CHALMERS
UNIVERSITY OF TECHNOLOGY

# Detecting and studying non-recurring congestion in Helsinki

A study of non-recurring congestion detection methods

Master's thesis in Space, Earth and Environment

Xavier Domènech Garcia

Department of Space, Earth and Environment
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2021

Master's thesis 2021

# Detecting and studying non-recurring congestion in Helsinki

A study of non-recurring congestion detection methods

Xavier Domènech Garcia

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Space, Earth and Environment
Chalmers University of Technology
Gothenburg, Sweden 2021

Detecting non-recurring congestion in Helsinki
A study of non-recurring congestion detection methods
Xavier Domènech Garcia



Supervisor: Wasim Shoman, Space, Earth and Environment
Examiner: Sonia Yeh, Space, Earth and Environment

Master's Thesis 2021
Department of Space, Earth and Environment
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Description of the picture on the cover page (if applicable)

Typeset in LaTeX
Gothenburg, Sweden 2021

Detecting non-recurring congestion in Helsinki
A study of non-recurring congestion detection methods
replace this with only Chalmers' logo
Department of Space, Earth and Environment
Chalmers University of Technology and University of Gothenburg

# Abstract

The traffic related data collected from sensors on the road has been growing tremendously. This trend requires new tools to process and analyse the records to extract useful information such as incident detection. For this reason, this thesis explores three different methods extracted from literature to detect road incidents (non-recurring congestion) in Helsinki and analyse them using delay as a traffic congestion metric with one year of data records (i.e., 2018). The three methods tested are based on: dynamic threshold, delay prediction, and outlier detection. The first method uses percentile score to set the thresholds that vary with time, whereas the last two methods are performed employing the algorithm k-Nearest Neighbours (k-NN). The methods validation is tested using a data source that contains part of the incidents occurred during the recordings. This data source is uncompleted as the major part of the incidents are usually not reported. The results shows that k-NN outlier detection outperform the other two methods as reach the 60% of accuracy with the most reasonable number of non-reported incidents (4,35%). Therefore, the research expose that outlier methods are one of the easiest ways to detect incidents in traffic data. Such methods trend to be simpler and obtain better results than more complex techniques. Finally, the incident analysis shows that the most severe accidents occur during the morning rush hour, whereas the afternoon rush hour concentrates the major number of incidents.

Keywords: Traffic, congestion, machine learning, outlier detection.

# Acknowledgements

Firstly, I would like to express my acknowledgements to Professor Sonia Yeh and Wasim Shoman for their constant guide, advice and support. Thanks also for the patience showed during the large amount of long meetings. Their new ideas, corrections and knowledge have marked the whole thesis.
Additionally, I would like to acknowledge the help provided by Antoni Guasch for assist me focusing the thesis. I wish to thank my family for giving me the opportunity to develop my master thesis abroad, and for the unconditional aid given. Finally, I want to express gratitude to my friends of Barcelona and to the Erasmus friends of Göteborg for all the moral support given during the thesis development.

Xavier Domènech Garcia, Gothenburg, September 2021

# Contents

# List of Figures

# List of Tables

# 1

# Introduction

During the last century, the percentage of people living in cities has continued to grow [4, 5]. As a result, cities have increased not only their internal mobility, but also the people trying to get in and out of them. Despite the recent COVID pandemic that has temporarily reduced the travel demand, road traffic has rebounded to its normal pre-pandemic level or even higher [6]. Meanwhile, every year millions of dollars are lost because of delays, extra fuel consumption, and incidents [7] [8]. Moreover, there is an environmental cost and psychological stress that are caused by traffic congestion [9]. Consequently, for more than 100 years, many governments, universities, and companies have investigated and developed systems to study and control traffic to understand and avoid major congestion [10] [11]. Despite the efforts made, the cost caused by congestion during 2018 at EU states member was approximately 100 billion €. And the forecast for 2050 is 150 billion € [12].

## 1.1 Background

For the last 20 years, many sensors measuring different parameters have been placed on road segments around the globe [13]. The amount of information collected combined with the increase of computational power has enabled machine learning to increase its prominence in traffic related studies [2, 3, 14, 15]. Despite of that, the analysis of large amounts of data hides many challenges such as handle big files, differentiate useful information from noisy one, and more [16].

Road incidents detection is an important research field in traffic studies [16]. A road incident is considered as an event such as an accident, bad weather, or construction works, that causes more traffic congestion than normal [17]. When analysing traffic data, congestion caused by incidents can be difficult to differentiate from the variability of usual traffic [18]. Accordingly, the congestion caused by incidents is named non-recurring congestion (NRC) . In contrast, recurring congestion can be defined as the congestion that appears seasonally during peak hours [19].

There have been many attempts to study how much of the total congestion is caused by NRC incidents [19, 20, 21], but there are not many studies focused on detecting and studying incidents. Recently, NRC detection has gained relevance due to the necessity of traffic centers to detect it, and take preventive action to mitigate its effects [16]. The importance of NRC relays on its unpredictable character, which introduce a random variable when people and companies try to plan their journeys,

meetings, and delivery times. Therefore, a quick mitigation of NRC contribute to strengthen the economy and make life easier to people [22].

Most of the NRC detection studies have focused on a limited subset of a given road network due to the necessity of evaluating many factors at the same time such as: velocity, weather conditions, incidents, and traffic capacity [17, 22]. That number of parameters can be hard to record in a large road network such as those found in cities.

In consequence, when studying NRC in urban networks, more challenges need to be dealt with. One is the lack of incidents information. It is not easy to record information of a big road network and usually there is no way to know if there was a real NRC episode or not [23]. In addition, the urban networks tend to be heterogeneous in terms of lengths and links [24]. Therefore, the NRC events may not have the same consequences depending on which road they take place at. As an example, short roads tend to have more variability in terms of traffic, while long roads do not [16]. That different behaviour makes it more difficult to detect NRC in both kinds of roads at the same time .

## 1.2 Aim of the project

The aim of this master thesis is to detect and study non-recurring congestion in Helsinki, Finland. The dataset consists in speed recordings obtained every 5 minutes during 2018 from HERE [25], a major online real-time traffic information provider. Therefore, the thesis analyze and study the historical data recorded during that year. The pillars for achieving the main goals are:

- Find a valid method to detect non-recurring congestion.
- Analyse non-recurring congestion episodes.

Within this objectives, it is aimed to facilitate future work on NRC episodes detection.

## 1.3 Thesis outline

The Master thesis is structured in six chapters. Chapter 1 contains the introduction to the topic and the thesis structure. On chapter 2, a background of the traffic data and the methods used to detect NRC are explained. Chapter 3 describes the data-sets used and the processing and cleaning steps applied. Chapter 4 explains the methodology followed to analyse the data, the three different NRC detection methods tested, and the analysis of the NRC episodes found. Chapter 5 presents the results obtained and their discussion. Finally, on chapter 6 the conclusions of the master thesis are exposed as well as the proposed future work.

# 2

# Background

This chapter include previous literature findings on congestion detection and study. Additionally, it introduces some crucial topics such as recurring/non-recurring congestion, traffic data collection methods and congestion metrics.

## 2.1 Recurring/Non-recurring congestion

Most traffic centers distinguish between two types of congestion: recurring and non-recurring [16]. Recurring congestion is the one that is expected to happen regularly as it is caused by an excess of demand like on rush hours. On the other hand, non-recurring congestion (NRC) is caused by unexpected events. Some examples could be accidents, vehicles breakdowns, or scheduled events like road works and football matches [17].

## 2.2 Data collection methods

Each traffic study is influenced by the data set used. At the same time, the data set is conditioned by the collection method used to sample the data. The collection of traffic data can be separated into two main groups: fixed sensors, and mobile devices.

### 2.2.1 Fixed sensors

The traditional source of traffic data has been the sensors that can be spread around the network. There are many kinds of sensors like traffic cameras, radar sensors, inductive loop detectors, or magnetic sensors. This infrastructure is usually more reliable than mobile devices. The drawback of fixed sensors is the cost and the huge infrastructure required to cover a significant part of the network. Also, it can be inaccurate when measuring different kinds of vehicles such as trucks and motorbikes [26].

Some of this sensors do not provide instant information. An example is traffic cameras. This kind of sensors report visual data of the road status. That information needs to be converted into some numeric data to be useful to work with. One of the technologies used is Automatic Number Plate Recognition (ANPR) cameras.

That system allows for the estimation of the aggregate of link journey time that is explained in section 2.3.

### 2.2.2 Mobile devices

During the last few years, the evolution of wireless technologies, cars, and mobile phones have changed data collection significantly. Nowadays, there are many devices collecting data that can be useful to analyze the road status. Consequently, new companies have emerged to collect and uniform that amount of data to make it profitable [1]. Some of those companies are HERE, INRIX and TomTom.

Those data providers usually divide the road network into segments using Traffic Message Channel (TMC) location reference [27]. For each segment at each instant of time, information about the traffic speed, time, and other complementary information is provided.

The advantage of this kind of data source is that it is available from any part of the network without extra infrastructure. The drawback is that this information is not always reliable and accurate. That is why many studies have focused on studying the reliability of these measurements, comparing the data with measurements taken with other technologies such as radar sensors [28].

## 2.3 Congestion measurement metrics

There are many metrics to evaluate the level of congestion. Each of them has its own advantages and disadvantages, but all of them should have common specific characteristics as it is specified at [26]:

- Easy to understand and unambiguous.
- Ability to describe the current state of traffic and its future changes.
- Ability to apply statistical techniques.
- Applicability to various facilities and time periods.

Below, you can find some of the congestion metrics used on NRC detection studies and their explanations. Notice that in most studies, the data used and how it has been collected set also the congestion metric.

#### Speed

Speed is one of the simplest metrics. It is easy to understand as usually it represents the average speed of a part of a road during a determined period of time. The lower the speed is compared to the road limit, the more congestion there is expected to be. Further details about the speed as a congestion metric can be found at [26]. Some NRC detection studios explained at 2.4 that use speed as a metric are [1, 22, 29]. On the other hand, speed metric cannot be directly used to compare roads that have different speed limit.

### Travel time

The travel time metric in congestion is used to measure the time that is required to go from one point to another [30]. It has the advantage of expressing congestion in terms of both space and time.

Many metrics are derived from travel time, such as Link Journey Time (LJT). The LJT is the estimated journey time through a link at an established time interval and is obtained by matching the ANPR cameras' readings [16, 18]. The procedure of the conversion from images to LJT is explained in [31].

Another research that uses travel time as a congestion metric is [32] where the travel time is recorded by sensors on the city main roads. After processing the information, they use causality trees [33] to find recurrent congestion patterns.

### Delay

The delay metric is calculated with the travel time. It is similar to the travel time and is defined as the additional travel time necessary to cover a distance compared to the travel time necessary in a non congested scenario. Consequently, and as it is the case with the travel time metric, it is not a direct measure and is mainly calculated using the speed.

Delay is a normalized metric that allows comparing different roads, and consequently facilitates analyzing and extracting conclusions from their values. Also, it can be useful when there is a lack of flow information. For these reasons, this is the metric chosen within this master thesis. The used formulas are shown below as explained in [34]:

$$t_{1_i} = \frac{l_i}{f_i} \cdot 60 \tag{2.1}$$

$$t_{0_i} = \frac{l_i}{s_i} \cdot 60 \tag{2.2}$$

$$Delay = max\left(\frac{t_{0_i} - t_{1_i}}{l_i}, 0\right) \tag{2.3}$$

The $t_{0i}$ is travel time in minutes of a vehicle running at free flow speed for the segment $i$. Where $l_i$ is the length of the segment in km, and $f_i$ is the speed limit. The $t_{1i}$ is travel time in minutes of a vehicle running at current speed for the same segment $i$. Where $s_i$ is the current speed of the segment $i$ at a given time.

Finally, the delay is expressed in min/km. Notice that if the current travel time is lower than the free flow travel time, it is counted as 0 min/km of delay. Delay can also be relativized to permit different kind of roads comparison. The drawback is that it can be difficult to understand as it is a dimensionless measure.

## 2.4 Incident detection methods

The methods used to detect incidents are strongly related to each research and the data set used. Therefore, it is recommended to know the whole context of the study to understand each method. The way to detect incidents is based on finding abnormal values of congestion, that is to say NRC. Most of the studies use statistical tools to develop their methods, but also some of them apply machine learning techniques. In this section a summary of methods found on the literature are explained.

### Dynamic threshold

The dynamic threshold method is based on a threshold that changes through time. The measures that have higher/lower values than the threshold are considered as suspicious of being part of a NRC episode. A dynamic threshold can be applied to many types of measures, the most common are speed and travel time.

**Speed based** An example of speed dynamic threshold application is [1]. The starting point of this study is a speed data source obtained by the data provider INRIX [35] every 1 minute of the region Iowa. After evaluating the data quality, they decided to focus their study only on the Iowa Interstates network as its quality was superior compared with the rest of the network. The dynamic threshold is calculated using the algorithm adopted by [36]. That algorithm uses the last two months of data from each segment and calculates a threshold every 15 minutes for each weekday as follows:

$$Threshold = Median\ Speed - 2 \cdot IQR^{1} \tag{2.4}$$

Additionally, another static threshold is taken into account: 45 mph. They realized that below 45 miles per hour (mph), the quality of the speed measurements started to decrease considerably. Finally, the detection of non-recurring congestion is summarized as:

1. The speed drops below 45 mph.
2. The speed drops below the threshold during more than 15 minutes.
3. A matching incident must be reported by Iowa Advanced Traffic Management System.

At the following figure extracted from that research, the detection is illustrated:

[1] IQR: Inter Quartile Range. Difference between the 75th and 25th percentiles.

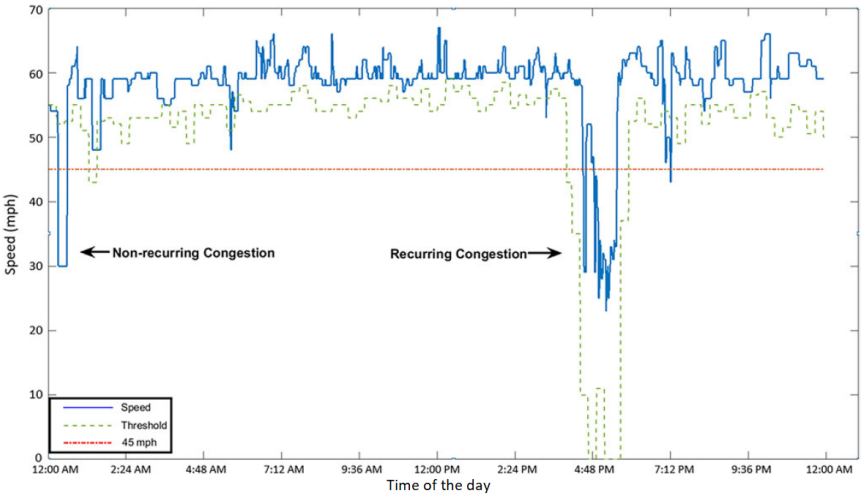


**Figure 2.1:** Example of the speed dynamic threshold detection used in [1]. Source: [1]

A relevant conclusion they get through is that the segments with a length lower than 0,64 km have more speed variability and are more difficult to study.

Another example of speed dynamic threshold is the study [22]. The method is based on an adjusted boxplot. The data source is real-time traffic conditions for a one minute interval from each TMC segment. Their proposal is based on the premise that any measure out of the following interval can be classified as a potential crash case:

$$[Q_1 - 1.5 \cdot IQR,\ Q_3 + 1.5 \cdot IQR] \tag{2.5}$$

Where $Q$ is the quantile, and $IQR$ the inter quantile limit. The interval described at 2.5 cannot be applied directly as the results could be distorted by if the data is skewed. On their case, many regular observations exceed the interval defined at thick-tailed symmetric distributions, while the contrary effect happens at thin tailed distribution [37]. To avoid that, they measure the skewness of their data using the medcouple (MC) [38]. Finally, they managed to adjust the skewness as follows:

$$[h_l(MC) = 1.5e^{aMC},\ h_u(MC) = 1.5e^{bMC} \tag{2.6}$$

Following what Chung and Recker studied in [39], two speed sections at a time can be defined:

- $S_i(t_n) <= Q_1 - h_l(MC)$ IQR: under crash impact area
- $S_i(t_n) > Q_1 - h_l(MC)$ IQR: crash-free area.

Consequently, the measurements located in the first section are defined as possible NRC episodes.

**Time travel based** Two examples of time travel dynamic threshold are explained in [16, 18]. Their study is based on Link Journey Time data obtained using ANPR

cameras in London. The dynamic threshold is based on the per cent point function method to consider the statistical distribution of the LJTs. The formula was extracted from [40]:

$$G(p) = exp(\mu + \sigma\phi^{-1}(p)) \tag{2.7}$$

P is the cumulative probability, $\phi^{-1}(p)$ is the per cent point function of the standard normal distribution function, mu and sigma are the mean and standard deviation of the underlying normal distribution, respectively.

The process calculates the pith percentile value of an LJT for each time. A measure is considered to belong to an NRC if the measure is higher than the percentile limit calculated for that link at that time. Then the LJT that is higher than its pith percentile are clustered with other LJTs that are linked to and that had higher measurements at the same time as well (Spatio-temporal overlap).

#### Expected traffic

This method is based on calculating the expected traffic status for each instant of time, and then compare the calculated values with the real ones. If the difference between the two measurements is too large, the method reports a NRC episode. An example of this method is also applied in studies [16, 18], that proposed a second method adapted from the expectation-based space-time regions (STSs) described in [41]. The procedure relies on three steps:

1. Creating space-time regions (STR)
2. Determining the likelihood ratio function using the LJT distribution.
3. Determining significant STRs by comparing the likelihood ratio scores with the observed data.

A fake scenario with no NRC episodes is created and compared with the observed LJT. LJTs that do not fit the fake scenario are classified as NRC. Finally, the same spatio-temporal overlap observations are made.

#### Machine learning outlier detection

Another way to approach NRC detection is to apply machine learning [42] to detect outlier values. There are many algorithms that can be used to detect outliers such as DBSCAN, k-NN, SVM, and more.

In [14] a distance based method using k-Nearest Neighbor (k-NN) is applied to detect outliers from the flow distribution probabilities. Their study can be structured in two main parts:

1. Building a flow distribution for each location for a given time interval using historical data.
2. Applying the outlier detection technique.

The historical distribution is used to detect outliers in new flow's distribution coming

in a streaming way. If the data is an inlier, it is included in the historical data. For the outlier detection, they used k-NN-FDP[2] algorithm [15]. The algorithm returned a Boolean that indicates if the new flow measure is an outlier or not. The algorithm calculate the distance using KL-divergence between the historical values and the new value. If it exceed a threshold, the measure is labeled as an outlier.

### Machine learning traffic prediction

During the background research, other studies based on machine learning traffic prediction, but not focused on NRC detection, were found. Some of those studies are explained below as one of the NRC detection method (expectation method) is based on calculating the expected traffic.

**k-NN flow prediction** An example of this technique is [2], where the algorithm k-NN is used for short-term traffic flow prediction. The basic idea is to use historical data to predict the flow of the next 5 minutes as follows:

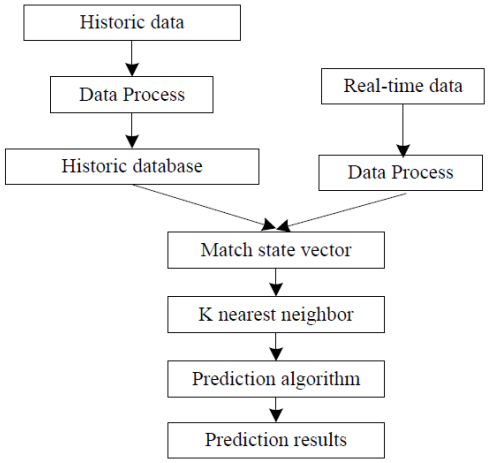


**Figure 2.2:** Scheme of the k-NN method used to predict the next 5 minutes using historical data. Source: [2]

The method is applied on urban expressways in Shanghai where they get the real-time flow for the last instant of time, build a vector of traffic flow, and compare it with the historical data. After testing with different number of neighbors (k) and vector lengths (q), they find out that using k=18 and q=4, they could reach an accuracy of 90% and a MAPE[3] of less than 10%.

**CNN image based prediction** A different approach to predict road network speed can be found in [3]. They took profit from the potential of Convolutional neural network (CNN) algorithm. CNN is considered a deep learning algorithm as

[2]FDP: Flow Distribution Probability
[3]MAPE: Mean Absolute Percent Error

it uses a complex convolutional function. It is mainly used in artificial vision and natural language processing.

Their idea is to transform the traffic speed values into an image using a matrix-based method. Therefore, for each instant of time, the network has a different image. As illustrated at figure 2.3, each column of the matrix represents an instant of time, while the rows represent the road sections. Finally, each value of the matrix corresponds to the traffic speed at its section $Q$ and time $N$. The process is illustrated by them on the following figure:

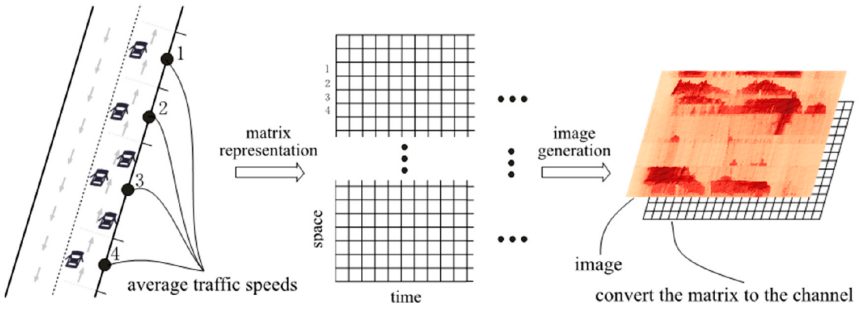


**Figure 2.3:** Methodology followed to transform speed recordings to image. Source: [3]

The method is implemented in two networks that group just one-way roads with different complexities. The technique is tested making predictions for 10 and 20 minutes using 30 and 40 minutes of previous speed recordings. Finally, the system developed is compared to other methods: OLS, k-NN, ANN, RF, SAE, RNN and LSTM NNs and it is found that CNN outperforms all of them with an average accuracy promotion of 42.91%. Even though, they also mention that the main drawback of the method is the training time.

# 3

# Data and data processing

In this chapter you can find all the information related to the original data sets: speed data set, and reported incidents data set. Furthermore, the cleaning and processing of these data sets, as well as the delay calculation are also explained.

## 3.1 Data source: HERE

The data used in this master thesis was obtained from HERE [25], a major online real-time traffic information provider. HERE provides real-time information of more than 83 countries such as traffic speed, incident and accident information, real-time weather and more. The data is obtained using an open application programming interface (API). In this case, even that this master thesis only study Helsinki, the data was collected from 45 different cities. The sampling was done every 5 minutes during 1 year for more than 300 thousand road segments.

The validation of the data is done previously in [34]. Their work is focused on describing and analyzing the traffic data obtained from 45 cities. To validate the data, they use traffic sensor data collected by the Swedish Transport Administration (STA). They compare some road segments with high data availability and find out that HERE data tend to match with the real data collected by STA. Finally, they conclude that the data source can be used for traffic research proposes.

## 3.2 Input data-sets

Despite the data available is composed of 45 cities, this study is only focused on the city of Helsinki. The reason to chose Helsinki is that it is the city that has the best incident reporting for 2018, and which data set size is still reasonable to handle with an common computer. The Helsinki speed data-set has a size of 13 GB with more than 231 million samples from 2.158 different road segments. The network segmentation is partitioned using TMC location reference. The file contains 7 columns, but only the following 6 are relevant:

- **Time:** The time column contains the exact time when the measurement was taken.
- **Segment name:** Name of the segment in TMC code.
- **Confidence:** It is a confidence indicator of the speed measurement that in-

dicates how reliable the sample is. The indicator is calculated taking into account the number of observations used for determine the speed value, and the variance of those observations.

- **Free flow speed:** The speed of the segment when there is no traffic.
- **Speed cap:** The speed caped with the road speed limit.
- **Speed uncap:** The speed registered without cap.

**Segment location data-set** The Helsinki location file has a size of 2.209 KB and contains the segments names with its associated GPS coordinates. The file has 4 columns with the following information:

- **Road name:** Real name of the road.
- **Segment name:** Name of the segment in TMC code.
- **GPS coordinates:** It is a list that contains many pairs of GPS points. Each pair of GPS points represents a part of the segment.

**Incident reported data-set** The Helsinki incidents file has a size of 6.284 KB with 32.505 samples. The file contains 11 columns, but only the following 4 are relevant:

- **Time:** Time when the incident is reported.
- **Start time:** Time when the incident started.
- **End time:** Time when the incident ended.
- **Segment name:** Segment where the incident is reported.

## 3.3 Cleaning and pre-processing

Three different data sets are the start point of the analysis, but first they need to be cleaned and processed. The following scheme illustrates what is expected from each one:

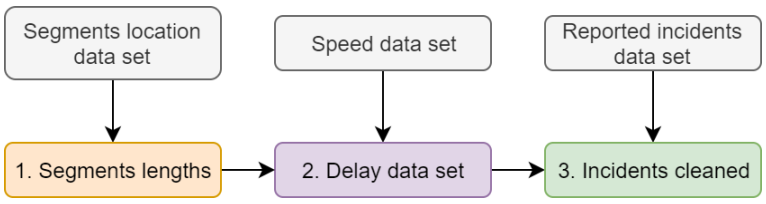


**Figure 3.1:** Cleaning and pre-processing scheme followed in this thesis. The arrows indicates the data sets used to obtain the next data set.

In grey, the original data sets. In orange, the first data set obtained. In purple, the delay data set obtained using the speed and segment lengths. Eventually, in green the final incidents, obtained from the reported incidents data set, and cleaned analyzing its impact on delay.

**Segment length calculation** The length of each segment is a sum of the distance between the pair of GPS points. This distance was calculated using the 'harvesine' formula for each pair of points:

$$a = sin^2\left(\frac{\triangle\phi}{2}\right) + cos(\phi 1) \cdot sin^2\left(\frac{\triangle\lambda}{2}\right) \tag{3.1}$$

$$c = 2 \cdot atan2(\sqrt{a},\sqrt{1-a}) \tag{3.2}$$

$$Distance = R \cdot c \tag{3.3}$$

Where $\phi$ is the latitude, $\lambda$ is the longitude, and R is the radium of the earth.

**Speed data-set cleaning** Relevant road segments in the city are chosen to work with fluently as Helsinki speed file is too large. To decide which road segments are relevant, the incident file is analyzed, and it is found out that only 85 segments have incidents reported after cleaning. As ground truth data is needed to test the methods performance, it is decided to work only with those 85 segments instead of all the 2.158. After cleaning the segments, the speed file reduce its size from 13 GB to 0.6 GB.

**Delay calculation** The delay for each sample of the cleaned speed data-set is calculated using the earlier introduced formula 2.3.

**Reported incidents data set cleaning** The reported incidents data set has 32.505 rows, but not each of them belong to a single incident. To clean the file only the last end hour recorded is kept as all of them have many consecutive end hours recorded. The result is 445 incidents registered in 1 year belonging to 124 segments in the Helsinki area.

Afterwards, an exploration of the incidents is done to validate if they are well reported. After the exploration, it is seen that not all the incidents are well reported, as it is exposed at figure 3.2. This figure shows the delay shape for an incident compared with the average delay shape. Additionally, the start and end hour for the incident is also indicated.

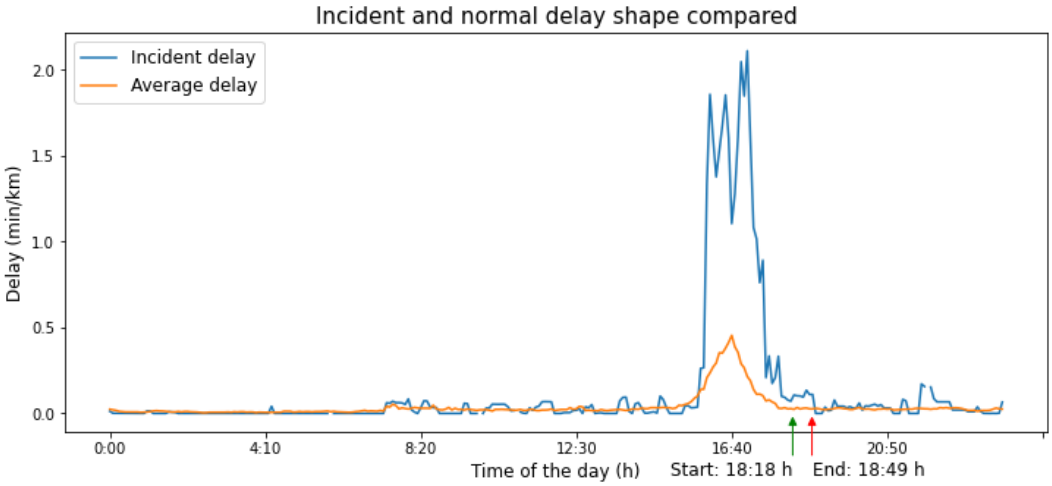


**Figure 3.2:** Day delay shape for an accident day compared with the average delay shape.

In order to clean the incidents, the incidents severity is explored plotting the distribution of the maximum delay value reached during each incident. Additionally the percentile of each maximum delay is calculated based on the samples registered at the same segment, at the same time and during the same week-group. That additional histogram allows us to analyze if the delays reached registered during each incidents are normal or not.

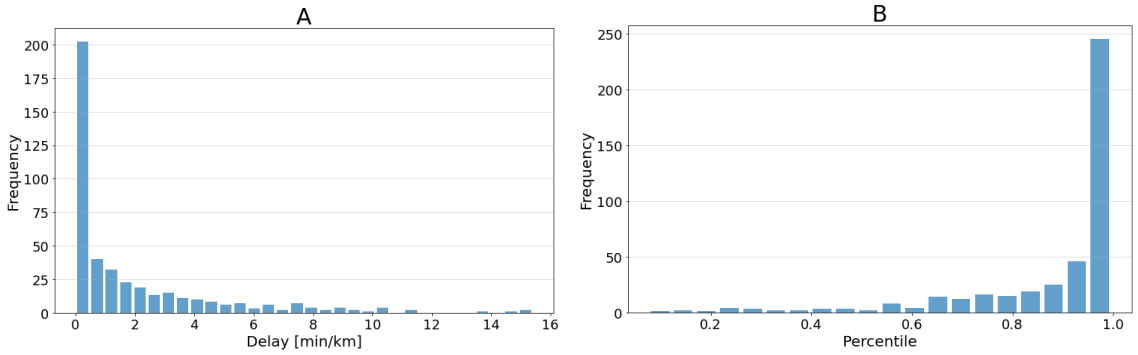


**Figure 3.3:** A: Distribution of the maximum delay reached during each incident. B: Distribution of the percentile of the maximum delay reached of each accident

Finally, after evaluating both plots in figure 3.3, it is decided to delete the minor incidents by applying two different layers of filtering:

- Delete the incidents with a maximum delay lower than 0,5 min/km.
- Delete the incidents with maximum delay percentile lower than 90%.

The first filter is applied because the accidents with a low delay are not that relevant for the study are hard to detect. The second filter is applied to guarantee that the delays registered during the incident are abnormal, and in consequence, that the incident is relevant. After the cleaning, only 185 incidents are kept from 85 different

segments.

# 4

# Methodology

This chapter is focused on explaining the methodology followed, and it is divided in six sections. The first section includes a brief description of the tools used to develop the thesis. The second section explains the time resolution chosen for the data and used during the whole master thesis. The third part exposes the process followed to find recurrent congestion patterns. In the fourth section, the NRC detection methods tested are explained in detail. The fifth section describe the process followed to analyse the NRC episodes detected. Ultimately, in the sixth part the strategy used to calculate the cost of the congestion is explained.

## 4.1 Research environment

The thesis is developed using Python programming language and its environment [43]. The main libraries employed are:

- **NumPy:** It allows users to operate mathematical functions to large multi-dimensional arrays and matrices [44].
- **Pandas:** It allows users to manipulate and analyze data using numerical tables and time series [45].
- **Daskdataframe:** Similar to pandas, but it also allows parallel computing to manage large amount of data [46].
- **Multiprocessing:** It allows generic parallel computing [47].
- **Scikit learn:** It allows users to apply machine learning algorithms to their analysis [48].
- **GeoPandas:** It allows users to work with geospatial data through pandas library [49].

## 4.2 Time resolution

The data set used is composed by samples recorded every 5 minutes. That is, it has a time resolution of 5 minutes. Nevertheless, the time resolution used at this master thesis has been 10 minutes due to two main reasons: not all the segments used for the study have measurements every 5 minutes, and working with a 10-minute window reduced data-set size, and in consequence, the time that the computer needs

to process the data. The last point is a key aspect to smooth the analysis, as many processes are executed to treat the data, study the methods, and analyse the results.

The procedure followed to change the time window resolution from 5 to 10 minutes is:

1. Divide the time into 10 minutes slots for each day [0:00, 0:10, ..., 23:50] resulting in 144 slots per day.
2. Calculate the delay of each 10 minute slot as the average of the delays sampled during each of those slots.

Therefore, each segment have 37.440 time slots (144 *per day* · 260 *working days*).

## 4.3 Recurring patterns

Before detecting NRC episodes, a preliminary analysis about the data given is carried out. The analysis aims to find the regular delay behaviour that is expected to improve NRC detection. During the analysis, two main paths are explored: finding time patterns and similarities between segments. Comparing and classifying is required in order to carry out the proposed analysis. These processes cannot be done directly as the data has to be standardized beforehand. The standardization and clustering techniques used are explained below.

**Standardization** The standardization have to be applied due to the different segment delay scale. Otherwise some similar delay shapes would be considered different, even if they share similar patterns as showed at figure 4.1. In consequence, the standard is applied by removing the mean and scaling to unit variance [48].

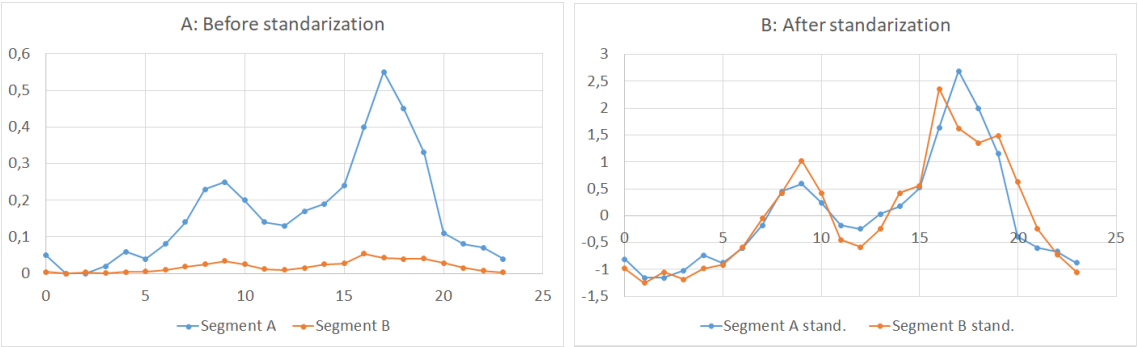


**Figure 4.1:** A: Average delay values per hour of the day for two different segments. B: Same average delay values, but standardized.

**Clustering** To classify the segments into groups the algorithm K-means is chosen. K-means separates samples into a predefined number of clusters with equal variance, minimizing the sum of squared error (SSE) [50]. Choosing the number of clusters is not trivial and the selection is usually done by looking at two measures:

- **Sum of squared error (SSE):** SSE is calculated as the sum of the squared

distance between centroid and each member of the cluster. The lower, the better.

- **Silhouette Coefficient:** It provides information regarding how well an object has been classified. It is ranged between -1 and 1, and the closest to 1, the better. If it is closer to 0 means that the clusters are overlapping between them. More information can be found in study [51].

A good clustering should have a Silhouette coefficient as closer to 1, with the lowest SSE possible.

## 4.4 Non-Recurring Congestion detection

In this section, three different NRC congestion detection methods are explained: one statistical-based method called dynamic threshold and two machine learning based methods called k-NN delay prediction and k-NN local outliers, respectively.

### 4.4.1 Considerations

Some basic aspects observed during the delay exploration, are taken into consideration for all three of the NRC detection methods tested (section 5.1). These rules can be summarized as follows:

- **Segment separation:** Each segment has different delay behaviors. Therefore, each segment is taken into consideration individually when applying all the methods.
- **Week-group separation:** The delay of each segment can be clustered into two different week-groups: weekdays and weekends. Due to there being more data available for the first group, only weekdays are used for this study.
- **Time of day:** The delay has a repeated pattern each 24 hours. In consequence, the time of day is taken into consideration in all the methods tested.
- **Time between two abnormal measurements:** To consider NRC episodes, at least two measurements have to be marked as abnormal in less than x minutes. The x values explored are 10, 20 and 30 minutes. That requirement, combined with the 10 minute time window, limit the methods to detect only the incidents that last 20 minutes or more.

### 4.4.2 Delay dynamic threshold

The first method tested to detect NRC episodes is focused on statistics as it is the most common technique find on literature as explained in section 2.4. Following the findings at [22], a dynamic threshold is proposed to detect NRC episodes. Delay dynamic threshold mark as possible non recurring congestion the values that exceed a threshold during a certain time. That threshold is calculated by adapting the

formula 2.4:

$$Threshold_{ij} = MD_{ij} + n \cdot IQR_{ij} \tag{4.1}$$

Where $MD$ is the median delay, IQR is the inter quantile range, $i$ is the time, $j$ the segment, and the parameter $n$ a number that need to be optimized. Figure 4.2 expose a sample of how dynamic threshold works.

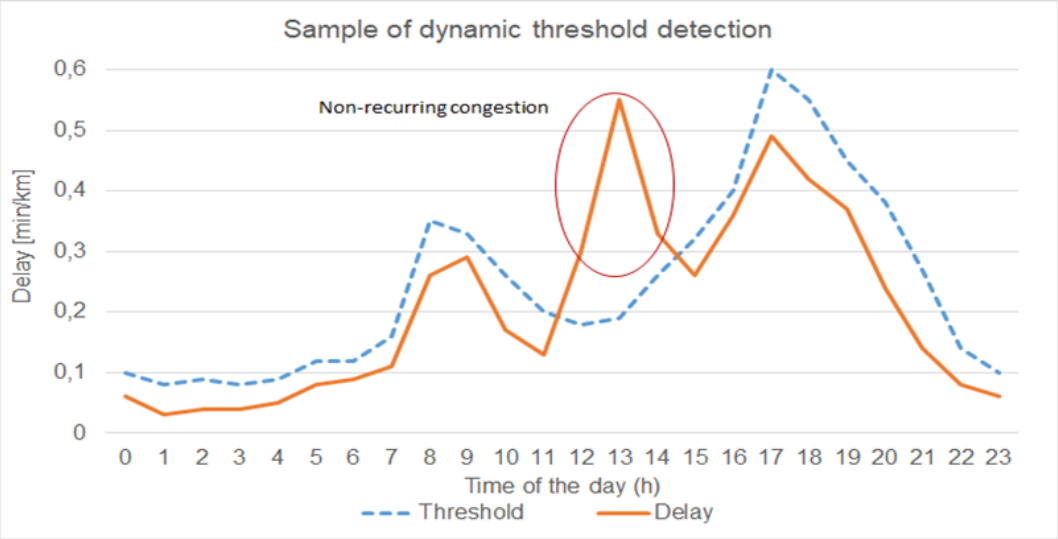


**Figure 4.2:** Example of how dynamic threshold detection works.

In consequence, the dynamic threshold is composed of different thresholds calculated for each different segment $i$ and at each time $j$ taking into account only weekdays. As it is worked with a 10-minute time resolution (6 slots per hour), 12.240 different thresholds are found for each $n$ value tested:

$$No^{1}\ of\ thresholds = 85 \cdot 24 \cdot 6 = \ 12.240 \tag{4.2}$$

Where 85 is the number of segments studied, 24 the hours in a day, and 6 the 10 minutes slots that 1 hour has.

The formula 4.1 can not be applied directly, because in case of a fat tail in the delay distribution, many observations will exceed the threshold defined, and would be considered as incidents without being really incidents [22]. For that reason, the skewness is studied using medcouple values. The medcouple value measures the skewness of an univariate distribution and it is limited between -1 and 1 [38]. Distributions that are skewed to the right have a positive medcouple, while distributions skewed to the left have a negative medcouple. Normal distributions have medcouple values close to 0. Additionally, as mentioned in section 4.4.1, a minimum of 2 abnormal measurements needs to be detected in less than 10, 20 or 30 minutes to consider an NRC episode.

[1]Number

### 4.4.3 Machine learning methods

As part of the literature work with machine learning techniques, two of these techniques are explored as methods to detect non recurring congestion: predict the typical delay and later compare it with the real measurements, and use outlier techniques to detect abnormal measurements. In this master thesis, both of the methods are applied using the algorithm k-Nearest Neighbors (k-NN), therefore the following section explains briefly how it works.

#### k-NN algorithm

The k-NN algorithm is a simple way to find a predefined number (k) of dataset samples, in this case historical values, that are closest in distance to the new point. Later, the closest samples can be used to predict the value from the new point. The historical data is usually named as training data, and each sample has numerical attributes that are used to calculate the distance [2]. In both cases the distance used is the Euclidean. For a two dimension point it is calculated as follows:

$$d(p,q) = \sqrt{(q_x - p_x)^2 + (q_y - p_y)^2} \tag{4.3}$$

Moreover, to evaluate the performance when using k-NN to predict, some additional metrics are commonly used:

- **Mean Absolute error (MAE):** It is calculated as the mean of all the absolute errors between the predicted and the real value. It give an idea of how precis is the prediction. The closer to 0, the better.

$$MAE(y, y_{pred}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} |y_i - y_{pred_i}| \tag{4.4}$$

- **R2 score:** It is the coefficient of determination and it measure the precision for new predictions using the proportion of explained variance [52]. Therefore, r2 measures how well the model can predict the next values. The closer to 1, the better.

#### NRC detection based on delay prediction

This method try to predict the expected delay by taking into account 20 to 60 minutes of previous delay values (2 to 6 values as the time resolution is 10 minutes). Taking as an example [2], k-NN is the algorithm chosen to predict the delay. The method can be described as follows:

1. Converting each sample into a point of 2-6 attributes (previous delay).
2. Finding the k closest points for each sample using k-NN.
3. Predict the delay value for the next 10 minutes of each sample.
4. Compare the predicted delay value with the real one, and decide if the real value is abnormal.

To evaluate point 4, a threshold is set using the natural logarithm ($ln$) of the division between the real delay and the predicted delay. The division quantifies how much bigger is the real value in comparison to the predicted one:

$$Difference = \ln \frac{Real\ delay}{Predicted\ delay} \tag{4.5}$$

The reason to apply $ln$ is because the division can highlight slight variations when the delay values are close to 0. Small differences can result in a high division number, whereas big differences can be hidden. The $ln$ smooths this effect as explained in table 4.1.

**Table 4.1:** Sample of natural logarithm smooth effect. A: Slight difference between delays that are close to 0 min/km. B: Big difference between large delays.

| | **Real delay [min/km]** | **Predicted delay [min/km]** | **Difference [min/km]** | **Real delay/Pred. delay (div)** | **ln(div)** |
|---|---|---|---|---|---|
| **A** | 0,1 | 0,04 | 0,06 | 2,5 | 0,92 |
| **B** | 7 | 5 | 2 | 1,4 | 0,34 |

As well as on the dynamic threshold, a minimum of 2 abnormal measurements needs to be detected in less than 10, 20 or 30 minutes to consider an NRC episode.

### k-NN Local outliers

Finally, a second method based on machine learning is tested. The technique called k-NN Local Outliers is very similar to the Local Outlier Factor (LOF) technique that is suggested in [14].

The algorithm k-NN Local Outliers is based on detecting the measurements that are isolated from the others. To detect them, the algorithm k-NN calculates the distance between the $k$ closest measurements and computes the average of the $k$ distances obtained. If the mean distance is larger than a threshold, it is considered as isolated from the other measurements. This process is repeated for each measurement of each segment. To apply this method, each sample is converted into a two dimensional point $(x, y)$ with the following information:

- $x$= The time of day transformed using the sine function to avoid a discontinuity between 0:00h and 23:59h, and in the end, normalized between 0 and 1. See figure 4.3.
- $y$= Delay normalized also between 0 and 1.

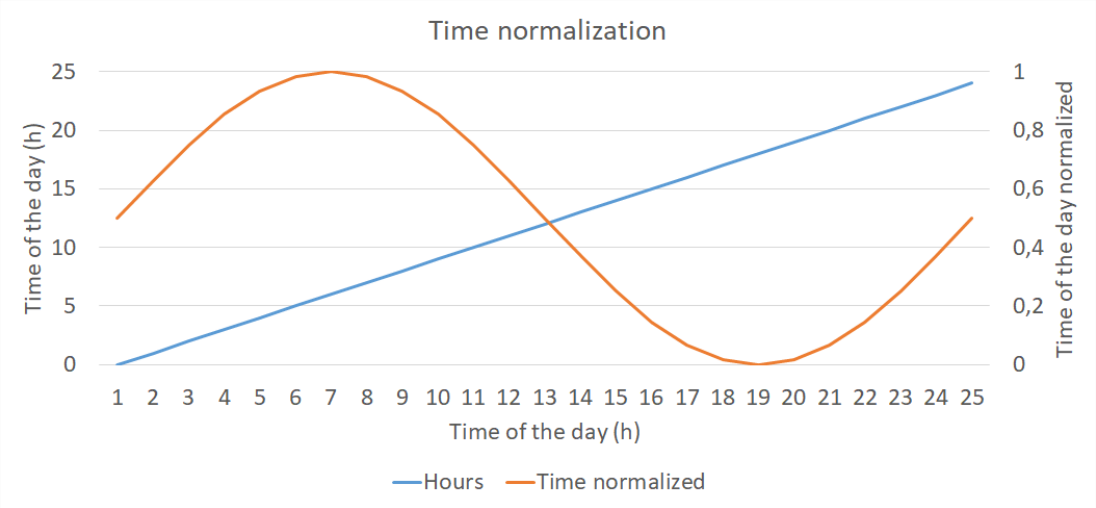


**Figure 4.3:** Discontinuity of the time in hours (blue), and the normalization applied to solve it (orange).

In the following example plot, each point represents a delay measurement for one segment at three different weekdays. The abnormal delay measurements that the method would detect are circled as they are isolated from the others.

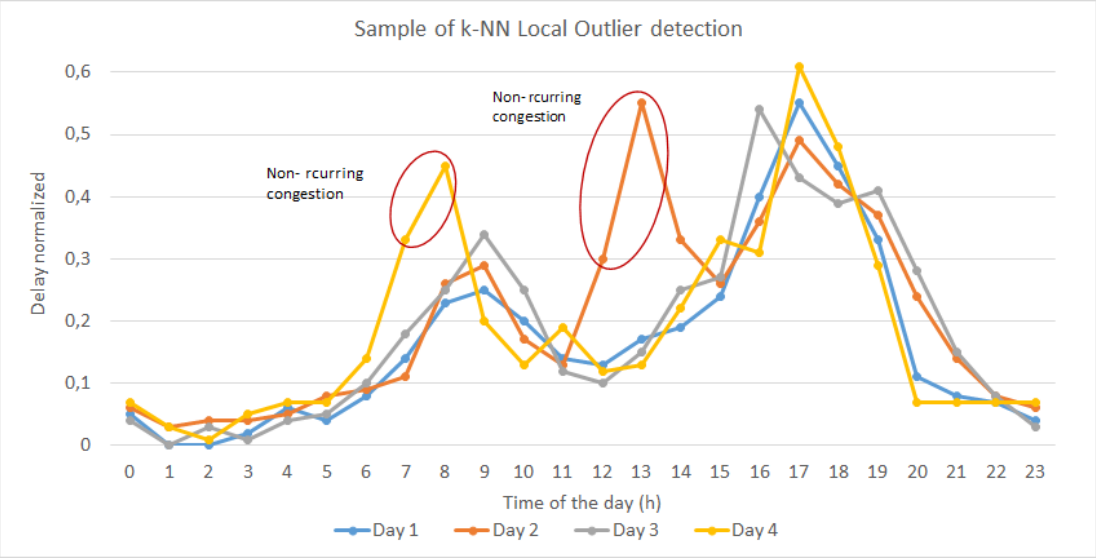


**Figure 4.4:** Sample of k-NN Local Outlier method incident detection.

This method has three different parameters that need to be optimized: number of neighbors, minimum time between two isolated measures, and minimum average distance to consider the delay measurement as isolated. That minimum average distance should not be the same for each segment, as each segment may have different measurements sparseness. For that reason, it is decided to create an average distance threshold calculated for each different segment:

$$A.D.\,Threshold_{seg.\,i} = \frac{Max(MDs_{seg.\,i})}{"d"} \tag{4.6}$$

Where the numerator $MDs$ is the maximum average distance obtained of each segment, and $d$ is a number that has to be optimized. Eventually, as the previous methods, a minimum of 2 abnormal measurements needs to be detected in less than 10, 20 or 30 minutes to consider an NRC episode.

### 4.4.4 Validations

In order to evaluate the methods, some uniformed metrics have to be predefined. As the only ground-truth data available is the incident data-set, the following two main metrics are examined:

1. Number of reported incidents (registered in the incident file) detected by the method. During the rest of the thesis, this metric is named as (1).
2. Number of extra incidents detected by the method at day scale. This number does not take into account the registered ones (1). During the rest of the thesis, this metric is named as (2).

The first metric aim to analyze if the method succeed in detecting the reported incidents. Whereas the second metric evaluate the number incidents detected by the method over the total possible incidents, taking into account the assumption that each segment can only register one incident per day (day scale). A high value for the second metric mean that the method is not able to distinguish NRC episodes from recurring traffic. The following table illustrates how metrics (1) and (2) are calculated:

**Table 4.2:** Sample of the table used to present the validation metrics results of each method.

| **Number of reported incidents detected** | **Percentage of reported incidents detected** | **Number of extra incidents detected** | **Percentage of extra incidents detected** |
|---|---|---|---|
| (1) | (1)/185 | (2) | (2)/22100 |

The percentages are calculated over the total number of cleaned incidents registered (185), and total number of possible incidents to be detected at day scale (260 week-days $\cdot$ 85 segments = 22100). Notice that an incident is considered as detected if the method detected an NRC episode during a period between the start and end registered in the incidents file.

#### Optimization

The optimization of the methods rely on reaching the maximum number of reported incidents detected with the lowest number of extra incidents detected. This criteria is set for two main reasons: there is no ground truth data about the real number of incidents occurred, and to ensure that the incidents detected had a real impact on the delay. Otherwise, if the method reach high values of reported incidents, but also high values of extra incidents, the method would have no criteria when distinguishing incidents from normal congestion.

When optimizing the methods a problem emerged: if the requirements for considering an NRC episode are low enough, the methods always reach the 100% of reported incidents detected, but at the same time a high number of extra incidents detected. For that reason, it is decided to set a 60% of reported incidents detected as it is the reasonable accuracy expected for this kind of studies [53].

## 4.5 Incident analysis

Finally, the incidents detected by the method chosen are analyzed. Therefore, as it can be seen in figure 4.5 from all the cleaned incidents (185), only the ones detected by the method are included in the analysis (blue and yellow).

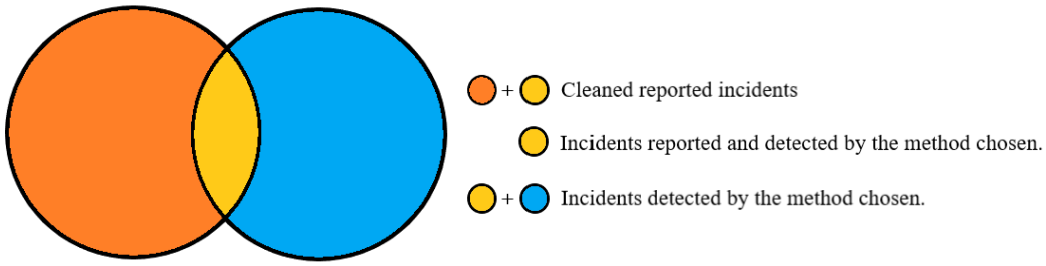


**Figure 4.5:** Scheme of the incidents selected for the analysis.

The analysis is focused on studying the characteristics of the incidents and on finding time and location patterns. The characteristics studied are the duration and severity. The duration is calculated using the start and end time of the incident. To evaluate the severity, the maximum delay value registered during each incident is taken as an indicator.

The time pattern study is focused on the frequency of the incidents in time. Therefore the number of incidents per hour, weekday and month, are studied. Also, the relation between duration, severity and time of day is studied. During the space analysis the distribution of the incidents is examined to visualize the most troubled road segments of Helsinki.

## 4.6 Congestion costs

In order to calculate the cost of the congestion in Helsinki, several assumptions are done. The main problem when calculating the cost using delay is the lack of flow information. For this reason, an external source of flow data is used. The flow data is obtained from Helsinki Region Infoshare [54]. The flow data corresponds to 2018 average traffic volumes of some roads. The estimation is done counting volumes from Monday to Thursday in September and October of each year. More information about the data set can be found in [55].

Total delay per passenger is chosen as a metric to calculate the cost of congestion. The total delay is calculated taking into account: the average delay weighted with

the hourly car flow, and the average distance travelled. Therefore the total delay per passenger is calculated as:

$$Total\ delay = Avg.D. \left[\frac{min}{km}\right] \cdot d \left[\frac{km}{passenger}\right] \tag{4.7}$$

Where $Avg.D$ is the average delay, and $d$ is the average car distance traveled in one year per passenger. In consequence, the results for this section are calculated assuming:

- The total delay per year is calculated taking into account only weekdays (Monday to Friday), 260 days per year.
- The travelled distance per passenger over a year is 13.600 km [56].

### 4.6.1 Extra costs

To calculate the extra costs, the following assumptions are taken into account:

- All vehicles in Finland use petrol as fuel [57].
- All car engines are idling during the lost time.
- The mean idling consumption for a car is 1,89 l/h [58].
- The average price for petrol in Finland in 2018 was 1,52 €/l [59].
- 1 liter of petrol produces 2,3 kg of CO2 when burned [60].

The equations used to calculate the extra costs are:

$$Extra\ fuel = Total\ delay\ [h] \cdot 1,89\ \left[\frac{l}{h}\right] \tag{4.8}$$

$$Extra\ cost = Extra\ fuel\ [l]\ \cdot 1,52\ \left[\frac{euros}{l}\right] \tag{4.9}$$

$$Extra\ CO_2 = Extra\ fuel\ [l] \cdot 2,3\ \left[\frac{CO_2\ kg}{l}\right] \tag{4.10}$$

# 5

# Results and discussion

Chapter 5 includes the results obtained during the thesis development, and the discussion extracted from these results. The chapter is structured in five different parts. The first part describes the recurrent congestion patterns found in Helsinki. The second part is focused on validating the NRC episode detection methods. The third part describes the results obtained when analyzing the NRC episodes detected by the method chosen. In the fourth part of the results an application for evaluating the cost of congestion is presented . Eventually, the last part exposes the data limitations and how they have influenced the results.

## 5.1 Recurrent congestion

Although finding recurrent congestion is not an aim for the thesis, knowing the recurring congestion patterns is important to understand the data used, to define a good NRC detection method, to perform the analysis of the NRC episodes, and to understand the analysis results.

### 5.1.1 Temporal patterns

The data set given cover the year 2018 allowing us to explore daily and monthly delay patterns. For that reason, two main charts are plotted at figure 5.1: monthly mean delay and daily mean delay.

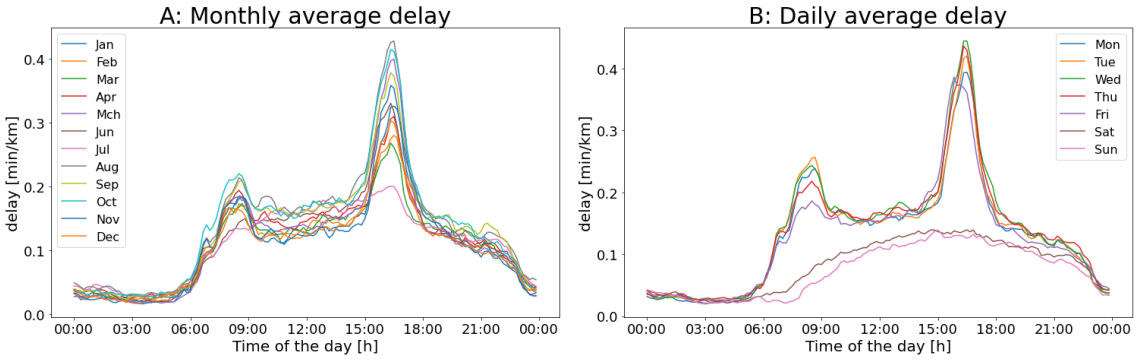


**Figure 5.1:** A: monthly average delay per hour of the day. B: daily average delay per hour of the day.

The conclusions extracted from figure 5.1 are summarized in the following points:

1. Both plots show that every day, except from Saturdays and Sundays, the delay shape has two marked peaks. These two peaks are expected to be caused by commuting, and delimits morning rush hour between 7h and 10h, and afternoon rush hour between 15h and 18h. It is also observed that the afternoon peak reaches higher values of delay, in comparison with the morning peak.
2. Plot A exposes that the month of the year do not has a clear impact on the delay behaviour, therefore the data can be grouped in one. In spite of this, July has a clear decrease in delay during afternoon due to vacation period in Helsinki.
3. Plot B shows that there are two marked clusters which delay behave different: Monday to Friday, and Saturday to Sunday. As the first group is considerable bigger, only Monday to Friday are considered for this thesis.

The clusters proposed at points 2 and 3 are confirmed using the algorithm k-means as detailed in figures A.1 and A.2, which can be found in the appendix.

### 5.1.2 Location patterns

The difference between segments is another main field that needed to be explored. The objective is to study if there is any kind of groups of segments that behave in a similar way. To expose hidden similarities, two different charts are presented: average delay of each segment, and standardized average delay of each segment. The standardized chart allows a better comparison between all segments to find hidden similar behaviours. Notice that each line corresponds to the average delay of one segment.

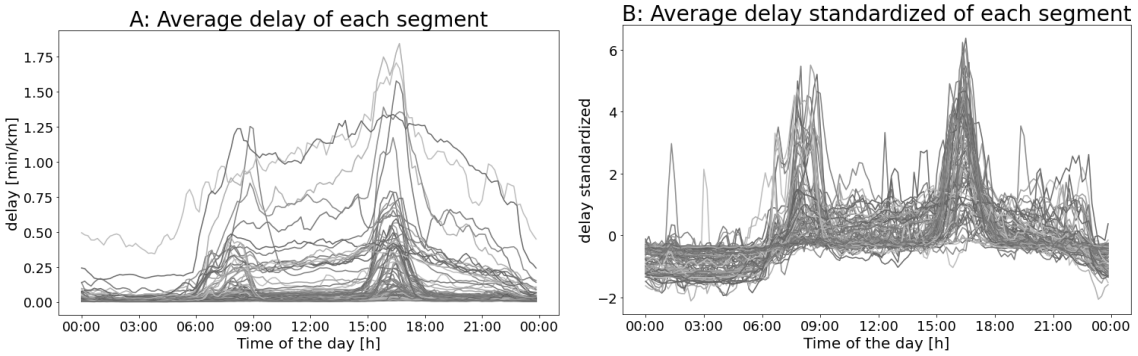


**Figure 5.2:** Each line represents an average delay shape for one segment. A: The raw average delay for each segment. B: The standardized average delay for each segment.

As displayed on figures 5.2, there are too many segments to obtain conclusions. For that reason, k-means is applied to the standardized data with the following results:

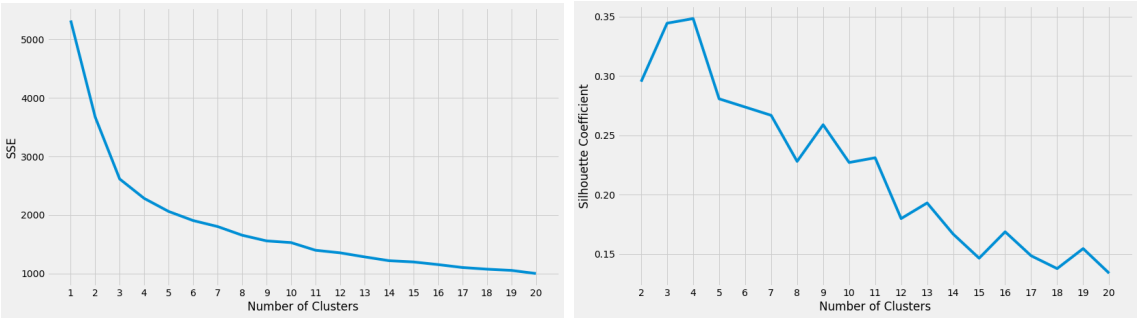


**Figure 5.3:** SSE and Silhouette coefficient obtained when clustering the segments.

The Silhouette coefficient and SEE obtain poor results in all the different clusters compared with the ones obtained at A.2. Therefore each segment is considered individually when applying the NRC detection methods selected.

## 5.2 NRC detection validation

In this section the performance results of the three different methods tested are presented. The performance is evaluated using the metrics described in section 4.4.4. In the end, the best method with the best metrics is selected to continue with the incident analysis.

### 5.2.1 Delay dynamic threshold

Before applying the dynamic threshold, a skewness study is done to ensure a correct performance of the method. As mentioned in 4.4.2, the medcouple is used to measure the skewness [38]. The medcouple is calculated for each segment at each time individually as well as the threshold. The average medcouple value obtained is 0,55, therefore the data is skewed to the right. To address that problem, two different options are evaluated: applying the square root or applying the natural logarithm[1] (LN). The medcouple average values obtained after the transformations are:

**Table 5.1:** Medcouple values for two different time periods. Medcouple 1: average of the medcouple values of the entire day. Medcouple 2: average of the medcouple values obtained between 6h to 22h.

| | **Raw Delay** | **SQRT(delay)** | **LN(delay)** |
|---|---|---|---|
| **Medcouple 1** | 0,554 | 0,396 | 0,206 |
| **Medcouple 2** | 0,419 | 0,230 | 0,010 |

The reason to calculate two different medcouple values is that during the night (22h-6h) the delay values are usually close to 0 min/km as seen in figure 5.1, and

[1]To avoid undefined numbers when applying the natural logarithm, a 0,001 min/km of delay is added to all samples that had 0 min/km as value.

consequently highly skewed, which distort the results of the mean medcouple. Furthermore, this period has a low relevance when studying NRC as few incidents are registered during that period. For that reason only the period between 6 and 22h is taken into account in the results.

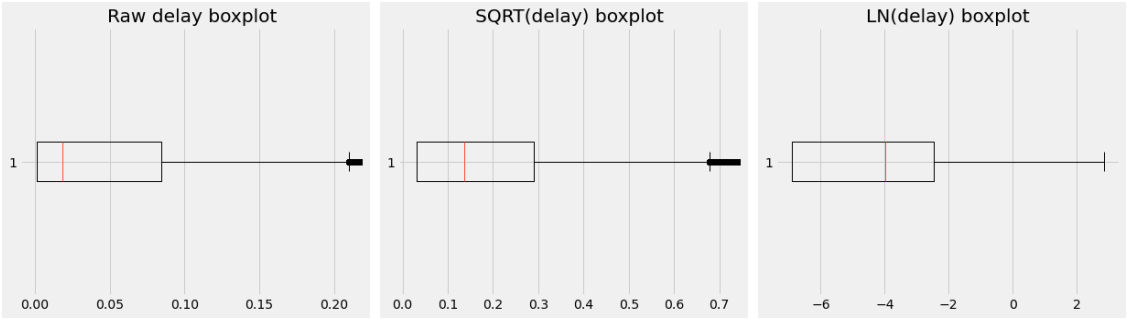


**Figure 5.4:** Box-plots before and after applying the square root and e-log to delay values.

In figure 5.4 the skewness of the data can be observed before and after applying the square root and LN. Notice that the raw and SQRT boxplots are zoomed in to particular sections. The whole plots can be seen in A.3 located at the appendix. In consequence, to define the dynamic threshold, three different parameters have to be optimized:

- $n$: Parameter defined in equation 4.1 that is optimized to reach the 60% of reported incidents detected.
- Skew correction: SQRT(delay) or LN(delay).
- Maximum Time (MT): Maximum time between two outlier measurements to consider a NRC episode.

**Table 5.2:** Validation metrics results for dynamic threshold method taking into account only the period between 6h and 22h:

| **Skew correction** | **Number of reported incidents detected** | **%** | **Number of extra incidents detected** | **%** |
|---|---|---|---|---|
| **Raw delay** | 111 | 60% | 2163 | 9,8% |
| **SQRT(delay)** | 111 | 60% | 2150 | 9,7% |
| **LN(delay)** | 111 | 60% | 3807 | 17,2% |

Finally the results are shown in table 5.2. The maximum time (MT) between two suspicious measurements to consider them as a NRC episode is fixed at 10 minutes as it turns out to has no relevant influence in the results. The whole results with different MT values can be seen in table A.1 located in the appendix. The best result obtained is 9,7% of extra incidents detected with 60% of reported incidents detected.

### 5.2.2 k-NN delay prediction

To apply k-NN delay prediction, two main steps have to be taken: predict the delay, and chose the abnormal measurements comparing the real delay values with the predicted ones. Three parameters are optimized during the whole process:

- $k$: Number of neighbors used to predict the delay.
- $d$: Number of previous samples of delay used to predict the next 10 minutes (attributes).
- Threshold value for the natural logarithm ($LN$) of the division used to decide if the measurement is abnormal.
- Maximum time (MT): Maximum time between two abnormal measurements to consider a NRC episode.

Only the results for $d = 4$ are presented. The other results are detailed in tables A.2 and A.3 located in the appendix. Also, the maximum time (MT) between two abnormal measurements is fixed again at 10 minutes due to the high number of incidents detected by the method.

To evaluate the performance only the period between 6h and 22h is considered. The decision is taken because the threshold used (even when applying $LN$) do not perform well with delays close to 0 as shown in table 4.1, leading to report false incidents. This kind of delays are common during night period as seen in figure 5.1. This is also done to ensure an equal comparison with the previous method tested (dynamic threshold).

**Table 5.3:** Validation metrics results for delay prediction method. Parameter $d$ fixed at 4 and taking into account only the aforementioned period between 6h and 22h:

| k | thd | Number of reported incidents detected | % | Number of extra incidents detected | % |
|---|---|---|---|---|---|
| **5** | 0,65 | 111 | 60,00% | 16649 | 75,19% |
| **10** | 0,605 | 111 | 60,00% | 16496 | 74,50% |
| **15** | 0,635 | 111 | 60,00% | 15221 | 68,74% |
| **20** | 0,65 | 111 | 60,00% | 14549 | 65,70% |
| **25** | 0,717 | 111 | 60,00% | 12744 | 57,55% |
| **30** | 0,71 | 111 | 60,00% | 12751 | 57,58% |

The results manifest that the method is unreliable due to the high number of extra incidents detected, which proved that the method do not have any accurate criteria when differentiating normal measurements from abnormal ones. The problem is caused by the high difference between the predicted and real delay. The following figure 5.5 show a sample of the predicted and real delay for a segment during 32 hours with parameter $k$ fixed at 25 and $d$ fixed at 4.

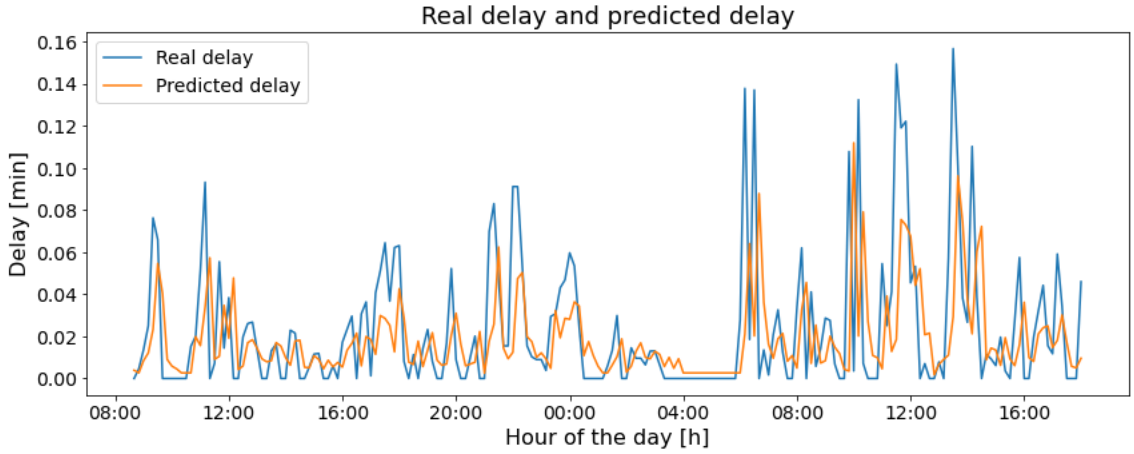


**Figure 5.5:** Sample of the delay predicted by k-NN compared with the real delay. $k = 25$ and $d = 4$.

As seen in figure 5.5, the algorithm tends to underestimate the delay, which causes a lot of false incidents detected. To prove that, the mean absolute error (MAE) and the r2 score are calculated for $k = 25$ and $d = 4$, as that combination has the lowest number of incidents detected:

**Table 5.4:** Prediction metrics for $k = 25$ and $d = 4$.

| | **MAE** | **R2** |
|---|---|---|
| **24 h** | 0,054 | 0,726 |
| **6-22h** | 0,07 | 0,726 |

The results in table 5.4 show a mean absolute error of 0,07 , which is not reasonable taking into account the range of delays values seen in figure 5.5 (0 - 0,16 [min/km]).

### 5.2.3 k-NN-LOC validation

To configure the k-NN Local Outliers method three different parameters needs to be defined and in consequence, optimized:

- $k$: Number of neighbors.
- $dn$: Division number.
- Maximum Time (MT): Maximum time between two abnormal measurements to consider a NRC episode.

To optimize the parameters, three explorations are carried out with the maximum time (MT) fixed at 10, 20 and 30 minutes. For each one of these explorations, different values for $k$ and $dn$ are proposed. After analyzing the results it is seen that the best option is to fix MT at 20 minutes as obtains the lowest number of extra incidents.

**Table 5.5:** Validation metrics results for k-NN Local Outlier method. Parameter MT fixed at 20 minutes:

| k | dn | Number of reported incidents detected | % | Number of extra incidents detected | % |
|---|---|---|---|---|---|
| **40** | **9,64** | 111 | 60,00% | 1005 | 4,54% |
| **45** | **9,25** | 111 | 60,00% | 983 | 4,44% |
| **50** | **8,96** | 111 | 60,00% | 963 | 4,35% |
| **55** | **8,94** | 111 | 60,00% | 990 | 4,47% |
| **60** | **8,95** | 111 | 60,00% | 1022 | 4,62% |

The results for MT fixed at 20 minutes and $k$ between 40 and 60 are exposed in table 5.5. The complete results with other $k$ and MT values can be seen in tables A.5, A.4 and A.6 located in the appendix. The lowest number of extra incidents detected is obtained, as highlighted in green, for $k = 50$ and $dn = 8,96$. Therefore, the total number of incidents detected detected by k-NN Local Outlier in Helsinki during 2018 is 1074.

## 5.3 NRC analysis

The results of this section are reached with the 1074 incidents detected by the method k-NN Local Outlier. This section is structured in three different parts: incident characteristics, temporal patterns, and space analysis.

### 5.3.1 Incident characteristics

The characteristics studied are duration and severity, that are measured as specified in section 4.5. Firstly, the distribution of the duration of the incidents can be seen in figure 5.6.

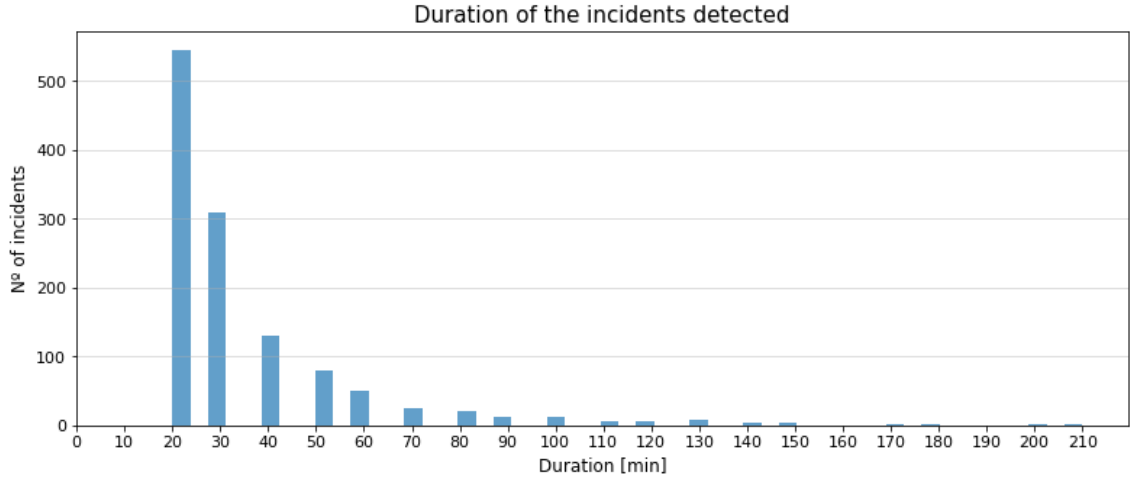


**Figure 5.6:** Number of incidents detected classified per duration for 2018 in Helsinki.

Figure 5.6 manifest that the great part of the incidents detected during 2018 lasted for 40 minutes or less. Notice that the method cannot detect incidents with a duration under 20 minutes as explained in section 4.4.1. Secondly, the severity is studied in figure 5.7, which displays a distribution with the number of incidents per maximum delay value reached during each incident.

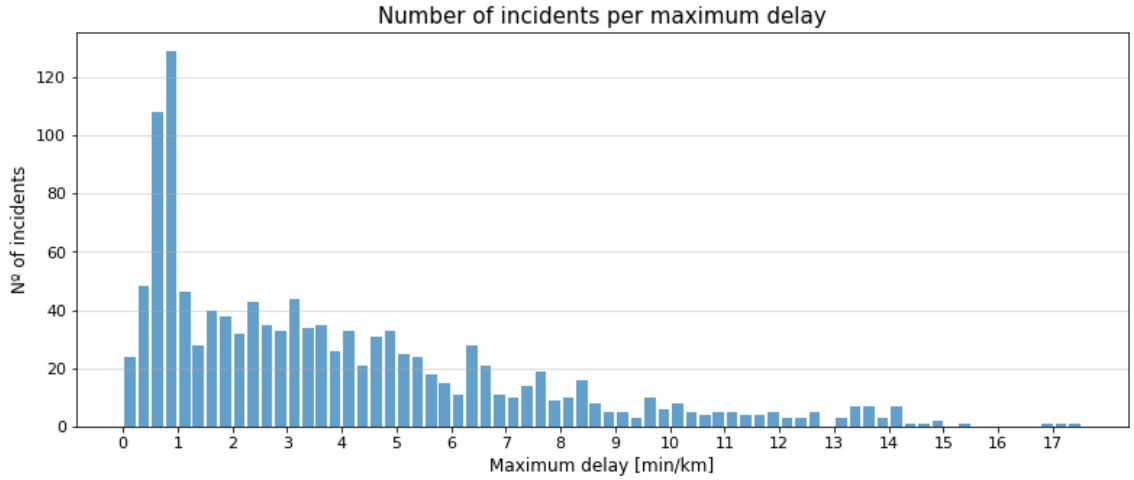


**Figure 5.7:** Incidents classified per maximum delay reached.

Figure 5.7 denotes a peak of incidents with a maximum delay not exceeding 1 min/km. It is also clear that the number of incidents with a maximum delay between 1 to 5 min/km is similar. Additionally, the relation between the duration and the maximum delay is explored on figure 5.8.

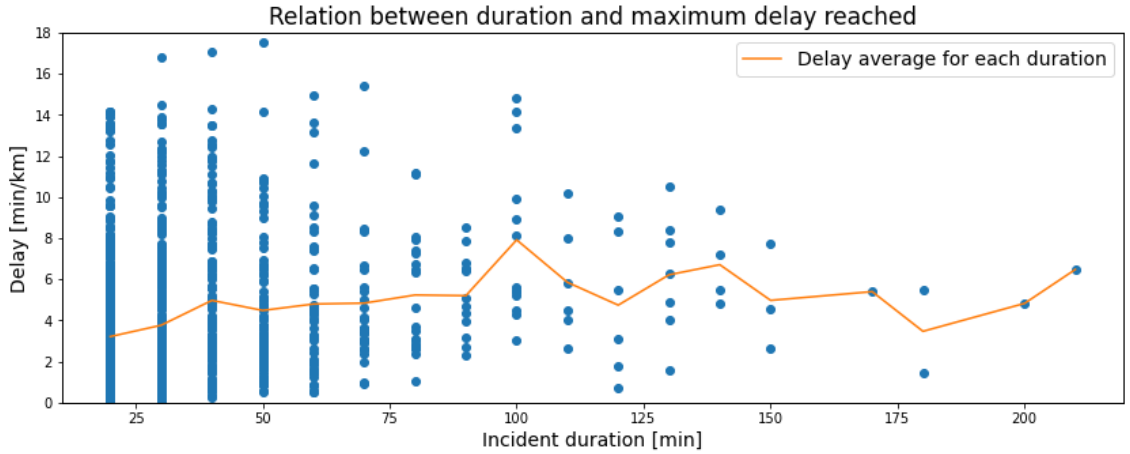


**Figure 5.8:** Relation between the duration of the incidents, and the maximum delay reached.

The figure 5.8 shows that the delay values reached during an incident does not has a clear effect on its duration.

### 5.3.2 Temporal patterns

The temporal patterns studied are based on three different analysis. The first one is focused on finding hourly patterns, the second on daily, and the last on monthly. Hourly patterns are analysed in figure 5.9, which shows the number of incidents detected and the average duration by time of day.

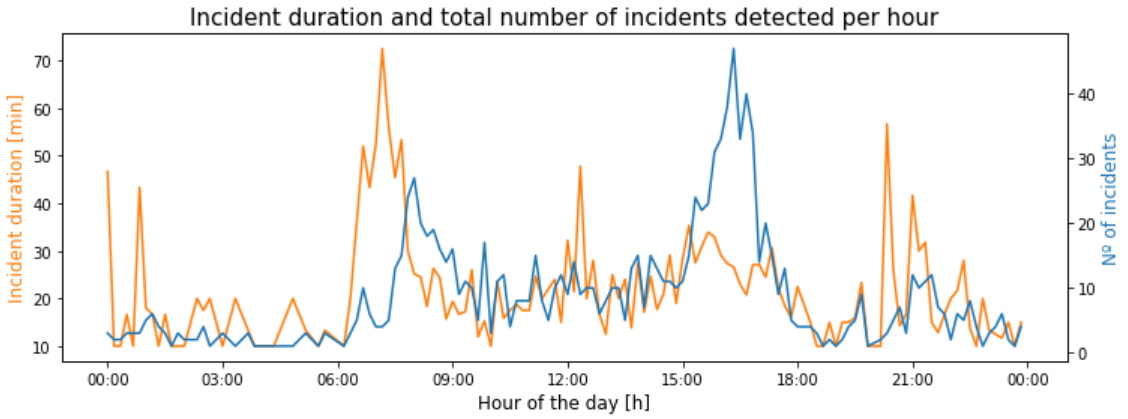


**Figure 5.9:** Incident duration per hour (blue) and total number of incidents detected (orange) according to time of day for the year 2018 in Helsinki.

As expected, the plot displaying number of incidents (blue) has two peaks corresponding to morning and afternoon rush hours. Additionally, the afternoon period concentrated a greater number of incidents than the morning one. On the other hand, the incident duration (orange) manifest that morning incidents tend to have a longer duration, even if most incidents occurred during the evening.

The daily results are summarized at table 5.6, which shows the number of incidents per weekday an its mean duration.

**Table 5.6:** Number of incidents detected per weekday, ,its average duration, and its average maximum delay (A.M.D.):

| | **Monday** | **Tuesday** | **Wednesday** | **Thursday** | **Friday** |
|---|---|---|---|---|---|
| **No of incidents** | 210 | 265 | 245 | 264 | 231 |
| **% of incidents** | 17,3% | 21,8% | 20,2% | 21,7% | 19,0% |
| **Avg. duration [min]** | 29,2 | 35,0 | 41,7 | 33,3 | 34,3 |
| **A. M. D. [min/km]** | 3,5 | 4,2 | 4,0 | 4,0 | 3,6 |

As seen in table 5.6, Tuesday, Wednesday and Thursday are the days with more congestion as have the higher number of incidents detected and the higher maximum delay averages. Moreover, Monday is the day with lower congestion as it has a slight lower number of incidents detected, and the lowest maximum delay average. Additionally, Wednesday has a significant higher average duration of the incidents.

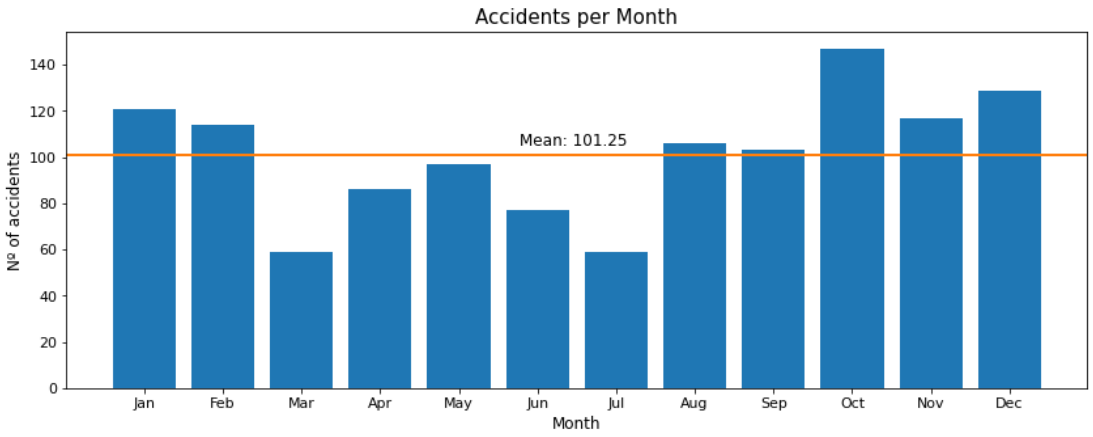


**Figure 5.10:** Incidents detected per month during 2018 in Helsinki.

At monthly scale the results expose that the months with snowy weather had more incidents detected: January, February, October, November and December. Additionally, June and July had a significantly low number of incidents, corresponding to the vacation period. Finally, March had an unexpected low number of incidents, as the 2018 winter vacation period in Finland was in February.

### 5.3.3 Space analysis

The last part of the analysis consist in visualizing the incident distribution through Helsinki. Figure 5.11 shows a heat-map of the number of incidents per segment.

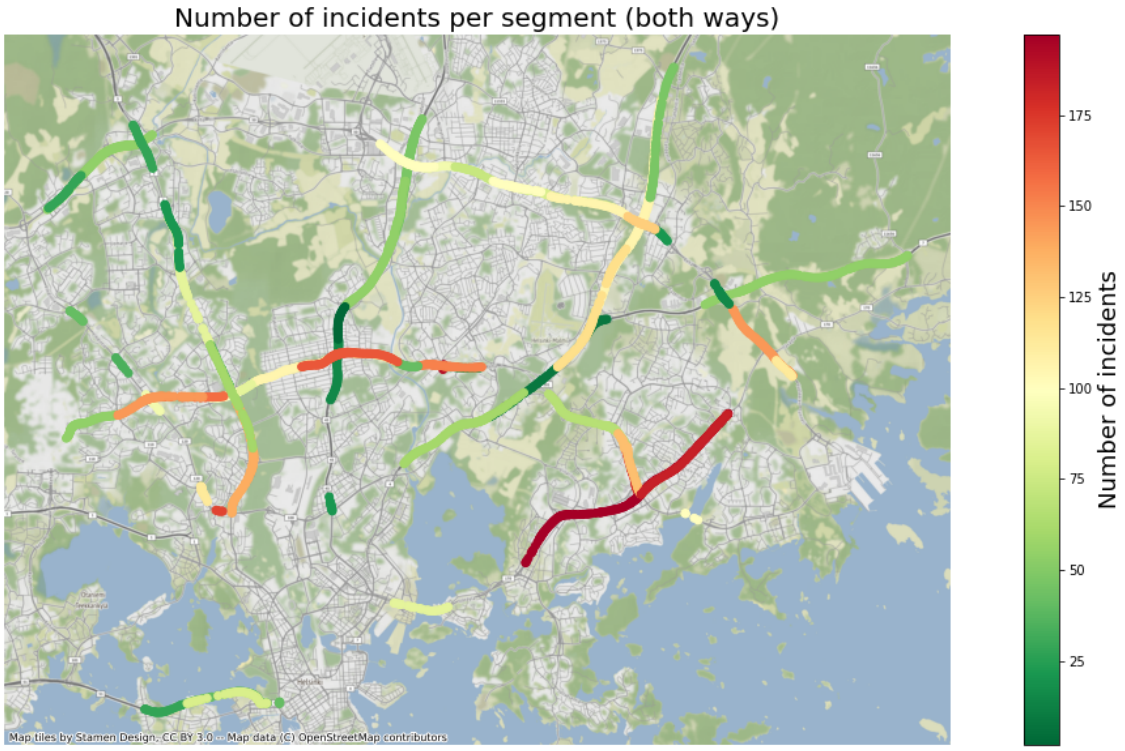


**Figure 5.11:** Heat map of the incidents detected during 2018 in Helsinki.

Notice that the number of incidents is calculated as the total incidents occurring in both road directions, and that only the 85 segments studied are displayed. The map shows that most of the 85 segments studied are highways, and as only the segments with incidents reported have been studied, it can be concluded that mainly only highways reported incidents in Helsinki during 2018. Additionally, despite of the fact that this map is incomplete, it can be observed a concentration of incidents at one of the east corridors of the capital.

## 5.4 Congestion cost

Before exposing the results, notice that the calculations are done for the year 2018 and taking into account the assumptions mentioned in 4.6. The total delay in Helsinki per passenger was 38,5 hours. The extra cost per passenger and year caused by congestion in Helsinki was: 72,8 liters of fuel consumption, with an extra direct cost of 110,7 €, and an extra emissions of 167,5 kg of C02.

## 5.5 Data limitation

The master thesis limitations have been caused mainly by the type of data used (speed recordings), the quality of the data, and the size of the data sets processed.

First of all, the data used consists of speed recordings. Speed recordings does not supply any information of the number of vehicles per time unit. In consequence, the estimation of congestion is done with delay. Secondly, some segments have gaps on the speed recordings, and the incidents registered are highly unreported. This last point has restricted the study, analysis, and validation of methods to detect non-recurring congestion. It has been problematic to find a valid method without any ground truth data of incidents.

Additionally, the size of the data sets has introduced extra challenges such as the time necessary to process the data. In spite of using parallel processing techniques to enable the computer apply several operations at the same time, and data chunking to split the data into smaller data sets, the specifications of the computer used have slowed and limited the research. These limitations can be summarized as: working with just a part of the city, reducing the time resolution from 5 to 10 minutes, and not performing all the test desired to optimize the methods.

Finally, other limitations have been caused by the nature of the traffic. Traffic behavior on urban networks can be random and unpredictable; not all incidents cause real congestion [24]. Urban networks are complex as there are many different roads types with different lengths and different intersections, which can have induced to errors when analyzing and interpreting the data.

# 6

# Conclusions

The aim of this study has been to find a method to detect NRC episodes and analyse them using delay. The data used belong to Helsinki area and is composed by speed recordings sampled every five minutes during 2018. Three different methods are tested and compared: delay dynamic threshold, k-NN delay prediction, and k-NN Local Outlier detection. To evaluate them, a data set of incidents registered is used. The results showed that the method with the best performance is k-NN Local Outlier detection. Finally, a analysis of the NRC episodes found with the method chosen is done. During the analysis many characteristics such as duration, severity and location are studied.

This study has show that outlier detection is a simple way to address NRC detection. Other methods such as dynamic threshold or traffic prediction are more complex and need more data processing. For example dynamic threshold needs a skew study and correction before applying the method. Additionally, delay prediction needs two steps to detect NRC: predict and compare the real measurements with the predicted ones looking for non matching scenarios. Therefore, it is needed to optimize this two parts separately. On the other hand, outlier detection has emerged as easy and faster to implement and capable of detecting 60% of reported incidents with a just 4,35% non reported incidents detected and 1074 incidents detected.

Secondly, the analysis of the NRC episodes detected by k-NN Local Outlier has highlighted many aspects. Referring to time patterns, slight differences between weekdays have been found such as Monday is the less congested weekday, whereas Tuesday, Wednesday and Thursday are equally the most congested ones. Even though, no major differences between weekdays have been found. The results, show that all of them can be treated as one, as they have matching characteristics and behave in a similar way. Also the number of incidents detected per weekday had no significant differences, with a share closer to 20% for all five of them. In contrast, the number of incidents detected per month showed that during snowy months (October-February) the number of incidents increase, whereas June and July had the lowest number of incident detected due to summer vacation. Regarding to NRC episodes characteristics, it is found that most of them last for less than 40 minutes, and that the number of NRC episodes that have a maximum delay registered between 1 and 5 [min/km] is very similar. Eventually, another interesting feature is that morning NRC episodes least for longer time, whereas the afternoon is the period that more NRC episodes occur.

Finally, during this study some issues have emerged. Most of them are related with the quantity of data and its quality. The large amount of data to process and the specifications of the computer used have forced to work with just a part of the data. Additionally, the lack of ground truth data for incidents, and the poor precision of the incidents reported have introduced additional challenges when evaluating the performance of the methods.

In the end, this study should help future researches to improve NRC detection methods, and to understand better how NRC behave and which characteristics does it has.

## 6.1 Limitations

The major limitation of this study is that it has been performed using just 85 of the 2158 road segments of Helsinki. Therefore some extra validations should be done to ensure that the results can be extrapolated to the rest of the city. Another limitation is the time resolution used. During the study it has been fixed to 10 minutes, what may be not enough to analyse deeply short NRC episodes. Finally, even traffic trends are difficult to change, the Covid-19 pandemic may have changed some of society's routines and some traffic patterns. Despite that, the data used for the study is from before the spread of the pandemic. Therefore, none of its effects have been taken into account.

## 6.2 Future work

This study is not yet completed and can be extended in many different ways as many questions have still no answer. Consequently, in this section a suggestion of possible improvements and future work is done.

The most obvious improvement is the application of the method k-NN Local Outlier to the entire Helsinki in order to detect incidents all over the city. That would permit a more complete study of the non recurring congestion, as many other incidents would be available to be analyzed. It would be also interesting to include the weekends on the study, and compare the differences between weekdays and weekends incidents. Besides, it is also suggested to apply the methods to other cities in order to verify if the method performs well in other environments different from Helsinki.

Another way to complete the study would be work with the highest time resolution available, 5 minutes. Therefore it could be compared if the method chosen perform better, or worse. Additionally, working with a resolution of 5 minutes would let the method detect incidents with a duration of 10 minutes and more. In order to introduce more complexity to the incident analysis, more variables could be taken into account such as the weather or how the segments are connected between them.

Finally, it is suggested to explore other local outlier detection algorithms. It would be interesting to verify the incident detection capacity of this kind of algorithms, and also carry out a comparison between them to which has a better performance.

# Bibliography


[1] V. Ahsani, M. Amin-Naseri, S. Knickerbocker, and A. Sharma, "Quantitative analysis of probe data characteristics: Coverage, speed bias and congestion detection precision," *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, vol. 23, no. 2, pp. 103–119, 3 2019.

[2] L. Zhang, Q. Liu, W. Yang, N. Wei, and D. Dong, "An Improved K-nearest Neighbor Model for Short-term Traffic Flow Prediction," *Procedia - Social and Behavioral Sciences*, vol. 96, pp. 653–662, 11 2013.

[3] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction," *Sensors (Switzerland)*, vol. 17, no. 4, 4 2017.

[4] N. Brenner and C. Schmid, "The 'urban age' in question," *International Journal of Urban and Regional Research*, vol. 38, no. 3, pp. 731–755, 5 2014. [Online]. Available: https://onlinelibrary.wiley.com/doi/full/10.1111/1468-2427.12115https://onlinelibrary.wiley.com/doi/abs/10.1111/1468-2427.12115https://onlinelibrary.wiley.com/doi/10.1111/1468-2427.12115

[5] A. Gaviria and E. H. Stein, "The Evolution of Urban Concentration Around the World: A Panel Approach," *SSRN Electronic Journal*, 1 2012. [Online]. Available: https://papers.ssrn.com/abstract=1817212

[6] "Post-pandemic traffic jams." [Online]. Available: https://www2.deloitte.com/ca/en/pages/technology-media-and-telecommunications/articles/post-pandemic-traffic-jams.html

[7] G. Weisbrod, D. Vary, and G. Treyz, "Measuring economic costs of urban traffic congestion to business," *Transportation Research Record*, no. 1839, pp. 98–106, 1 2003. [Online]. Available: https://journals.sagepub.com/doi/abs/10.3141/1839-10?casa_token=QOMJs59dxvcAAAAA%3AP4080ulcznvrFFl66-I1yFrwV7A_vL65jAqUK9V6-7fuTiz9Q93TTuUpqzyevhN2FEdZv6wx2-s3Rw

[8] S. A. Jayasooriya and Y. M. Bandara, "Measuring the Economic costs of traffic congestion," in *3rd International Moratuwa Engineering Research Conference, MERCon 2017*. Institute of Electrical and Electronics Engineers Inc., 7 2017, pp. 141–146.

[9] G. W. Evans and S. Carrère, "Traffic Congestion, Perceived Control,

and Psychophysiological Stress Among Urban Bus Drivers," *Journal of Applied Psychology*, vol. 76, no. 5, pp. 658–663, 1991. [Online]. Available: /record/1992-07324-001

[10] G. Lyons, "Getting smart about urban mobility – Aligning the paradigms of smart and sustainable," *Transportation Research Part A: Policy and Practice*, vol. 115, pp. 4–14, 9 2018.

[11] M. Barthelemy, "The Structure and Dynamics of Cities: Urban Data Analysis and Theoretical Modeling," 2016.

[12] "The true cost of congestion - International Fleet World." [Online]. Available: https://internationalfleetworld.com/the-true-cost-of-congestion/

[13] S. Faye and C. Chaudet, "Characterizing the Topology of an Urban Wireless Sensor Network for Road Traffic Management," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 7, pp. 5720–5725, 7 2016.

[14] Y. Djenouri, A. Belhadi, J. C. W. Lin, and A. Cano, "Adapted K-Nearest neighbors for detecting anomalies on spatio-temporal traffic flow," *IEEE Access*, vol. 7, pp. 10 015–10 027, 2019.

[15] S. Ramaswamy, R. Rastogi, and K. Shim KAIST, "Efficient Algorithms for Mining Outliers from Large Data Sets," Tech. Rep., 2000. [Online]. Available: www.bell-labs.com/projects/serendip

[16] B. Anbaroglu, B. Heydecker, and T. Cheng, "Spatio-temporal clustering for non-recurrent traffic congestion detection on urban road networks," *Transportation Research Part C: Emerging Technologies*, vol. 48, pp. 47–65, 11 2014.

[17] C. L. Lan, R. Venkatanarayana, and M. D. Fontaine, "Development of a Methodology for Determining Statewide Recurring and Nonrecurring Freeway Congestion: Virginia Case Study," *Transportation Research Record*, vol. 2673, no. 6, pp. 566–578, 6 2019.

[18] B. Anbaroğlu, T. Cheng, and B. Heydecker, "Non-recurrent traffic congestion detection on heterogeneous urban road networks," *Transportmetrica A: Transport Science*, vol. 11, no. 9, pp. 754–771, 10 2015.

[19] R. Dowling, A. Skabardonis, M. Carroll, and Z. Wang, "Methodology for Measuring Recurrent and Nonrecurrent Traffic Congestion," *Transportation Research Record*, vol. 1867, no. 1, pp. 60–68, 2004. [Online]. Available: https://doi.org/10.3141/1867-08

[20] A. H. Chow, A. Santacreu, I. Tsapakis, G. Tanasaranond, and T. Cheng, "Empirical assessment of urban traffic congestion," *Journal of Advanced Transportation*, vol. 48, no. 8, pp. 1000–1016, 12 2014.

[21] A. Skabardonis, P. P. Varaiya, K. F. Petty, and E. Org, "UC Berkeley Earlier Faculty Research Title Measuring Recurrent and Non-Recurrent Traffic Congestion Publication Date," Tech. Rep., 2008. [Online]. Available: https://escholarship.org/uc/item/3nh629g9

[22] H. Park, A. Haghani, and M. Hamedi, "Quantifying non-recurring congestion impact on secondary incidents using probe vehicle data," Transportation Research Forum, Tech. Rep. 206944, 3 2013. [Online]. Available: https://ideas.repec.org/p/ags/ndtr13/206944.html

[23] T. Thomas and E. C. Van Berkum, "Detection of incidents and events in urban networks," *IET Intelligent Transport Systems*, vol. 3, no. 2, pp. 198–205, 2009.

[24] S. Aghabozorgi, A. Seyed Shirkhorshidi, and T. Ying Wah, "Time-series clustering - A decade review," *Information Systems*, vol. 53, pp. 16–38, 5 2015.

[25] "HERE Technologies | The world's #1 location platform." [Online]. Available: https://www.here.com/

[26] A. Mohan Rao and K. Ramachandra Rao, "MEASURING URBAN TRAFFIC CONGESTION-A REVIEW," *International Journal for Traac and Transport Engineering*, vol. 2, no. 4, 2012.

[27] P. Davies, C. Hill, and G. Klein, "Standards for the Radio Data System - Traffic Message Channel," *SAE Technical Papers*, 8 1989. [Online]. Available: https://www.sae.org/publications/technical-papers/content/891684/

[28] V. Verendel and S. Yeh, "Measuring Traffic in Cities Through a Large-Scale Online Platform," *Journal of Big Data Analytics in Transportation*, vol. 1, no. 2-3, pp. 161–173, 12 2019. [Online]. Available: http://link.springer.com/10.1007/s42421-019-00007-7

[29] S. Devi and T. Neetha, "Machine Learning based traffic congestion prediction in a IoT based Smart City," *International Research Journal of Engineering and Technology*, 2017. [Online]. Available: www.irjet.net

[30] X. J. Ban, Y. Li, A. Skabardonis, and J. D. Margulici, "Performance Evaluation of Travel-Time Estimation Methods for Real-Time Traffic Applications," *http://dx.doi.org/10.1080/15472451003719699*, vol. 14, no. 2, pp. 54–67, 4 2010. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/15472451003719699

[31] R. Steve and P. John, "Overtaking Rule Method for the Cleaning of Matched License-Plate Data," *Journal of Transportation Engineering*, vol. 132, no. 8, pp. 609–617, 8 2006. [Online]. Available: https://doi.org/10.1061/(ASCE)0733-947X(2006)132:8(609)

[32] H. Nguyen, W. Liu, and F. Chen, "Discovering Congestion Propagation Patterns in Spatio-Temporal Traffic Data," *IEEE Transactions on Big Data*, vol. 3, no. 2, pp. 169–180, 7 2016.

[33] S. Athey, G. W. Imbens, and V. Ramachandra, "Machine Learning Methods for Estimating Heterogeneous Causal Effects Machine Learning Methods for Estimating Heterogeneous Causal Effects *," 2015. [Online]. Available: https://www.researchgate.net/publication/274644919

[34] V. Verendel and S. Yeh, "Traffic congestion of large cities through a large-scale online platform," 2019.

[35] "Home - INRIX." [Online]. Available: https://inrix.com/

[36] P. Chakraborty, J. R. Hess, A. Sharma, and S. Knickerbocker, "Outlier mining based traffic incident detection using big data analytics," in *Transportation Research Board 96th Annual Meeting Compendium of Papers*, 2017, pp. 8–12.

[37] M. Stuart, "Understanding Robust and Exploratory Data Analysis," *Journal of the Royal Statistical Society: Series D (The Statistician)*, vol. 33, no. 3, pp. 320–321, 1984. [Online]. Available: https://rss.onlinelibrary.wiley.com/doi/abs/10.2307/2988240

[38] G. Brys, M. Hubert, and A. Struyf, "A Comparison of Some New Measures of Skewness," *Developments in Robust Statistics*, pp. 98–113, 2003. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-57338-5_8

[39] Y. Chung and W. W. Recker, "A Methodological Approach for Estimating Temporal and Spatial Extent of Delays Caused by Freeway Accidents," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1454–1461, 4 2012.

[40] W. Pu, "Analytic relationships between travel time reliability measures," *Transportation Research Record*, no. 2254, pp. 122–130, 12 2011.

[41] D. B. Neill, "Expectation-based scan statistics for monitoring spatial time series data," *International Journal of Forecasting*, vol. 25, no. 3, pp. 498–517, 7 2009.

[42] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 7 2015. [Online]. Available: https://science.sciencemag.org/content/349/6245/255https://science.sciencemag.org/content/349/6245/255.abstract

[43] "Welcome to Python.org." [Online]. Available: https://www.python.org/

[44] "NumPy." [Online]. Available: https://numpy.org/

[45] "pandas - Python Data Analysis Library." [Online]. Available: https://pandas.pydata.org/

[46] "DataFrame — Dask documentation." [Online]. Available: https://docs.dask.org/en/latest/dataframe.html

[47] "multiprocessing — Process-based parallelism — Python 3.9.6 documentation." [Online]. Available: https://docs.python.org/3/library/multiprocessing.html

[48] "sklearn.preprocessing.StandardScaler — scikit-learn 0.24.2 documentation." [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

[49] "GeoPandas 0.9.0 — GeoPandas 0.9.0 documentation." [Online]. Available: https://geopandas.org/index.html

[50] D. Arthur and S. Vassilvitskii, "k-means++: The Advantages of Careful Seeding."

[51] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, no. C, pp. 53–65, 11 1987.

[52] "3.3. Metrics and scoring: quantifying the quality of predictions — scikit-learn 0.24.2 documentation." [Online]. Available: https://scikit-learn.org/stable/modules/model_evaluation.html#r2-score

[53] J. Xiao, "SVM and KNN ensemble learning for traffic incident detection," *Physica A: Statistical Mechanics and its Applications*, vol. 517, pp. 29–35, 3 2019.

[54] "Helsinki Region Infoshare - Open data service." [Online]. Available: https://hri.fi/en_gb/

[55] "Traffic data from Helsinki - Traffic volumes in CSV - Helsinki Region Infoshare." [Online]. Available: https://hri.fi/data/en_GB/dataset/liikennemaarat-helsingissa/resource/a652b375-548a-4ebe-a396-18500595244f

[56] "Change in distance travelled by car | ODYSSEE-MURE." [Online]. Available: https://www.odyssee-mure.eu/publications/efficiency-by-sector/transport/distance-travelled-by-car.html

[57] "Most EU cars run on petrol - Products Eurostat News - Eurostat." [Online]. Available: https://ec.europa.eu/eurostat/web/products-eurostat-news/-/DDN-20191024-1

[58] S. Dhec, "Idling: Why It's a Problem and What You Can Do." [Online]. Available: http://www2.epa.gov/cleandiesel/clean-school-bus

[59] "• Gasoline prices Finland 2000-2021 | Statista." [Online]. Available: https://www.statista.com/statistics/598024/unleaded-gasoline-prices-finland/

[60] N. Resources Canada, "Learn the facts: Fuel consumption and CO 2." [Online]. Available: www.4cleanair.org

# A

# Appendix 1

## A.1 Results

### A.1.1 First exploration

The results for monthly clustering suggest that the best option is to take one or two clusters. In spite of that, the Silhouette coefficient vales are not as high as in A.2. Therefore, to simplify the methodology, one group is chosen.

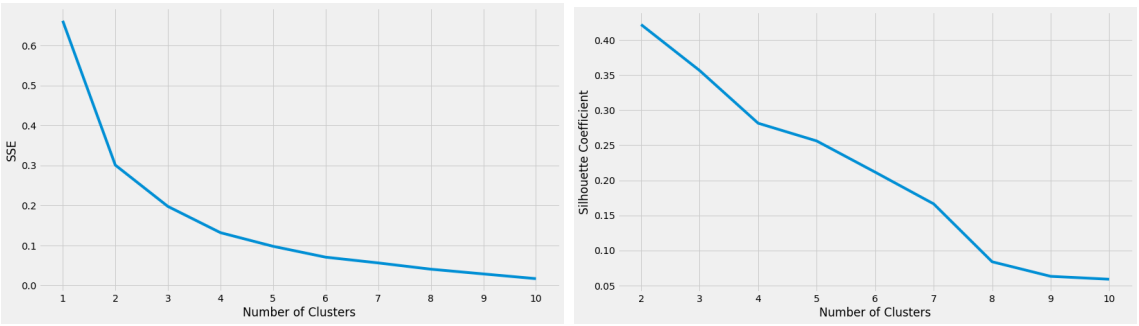


**Figure A.1:** SSE and Silhouette coefficients results for monthly k-means clustering.

On the other hand, the daily clustering has a clear optimum in 2 clusters.

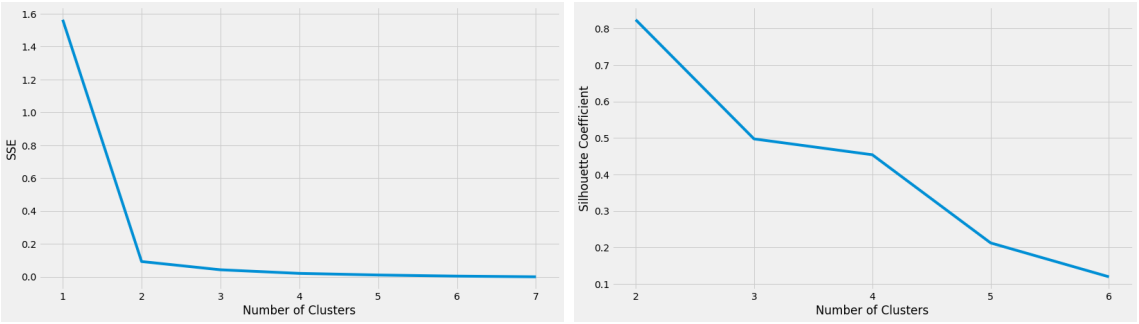


**Figure A.2:** SSE and Silhouette coefficients results for daily k-means clustering.

### A.1.2 Dynamic threshold

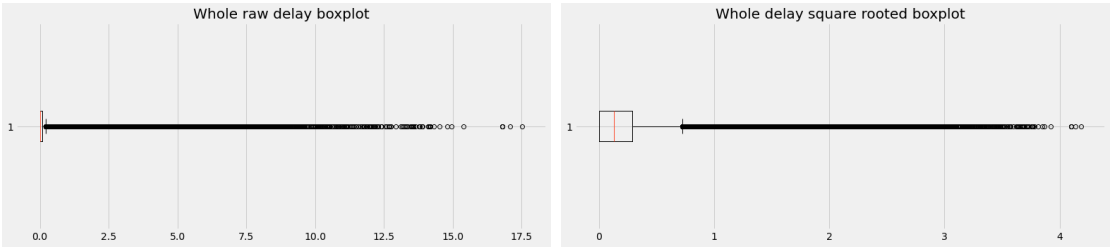


**Figure A.3:** Whole boxplots before and after applying square root to delay values.

**Table A.1:** Validation metrics results for dynamic threshold method for different maximum time (MT) values:

| | **Max. time** | **Number of reported incidents detected** | **%** | **Number of extra incidents detected** | **%** |
|---|---|---|---|---|---|
| **Raw** | **10** | 162 | 87,57% | 9317 | 42,08% |
| | **20** | 162 | 87,57% | 9715 | 43,87% |
| | **30** | 163 | 88,11% | 9947 | 44,92% |
| **SQRT** | **10** | 147 | 79,46% | 5073 | 22,91% |
| | **20** | 147 | 79,46% | 5232 | 23,63% |
| | **30** | 147 | 79,46% | 5312 | 23,99% |
| **LN** | **10** | 93 | 50,27% | 2437 | 11,01% |
| | **20** | 96 | 51,89% | 2541 | 11,48% |
| | **30** | 97 | 52,43% | 2595 | 11,72% |

### A.1.3 k-NN delay prediction

**Table A.2:** Results for d=2 and taking into account only incidents detected from 6h-22h:

| k | thd | Number of reported incidents detected | % | Number of extra incidents detected | % |
|---|---|---|---|---|---|
| **5** | 0,61 | 111 | 60,00% | 17251 | 77,91% |
| **10** | 0,59 | 111 | 60,00% | 16927 | 76,44% |
| **15** | 0,62 | 111 | 60,00% | 15660 | 70,72% |
| **20** | 0,645 | 111 | 60,00% | 14712 | 66,44% |
| **25** | 0,655 | 111 | 60,00% | 14379 | 64,94% |
| **30** | 0,65 | 111 | 60,00% | 14329 | 64,71% |

**Table A.3:** Results for d=6 and taking into account only incidents detected from 6h-22h:

| k | thd | Number of reported incidents detected | % | Number of extra incidents detected | % |
|---|---|---|---|---|---|
| **5** | 0,627 | 111 | 60,00% | 17524 | 79,14% |
| **10** | 0,68 | 111 | 60,00% | 15347 | 69,31% |
| **15** | 0,705 | 111 | 60,00% | 14190 | 64,08% |
| **20** | 0,7 | 111 | 60,00% | 13996 | 63,21% |
| **25** | 0,735 | 111 | 60,00% | 12939 | 58,43% |
| **30** | 0,735 | 111 | 60,00% | 12771 | 57,68% |

### A.1.4 k-NN Local outliers

**Table A.4:** Results for k-NN Local Outlier, maximum time (MT) parameter: 10 minutes

| k | d | Number of reported incidents detected | % | Number of extra incidents detected | % |
|---|---|---|---|---|---|
| **5** | **17** | 111 | 60,00% | 1900 | 8,58% |
| **10** | **15** | 111 | 60,00% | 1549 | 7,00% |
| **15** | **12.5** | 111 | 60,00% | 1209 | 5,46% |
| **20** | **11.65** | 111 | 60,00% | 1145 | 5,17% |
| **25** | **11.3** | 111 | 60,00% | 1123 | 5,07% |
| **30** | **10,8** | 111 | 60,00% | 1085 | 4,90% |
| **35** | **10,3** | 111 | 60,00% | 1050 | 4,74% |
| **40** | **10,1** | 111 | 60,00% | 1056 | 4,77% |
| **45** | **10,4** | 111 | 60,00% | 1142 | 5,16% |
| **50** | **10,1** | 111 | 60,00% | 1119 | 5,05% |
| **55** | **9,8** | 111 | 60,00% | 1094 | 4,94% |
| **60** | **9,5** | 111 | 60,00% | 1074 | 4,85% |
| **65** | **9,5** | 111 | 60,00% | 1102 | 4,98% |

**Table A.5:** Results for k-NN Local Outlier, maximum time (MT) parameter: 20 minutes

| k | d | Number of reported incidents detected | % | Number of extra incidents detected | % |
|---|---|---|---|---|---|
| **5** | **14,7** | 111 | 60,00% | 1598 | 7,22% |
| **10** | **13,6** | 111 | 60,00% | 1379 | 6,23% |
| **15** | **12** | 111 | 60,00% | 1174 | 5,30% |
| **20** | **11,12** | 111 | 60,00% | 1089 | 4,92% |
| **25** | **11** | 111 | 60,00% | 1114 | 5,03% |
| **30** | **10,21** | 111 | 60,00% | 1024 | 4,62% |
| **35** | **9,8** | 111 | 60,00% | 998 | 4,51% |
| **40** | **9,64** | 111 | 60,00% | 1005 | 4,54% |
| **45** | **9,25** | 111 | 60,00% | 983 | 4,44% |
| **50** | **8,96** | 111 | 60,00% | 963 | 4,35% |
| **55** | **8,94** | 111 | 60,00% | 990 | 4,47% |
| **60** | **8,95** | 111 | 60,00% | 1022 | 4,62% |

**Table A.6:** Results for k-NN Local Outlier, maximum time (MT) parameter: 30 minutes

| k | d | Number of reported incidents detected | % | Number of extra incidents detected | % |
|---|---|---|---|---|---|
| **5** | **13,1** | 111 | 60,00% | 1428 | 6,45% |
| **10** | **11,6** | 111 | 60,00% | 1223 | 5,52% |
| **15** | **10,7** | 111 | 60,00% | 1156 | 5,22% |
| **20** | **9,8** | 111 | 60,00% | 1073 | 4,85% |
| **25** | **9,3** | 111 | 60,00% | 1033 | 4,67% |
| **30** | **9,06** | 111 | 60,00% | 1032 | 4,66% |
| **35** | **8,7** | 111 | 60,00% | 1009 | 4,56% |
| **40** | **8,4** | 111 | 60,00% | 993 | 4,48% |
| **45** | **8,5** | 111 | 60,00% | 1045 | 4,72% |
| **50** | **8,5** | 111 | 60,00% | 1089 | 4,92% |
| **55** | **8,6** | 111 | 60,00% | 1128 | 5,09% |
| **60** | **8,7** | 111 | 60,00% | 1182 | 5,34% |
| **65** | **8,65** | 111 | 60,00% | 1201 | 5,42% |