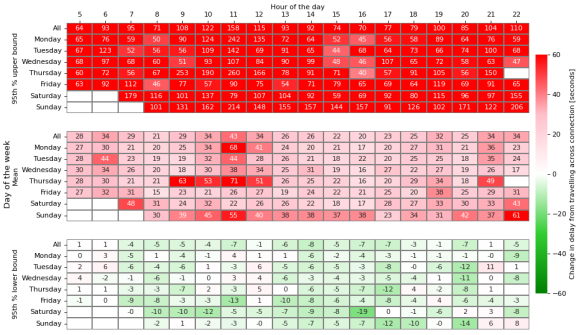


(h) Connection D11 to D10.

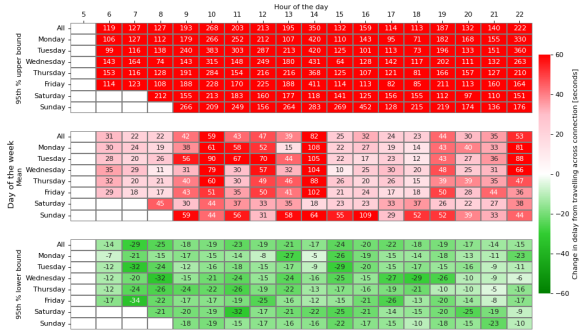
Figure B.14: Second model results from D08 to D07 to D11 to D10.



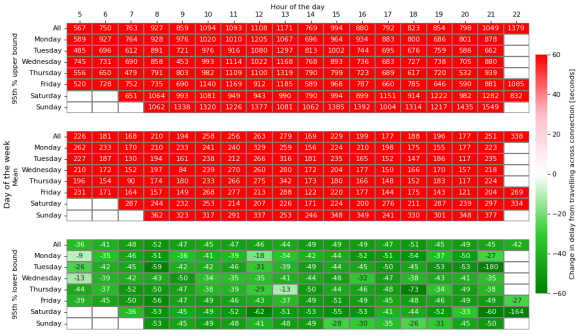
(a) Connection D11 to D12.



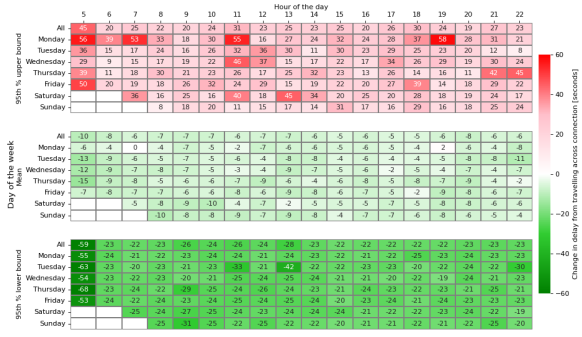
(b) Connection D12 to D11.



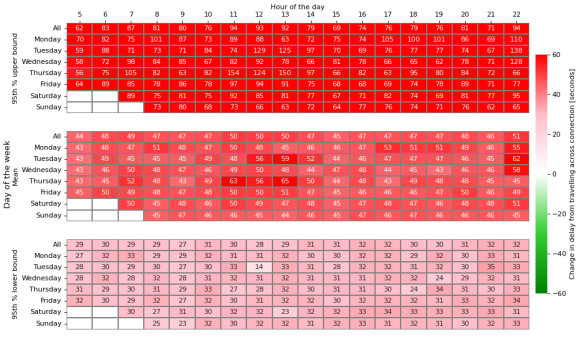
(c) Connection D12 to D13.



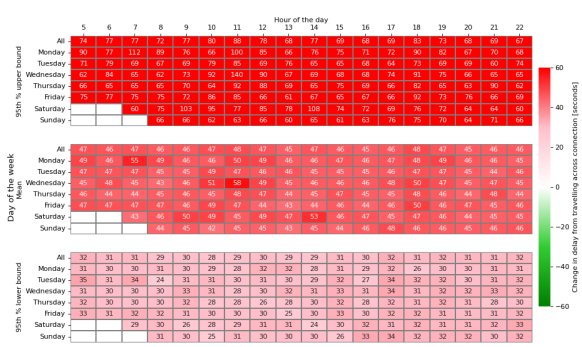
(d) Connection D13 to D12.



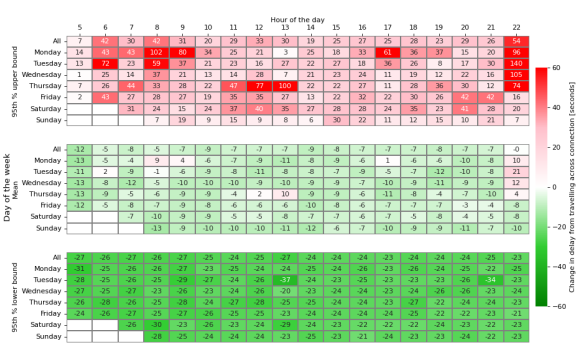
(e) Connection E01 to B01/F01.



(f) Connection E01 to E02.

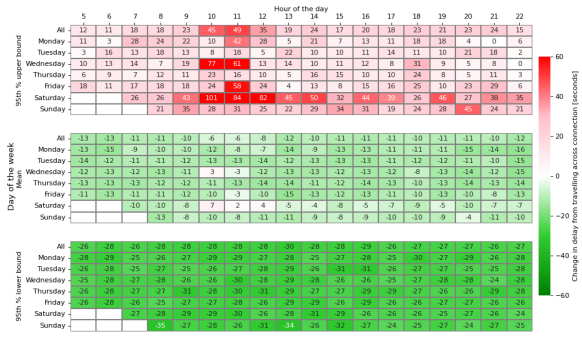


(g) Connection E02 to E01.

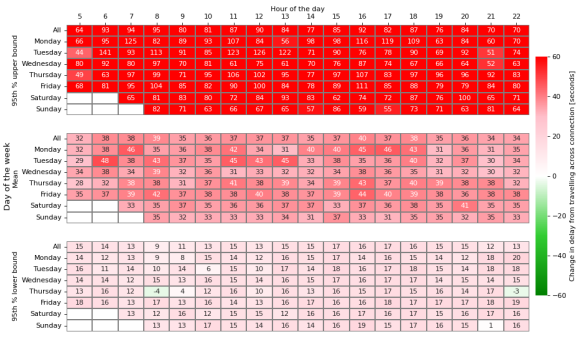


(h) Connection E02 to E03.

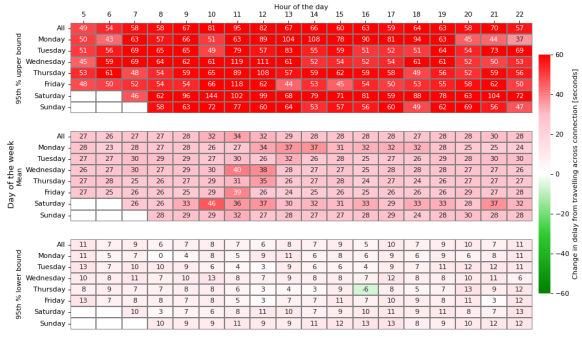
Figure B.15: Second model results from D11 to D12 to E02 to E03.



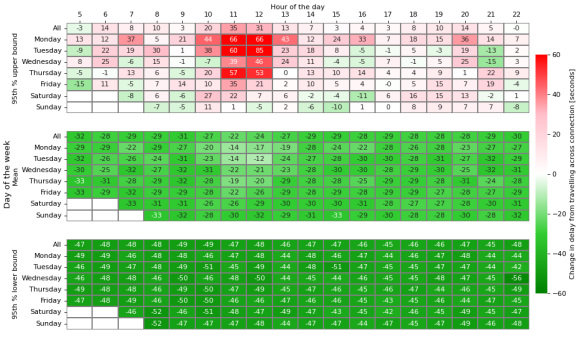
(a) Connection E03 to E02.



(b) Connection E03 to E04.

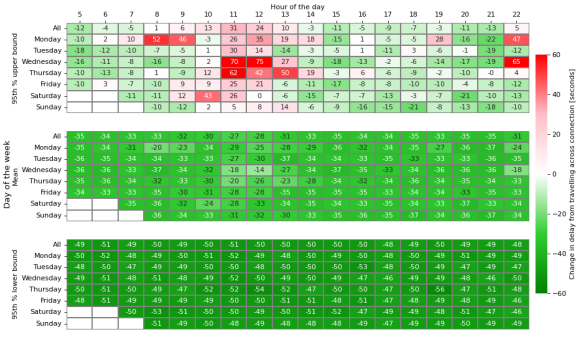


(c) Connection E04 to E03.

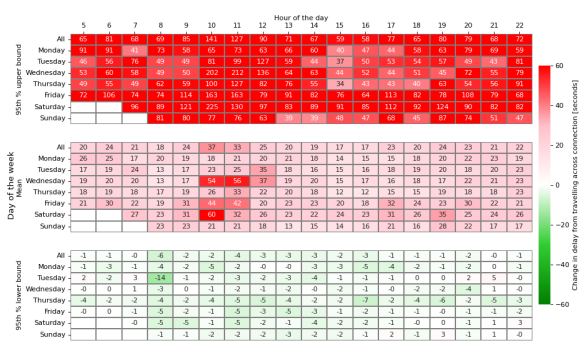


(d) Connection E04 to E05.

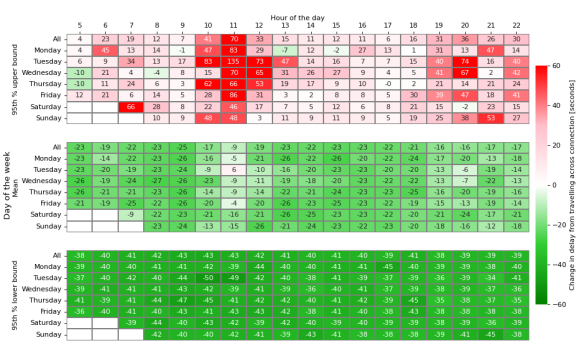
(e) Connection E05 to B06/E06.



(f) Connection E05 to E04.

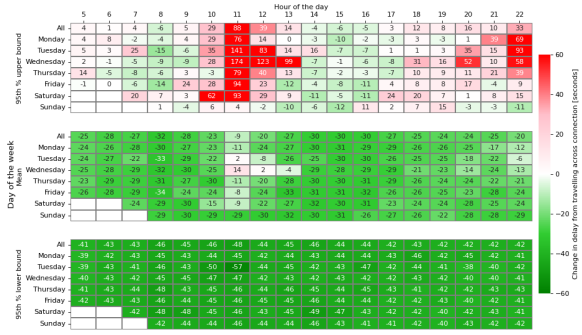


(g) Connection E07 to B06/E06.

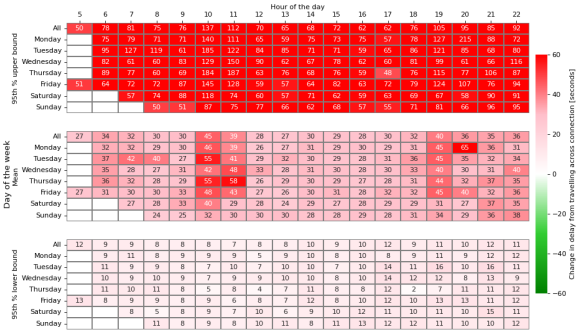


(h) Connection E07 to E08.

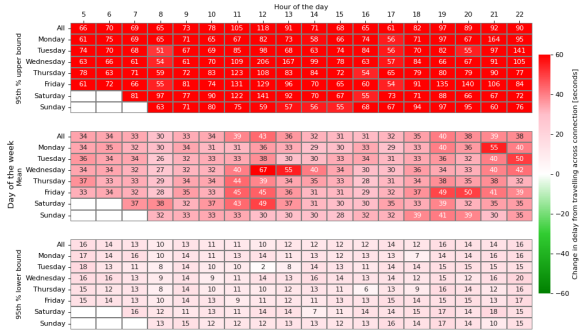
Figure B.16: Second model results from E03 to E02 to E07 to E08.



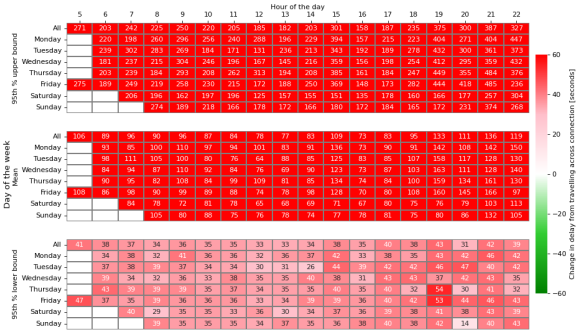
(a) Connection E08 to E07.



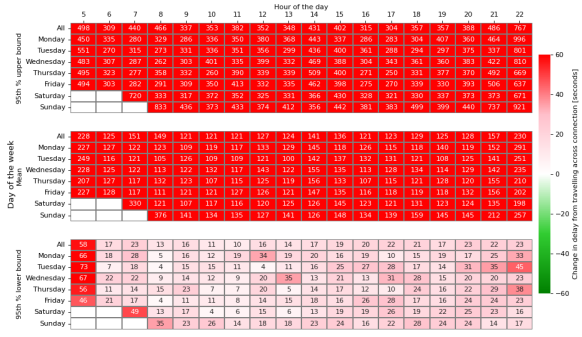
(b) Connection E08 to E09.



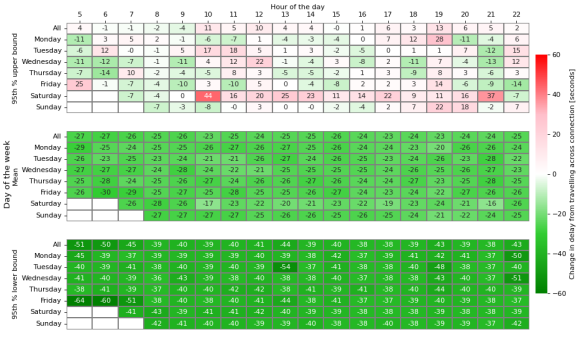
(c) Connection E09 to E08.



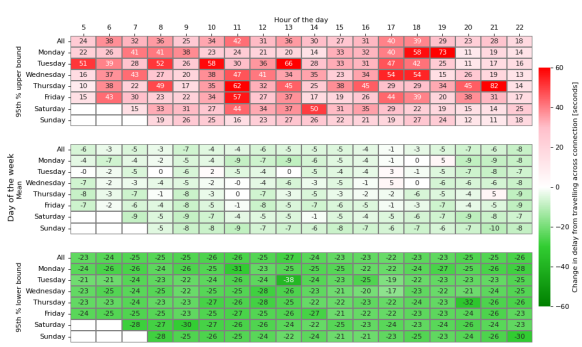
(d) Connection E09 to E10.



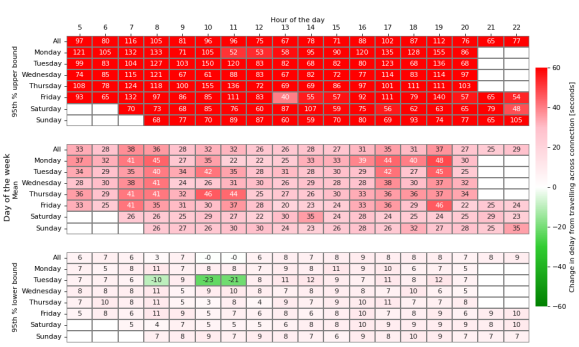
(e) Connection E10 to E09.



(f) Connection F02 to B01/F01.

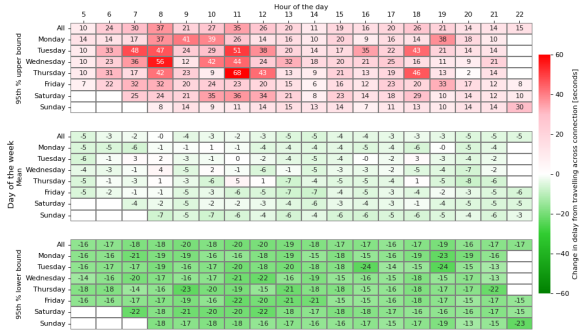


(g) Connection F02 to D03/F03.

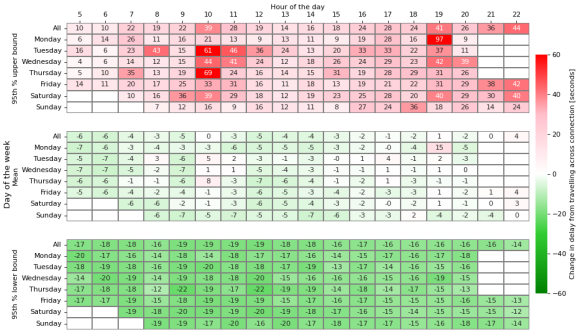


(h) Connection F04 to D03/F03.

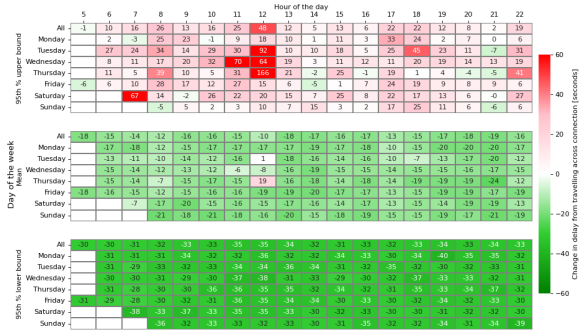
Figure B.17: Second model results from E08 to E07 to F04 to D03/F03.



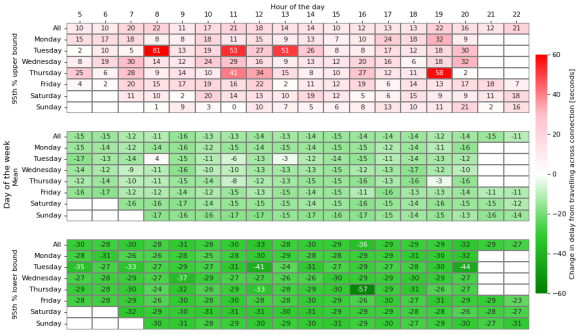
(a) Connection F04 to F05.



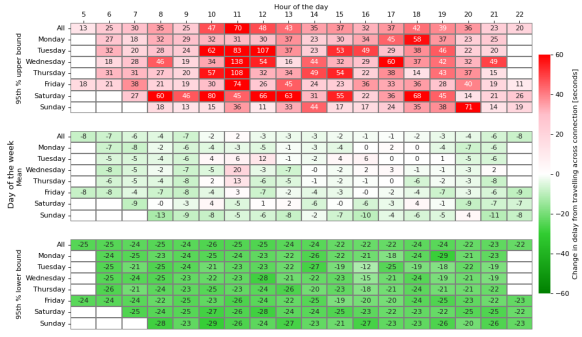
(b) Connection F05 to F04.



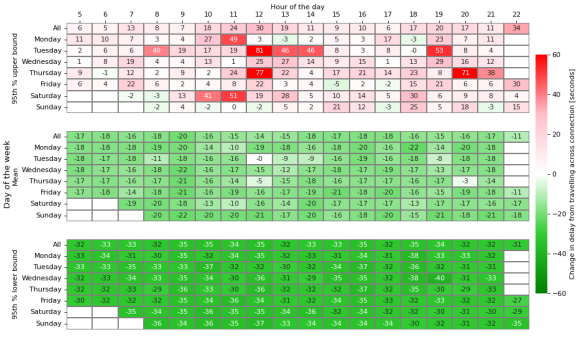
(c) Connection F05 to F06.



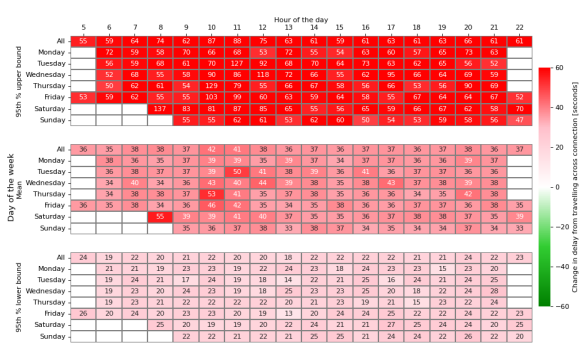
(d) Connection F06 to F05.



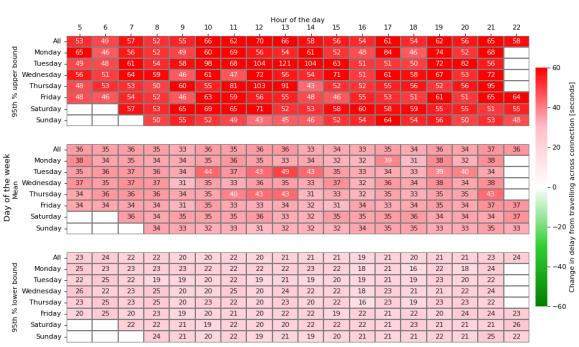
(e) Connection F06 to F07.



(f) Connection F07 to F06.

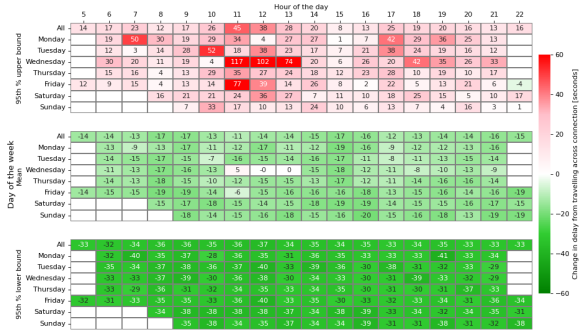


(g) Connection F07 to F08.

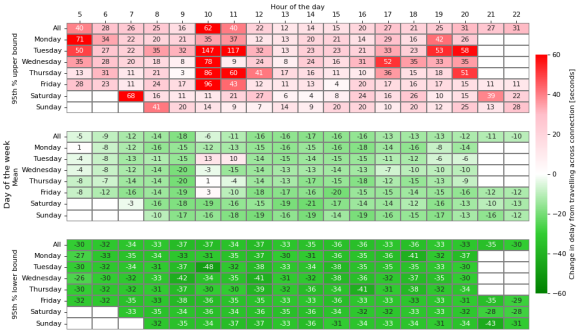


(h) Connection F08 to F07.

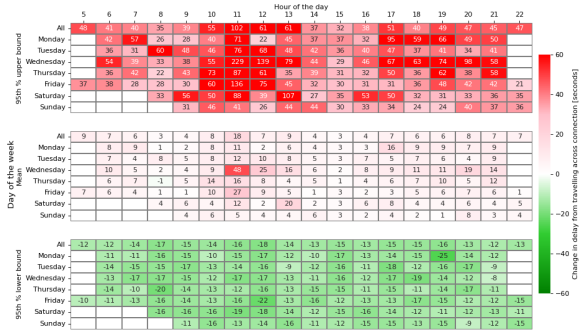
Figure B.18: Second model results from F04 to F05 to F08 to F07.



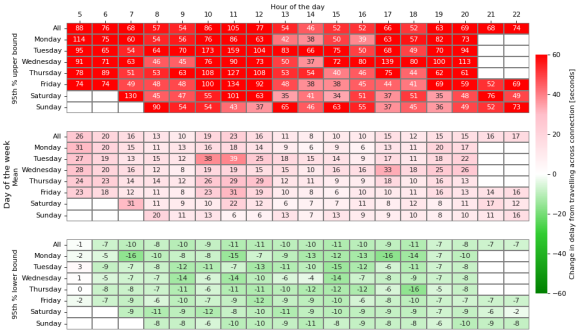
(a) Connection F08 to F09.



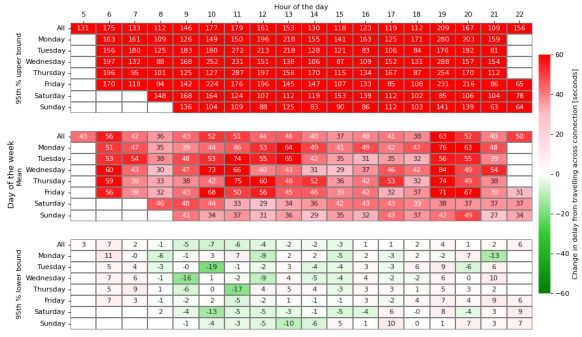
(b) Connection F09 to F08.



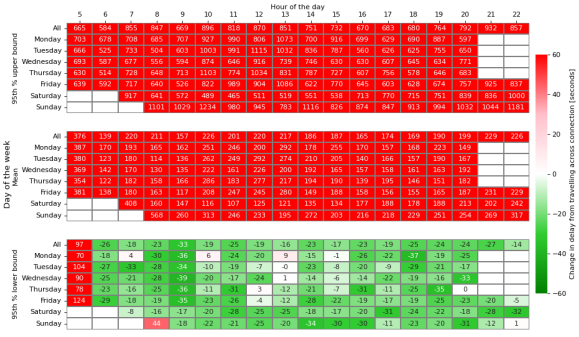
(c) Connection F09 to F10.



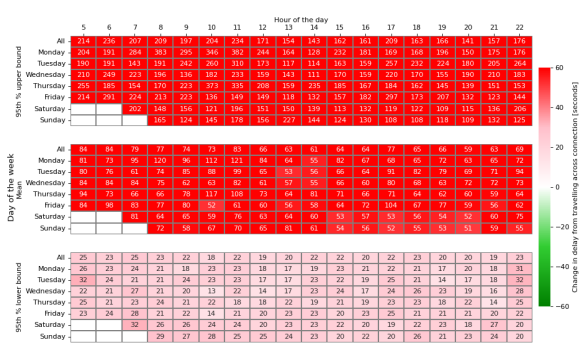
(d) Connection F10 to F09.



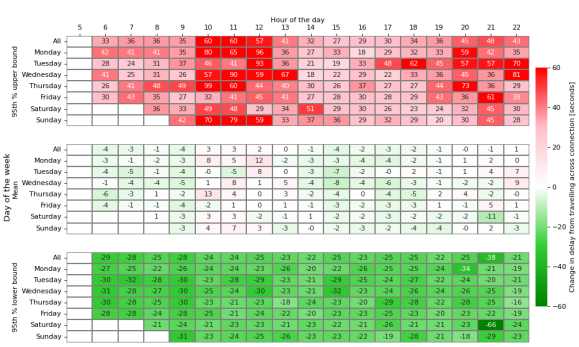
(e) Connection F10 to F11.



(f) Connection F11 to F10.

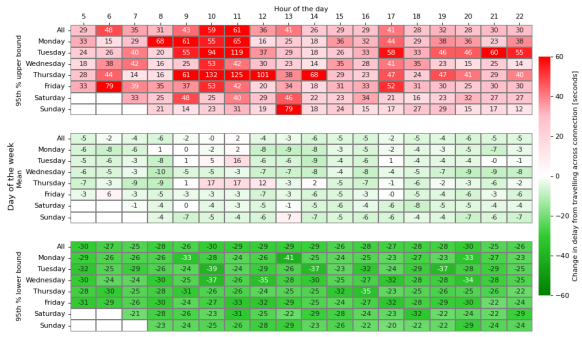


(g) Connection G01 to D08.

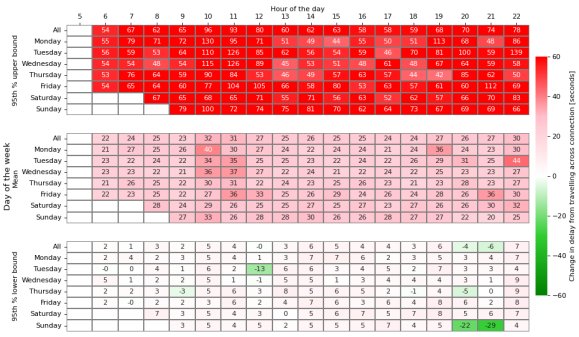


(h) Connection G01 to G02.

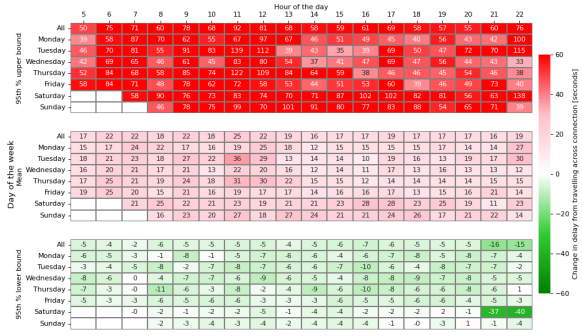
Figure B.19: Second model results from F08 to F09 to G01 to G02.



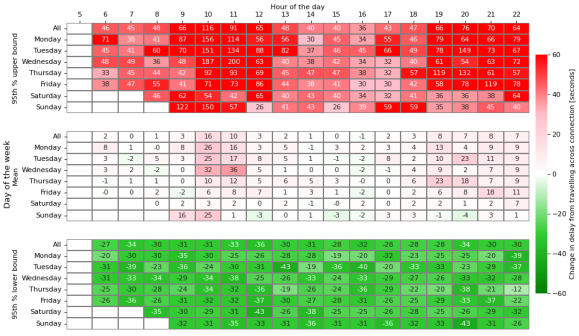
(a) Connection G02 to G01.



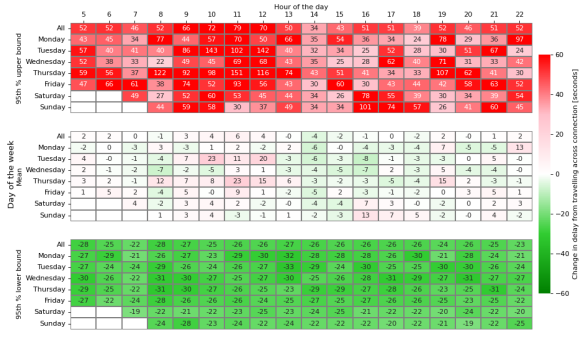
(b) Connection G02 to G03.



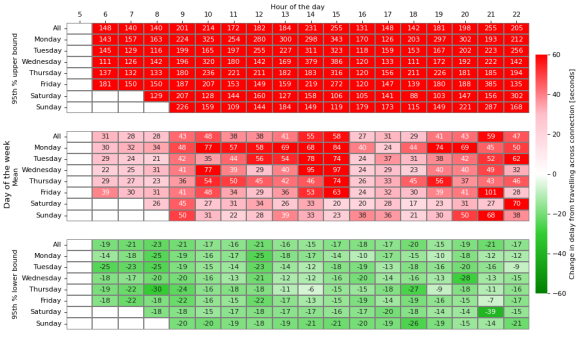
(c) Connection G03 to G02.



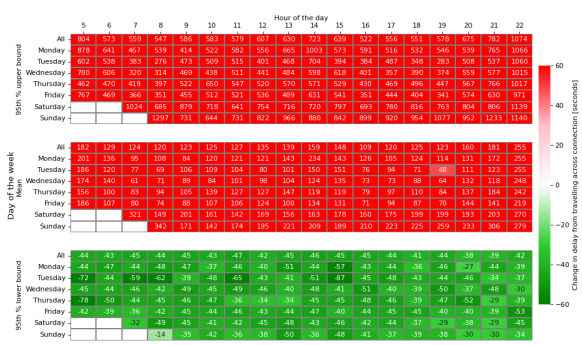
(d) Connection G03 to G04.



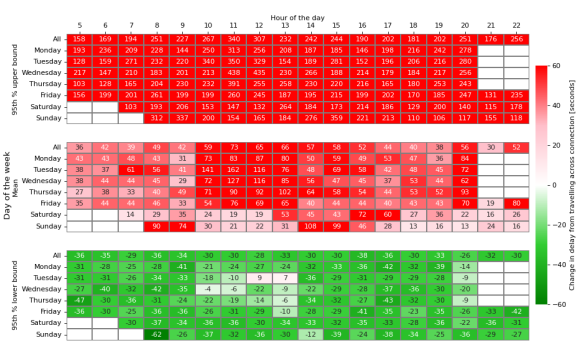
(e) Connection G04 to G03.



(f) Connection G04 to G05.

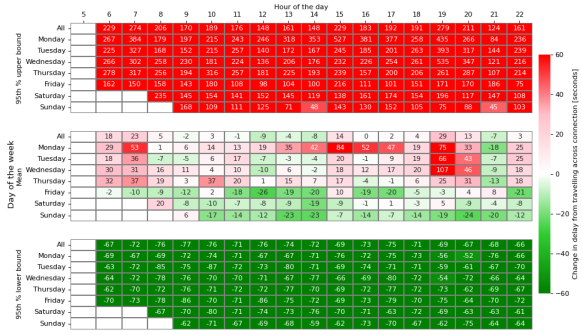


(g) Connection G05 to G04.

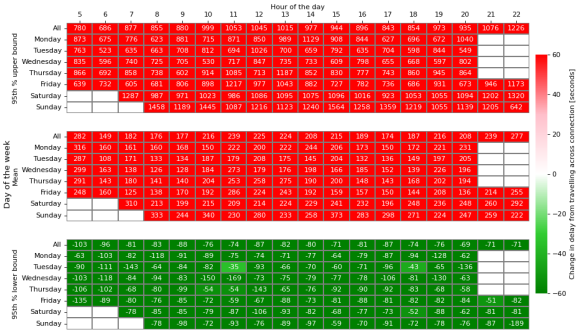


(h) Connection J02 to C13.

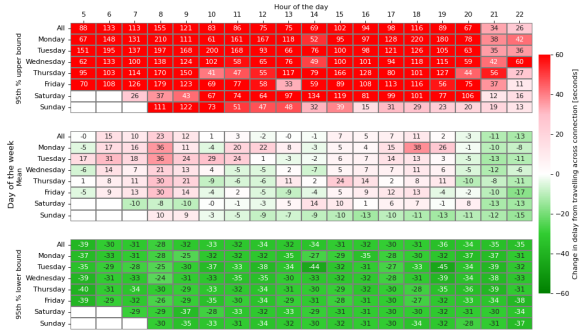
Figure B.20: Second model results from G02 to G01 to J02 to C13.



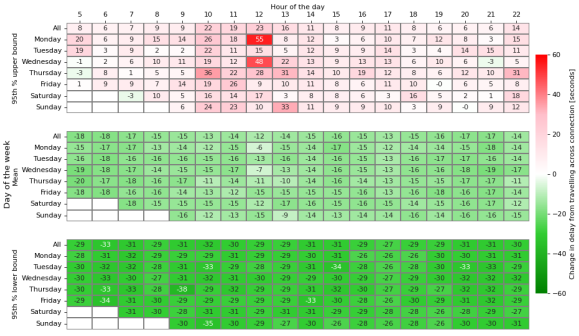
(a) Connection J02 to J03.



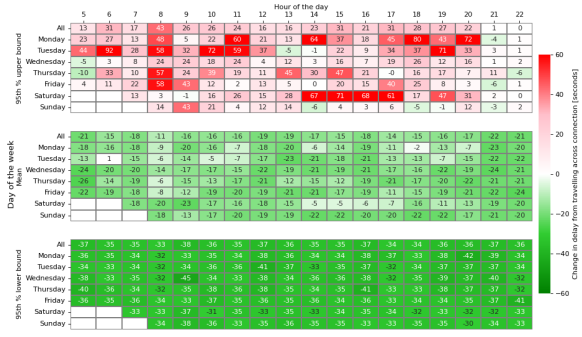
(b) Connection J03 to J02.



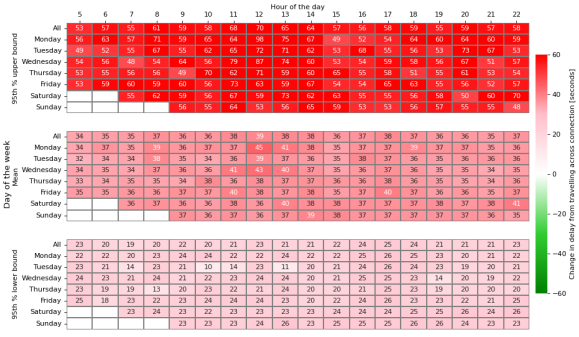
(c) Connection K01 to C05.



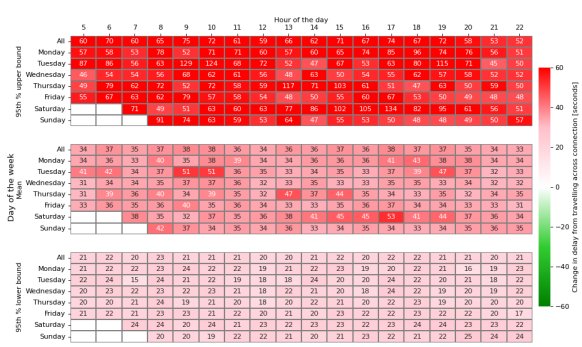
(d) Connection K01 to K02.



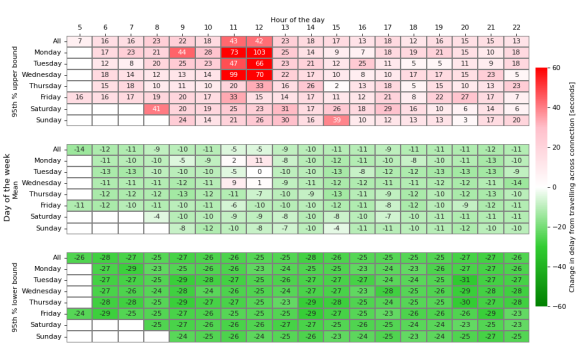
(e) Connection K02 to K01.



(f) Connection K02 to K03.

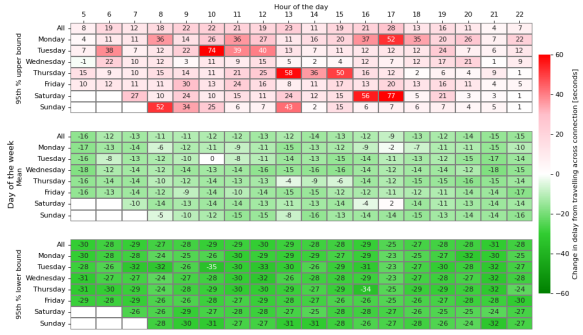


(g) Connection K03 to K02.

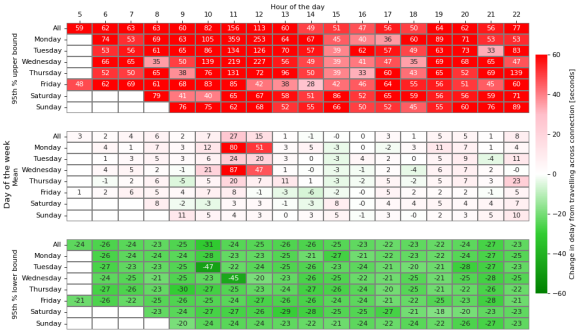


(h) Connection K03 to K04.

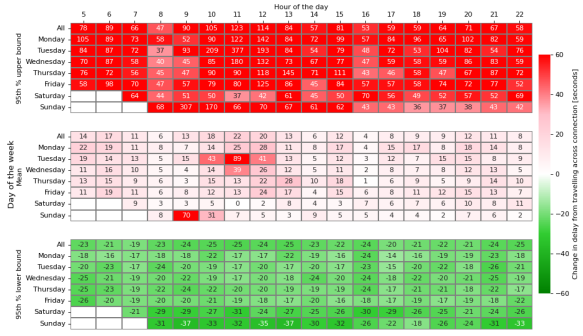
Figure B.21: Second model results from J02 to J03 to K03 to K04.



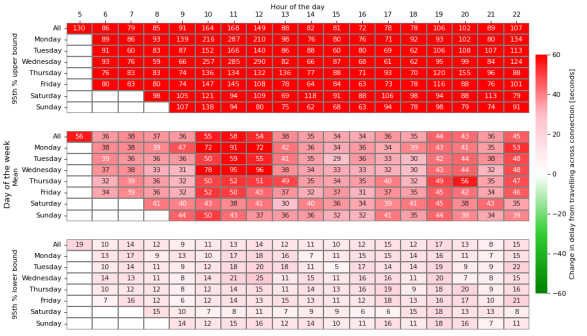
(a) Connection K04 to K03.



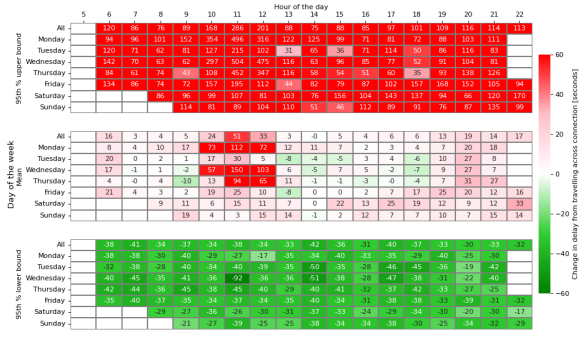
(b) Connection K04 to K05.



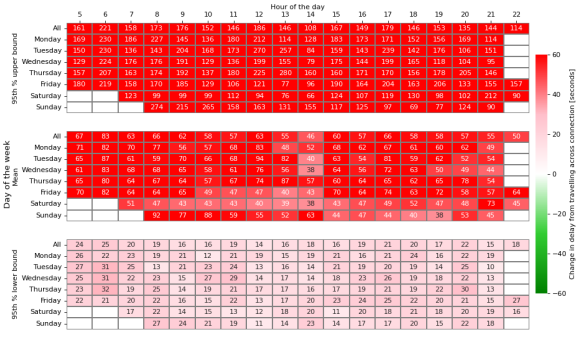
(c) Connection K05 to K04.



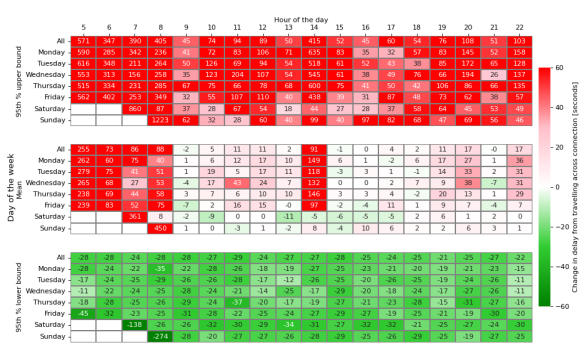
(d) Connection K05 to K06.



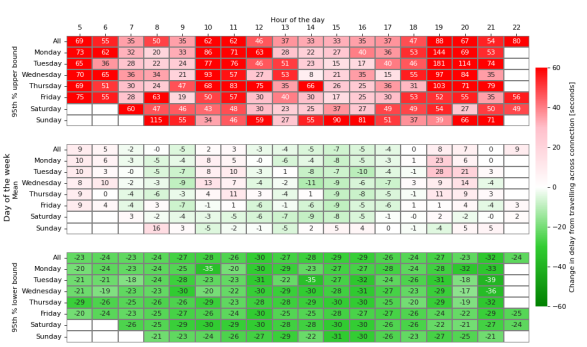
(e) Connection K05 to N01.



(f) Connection K06 to K05.

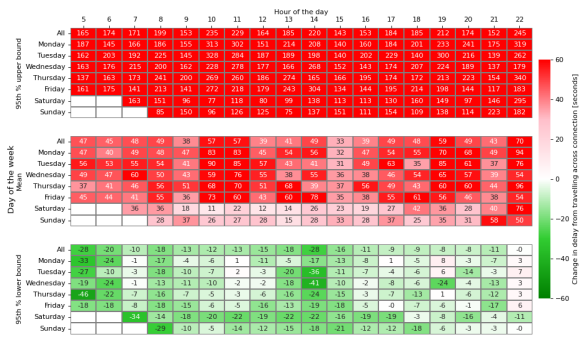


(g) Connection K06 to K07.

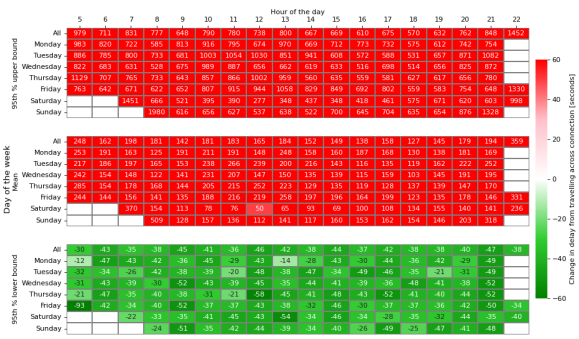


(h) Connection K07 to K06.

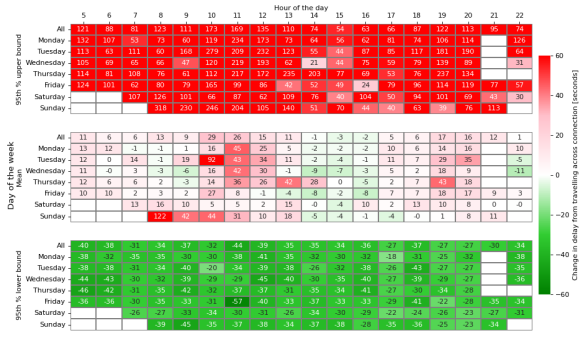
Figure B.22: Second model results from K04 to K03 to K07 to K06.



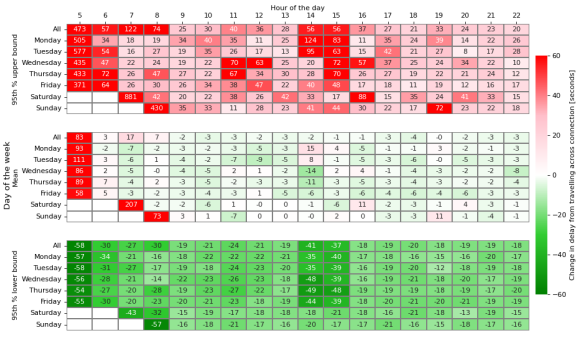
(a) Connection K07 to K08.



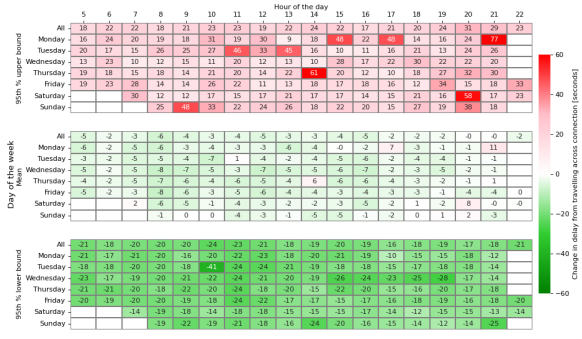
(b) Connection K08 to K07.



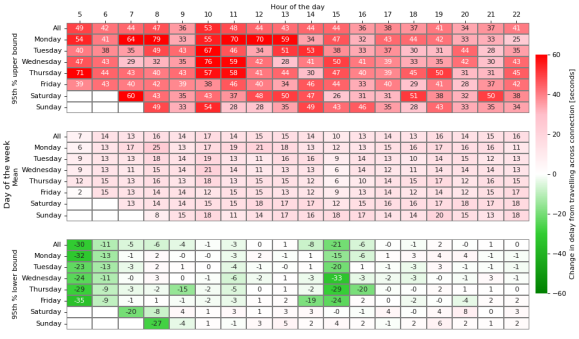
(c) Connection N01 to K05.



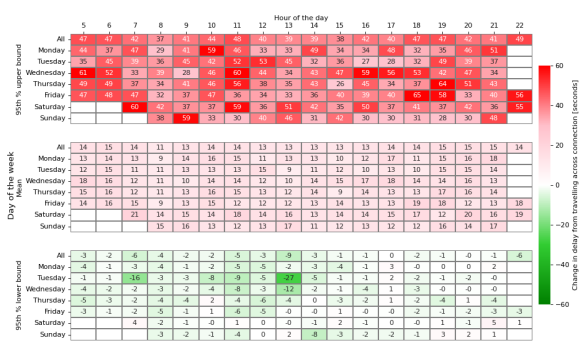
(d) Connection N01 to N02.



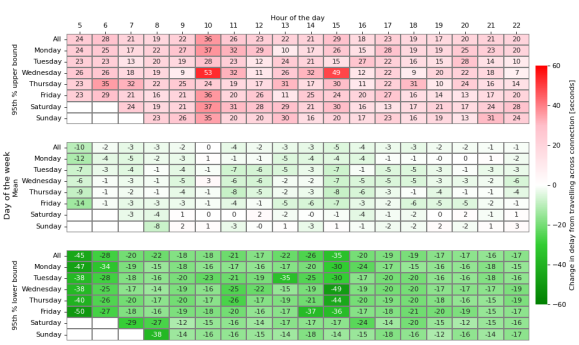
(e) Connection N02 to N01.



(f) Connection N02 to N03.

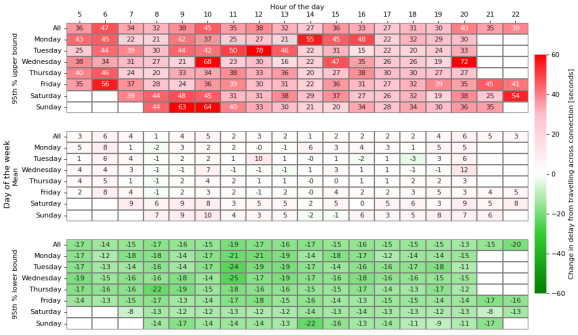


(g) Connection N03 to N02.

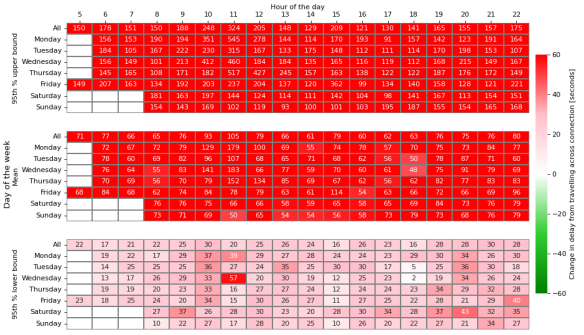


(h) Connection N03 to N04.

Figure B.23: Second model results from K07 to K08 to N03 to N04.



(a) Connection N04 to N03.



(b) Connection N04 to N06.

(c) Connection N06 to N04.

Figure B.24: Second model results from N04 to N03 to N06 to N04.

C

Average Propagation Behaviour Lower and Upper Bound Maps

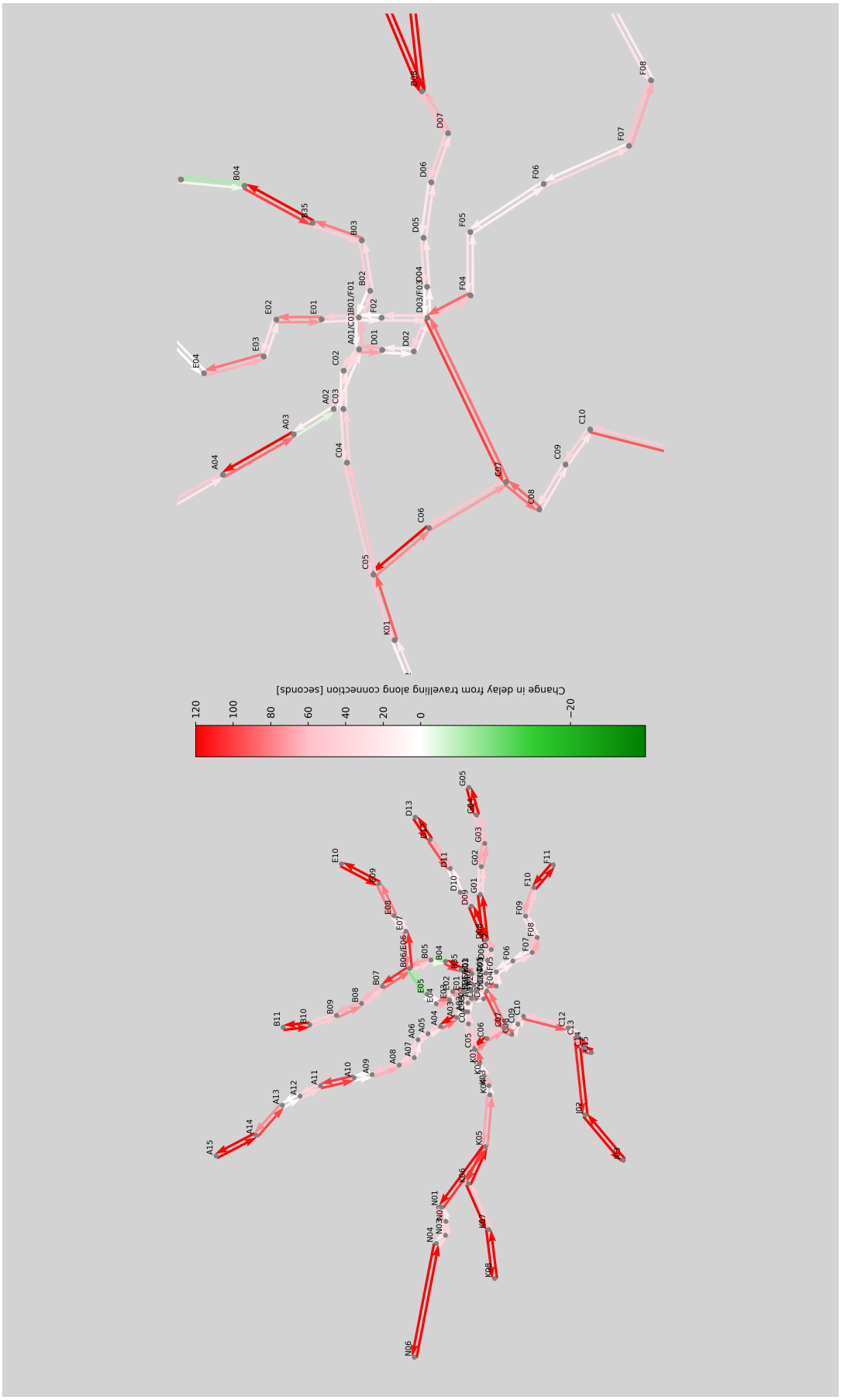


Figure C.1: Overall 95th percentile of confidence range higher bound delay propagation behaviour per connection, with the total metro network on the left, and zoomed in on the network's centre on the right.

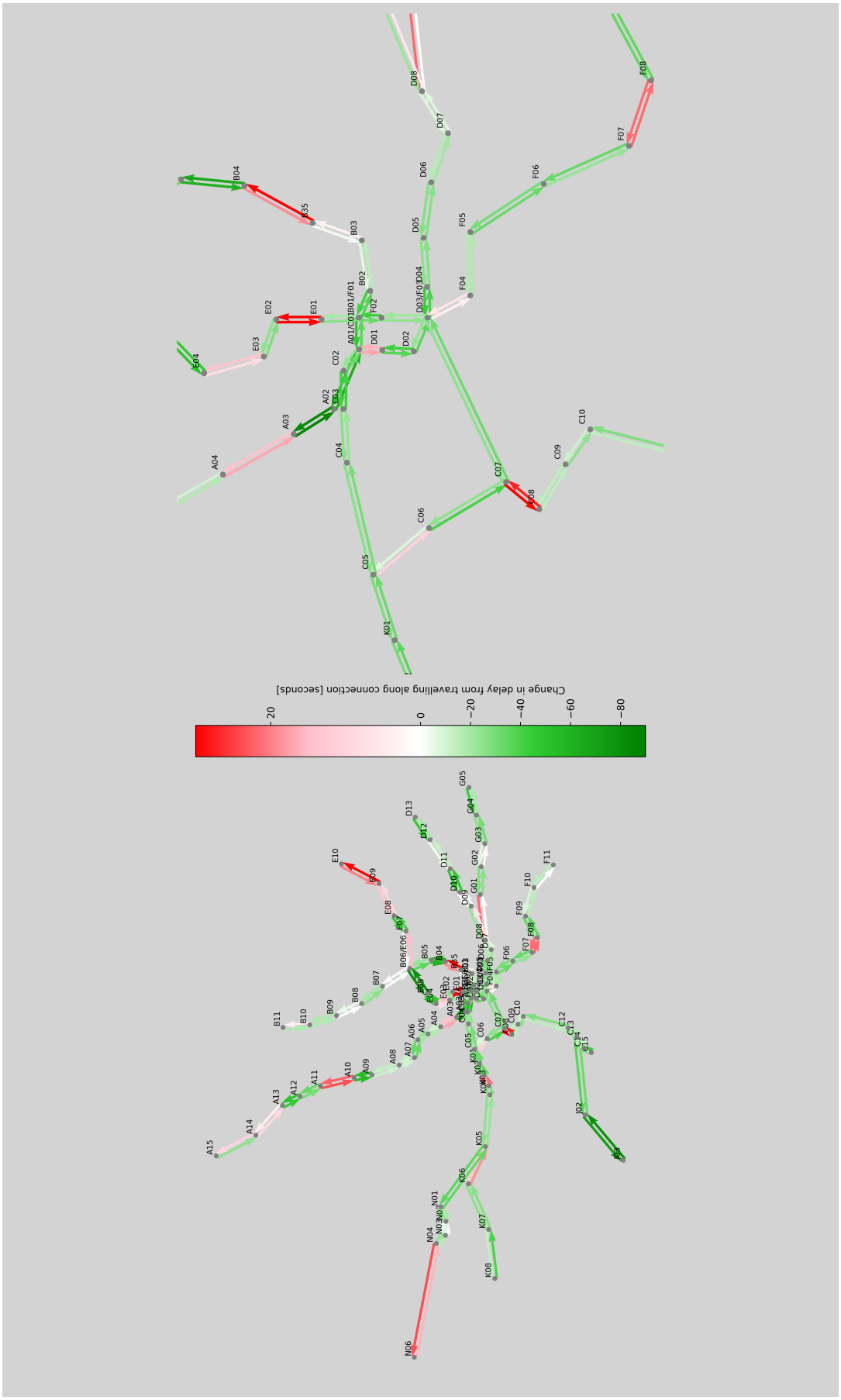


Figure C.2: Overall 95th percentile of confidence range higher bound delay propagation behaviour per connection, with the total metro network on the left, and zoomed in on the network's centre on the right.

D

Unfiltered Cross Examinations

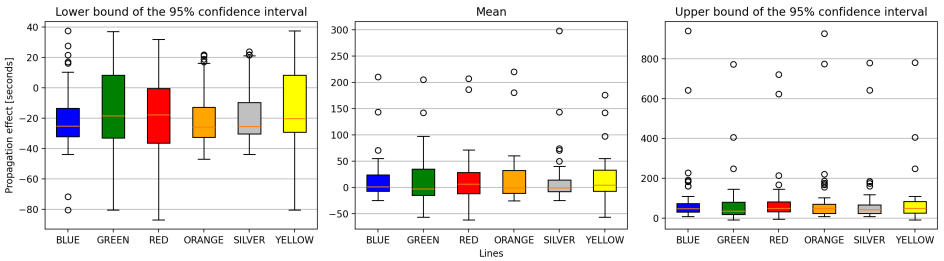


Figure D.1: Propagation behaviour per line all connections.

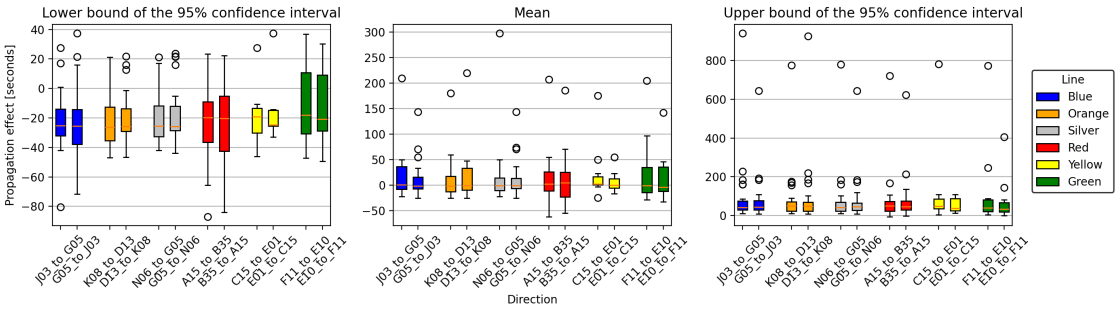


Figure D.2: Propagation behaviour per line per direction all connections.

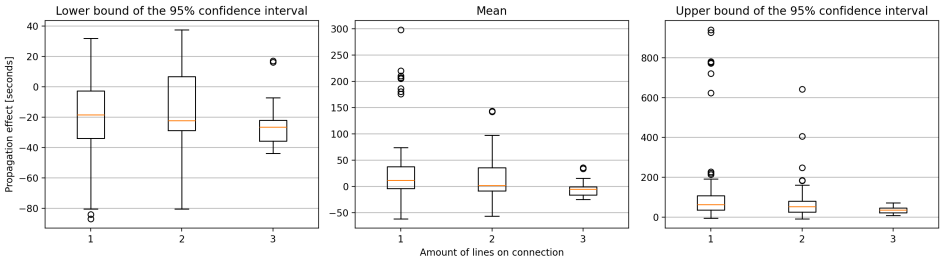


Figure D.3: Propagation behaviour per number of lines on the connection all connections.

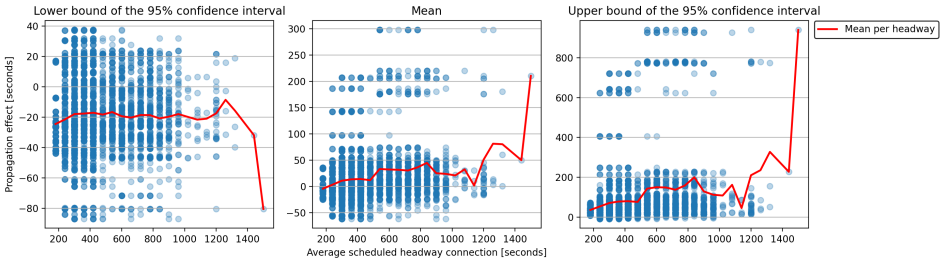


Figure D.4: Propagation behaviour per average scheduled headway all connections.

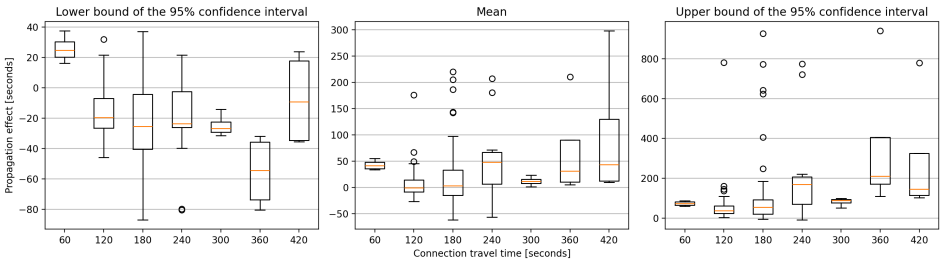


Figure D.5: Propagation behaviour per amount of allocated travel all connections.

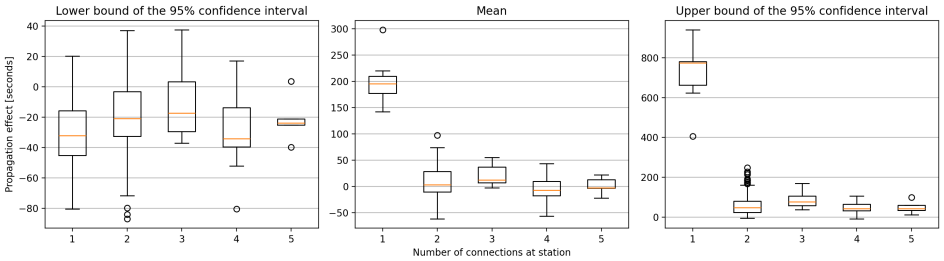


Figure D.6: Propagation behaviour per number of connection at a station all connections.

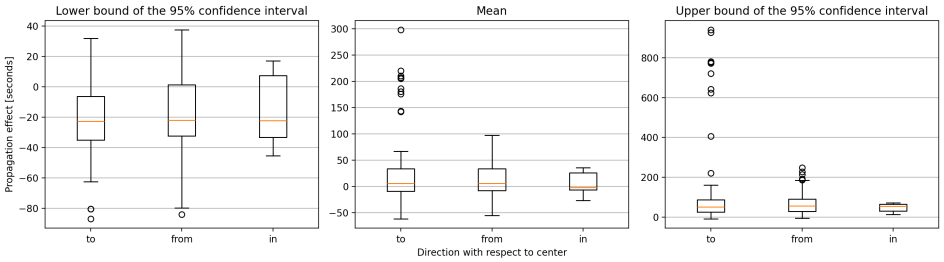


Figure D.7: Propagation behaviour per direction with respect to the network centre all connections.

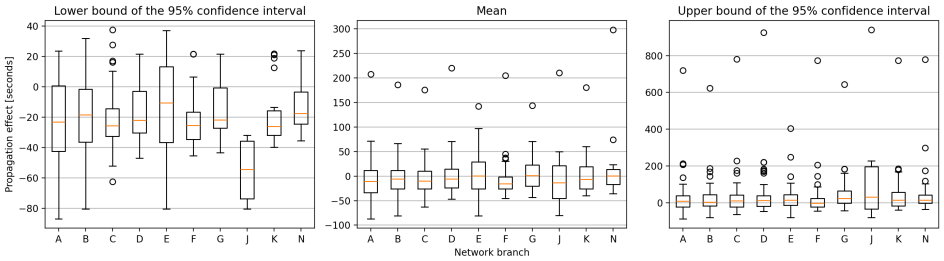


Figure D.8: Propagation behaviour per network branch all connections.