
A data-driven method to automate the detection of traffic control systems that do not perform as intended

Master Thesis

Victor Brouwer

S130375

November, 2019

Civil Engineering

University of Twente, The Netherlands

Internal Supervisor:

prof.dr.ir. E.C. van Berkum (University of Twente)

dr. K. Gkiotsalitis (University of Twente)

External Supervisor:

ing. E. Jongenotter (Witteveen + Bos)

S.A. Veenstra MSc (Witteveen + Bos)

**UNIVERSITY
OF TWENTE.**



Abstract

Due to the limited available resources and the sheer number of Traffic Control Systems (TCS) used in contemporary cities, the frequency of updating TCS' timing program is often low or sporadic. An outdated timing program means a less than optimal performance of the TCS, resulting in longer travel times and unnecessary travel costs. Past literature has investigated how the TCS can be improved through retiming, but there are limited studies on determining for which TCS retiming is most valuable. This study fills this research gap by investigating the performance of machine learning methods for identifying TCS that needs retiming. The performance indicators that monitor the performance of a TCS are often influenced by the policy of the road authority or geographical characteristics of the TCS. To enable the unbiased comparison of different TCS, this study uses policy- and geographically- neutral performance indicators, such as double stops and red-light runners. Then, we test the performance of unsupervised learning methods (Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and Isolation Forest) on a six-month case study in three regions of the Netherlands (province of North-holland, the city of the Hague, the city of Deventer). This case study demonstrates the benefit that the differences in the TCS performance helps by providing targeted maintenance. All the 11 TCS which are detected as anomalous by the DBSCAN have at least one performance indicator with a statistically extreme value. The Isolation Forest detects 17 TCS as anomalous, where 2 anomalies do not have a statically extreme value for one of the performance indicators. In total 38 of the 125 TCS had at least one performance indicator with a statistically extreme value. This work supports the introduction of automated methods for identifying problematic TCS by providing the first step in this direction.

Contents

List of Figures	v
List of Tables	vi
1 Introduction	1
1.1 Problem description	1
1.2 Research objectives	2
2 Literature Review	5
2.1 Retiming	5
2.2 Existing KPIs for the performance of TCS	7
2.3 Methods to detect TCS that does not perform as intended	8
3 Potential policy and geographical- neutral performance indicators	12
4 Method	15
4.1 Cluster and anomaly detection methods	15
4.2 Density-Based Spatial Clustering of Applications with Noise (DBSCAN)	16
4.3 Isolation forest	19
5 Case study	22
5.1 Data preparation	22
5.2 Comparison of the performance of multiple TCS	23
5.3 Clustering and anomaly detection	28
5.4 Performance and validation	30
6 Discussion and limitations	35
7 Conclusions and recommendation	38
References	40
Appendices	44
A Savings in user costs due to TCS retiming	45

CONTENTS

B	Research classification and questions	46
C	Theoretical background for the case study	48
D	Calculation of the KPIs	56
E	KPI values during the day	62

List of Figures

1.1	Number of performance checks conducted at traffic control systems by Dutch road authorities	2
2.1	Performance indicators used in the literature.	7
4.1	Explanation of the DBSCAN algorithm	18
4.2	Explanation of the Isolation Forest algorithm	19
4.3	The Isolation Forest tree	20
5.1	Trajectory of the data	23
5.2	mean value of a KPI during the day.	26
5.3	mean value of a KPI during the week.	27
5.4	Cluster method with a week of data and the safety KPI group.	29
5.5	Isolation Forest during the evening peak and all KPI's are taken into account.	30
A.1	Savings in user costs due to TCS retiming (Sunkari, 2004).	45
C.1	Signal group labeling (Wegenwiki, 2015)	49
C.2	Position and describing of the detector loops(Veenstra, Geurs, Thomas & van den Hof, 2016)	50
C.3	Primary realization of an intersection(Katwijk, 2008)	51
C.4	Alternative realization of an intersection(Katwijk, 2008)	52
C.5	V-log data visualization	53
C.6	Phase cycle of a signal group	54
D.1	Entry at red and leave during a red light	56
D.2	Visualization why prolonging green is not unnecessary green	57
D.3	Situation 1	58
D.4	Situation 2	58
D.5	Example of a double stop	59
E.1	mean value of a KPI during the day.	62
E.2	mean value of a KPI during the day.	62
E.3	mean value of a KPI during the day.	63

List of Tables

2.1	Literature summary	10
3.1	KPI summary	13
4.1	DBSCAN Algorithm	17
5.1	Correlations	24
5.2	The standard deviation of the correlation between the KPI's	24
5.3	The mean standard correlation and standard deviation with the intensity	25
5.4	Anomalies per group	31
5.5	number the TCS is counted as anomaly	31
5.6	Performance of the anomaly detection methods	33
5.7	Characteristics of the anomalies	33
B.1	Problem description and how this problem is approached	46
C.1	Modalities and corresponding labels	48

Chapter 1

Introduction

1.1 Problem description

The systematic optimization of traffic control systems (TCS) timing programs represents an essential element of traffic management (Dunn Engineering Associates, 2005). The optimal functioning of TCS has many benefits, among which reducing travel time and safety for motorists (Sunkari, 2004). The re-optimization of traffic control systems is costly and labor-intensive since every TCS needs individual treatment. In addition, there is a very large number of TCS that need to be re-optimized periodically to adjust to the traffic demand changes. As a result, the frequency of updating TCS' timing programs is often limited by the resources available (Koningsbruggen, 2016).

Ineffective timing programs impact road users at congested, as well as at uncongested, routes. In general, the safety and travel time of all drivers and their passengers depends on a TCS' timing program (Dijkstra, 2014). Travelers expect TCS to be efficiently managed by the road authority. Drivers usually assume that the responsible agency can efficiently operate the TCS, thus they only report the most obvious failures (Dunn Engineering Associates, 2005). Most of the time, inefficient TCS timing programs do not lead to public complaints. However, inefficiencies silently increase the operational costs of the travelers through longer trip times and increased fuel costs (Baneke, 2016).

Due to changes in local traffic demand, many TCS programs are not well adjusted to the current traffic conditions. In the past few years, several research papers have reported that outdated TCS programs lead to unnecessary red-light times, which can cause frustration among car drivers (ANWB, 2018). In some cases, this can even lead to ignorance of the red light, which could create dangerous situations (Rijkswaterstaat, 2016).

In 2015, a survey for all the Dutch road authorities assessed the current level of performing a

A data-driven method to automate the detection of traffic control systems that do not perform as intended 1

quality observation of the Dutch TCS (DTV Consultants B.V. & Willekens, 2016). Figure 1.1 shows the frequency of authorities testing the TCS' performance according to the current traffic situation.

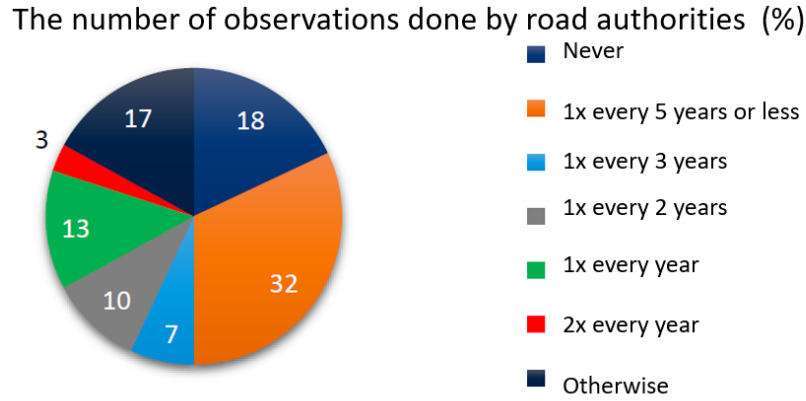


Figure 1.1: Number of performance checks conducted at traffic control systems by Dutch road authorities

Figure 1.1 shows that 18% of the road authorities never perform a performance check while 32% check the TCS performance less than once in five years. Due to this fact, significant improvements might be achieved by retiming the TCS program (Dunn Engineering Associates, 2005). The direct benefit of retiming a TCS is the reduced delay for the motorist. Besides the direct benefits, retiming provides several indirect benefits such as reduced frustration, fuel consumption, emissions and improved safety (Sunkari, 2004). In figure A.1 the savings in user costs (reduced delays, stops, fuel consumption) for a retimed TCS are illustrated. Due to the high number of TCS and the need for individual assessment of their performance, it is important to compare multiple TCS to identify which TCS do not perform as intended. This will facilitate targeted maintenance which leads to reduced travel and maintenance costs. The fundamental understandings of the current maintenance and retiming policies are explained in section 2.1.

1.2 Research objectives

In the literature, different approaches to increase the performance of a TCS have been studied. Typically, the optimization of TCS is done by the use of computer simulations and models that test the suggested retiming strategies (Pop, 2018)(Yousef et al., 2018) (Kumar et al., 2018).

By the same token, newly available technologies, such as GPS data obtained by smartphones and cars, provide new input data for the TCS re-timing which can be used in

the optimization process. In addition, the location of congestion in a road network can be predicted with the use of GPS data and the TCS timing programs can be adjusted accordingly. (Kan et al., 2018) (Andrea & Marcelloni, 2016) (Munoz-organero, Ruiz-blaquez & Sánchez-fernández, 2017).

Another approach to improve the TCS performance studied in the literature is via car-to-infrastructure communication, also described as talking TCS. A talking TCS communicates with the nearby approaching traffic. This traffic then receives information and the vehicles adjust their speed so they will enter the intersection when the corresponding direction light displays green (Stahlmann & Malte, 2018) (Litescu, Viswanathan, Aydt & Knoll, 2019).

Although there is an extensive literature on optimizing the performance of a TCS, determining the current performance of TCS with a data-driven approach and suggesting targeted TCS maintenance is not extensively studied. The problem with the TCS maintenance is the lack of available resources to retune every TCS. The process of retuning a TCS's program is still costly and labor-intensive, therefore it is important to know which TCS does not perform as intended.

In this research Machine Learning methods are used to identify TCS that performs below average and require maintenance. Some performance indicators that are commonly used (such as waiting time and throughput) cannot be used in this research. For example, a TCS with a high traffic demand is more likely to have a higher throughput since the total throughput is dependent on the traffic demand of the location. The waiting time for a specific modality depends on the policy used by the road authority. For example, in some regions, cyclists receive more green time since a cyclist-friendly policy is applied. Therefore, this study uses geographically and policy-neutral key performance indicators (KPI) so that all the TCS can be compared with each other. An example of a KPI which is geographically and policy-neutral is the number of red-light runners. No matter the policy or geographical design of a TCS, a higher value of red-light runners indicates a configuration problem.

The contributions of this research are as follows:

- The introduction of geographical- and policy-neutral performance indicators for performing an unbiased comparison of the performance of multiple traffic control systems.
- The introduction of unsupervised machine learning methods to automate the detection of traffic control systems that do not perform as intended.

In chapter 2 a literature review is conducted which focuses on related literature and the relevance of this research. In chapter 3 the findings of the interviews with experts are presen-

ted and the used policy- and geographically- neutral performance indicators are presented. Chapter 4 the anomaly detection method Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and Isolation Forest will be explained. In chapter 5 the obtained novel method, to identify which TCS do not perform as intended, is tested for a case study in the Netherlands. The thesis concludes with the discussion and limitations in chapter 6 and conclusions and recommendations in chapter 7.

Chapter 2

Literature Review

To identify which TCS does not perform as intended, the performance indicators and the reasons that trigger a retiming need to be investigated. For this reason, this chapter is divided into three parts: 1) Fundamental understandings of retiming 2) Existing KPIs for the performance of TCS 3) Methods to detect TCS with a performance below average.

2.1 Retiming

Parameters which influence the performance of the TCS may change over time, such as traffic volumes. The traffic conditions have day-to-day changes and even during the day several traffic conditions may occur (Yang, Yu, Wang, Quddus & Xue, 2018). For this reason, the values of the KPIs during different time segments (weekday/weekend, peak hour/off-peak) are compared to check whether the used KPI also changing during the day.

It is important to know which factors lead to the technical need for a TCS program retiming. (Sunkari, 2004) and (Dunn Engineering Associates, 2005) reported six important factors which lead to the need for retiming a TCS program:

1. Changes in local or area-wide traffic demands.
2. Changes in peak period volumes.
3. Changes in the directional flow.
4. Local land-use changes.
5. Change in intersection geometry.
6. Change in the number of used lanes

Currently, major events, like factors four, five and six usually directly lead to retiming the TCS program. Therefore, for this research, only the first three points are relevant. These three factors change (slowly) over time and do not directly lead to a functioning check of the TCS (Sunkari, 2004). The three incentives which lead to checking whether retiming of a TCS is needed are (Gordon L, 2010):

1. An accident experience
2. Comments and complaints by the public
3. Observation of the performance

One of the goals of this research is to show with a data-driven analysis which TCS performance is below average.

The U.S. Department of Transportation defined four important performance indicators to evaluate the performance of the TCS during an observation:

1. Cycle failure (inability of a vehicle to pass through the intersection in one signal cycle) is a key indicator of a saturated phase.
2. Spillback from turning bays into general use lanes.
3. Delays that may be incompatible with the volume to capacity ratio (V/C). For example, unduly long cycle lengths or improper splits may lead to excessive delay when the minimal flow is observed during other portions of the green time for the phase.
4. Imbalance in green time (high demand approach versus low demand approach).

For this research, it is not possible to take the second indicator into account, because it is not possible to calculate this spillback with the data generated by the TCS. The other 3 performance indicators are taken into account in one or more policy-neutral performance indicators: double stops indicate cycle failure, 90 seconds waiting time indicate a wrong V/C ratio and the imbalance in green time is taken into account in the no use of fixed green and unnecessary green. A more extended description of the KPI is given later on in chapter 3 and Appendix C and D.

It is important to review the functioning of a TCS within five years because the probability is high that retiming the TCSs program provides significant improvements in most cases (Dunn Engineering Associates, 2005). Especially in areas where the traffic volumes grow or change, frequent observations are needed.

2.2 Existing KPIs for the performance of TCS

TCS are important to provide well-organized traffic management, but their timing programs are often outdated. Nowadays the outdated TCS can be optimized for a single intersection or network if the traffic flow is known or static (R. I. Blokpoel, Caselli, Härrä, Niebel & Leich, 2015). However, the traffic flows are not static but are changing over time, hence the TCS timing program regarding the phase orders, green time duration and cycle lengths might be outdated (Kant & Koenen, 2017). Besides outdated input data for TCS timing program algorithm, new technologies and data sources are available to improve the TCS algorithm (Krajzewicz, Blokpoel & Cornelia, 2014) which could be an incentive for upgrading the current TCS.

Since there are not enough resources available to update all TCS, only some TCS can be updated at a given time, therefore it is important to know the performance of every TCS. (R. I. Blokpoel et al., 2015) reviewed 50 publications and noted the scenario and performance indicators used in each instance (see figure 2.1). They concluded that the different TCS programs are hardly comparable due to the different scenario set-ups and performance indicators.

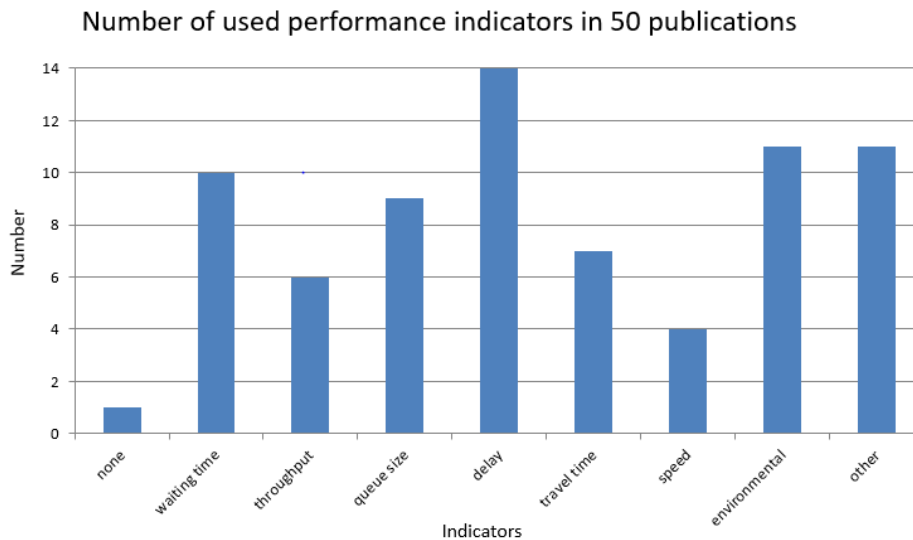


Figure 2.1: Performance indicators used in the literature.

Figure 2.1 shows that the most frequent performance indicator is delay, but it is only used in 35% of the publications. This implies that there is no universal performance indicator or benchmark to assess the performance of TCS. From figure 2.1, (R. Blokpoel, Vreeswijk, Krajzewicz & Kless, 2014) concludes that using all the measurements and summing them up will not always indicate how well a TCS performs. Some indicators are more important for a particular policy. Adding a weight factor to the indicators is a possible solution to

this problem where the road authority can decide the weight factor.

In the Netherlands, the most used indicator is the waiting time (R. Blokpoel et al., 2014). However, the actual waiting time can differ from the drivers perceived waiting time. Moving and stopping several times at the same intersection results in lower perceived waiting times, compared to a long standstill (Bijl, Vreeswijk, Bie & Berkum, 2011). A common behavior of road users is to slow down if they are reaching the end of the queue or stop line and this will not influence the waiting time. For this reason, only considering the objective waiting time would make the performance judgments of the TCS too positive (R. J. Blokpoel, Krajzewicz & Nippold, 2010). Therefore, other performance indicators to evaluate a TCS must be also taken into account.

To conclude, multiple KPIs must be taken into account to evaluate the performance of a TCS. The comparison of the performances of different traffic control systems is still difficult because different traffic control systems might have different objectives. For instance, a traffic control system might intentionally limit the throughput of an intersection. Hence, the unbiased comparison of the performance of multiple TCS is still a gap in literature. As established in chapter 1, the used KPIs must be geographical and policy-neutral. The seven performance indicators mentioned in figure 2.1, are not directly geographical- and policy-neutral. In each location, the maximum allowed speed, the presence of freight traffic and the traffic demand can be different. This is why the performance indicators speed, environmental, throughput and queue size are not independent. Road authorities differ in prioritizing the modalities depending on their policy. For this reason the waiting time, travel time and delay are not independent. Most of the KPIs used in this research are related to one of the KPIs mentioned in figure 2.1 and the KPIs will be separated in different policy groups. This will be explained further in chapter 3.

2.3 Methods to detect TCS that does not perform as intended

In this section other works that detected congested roads or abnormal traffic patterns are described and the used methods are briefly explained. (Jin, Zhang, Li & Hu, 2008) used a robust Principal Components Analysis-based method to detect abnormal traffic flow patterns using loop detector data. First, a principal component analysis is conducted and after that, three abnormality isolation strategies are used to detect the abnormal traffic. These strategies can be used to identify the cause of these abnormal traffic patterns.

(Alesiani, Moreira-Matias & Faizrahnemoon, 2018) also analysed loop detector data. First, the detection loops based on similar behavior with a distance matrix built upon a statist-

ical metric are clustered. Then a principal coordinates analysis is conducted to detect the anomalous detector loops. Thereafter, a fundamental diagram that discovers the critical density of a road section or spot is developed. With seven different learning methods, the occurrence of the critical density was forecasted. (Rossi, Gastaldi, Gecchele & Barbaro, 2015) analyse loop detectors data with a fuzzy logic-based method to detect incidents.

(Zhang, He, Tong, Gou & Li, 2016) detect spatial-temporal traffic flow patterns based on loop detector data with a dictionary-based compression theory. For this theory, five different features are used: the county, the sub-region, intersection level, traffic flow and traffic occupancy. Each feature is divided into several categories. The combination of the different category values per detector loop leads to different traffic patterns.

(Guardiola, Leon & Mallor, 2014) used a Functional Data Analysis (FDA), a collection of statistical techniques for the analysis of information, to determine a traffic profile corresponding to a single datum. With the use of the Principal Component Analysis (PCA) for multiple days between 2004 and 2011 the traffic profile is determined and compared. (Zhong et al., 2017) predict travel time by using a Functional Principal Component Analysis based on historical data and real-time measurements to assess the effects of abnormal traffic conditions. (Maghrour Zefreh Torok, 2018) describes a method that detects bad loop detectors data samples and based on many time series samples the holes caused declared bad, are filled.

The use of detector loop data is primarily used to check the performance of a traffic network instead of individual TCS. (Leclercq, Chiabaut & Trinquier, 2014) compares several existing estimation methods for the Macroscopic Fundamental Diagram (MFD). This is a diagram where the flow (veh/sec/lane) is set out against the density (veh/m/lane) of the network, to detect critical situations. Loop detectors fail to provide a good estimation for mean network speed or density because they cannot capture the traffic spatial dynamics over links. (Leclercq et al., 2014) used a simple adjustment technique in order to reduce the discrepancy when only loop detectors are available. (Ambühl & Menendez, 2016) combines the loop detectors data with floating car data to determine the MFD, this method reduces estimation errors significantly.

(Novikov, Novikov, Katunin & Shevtsova, 2017) also faced the problem of the changing traffic flows. Novikov improved the TCS program by using more information provided by various sensors and measuring devices, transport detectors, video surveillance systems, GPS-GLONASS systems and mobile devices. The new TCS program algorithm adapts automatically to the changes in traffic conditions. Table 1 provides a summary of the literature about methods which are used to detect abnormality in traffic situation.

Table 2.1: Literature summary

Authors	Objective	Algorithm
Guardiola, Leon, & Mallor	Determine a traffic profile corresponding to a single date for multiple years. (long term decisions, traffic changes)	Several FDA techniques: 1) Generalized Cross Validation (GCV) to determine the K for data smoothing 2) PCA to reduce the dimensionality. 3) Hotelling T2 and Multivariate Exponentially Weighted Moving Average (MEWMA) for traffic monitoring of the daily flow pattern
Zhong, et al.,	Predict travel time to assess the effects of abnormal traffic conditions.	Functional Principal Component Analysis
Rossi, Gastaldi, Gecchele, & Barbaro	Detect incidents with loop detectors data (LDD).	Fuzzy Inference Systems
Xuexiang, Zhang, Li, & Hu.	Detect abnormal traffic flow patterns using LDD.	Robust principal components analysis-based method.
Alesiani, Moreira-Matias, & Faizrahneemooon	Forecast critical density for a specific location.	1) Distance matrix for clustering 2) Principal coordinates analysis to detect anomalous loops 3) Learning methods to forecast critical density
1) Ambuhl Menendez 2) Leclercq, Chiabaut, & Trinquier	Determine the MFD with, among others, LDD.	The use of V (speed) , T (time), Q (flow) and K (density) Based on the fundamental equation $Q = K * V$
Novikov, Novikov, Katunn, & Shevtsova	Improving the TCS' program algorithm.	An improved algorithm based on the calculation method of the Russian Federation.

Summarizing the findings of these studies, it is possible to identify incidents, forecast critical density and abnormal traffic conditions with the use of detector loop data if the problem is supervised. A common problem is the occurrence of faults in the detector loop data. The principal component analysis is a commonly used tool to handle multiple dimensions/performance indicators. The remaining question for this research is as follows: is it possible to automatically identify TCS which performs below average with unsupervised machine learning methods?. Unsupervised learning is important since no labels are needed. It is not needed to define normal and abnormal before the analysis. These labels are automatically given after the analysis.

The closest work to our problem is the one by (P. Chen, Yu, Wu, Ren & Li, 2017). In this research, the data which is produced by the TCS was examined. In other words, the loop detectors data and the data about the change in green traffic light directions are combined. Chen, Yu, Wu, Ren, Li identified the number of red-light-running (RLR) and examined the influential factors associated with RLR since RLR leads to intersection-related crashes and endangers intersection safety. First, data preparation is done in three steps: 1) collect high-resolution traffic and signal event data. 2) Identify the RLR event by using stop bar detectors and down-stream entrance detectors. 3) Match the identified RLR cases to the detectors to determine the exact location. After the data preparation, the analysis of influential factors, correlation analysis, a regression model and validation is conducted.

In the research proposed here, the data generated by TCS will be analysed and is focused on factors that are changing during the time (peak volumes, directional flows and traffic demand) instead of the appearance of a major event (change in number of lanes or change in intersection geometry). This research is a new incentive for conducting a performance check at a TCS besides the occurrence of an accident or complaints by public. The goal of this analysis is to detect abnormal TCS conditions, from a negative perspective, at intersections. The focus will be on performance indicators which are policy-neutral, so all the TCS can be compared to one another. Most of the used performance indicators are related to commonly used performance indicators which are mentioned in figure 2.1. The researches listed in section 2.3 mostly detect abnormal traffic flows/patterns in a road network to identify the congested roads and intersections and not to detect abnormal TCS. Moreover, the above-mentioned methods are supervised since the abnormal traffic patterns with corresponding location and time were known. Concluding these points, the contributions of this research are as follows:

- The introduction of geographical- and policy-neutral performance indicators for performing an unbiased comparison of the performance of multiple traffic control systems.
- The introduction of unsupervised machine learning methods to automate the detection of traffic control systems that do not perform as intended.

Chapter 3

Potential policy and geographical-neutral performance indicators

In chapter 2 the relevant literature is presented for detecting negatively abnormal TCS. Detecting abnormal TCS is still a gap in the literature and the comparison between multiple TCS is barely investigated. Hence, a survey with experts is conducted to obtain more information. The data used is based on the Dutch labeling of an intersection. This labeling makes a distinction between modalities and driving direction. In addition, different types of detector loops are used in the Netherlands which are placed at different locations and with a different function. The Dutch road authorities have their own approach of controlling a TCS which takes all the different modalities and directions into account. This is called the Routine and Wintermaintenance Service Code (RWSC) where several signal groups of a TCS are placed in different blocks. The chosen sequence of these blocks is the primary realization of the TCS. For further optimization, the internal phase of a signal group is taken into account. The Dutch data which is saved by the TCS also differs from the international standard message format SPaT (Signal Phase and Timing data), because in the Netherlands V-log (verkeerskundig log) is used. An extensive explanation of the Dutch TCS is given in appendix C.

The performance indicators in this research are based on performance indicators that are used during observation to check the performance of a TCS (sub-section 2.1), the commonly used performance indicators in literature (sub-section 2.2) and expert interviews. Since the case study takes place in the Netherlands, Dutch road authorities are interviewed. Interviewees are asked about which performance indicators they used to check the performance of a TCS and their views are compared against the KPIs mentioned in the literature. The following experts are interviewed and data from the corresponding region is obtained:

- Erik Jongenotter, project manager and senior traffic consultant at Witteveen + Bos.
- Nico van Beugen, traffic manager of the municipality of Deventer

- Rogier Hoek, the traffic manager of the city of The Hague
- Dimitri Poncin, traffic manager of the province of North-Holland

The selected KPIs are summarized in table 3.1. The column labelled Expert shows which experts mentioned the corresponding KPI. In the column Literature are the related performance indicators from literature. In the column Policy group, this study lists the related policy of each particular performance indicator.

Table 3.1: KPI summary

Kpi	Policy group	Expert	Literature
Red-light running	Safety	all	P. Chen, Yu, Wu, Ren & Li
The unnecessary green time gave	Throughput	all	1) Dunn Engineering Associates 2) R. I. Blokpoel et al
No use of fixed green	Outdated design	all	1)Dunn Engineering Associates 2) R. I. Blokpoel et al
The number of double stops	Throughput	all	1) Dunn Engineering Associates 2) Kant & Koenen
The use of prolonging green	Outdated design	1	1) Katwijk 2) Scheepjens
Flutter behavior	Outdated design	2	1)Alesiani, Moreira-Matias & Faizrahnemoon 2) Maghrour Zefreh & Torok
Waiting time longer than 90 seconds	Throughput	all	1) Ministerie van Infrastructuur en Milieu 2) Dunn Engineering Associates
Early starters	Safety	2	P. Chen, Yu, Wu, Ren & Li

In table 3.1 eight KPIs are mentioned, with the help of the experts knowledge and literature. For each KPI will briefly be explained what a high value could indicate. A high number of Red-light runners and earlier starters lead to intersection-related crashes and endanger intersection safety (P. Chen et al., 2017). If unnecessary green time after the last vehicle passes the green light occurs often at a TCS, increase the total travel time. The unused fixed green phase of a direction at an intersection might cause extra waiting time for other directions. A high number of double stops cause more waiting time and more emissions/fuel costs since the starting and braking are fuel costly. The prolonging green phase only occurs in the alternative realization of a TCS, so if this phase often occurs, the primary realization of the TCS could be outdated. The fluttering of a loop detector is a common problem (Alesiani et al., 2018) that might cause unnecessary green-light requests. A waiting time longer than 90 seconds is too long for a TCS and in contrast with the Dutch guidelines (Ministerie van Infrastructuur en Milieu, 2014). How these KPIs are

exactly calculated is described in the Appendix D.

Concluding this chapter, the selected policy and geographically-neutral KPIs are based on literature and the knowledge of experts. The selected KPIs are divided into three groups: safety, throughput and outdated design. Furthermore, all the KPIs are standardized, this explained in Appendix D ,so the value of different TCS can be compared. All the KPIs will be taken into account, so the road authority can decide which KPI is the most valuable for their policy.

Chapter 4

Method

In the previous chapter, the theoretical background and the definitions of the performance indicators are given. In this chapter the cluster method and anomaly detection methods are described and it is explained how cluster and anomaly are defined. With these methods, groups and individual TCS which perform below average can be detected.

4.1 Cluster and anomaly detection methods

The problem in this research belongs to the area of unsupervised learning. The reason for this is that there is no data which can be marked as anomaly, since what consists an anomaly is not known beforehand. A cluster is a group of TCS that perform in the same way according to the values of the performance indicators. The traffic control systems which do not belong to one of the clusters are marked as an anomaly. The occurrence of clusters can indicate that some TCS groups perform in a homogeneous manner. In contrast, an individual TCS which does not belong to a cluster exhibits an abnormal behaviour compared to other TCS and can be marked as an anomaly.

The number of clusters is still unknown due to the unknown values of the data. This makes the use of clustering algorithms with a pre-set number of clusters complicated (Zong et al., 2018) (Al Tabash & Happa, 2018). Therefore, it was decided not to use unsupervised clustering algorithms such as K-means or Gaussian Mixture that require a pre-set number of clusters.

This results in two possible types of clustering, density-based and hierarchical based. The hierarchical cluster analysis treats each data point as a single cluster of the dataset and then successively merges pairs of clusters until all clusters have been merged into a single cluster that contains all data points (Seif, 2018) . Detection of anomalies in hierarchical

cluster analysis is often conducted with a density-based algorithm (Almeida, Barbosa, Pais & Formosinho, 2007)(Dey & Barai, 2017). A density-based clustering method can be used for both clustering and anomaly detection, therefore a density-based cluster method is used. The Density-Based Spatial Clustering of Applications with Noise (DBSCAN) will be used for clustering and anomaly detection and the outcome of the DBSCAN will be optimized and validated with the Silhouette coefficient. In addition, the Isolation Forest method is also used for anomaly detection. With the Isolation Forest method, an anomaly detection method, the TCS that can be isolated the easiest are identified as an anomaly, this is explained more extensively later on in this chapter.

4.2 Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

In the literature, several clustering methods are used to determine anomalies. The clustering method DBSCAN is often used in research with accurate results (Ranjith, Athanesious & Vaidehi, 2015) and yields more accurate results than the k-means method (Z. Chen & Li, 2011). In addition, the DBSCAN can handle scalability of the KPIs (in case there is a big difference between high and low-intensity TCS). Furthermore, there is no pre-set number of clusters needed (Seif, 2018). For these reasons, the DBSCAN will be used to cluster the data and detect anomalies. The DBSCAN algorithm is explained in table 4.1. The dataset (D) is in this research the list of 125 TCS with the corresponding calculated KPI values and p is one of the TCS. Since multiple KPIs/dimensions are used, the euclidean distance is useful to determine the distance between data points with multiple dimensions(Mumtaz, Studies & Nadu, 2010).

The following formula is used to calculate the euclidean distance:

$$d(a, b) = \sqrt{((a_1 - b_1)^2 + (a_2 - b_2)^2 + \dots + (a_n - b_n)^2)} \quad (4.1)$$

Simplified to:

$$d(a, b) = \sqrt{\sum_{i=1}^n (a[i] - b[i])^2}. \quad (4.2)$$

with:

1. a and b = a data point (TCS) with multiple dimensions (The KPIs)
2. $d(a,b)$ = the distance between point a and b .
3. n = the total number of used KPIs
4. i = the number of the KPI

The euclidean distance as described above, is used for the DBSCAN algorithm and this is described in table 4.1.

DBSCAN Algorithm
<p>Inputs:</p> <ul style="list-style-type: none"> • Dataset (D) • The distance of the neighbourhood (dist) • Minimum number of points (MinP)
<p>Output:</p> <ul style="list-style-type: none"> • Clusters • Anomalies
<p>Algorithm:</p> <p><i>Step 1:</i> Begin with one random data point from D, now called p the other points are q.</p> <p><i>Step 2:</i> Calculate the points within epsilon distance from p now called k, for k1 to ki.</p> $k1..i = euclidean\ distance(p, q) \leq eps(p)$ <p><i>Step 3:</i> Form a cluster with starting point p and neighbors k if the minimum number of points is reached and mark p as visited.</p> $i \geq MinP$ <p><i>Step 4:</i> Then repeat steps 2 and 3 for the neighbors, now called p, until all neighbors are reached.</p> <p><i>step 5:</i> Go to step 1 and pick a p which is not visited, until all the points in data set D are visited</p>

Table 4.1: DBSCAN Algorithm

The algorithm described in table 4.1 is visualised in figure 4.1. The radius of the circles in figure 4.1 represents the chosen epsilon and in this example the minimum number of points in this example is four. For every data point in the data-set, a circle is drawn and each point where the minimum number of points is reached can be marked as a core point. Data points with less than the minimum points in their circle but located in the circle of a core point are marked as border points (the yellow points in figure 4.1). The points with fewer points than the minimum points in their circle and not located in the circle of a core point are marked as an anomaly (the blue noise point in the example).

As mentioned in table 4.1 the DBSCAN has two input variables apart from the data-set: the distance to other data points and the minimum number of points. In this research the

A data-driven method to automate the detection of traffic control systems that do not 17 perform as intended

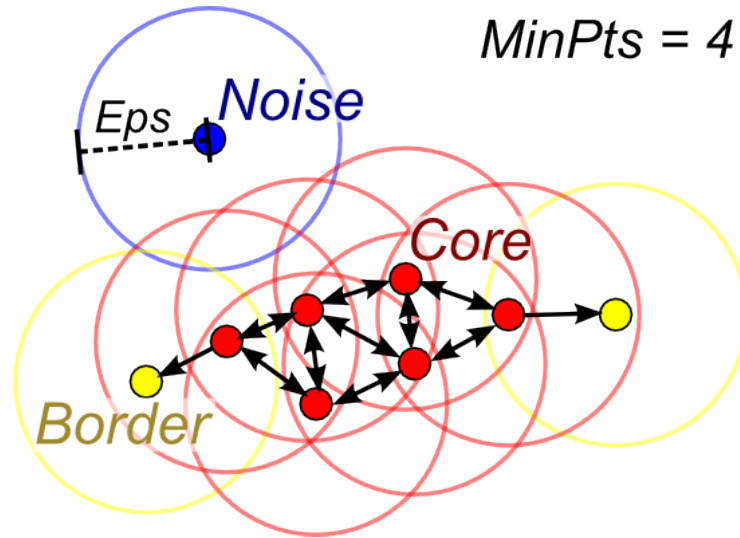


Figure 4.1: Explanation of the DBSCAN algorithm

distance is optimized with the Silhouette coefficient and differs for each group of KPIs the cluster method is conducted on.

The silhouette coefficient

The performance of the DBSCAN can be measured and if possible optimized by the silhouette score. The Silhouette score measure how well a data point is assigned to its cluster and to the other clusters (Evin Lutins, 2017). The silhouette score can be calculated for each data point (x) with the following formula (Rousseeuw, 1987):

$$s(x) = \begin{cases} 1 - \frac{a(x)}{b(x)}, & \text{if } a(x) < b(x) \\ 0, & \text{if } a(x) = b(x) \\ \frac{b(x)}{a(x)}, & \text{if } a(x) > b(x) \end{cases} \quad (4.3)$$

And can be summarized in one formula:

$$s(x) = \frac{b(x) - a(x)}{\max(a(x), b(x))} \quad (4.4)$$

With:

- $a(x)$ = the average distance between x , one TCS in this case, and all other data points, other TCS, within the same cluster.
- $b(x)$ = the smallest average distance of x , one TCS in this case, to all data points, other TCS, in any other cluster.
- $s(x)$ = silhouette coefficient

Based on the above definition:

$$-1 \leq s(x) \leq 1 \quad (4.5)$$

A score of $s(x)$ close to one means that the data is appropriately clustered. If $s(x)$ is a negative value, the data point x would be more appropriate if it was clustered in its neighbor cluster. When the value is about zero the $a(x)$ and $b(x)$ are approximately equal, it is not clear whether it should be assigned to cluster A or B (Rousseeuw, 1987). In this research, the hyper-parameters of the DBSCAN will be optimized based on the Silhouette coefficient.

4.3 Isolation forest

Since the problem is unsupervised and the data consist of multiple dimensions, an anomaly detection method is chosen which can deal with these conditions. Furthermore, for this research the classification of an outlier is important to detect TCS that perform below average. The isolation forest method is chosen since this method can handle unsupervised problems with multiple dimensions (Liu, Ting & Zhou, 2012). The isolation forest algorithm is an algorithm that uses isolation to detect anomalies instead of density or distance measures. The isolation forest are relative insensitive to redundant features, also it is rather unsensitive to the feature scaling (contrary to a distance based method such as DBSCAN). Several studies showed that isolation is a better indicator for anomaly detection than distance and density-based methods (Stripling, Baesens, Chizi & vanden Broucke, 2018) (Sun, Versteeg, Boztas & Rao, 2016).

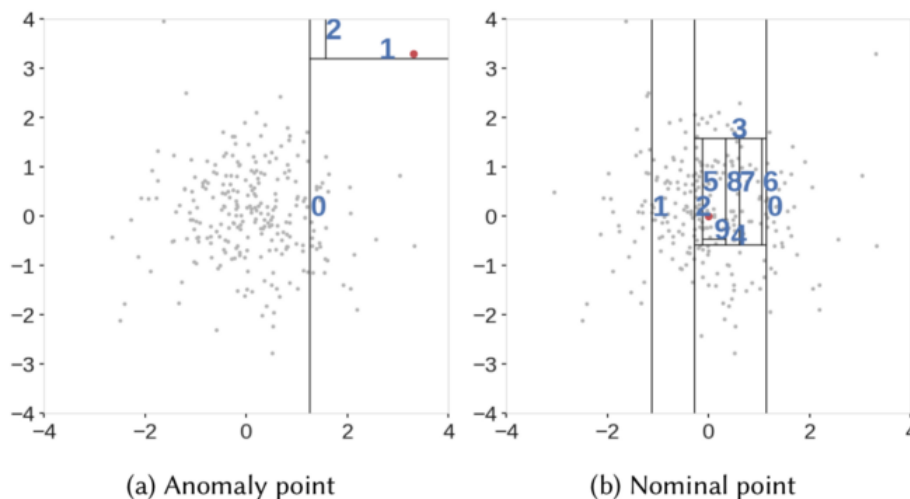


Figure 4.2: Explanation of the Isolation Forest algorithm

In figure 4.3 the explanation of the isolation forest is visualized. On the left side, only 3 splits are needed to isolate the red dot in contrast with 9 splits on the right side. The fewer

A data-driven method to automate the detection of traffic control systems that do not perform as intended

splits are needed to isolate a data point, the more likely it is an anomaly. The data will be split until all points are isolated. This is visualised in figure 4.3, where the small red path length identifies an anomaly and the blue longer path length a nominal data point.

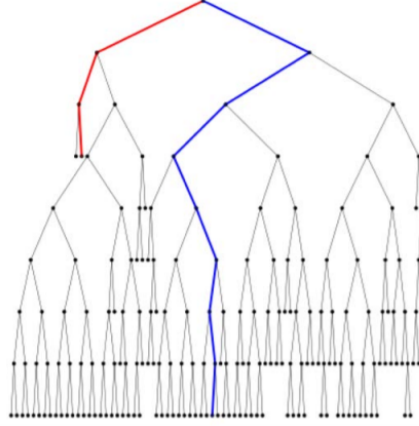


Figure 4.3: The Isolation Forest tree

This tree will be created 1000 times in this research, so the mean path length ($E(h(x))$) of all these trees can be calculated. Conducting this algorithm just once creates a biased outcome since the splits are done randomly. Therefore it can happen that a nominal point is isolated very quickly. Re-running the algorithm 1000 times prevents a biased outcome. The TCS with the highest anomaly score $S(x,n)$ (see equation 4.6), will be marked as an anomaly. In figure 4.3 a 2D example is given, in this research the dimension depends on the number of KPIs which are taken into account. In this research each KPI represents one dimension.

For this anomaly detection method an anomaly score is needed. For the isolation forest the following formula is used to calculate the anomaly score (Hariri, Kind & Brunner, 2018):

$$S(x, n) = 2^{-E(h(x,m,N))/c(n)} \quad (4.6)$$

The $c(n)$ can be calculated with the following formula:

$$c(n) = 2H(n - 1) - (2(n - 1)/n) \quad (4.7)$$

with:

- x = data point of the dataset, in this case, one TCS with corresponding KPIs.
- $h(x)$ = the path length of a data point until isolation.
- $E(h(x))$ = the mean path length of a TCS (x), of the 1000 isolation trees.

- N = Given dimensions of the dataset, number of used KPIs(8).
- n = number of data points used in the isolation tree.
- M = the total number of splits.
- k = total number of data points in the data frame, the 125 analysed TCS
- $H = \ln(i) + 0.577216$ (Euler's constant)
- $i = i \in (1, 2, \dots, N)$

Visualization

Both anomaly detection methods take every KPI, or a combination of KPIs (a policy group), into account. The different policy groups are explained in the next chapter. To enable 2D visualization, the dimensionality is reduced with PCA. With these two principal components, the outcome of the anomaly detection method will be visualized. The PCA algorithm will be briefly explained now. The first step is to transform the n (observations) \times m (variables) data matrix to a $m \times m$ covariance matrix. The second step is to extract the eigenvectors and eigenvalues of the covariance matrix. The eigenvectors are the principal components (PC) and give the direction of the PC. The eigenvalues represent the magnitude of PCs with other words the amount of variance explained by each PC.

$$Cx \cdot U = \lambda \cdot U \quad (4.8a)$$

$$Cx = (n - 1)^{-X^* \cdot X} \quad (4.8b)$$

with:

- Cx = The covariance matrix of the 125 TCS by 8 KPI matrix.
- X^* = data matrix with mean removed
- U = eigenvectors
- n = number of data points, the 125 analysed TCS.
- λ = eigenvalues (contains the amount of variance attributed to each PC)

For this research, the PCA package from sklearn is imported in python to conduct the PCA analysis. This method is only used to visualize the outcome of the anomaly detection methods and the method is applied to the data after the anomalies are calculated since information is lost by using the PCA.

Chapter 5

Case study

In this chapter, the results of the case study in the Netherlands are reported. In total 125 TCS are investigated, the data of one week was analysed. For all the TCS the week from 5 to 12 November 2018 and only the hours from 07:00 till 20:00 are used. The chapter starts with a section that explained how the data is prepared so the cluster and anomaly detection methods can be conducted. After that the performance of several TCS groups such as region, intensity and the presence of public transport are compared. The chapter ends with the result of the DBSCAN and Isolation Forest.

5.1 Data preparation

Since the format of the raw data is a binary and ASCII compressed V-log, the data follows a quite long trajectory before it can be used for anomaly detection. The path of the data is visualized in figure 5.1. During this path the data format is changed twice, from V-log to CSV file and from a CSV file to PostgreSQL table. During the analysis the useful data frames are saved as pickle, a compressed Python format that is fast in use. This saves a lot of time since retrieving data from the PostgreSQL server is time-consuming.

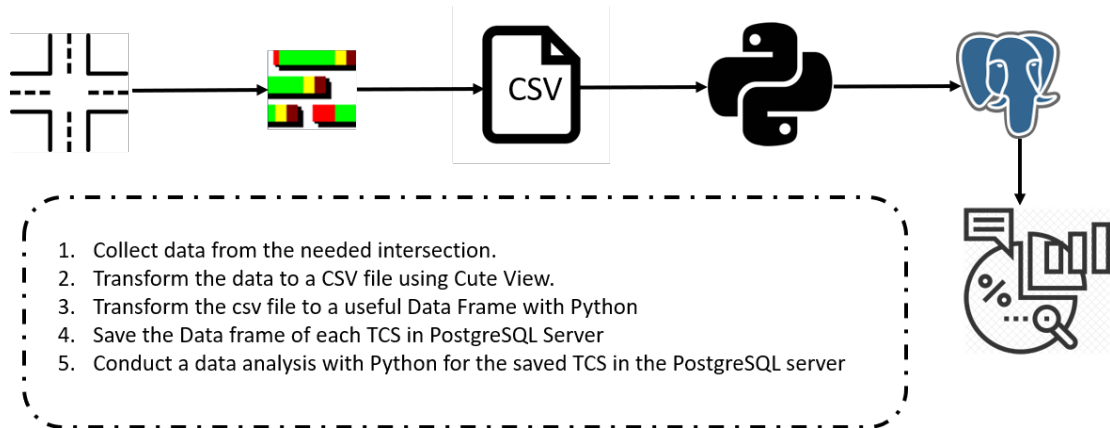


Figure 5.1: Trajectory of the data

5.2 Comparison of the performance of multiple TCS

For all the TCS the value of the KPIs are calculated for one week every hour between 07:00 and 20:00. Data of the evening or night cannot be taken into account since multiple TCS are in night modus and in this modus no KPI can be measured. For each hour the KPIs are calculated which corresponds with 91 measure points per KPI per TCS.

To compare multiple TCS, first the correlation between the KPIs is calculated to check whether all KPIs can be taken into account or not. The correlation gives info on which KPIs provide similar information, and therefore results get biased. In that case, one of the two KPIs must be removed from the subsequent analysis. A correlation could be negative, meaning that when one variable value increases, the other variables values decrease, or positive, meaning both variables move in the same direction. The correlation can also be zero or neutral, meaning that the variables are unrelated (Brownlee, 2018). The Pearson's correlation formula will be used:

$$\text{Pearson's correlation coefficient} = \frac{cov_{x,y}}{\sigma(X) * \sigma(Y)} \quad (5.1)$$

For each TCS the correlation between the KPIs is calculated, the mean value of these correlations is given in table 5.1. The table shows there are especially weak correlation recognizable between the KPIs, where 1 stands for a positive strong correlation and minus 1 for a negative strong correlation. Hence, it can be seen that the correlation between Fixed green and Early starters is the highest negative value (-0.67). This means that if the number of unused fixed green phases increases, the number of early starters for the same intersection decreases. Furthermore, the number of unused fixed green is the KPI which is highly correlated most often. It can be concluded that if the number of early starters, double stops, and unnecessary green given (after the last vehicle passes the intersection) is decreasing, the number of unused green increases at the intersection.

A data-driven method to automate the detection of traffic control systems that do not 23 perform as intended

Table 5.1: Correlations

KPI	RLR	90	FG	PG	UG	DS	ES	Flut
Red light runners (RLR)	1	-0.04	0.20	0.15	-0.14	-0.14	-0.17	0.16
90 seconds waiting (90)	-0.04	1	-0.29	-0.18	0.17	0.46	0.28	0.12
Fixed green (FG)	0.20	-0.29	1	0.57	-0.62	-0.64	-0.67	-0.23
Prolonging green (PG)	0.15	-0.18	0.57	1	-0.41	-0.43	-0.44	-0.18
Unnecessary green (UG)	-0.14	0.17	-0.62	-0.41	1	0.45	0.55	0.22
Double stops (DS)	-0.14	0.46	-0.64	-0.43	0.45	1	0.53	0.18
Early starters (ES)	-0.17	0.28	-0.67	-0.44	0.55	0.53	1	0.24
Fluttering (Flut)	0.16	0.12	-0.23	-0.18	0.22	0.18	0.24	1

The next step is to calculate the standard deviation of the correlations to check whether the correlation is consistent or not. The standard deviation is calculated and displayed in table 5.2.

Table 5.2: The standard deviation of the correlation between the KPI's

KPI	RLR	90	FG	PG	UG	DS	ES	Flut
Red light runners (RLR)	0.0	0.255	0.327	0.306	0.305	0.313	0.293	0.338
90 seconds waiting (90)	0.255	0.0	0.235	0.215	0.300	0.292	0.283	0.252
Fixed green (FG)	0.327	0.235	0.0	0.2657	0.301	0.182	0.272	0.292
Prolonging green (PG)	0.306	0.215	0.266	0.0	0.290	0.210	0.309	0.244
Unnecessary green (UG)	0.305	0.300	0.301	0.290	0.0	0.357	0.332	0.283
Double Stops (DS)	0.313	0.292	0.182	0.210	0.357	0.0	0.276	0.296
Early Starters (ES)	0.293	0.283	0.272	0.309	0.332	0.276	0.0	0.293
Fluttering (Flut)	0.338	0.252	0.292	0.244	0.283	0.296	0.293	0.0

This table shows that most of the standard deviation values are around 0.3. This is a quite high number since the value of the correlation is always between -1 and 1. The lower the value of the standard deviation is, the more consistent the correlation is between the KPIs. The mean value of the correlation in table 5.1 plus-minus the standard deviation gives a confidence interval of 69%. The mean value plus-minus two times plus-minus the standard deviations give a confidence interval of 95%. In this case, the standard deviation is around the 0.3, two time plus-minus corresponds with a range of 1.2. This high range for the correlation implies that the correlation between the KPIs for the different TCS is uncertain. According to these tables, there is no strong consistent correlation recognizable between KPIs. That is, all the KPIs have to be taken into account for detecting the anomalies and clusters.

The number of KPIs is known so the influence of the changing traffic demand during the day on the TCS can be calculated. Literature shows multiple times that there is a big

difference in traffic demand over time (Dunn Engineering Associates, 2005) (Lajunen & Özkan, 2011) (Zedgenizov & Burkov, 2017). To check whether the data must be separated in multiple time segments before applying the anomaly detection methods.

To check whether the data must be separated two analysis are conducted. First, the mean correlation with the intensity and standard deviation is calculated in table 5.3. Remarkable is that 6 of the 8 KPIs are relatively strongly correlated with the intensity. Only the percentage of red-light runners and fluttering are barely correlated with the intensity of the intersection. The explanation is obvious since the percentage of red-light runners are standardized by the intensity and fluttering is an error in the detection loop which can occur at every intersection. In addition, the standard deviation of these KPIs is quite high. This means the small correlation is very uncertain. The early starters, waiting time over 90 seconds, unnecessary green given after the last car enters the intersection and double stops are positively correlated with the intensity. That is to say, when the intensity is increasing at an intersection, the values of these KPIs will also increase. On the other hand, the number of unused fixed green and the value of the prolonging green use decrease. A decrease in the value of prolonging green use means a more effective way of using this phase. The standard deviation for the highest correlated KPIs is relatively low, hence it can be concluded that the intensity of the intersection influence the value of the KPI. This consistent correlation implies that the values of the KPIs are higher during the peak hours of the day.

Table 5.3: The mean standard correlation and standard deviation with the intensity

KPI	mean correlation intensity	mean standard deviation
Red light runners (RLR)	-0.19	0.34
90 seconds waiting (90)	0.42	0.28
Fixed green (FG)	-0.76	0.21
Prolonging green (PG)	-0.56	0.27
Unnecessary green (UG)	0.61	0.35
Double Stops (DS)	0.77	0.19
Early Starters (ES)	0.71	0.30
Fluttering (Flut)	0.29	0.36

Second, the findings of table 5.3 are visualized in figure 5.2. In this figure the mean value per hour of four KPIs during one week is calculated and plotted. The values for the other four KPIs are higher, so these are plotted in other figures which are placed in Appendix D. Furthermore, figure 5.2 shows that the values of most KPIs have a peak during the morning and evening hours. This finding confirms the previous finding and shows there are significant differences during the day. Therefore the data is separated in multiple time segments before applying the anomaly detection methods.

The used data is from three different regions, therefore the differences are briefly explained. Deventer is the smallest city with no presence of trams and the maximum speed is not

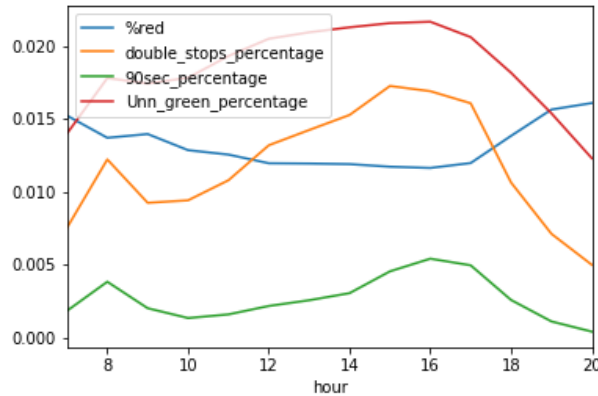


Figure 5.2: mean value of a KPI during the day.

higher than 50 km/h. Den Hague is a bigger city where the intensity at the TCS is higher and trams are often present. All the TCS are located in a city where the maximum speed is not higher than 50km/h. The TCS in the province of North-Holland are not located in the city centers, so the maximum speed is higher than 50km/h and there are no trams.

The next step is to calculate the difference in the performance of the TCS in the different regions. If the performance of a region is significantly worse, this might be interesting for the corresponding road authority. In addition, the influence of the presence of public transport and cyclists and traffic intensity is checked. A very useful tool to see correlation between KPIs and also split the KPIs into groups is the pairwise scatter plot.(Shao et al., 2016). For this scatter plot, the values of the KPI are presented on the x-axis and y-axis. The diagonal axis will present a density diagram for each performance indicator.

In figure 5.3 the mean value of the KPIs for the whole week from 07:00 until 20:00 is used. In the distribution of the dots (TCS) is no pattern recognizable, this confirms the statement that there is no significant correlation between the KPIs. Furthermore, the TCS are separated by region: each color represents another region. For the recognition of the differences between the regions, the three different density diagram per KPI are compared. The following remarkable differences based on the regions are:

1. In the province of Noord Holland, the unused fixed green phase occurs less.
2. In the province of Noord Holland there are less double stops.
3. In the city of Deventer there are less early starters.
4. In the city of Deventer, fewer vehicles have to wait more than 90 seconds for the TCS.

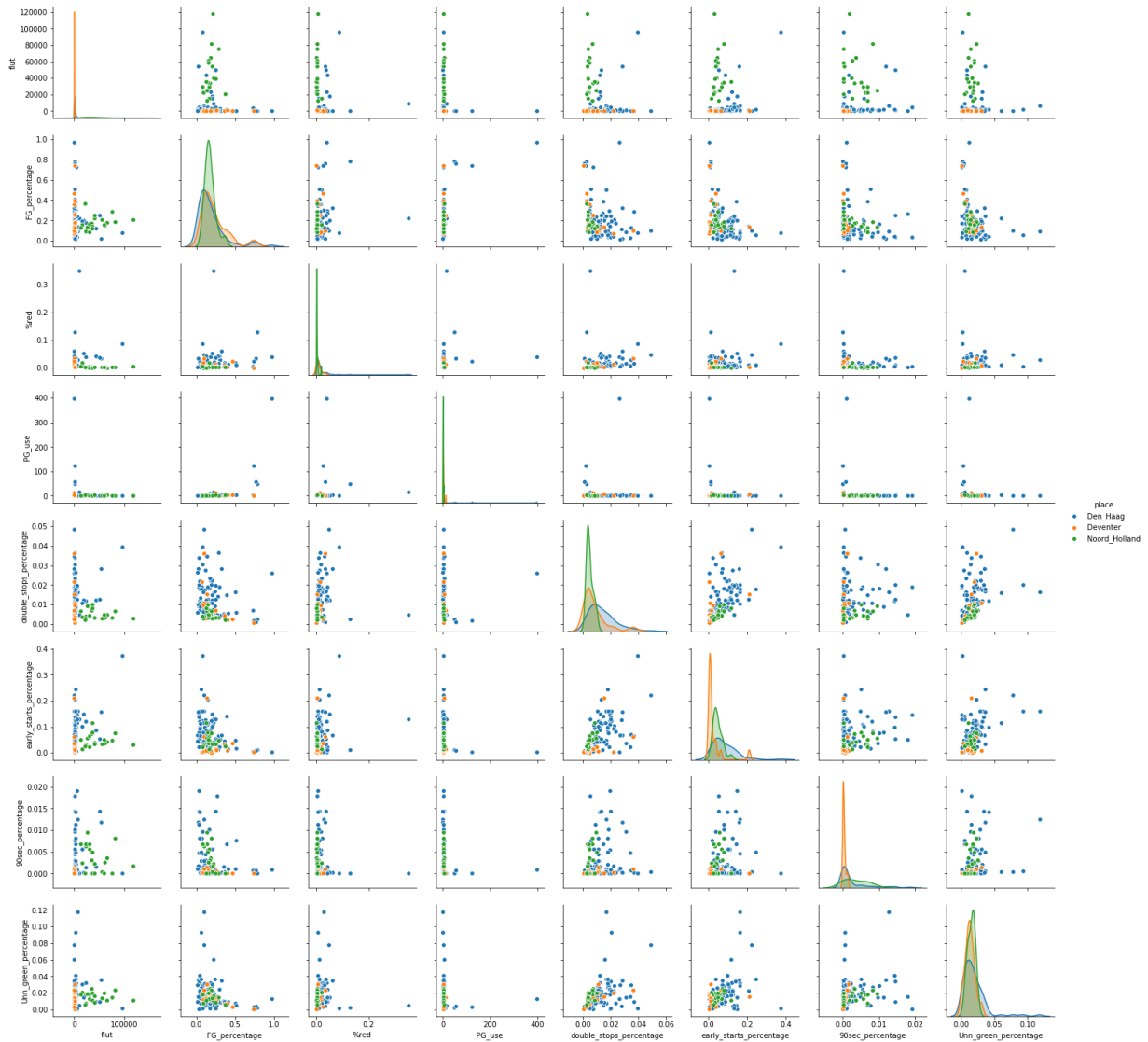


Figure 5.3: mean value of a KPI during the week.

The pairwise scatter plot is also made and analysed for the intensity group, presence of public transport and cyclists. For intensity the TCS are separated into 3 groups: high, medium and low intensity, with at least 40 of the 125 TCS in each group. For each factor, the remarkable findings are summed up. The presence of public transport:

1. The fluttering is lower with the presence of public transport.

The presence of cyclist:

1. The fluttering is lower with the presence of cyclist.

2. The number of double stops is higher with the presence of cyclist.
3. The number of early starters is higher with the presence of cyclist.

The intensity level of the TCS:

1. The fixed green phase is more often not used at TCS with low intensity.
2. The percentage of double stops is lower at an intersection with a high-intensity level.
3. The number of early starters is higher at TCS with medium intensity.
4. For TCS with a low-intensity level, fewer vehicles have to wait more than 90 seconds for the TCS.

5.3 Clustering and anomaly detection

The DBSCAN algorithm detects the points in regions with low density as an anomaly. The anomalies of the DBSCAN are the points that do not fit in a cluster. One constraint is added for the optimization of the anomaly detection, at least 2 of the 125 TCS must be detected as an anomaly. Since a percentage of anomalies below 1% is unlikely, keep in mind that the current maintenance level is not that high (see figure 1.1). The validation and optimization of the DBSCAN are done with the Silhouette coefficient.

Since there is a significant difference in the value of the KPIs during the day, the cluster and anomaly detection method is conducted for different time segments. In addition, a distinction between the three groups, mentioned in table 3.1, was made (safety, throughput, outdated design, altogether). So in total, the methods are conducted sixteen times to check the occurrence of clusters and/or anomalies. The cluster method is validated with the Silhouette coefficient, this is where the Silhouette has reached the maximum value. This maximum value of Silhouette coefficient is reached by changing the value of the epsilon, the value of the minimum number of TCS needed for a cluster is constant with a value of three.

The minimum number of points is set as a constant variable of 3 because the results of the previous question show that some extreme values occur by one or two TCS. By setting the minimum points to 3 the extreme values will be counted as anomaly and not as cluster, which is more suitable in this situation. The algorithm described in table 4.1 is used in Python. Therefore the DBSCAN package from sklearn is imported in Python and the data frame is first transformed to a data frame with a standard scale by using the StandardScaler from sklearn. With this standardized data frame, the cluster method is conducted to check whether there are clusters or not.

The outcome for every conducted cluster method is consistent and in all situations, no clusters occur. The Silhouette reached the maximum value when all the TCS are clustered into one cluster. In figure 5.4 one of the sixteen times that the DBSCAN is conducted is shown and indicate there is only one cluster. The calculated output of the DBSCAN:

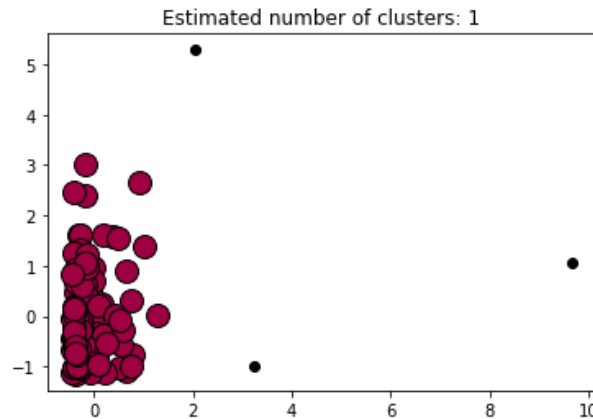


Figure 5.4: Cluster method with a week of data and the safety KPI group.

1. Number of clusters: 1
2. Number of noise points: 3
3. Silhouette Coefficient: 0.802

The fifteen other times the DBSCAN is conducted, are similar to the one shown in figure 5.4. This corresponds with the outcome shown in figure 5.3 since in this figure no patterns or groups are recognizable too. The anomaly detection method based on the DBSCAN is also shown in figure 5.4, the noise points of the cluster method are the anomalies.

For the visualization of the Isolation forest method, the Principle Component Analysis (PCA) is used. It is important to conduct the Isolation forest before the PCA because some characteristics of the data change after the PCA. By checking the plots visually, can be determined whether the anomalies are likely or not (located in an area with fewer other data points nearby). An example is given in figure 5.5.

Both anomaly detection methods show that most of the data points are located together in one group. Most of the anomalies are very isolated which confirms abnormal behavior of a TCS. For both anomaly detection methods the outcome is plotted and manually (visual) checked whether the anomaly is an extreme value.

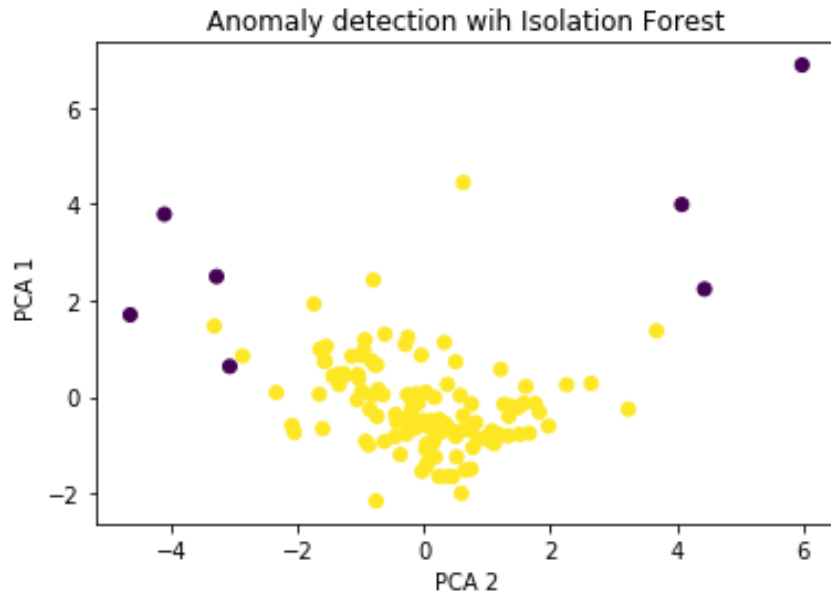


Figure 5.5: Isolation Forest during the evening peak and all KPI's are taken into account.

Concluding:

1. Clear anomalies can be found using these methods.
2. All "normal" behaviors cluster into 1 cluster, just as you would expect from policy and geographically neutral KPIs

5.4 Performance and validation

The performance of the TCS is checked in multiple ways. The data is separated in several time segments, the morning peak (07:00-09:00), the noon hours (12:00-1500), evening peak (16:00 19:00) and the whole day 07:00-20:00). In addition, the KPIs are divided into the groups classified in chapter 3 (safety, throughput and outdated design) and all the KPIs together (all group in table 5.4). Due to this separation, both anomaly detection methods are conducted sixteen times. The performance of the TCS in difference groups will be compared.

In table 5.4 all the anomalies which are detected per method and group are listed. Big differences in anomalies between the different policy groups are recognizable. This indicates that a single TCS does not perform the worst in multiple groups. This is in line with the data, where for every KPI different TCS score the worst. The differences between the time segments are minimal so if a TCS performs worse it is consistent during the day.

Table 5.4: Anomalies per group

Time/KPI combination	DBSCAN	Isolation Forest
whole day - all	0, 1,13,19,20,36,38	0, 1,13,19,20,38,79
whole day - safety	20,36,38	1, 36, 38
whole day -throughput	0,1,7	1,11,19,31
whole day - outdated	13,44	13,44
morning - all	0,1,7,13,36,38	0,13,19,20,36,38,79
morning - safety	36,38	36,38
morning - throughput	0,1,7	0,19,33
morning - outdated	13,44,118	13,44
noon - all	1,11,13,20,36,38	0,1,13,19,20,30,38
noon - safety	20,36,38	1,36,38
noon - throughput	0,1,11	0,11,19
noon - outdated	13,44,118	13,44,20
evening - all	0,1,13,20,36,38	0,1,13,19,20,38,43
evening - safety	20,36,38	1,38,58
evening - throughput	0,1,7	0,11,19
evening - outdated	13,44	13,44,110

In total 19 of the 125 TCS are detected as an anomaly. This is a percentage of 15.2%. Interestingly, the Isolation Forest method detects more different TCS than the DBSCAN. To make this more clear, the number of times a TCS is counted as anomaly is summarized in table 5.5. This shows that all the TCS which are detected as anomaly once are detected by the Isolation Forest. During the discussion with experts and road authorities, the

Table 5.5: number the TCS is counted as anomaly

Anomaly	count as anomaly
38,13	16
1	15
36,0	12
20	11
19	9
44	8
11	5
7	4
118,79	2
110,58,43,33,31,30,	1

remarkable findings in Cuteview (a program to visualise the data of 1 TCS). For this reason, the outcome is visually checked with Cuteview, this provides as much additional information about the TCS performance. After this analysis the question: is it possible to automatically identify TCS which performs below average with unsupervised machine learning methods? can be answered.

Visual Cuteview analysis

All the 125 TCS are checked and especially the TCS with KPIs with higher values than average shows deviant behaviour. These results are discussed with the experts concluding that TCS settings might be wrong in these situations:

1. The high value of the use of the prolonging green phase (PG) is caused by switching the prolonging green phase to the waiting green phase if a vehicle hits the stop line detector loop. Both high values occur at intersections with low intensity. (anomaly 13, 23 and 44)
2. The number of double stops for anomaly 1 and 38 is high because the maximum green time is not long enough to provide green for every vehicle. But the percentage of waiting time longer than 90 seconds is 0.
3. The high percentage of unnecessary green is higher for anomaly 0 and 7 because two long loop detectors are constantly occupied. Due to this occupancy, the extending green phase reached always the maximum time and almost every time unnecessary green occurs.
4. The high number of red-light runners for anomaly 36 is caused by fluttering.
5. Anomaly number 20 is a TCS with very low intensity-level, which often causes an unused fixed green phase. This low-intensity level also causes a high percentage of red-light runners, people are more likely to hit the red light when no other vehicles are detected at the intersection.
6. The high percentage of vehicles which has to wait over 90 seconds for anomaly 11 is probably caused by the use of extending and prolonging green phase which is always 0.1 seconds (only fixed and waiting green is used). In addition, no green request is recognizable. Due to this, all the signal groups give a long green time which causes a waiting time of 90 seconds quite often.

Comparison between the anomaly detection methods

To make a comparison between the different anomaly detection methods, the detected anomalies are divided into three groups: detected with the DBSCAN, the Isolation Forest en detected by both methods. The detection rate and False detection rate are determined as follows:

$$\text{Detection rate} = \text{true detections} / \text{true anomalies}$$

$$\text{False detection rate} = \text{false detections} / \text{all detections}$$

Table 5.6: Performance of the anomaly detection methods

	DBSCAN	Isolation Forest	Both
Detection rate	9 out 11	10 out 11	9 out 11
False detection rate	2 out 11	6 out 16	1 out 10

Table 5.6 shows that the Isolation Forest has the highest detection rate but also the False detection rate is also quite high. The detection rate of the DBSCAN and the anomalies detected with both methods is the same, but the False detection rate is higher for the DBSCAN. Since the false detection rate of the isolation forest is quite high, can be concluded that using both methods together gives the best results. The answer to the question; is it possible to automatically identify TCS which performs below average with unsupervised machine learning methods? is yes. Using machine learning will save a lot of time because no visual check, which is labor-intensive, is needed.

Characteristics of the anomalies

For all the 11 anomalies the characteristics are compared to check whether there are groups of TCS which are more likely to be an anomaly.

Table 5.7: Characteristics of the anomalies

City	presence of:	Intensity			
The Hague 11/11	Public transport 6/11	Cyclist 11/11	Low 6	Medium 4	High 1

Concluding the findings of table 5.7:

- All the anomalies are located in the city of the Hague. It was expected that most of the anomalies would be located in the city of The Hague since 80 of the 125 analysed TCS are located in this city. However, anomalies in other cities were expected too since 1/3 of the data is from the other cities.

- The presence of public transport does not influence whether the TCS is an anomaly or not since only at 6 of the 11 anomalies public transport is presented (for around 60% of the 125 TCS public transport is presented).
- At all the anomalies cyclist are present which indicate that the presence of cyclist has a negative influence on the performance of the TCS (for around 65% of the TCS cyclist are presented).
- The TCS are divided into 3 groups based on their intensity-level, each group with at least 40 TCS attended. however, only 1 TCS with a high intensity-level is identified as an anomaly.

Chapter 6

Discussion and limitations

The problem researched in this study is the deterioration of the TCS performance over time in combination with a lack of maintenance. Comparing multiple TCS and detecting which TCS performs under average is still a gap in literature. In this research, unsupervised data-driven machine learning methods were used to fill this gap.

The high value of the standard deviation of the KPIs shows that the TCS does not behave in the same way, and this is in line with the expectations and knowledge of the experts. There is no TCS with exactly the same characteristics such as the traffic demand, policy, and local specifications. This finding confirmed the need for data of multiple TCS (100+), so the TCS can be divided into several groups with equal characteristics. This makes a comparison between the groups possible which can lead to more general conclusions and recommendations according to targeted maintenance.

The results of the pairwise scatter plots indicate that there are significant differences in the performance of the TCS in different groups. These findings are hard to compare or validate with other methods since the comparison of multiple TCS is still a gap in the literature. The significant differences in the performance of TCS of different regions can be a good incentive for the road authority to examine what causes these differences. For example, the double stops in the province North-Holland are significantly lower than the other regions (a low chosen maximum green time for a direction at the TCS could cause unnecessary double stops). A notable finding is that the percentage double stop at TCS with high-intensity levels are significantly lower, since a higher value was expected. According to the experts, it is easier to optimize the performance of the TCS when the high intensities are expected. These results suggest that TCS where normally the intensity is low or on medium-level cannot handle extreme peak (high traffic demand) moments well.

The cluster method shows that even if the data is divided into multiple groups there are no

clusters recognizable. This indicates that there is no factor that influences the performance of the TCS in such a way that a distinction between the TCS is recognizable. Probably there are contradictory factors (for example, the region and intensity level could be contradictory) that influence the value of the KPIs and thus create a more homogeneous dataset. So can be concluded that the policy- and geographically-neutral performance indicators fulfil their function to exclude the influence of policy and geographical characteristics.

To question: is it possible to automatically identify TCS which performs below average with unsupervised machine learning methods?, can be answered with yes. The pairwise scatter plot (see figure 5.3), the cluster method (see figure 5.4), and the anomaly detection method (see figure 5.5) shows that most of the TCS are located in one big group with a few extraordinary values outside this group. This indicates that for every data-group the anomalies represent extreme values. Due to the use of the data of more than 100 TCS these extreme values are a good incentive to check what causes these extreme values. The presence of these extreme values of a KPI indicates that a data-driven method could be useful to provide targeted maintenance.

The outcome in table 5.4 reveals that the differences between the time segments are minimal. This indicates that if a TCS performs worse, this is consistent during the day. This is in contrast with the expectation of some experts, who thought that some TCSs are optimized with a focus on the peak hours and that the TCS will perform worse during the off-peak hours. This statement still cannot be disproven since the values of the anomalies are consequently higher during the whole day, but larger difference between peak and noon hours was expected.

The performance of the anomaly detection methods presented in table 5.6 shows that the detection rate of the isolation forest is 9% higher. On the other hand, the false detection of the isolation forest is 18% higher than the DBSCAN and 28% higher in comparison with the anomalies identified by both methods. Concluding these findings, the most useful way to detect anomalies is to use both methods and mark the TCS which are detected by both methods as an anomaly. For this method the false detection rate is the lowest which fits the problem of limited resources available for retiming. When the false detection rate and the number of detected anomalies are high, the available resources could still be a problem.

The most remarkable finding from table 5.7 is that only one TCS with high intensity is identified as an anomaly and for 6 out of 11 anomalies, the intensity-level was low. A possible explanation is that the road authorities focus is especially on the busy intersection of the city and the intersections with a low intensity-level do not have the priority for conducting maintenance.

Limitations

- The used V-log data is not easy in use and before the data analysis can be performed the format of the data has to change multiple times. Due to the changes in extension, the probability to lose or change some characteristics of the data will increase. In addition, CuteView is used for the first step in the conversion of the data, how the program converts the data is a black box and was assumed to work properly here.
- The missing benchmark for the performance of TCS creates an unsupervised problem. The validation of the outcome can only be done with the knowledge of experts and the comparison between the analysed TCS. This makes optimization of the used methods challenging.
- With CuteView it is only possible to convert data where the date-time is consecutive. Furthermore, very often it was not possible to load more than one week of data into CuteView. Due to this limitation the analysed period of all TCS is only one week.
- The saved V-log data is quite often not complete for the entire week. Multiple TCS are removed from the dataset due to this incompleteness. The data files for every time segments are present, but sometimes these files are empty or the data of the internal phases of the signal groups are missing. This excluded multiple TCS from the analysis.
- The running time for the calculation of the performance indicators is very long, small changes in analysis take a long time to calculate for all the data. In addition, the time for this research was limited. Due to these reasons, there was not enough time to conduct some further analysis. In the recommendations, the analysis which would be interesting for future research is shown.

Chapter 7

Conclusions and recommendation

The method proposed in this research represents a good and quick view of the performance of several pre-selected groups such as the region. By analysing the difference in the performance of the different groups of TCS, several general conclusions can be made about the performance. For example, the higher percentage of a double stop at intersections with a low-intensity level. This can be an incentive to check whether the maximum green time of these intersections is long enough. The anomaly detection method is a suitable solution to detect TCS with unusual performance values since several TCSs with extreme value are presented in the data. This extreme value does not guarantee that for this TCS retiming would be the most valuable, but it is very likely (according to the data and experts). To improve this method, the improvement of the TCS performance after retiming must be logged.

By dividing the KPI into three policy groups, the road authority can choose which performance indicators are more valuable for their policy. This method can help by providing targeted maintenance for TCS, and it can be used as a quick performance scan. After this method is used the choice can be made to investigate the anomalies further to find the reason which causes that the TCS is an anomaly and then to conduct maintenance or not.

The three main conclusion of this research:

1. Checking the performance for pre-selected groups of TCS (region and intensity) interesting findings according to the performance can be found.
2. All normal behaviors cluster into 1 cluster, just as you would expect from policy and geographically neutral KPIs
3. Clear anomalies can be found using unsupervised learning methods.

The practical implementation of the method

Multiple parties use V-log data and perform analysis with this data. Each party uses their

own programs and codes. For business reasons, most of the parties do not share these programs. This holds back fast development in data-driven analysis to provide targeted maintenance for TCS. With good cooperation between several parties, a big step forward to a versatile data analysis of TCS can be made.

An additional issue is a missing benchmark for the performance of the TCS, the only written directive is the maximum cycle length of 90 seconds. This research can help by setting up a benchmark since in this research the data of 125 TCS was compared. In addition, the TCSs are located in different regions of the Netherlands. Due to this missing benchmark, it is hard to assess the performance of an individual TCS.

The code behind the analysis is probably not easy to understand for the policymakers and changing input variables is impossible for people who do not know the programming language. A very clear dashboard is needed for the policymakers so they can change the input variables as they seem fit. In addition, the meaning of the output of some analysis must be explained very clear since machine learning is not well-known to all the road authorities. With this feature tailor-made targeted maintenance planning can be made.

Future research

If the improvement of the TCS performance (travel costs) after retiming is logged, this data can be analysed. The data before and after a retiming of multiple TCS in combination with the improvement in travel costs can be analysed. This provides a method to decrease the total travel costs by carrying out targeted maintenance at TCS.

In the discussion, it was already mentioned that only data of consecutive days can be converted at the same time. For future research it would be interesting to use data of multiple years, so the deterioration during this period can be taken into account. In this research only TCS with higher KPI values are detected as anomaly. It can happen that a TCS with a consequent higher value of the KPI has a deterioration of only 1 % during the year and a TCS with lower value of the KPI has a deterioration of 25%. In this case, the TCS with a deterioration of 25 % could be more interesting.

In this research already some machine learning methods are used which can handle unsupervised problems. If there are true values available (i.g. a TCS which performance is proven to be not sufficient) or a benchmark, supervised machine learning methods are possible. which create a lot more possibilities. For example, forecasting when a TCS perform beneath a certain level is possible. With this feature tailor-made targeted maintenance planning can be made.

References

- Al Tabash, K. & Happa, J. (2018). Insider-threat detection using Gaussian Mixture Models and Sensitivity Profiles. *Computers & Security*, *77*, 838–859. Retrieved from <https://doi.org/10.1016/j.cose.2018.03.006> doi: 10.1016/j.cose.2018.03.006 15
- Alesiani, F., Moreira-Matias, L. & Faizrahnemoon, M. (2018). On learning from inaccurate and incomplete traffic flow data. *IEEE Transactions on Intelligent Transportation Systems*, *19*(11), 3698–3708. doi: 10.1109/TITS.2018.2857622 8, 13
- Almeida, J. A. S., Barbosa, L. M. S., Pais, A. A. C. C. & Formosinho, S. J. (2007). Improving hierarchical cluster analysis : A new method with outlier detection and automatic clustering. , *87*, 208–217. doi: 10.1016/j.chemolab.2007.01.005 16
- Ambühl, L. & Menendez, M. (2016). Data fusion algorithm for macroscopic fundamental diagram estimation. *Transportation Research Part C: Emerging Technologies*, *71*, 184–197. doi: 10.1016/j.trc.2016.07.013 9
- Andrea, E. D. & Marcelloni, F. (2016). Detection of Traffic Congestion and Incidents from GPS Trace Analysis. *Expert Systems With Applications*. Retrieved from <http://dx.doi.org/10.1016/j.eswa.2016.12.018> doi: 10.1016/j.eswa.2016.12.018 3
- ANWB. (2018). *No Title*. Retrieved from <https://www.anwb.nl/experts/juridisch/23/onlogische-afstelling-verkeerslichten-in-nl> 1
- Baneke, I. (2016, feb). *Laks beheer verkeerslichten leidt tot onnodige ongelukken*. 1
- Bijl, B. V. D., Vreeswijk, J., Bie, J. & Berkum, E. V. (2011). Car Drivers' Perception and Acceptance of Waiting Time at Signalized Intersections. , 1–6. 8
- Blokpoel, R., Vreeswijk, J., Krajzewicz, D. & Kless, T. (2014). UNIFIED EVALUATION METHOD FOR TRAFFIC CONTROL ALGORITHMS. In *Conference: Its world congress 2014*. 7
- Blokpoel, R. I., Caselli, F. U., Härri, J. E., Niebel, W. D. & Leich, A. D. (2015). Cooperative Self-Organizing System for low Carbon Mobility at low Penetration Rates. 7
- Blokpoel, R. J., Krajzewicz, D. & Nippold, R. (2010). Unambiguous metrics for evaluation of traffic networks. , 1277–1282. 8
- Brownlee, J. (2018). *How to Use Correlation to Understand the Relationship Between Variables*. Retrieved from <https://machinelearningmastery.com/how-to-use-correlation-to-understand-the-relationship-between-variables/> 23

- Chen, P., Yu, G., Wu, X., Ren, Y. & Li, Y. (2017). Estimation of red-light running frequency using high-resolution traffic and signal data. *Accident Analysis and Prevention*, 102, 235–247. Retrieved from <http://dx.doi.org/10.1016/j.aap.2017.03.010> doi: 10.1016/j.aap.2017.03.010 11, 13
- Chen, Z. & Li, Y. (2011). Anomaly Detection Based on Enhanced DBScan Algorithm. *Procedia Engineering*, 15, 178–182. Retrieved from <http://dx.doi.org/10.1016/j.proeng.2011.08.036> doi: 10.1016/j.proeng.2011.08.036 16
- College, S. M. (2001). SYSTEMS ENGINEERING FUNDAMENTALS. (January). 46
- Dey, L. & Barai, A. (2017). No Title. , 24–29. doi: 10.13189/wjcat.2017.050202 16
- Dijkstra, A. (2014). *Naar meer veiligheid op kruispunten* (Tech. Rep.). 1
- DTV Consultants B.V. & Willekens, M. (2016). Verkeerslichten in Nederland 2015. 2, 51
- Dunn Engineering Associates. (2005). TRAFFIC CONTROL SYSTEMS HANDBOOK. *Federal Highway Administration*(October). 1, 2, 5, 6, 25
- Evin Lutins. (2017). *DBSCAN: What is it? When to Use it? How to use it.* Retrieved from <https://medium.com/@elutins/dbscan-what-is-it-when-to-use-it-how-to-use-it-8bd506293818> 18
- Gordon L, R. (2010). *Traffic Signal Retiming Practices in the United States.* doi: 10.17226/22915 6
- Guardiola, I. G., Leon, T. & Mallor, F. (2014). A functional approach to monitor and recognize patterns of daily traffic profiles. *Transportation Research Part B: Methodological*, 65, 119–136. Retrieved from <http://dx.doi.org/10.1016/j.trb.2014.04.006> doi: 10.1016/j.trb.2014.04.006 9
- Hariri, S., Kind, M. C. & Brunner, R. J. (2018). Extended Isolation Forest. , 1–11. Retrieved from <http://arxiv.org/abs/1811.02141> 20
- Jin, X., Zhang, Y., Li, L. & Hu, J. (2008). Robust PCA-Based Abnormal Traffic Flow Pattern Isolation and Loop Detector Fault Detection. *Tsinghua Science and Technology*, 13(6), 829–835. doi: 10.1016/S1007-0214(08)72208-9 8
- Kan, Z., Tang, L., Kwan, M.-p., Ren, C., Liu, D. & Li, Q. (2018). Computers , Environment and Urban Systems Traffic congestion analysis at the turn level using Taxis ' GPS trajectory data. *Computers, Environment and Urban Systems*(April), 0–15. Retrieved from <https://doi.org/10.1016/j.compenvurbsys.2018.11.007> doi: 10.1016/j.compenvurbsys.2018.11.007 3
- Kant, M. & Koenen, Y. (2017). *Verkeersstudie VRI's Soest-Zuid* (Tech. Rep.). Soast. 7
- Kaparias, I., Zavitsas, K. & Tomassini, M. (2010). State-of-the-art of urban traffic management policies and technologies. (1). 49
- Katwijk, R. T. V. (2008). *Multi-Agent Look-Ahead Traffic-Adaptive Control.* v, v, v, v, v, v, 49, 51, 52
- Koningsbruggen, P. v. (2016). *Tijd voor een nieuwe generatie VRI's.* Retrieved from <https://www.technolution.eu/nl/over-ons/publicaties/407-de-noodzaak-tot-innovatie-van-verkeerslichten.html> 1
- Krajzewicz, D., Blokpoel, R. & Cornelia, H. (2014). Towards a unified Evaluation of Traffic Light Algorithms. (July). 7

- Krol, L., Prinsen, L. & Misdom, L. (2018). Onderzoek detectieconfiguratie en signaalgroepafhandeling. 50
- Kumar, P. M., G, U. D., Manogaran, G., Sundarasekar, R., Chilamkurti, N. & Varatharajan, R. (2018). Ant colony optimization algorithm with Internet of Vehicles for Intelligent Traffic Control System. *Computer Networks*. Retrieved from <https://doi.org/10.1016/j.comnet.2018.07.001> doi: 10.1016/j.comnet.2018.07.001 2
- Lajunen, T. & Özkan, T. (2011). Traffic Behavior. (1997). 25
- Leclercq, L., Chiabaut, N. & Trinquier, B. (2014). Macroscopic Fundamental Diagrams: A cross-comparison of estimation methods. *Transportation Research Part B: Methodological*, 62, 1–12. Retrieved from <http://dx.doi.org/10.1016/j.trb.2014.01.007> doi: 10.1016/j.trb.2014.01.007 9
- Litescu, S. C., Viswanathan, V., Aydt, H. & Knoll, A. (2019). Information Dynamics in Transportation Systems with Traffic Lights Control. *Procedia - Procedia Computer Science*, 80, 2019–2029. Retrieved from <http://dx.doi.org/10.1016/j.procs.2016.05.522> doi: 10.1016/j.procs.2016.05.522 3
- Liu, F. T., Ting, K. M. & Zhou, Z.-H. (2012). Isolation-Based Anomaly Detection. *ACM Transactions on Knowledge Discovery from Data*, 6(1), 1–39. Retrieved from <http://dl.acm.org/citation.cfm?doid=2133360.2133363> doi: 10.1145/2133360.2133363 19
- Ministerie van Infrastructuur en Milieu. (2014). *Basis voor een Nota Verkeerslichten* (Tech. Rep.). 13, 60
- Mumtaz, K., Studies, M. & Nadu, T. (2010). An Analysis on Density Based Clustering of Multi Dimensional Spatial Data. *Indian Journal of Computer Science and Engineering*, 1(1), 8–12. 16
- Munoz-organero, M., Ruiz-blaquez, R. & Sánchez-fernández, L. (2017). Automatic detection of traffic lights, street crossings and urban roundabouts combining outlier detection and deep learning classification techniques based on GPS traces while driving. *Computers, Environment and Urban Systems*(September), 0–1. Retrieved from <http://dx.doi.org/10.1016/j.compenvurbsys.2017.09.005> doi: 10.1016/j.compenvurbsys.2017.09.005 3
- Novikov, A., Novikov, I., Katunin, A. & Shevtsova, A. (2017). Adaptation Capacity of the Traffic Lights Control System (TSCS) as to Changing Parameters of Traffic Flows Within Intellectual Transport Systems (ITS). *Transportation Research Procedia*, 20(September 2016), 455–462. Retrieved from <http://dx.doi.org/10.1016/j.trpro.2017.01.074> doi: 10.1016/j.trpro.2017.01.074 9
- Pop, M.-D. (2018). Traffic Lights Management Using Optimization Tool. *Procedia - Social and Behavioral Sciences*, 238, 323–330. Retrieved from <https://doi.org/10.1016/j.sbspro.2018.04.008> doi: 10.1016/j.sbspro.2018.04.008 2
- Ranjith, R., Athanesious, J. J. & Vaidehi, V. (2015). Anomaly Detection using DBSCAN Clustering Technique for Traffic Video Surveillance. *Seventh International Conference on Advanced Computing (ICoAC)*, 1–6. doi: 10.1109/ICoAC.2015.7562795 16
- Rijkswaterstaat. (2016). Inventarisatie van roodlichtnegatie bij bruggen door langzaam

verkeer.

1

- Rossi, R., Gastaldi, M., Gecchele, G. & Barbaro, V. (2015). Fuzzy logic-based incident detection system using loop detectors data. *Transportation Research Procedia*, 10(July), 266–275. Retrieved from <http://dx.doi.org/10.1016/j.trpro.2015.09.076> doi: 10.1016/j.trpro.2015.09.076 9
- Rousseeuw, P. J. (1987). Silhouettes : a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20, 53–65. 18, 19
- Scheepjens, R. (2016). *Algorithm Design for Traffic Signal Timings Predictions of Vehicle-Actuated Controlled Intersections using Support Vector Regression* (Unpublished doctoral dissertation). Delft University of Technology. 51, 53, 54
- Seif, G. (2018). *The 5 Clustering Algorithms Data Scientists Need to Know*. Retrieved from <https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68> 15, 16
- Shao, L., Schleicher, T., Behrisch, M., Schreck, T., Sipiran, I. & Keim, D. A. (2016). Guiding the exploration of scatter plot data using motif-based interest measures. *Journal of Visual Language and Computing*, 36, 1–12. Retrieved from <http://dx.doi.org/10.1016/j.jvlc.2016.07.003> doi: 10.1016/j.jvlc.2016.07.003 26
- Stahlmann, R. & Malte, M. (2018). Exploring GLOSA Systems in the Field : Technical Evaluation and Results.
- 3
- Stripling, E., Baesens, B., Chizi, B. & vanden Broucke, S. (2018). Isolation-based conditional anomaly detection on mixed-attribute data to uncover workers' compensation fraud. *Decision Support Systems*, 111, 13–26. Retrieved from <https://doi.org/10.1016/j.dss.2018.04.001> doi: 10.1016/j.dss.2018.04.001 19
- Sun, L., Versteeg, S., Boztas, S. & Rao, A. (2016). Detecting Anomalous User Behavior Using an Extended Isolation Forest Algorithm: An Enterprise Case Study. Retrieved from <http://arxiv.org/abs/1609.06676> 19
- Sunkari, S. (2004). The Benefits of Retiming Traffic Signals. *ITE JOURNAL*(April). v, v, v, 1, 2, 5, 6, 45
- Veenstra, S., Geurs, K., Thomas, T. & van den Hof, R. (2016). *Alle lichten op groen voor fietsmonitoring in Enschede*. v, v, v, 50
- Vialis. (2012). *V-Log protocol en definities*. 53
- Wegenwiki. (2015). *Verkeersregeling*. Retrieved 2019-02-22, from <https://www.wegenwiki.nl/Verkeersregeling> v, v, v, 49
- Yang, K., Yu, R., Wang, X., Quddus, M. & Xue, L. (2018). How to determine an optimal threshold to classify real-time crash-prone traffic conditions ? *Accident Analysis and Prevention*, 117(April), 250–261. Retrieved from <https://doi.org/10.1016/j.aap.2018.04.022> doi: 10.1016/j.aap.2018.04.022 5
- Yousef, K. M. A., Shatnawi, A., Latayfeh, M., Ahmad, K. M., Shatnawi, A. & Latayfeh, M. (2018). information Intelligent Traffic Light Scheduling Technique Using Calendar-Based History Information. *Future Generation Computer Systems*. Retrieved from

- <https://doi.org/10.1016/j.future.2018.08.037> doi: 10.1016/j.future.2018.08.037 2
- Zedgenizov, A. & Burkov, D. (2017). Methods for the Traffic Demand Assessment Based on the Quantitative Characteristics of Urban Areas Functioning. *Transportation Research Procedia*, 20(September 2016), 724–730. Retrieved from <http://dx.doi.org/10.1016/j.trpro.2017.01.117> doi: 10.1016/j.trpro.2017.01.117 25
- Zhang, Z., He, Q., Tong, H., Gou, J. & Li, X. (2016). Spatial-temporal traffic flow pattern identification and anomaly detection with dictionary-based compression theory in a large-scale urban network. *Transportation Research Part C: Emerging Technologies*, 71, 284–302. Retrieved from <http://dx.doi.org/10.1016/j.trc.2016.08.006> doi: 10.1016/j.trc.2016.08.006 9
- Zhong, R. X., Luo, J. C., Cai, H. X., Sumalee, A., Yuan, F. F. & Chow, A. H. (2017). Forecasting journey time distribution with consideration to abnormal traffic conditions. *Transportation Research Part C: Emerging Technologies*, 85, 292–311. Retrieved from <http://dx.doi.org/10.1016/j.trc.2017.08.021> doi: 10.1016/j.trc.2017.08.021 9
- Zong, B., Song, Q., Min, M. R., Cheng, W., Lumezanu, C., Cho, D. & Chen, H. (2018). DEEP AUTOENCODING GAUSSIAN MIXTURE MODEL FOR UNSUPERVISED ANOMALY DETECTION. *ICLR 2018*, 1–19. 15

Appendix A

Savings in user costs due to TCS retiming

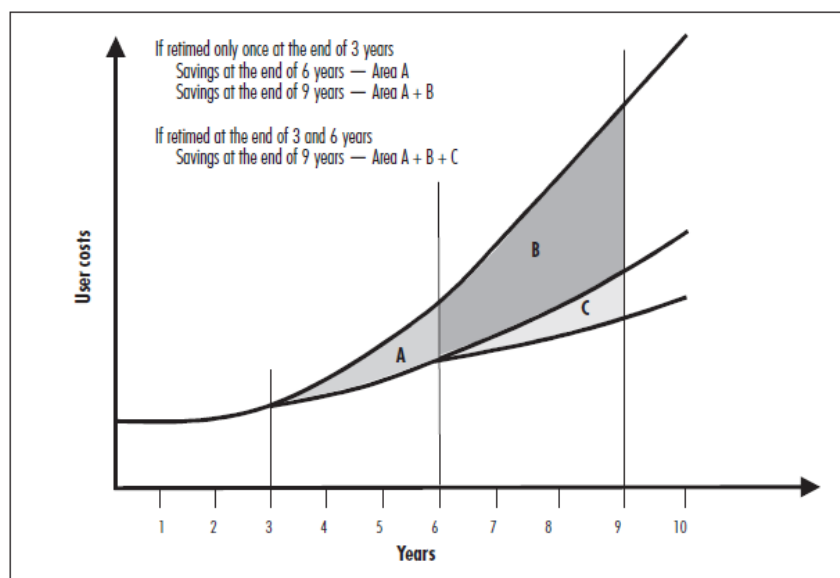


Figure A.1: Savings in user costs due to TCS retiming (Sunkari, 2004).

Appendix B

Research classification and questions

Since the goal of this research is to develop an algorithm/method (i.e. the design of a new 'product') the systems engineering approach is a good guideline to conduct this research process (College, 2001). In the simplified version of the systems engineering approach, the research can be divided into the three phases mentioned in the previous paragraph. Each of the mentioned phases has different and associated research questions and corresponding sub-questions. The main question of this research can be formulated as follows:

Is there one or a combination of multiple policy neutral KPI's that could be used to automatically indicate which TCS show abnormal behavior?

To come up with a solution for this question the following problems are dealt with. The problem and the corresponding approach to this problem is described in table B.1.

Table B.1: Problem description and how this problem is approached

Problem	Approach
Compare TCS with different policies and geographical characteristics.	The use of policy and geographical neutral performance indicators.
The missing quality benchmark for a TCS.	The performance indicators are count values, where a higher number is worse than a lower number.
Which TCS performs the worst/ for which TCS is retiming the most urgent	Detect anomalies in a data set of TCS.

To come up with an answer to the main question several sub-questions are set up per phase. The following questions (sorted by phase) need to be answered:

Phase 1 The requirements: Identify the policy-neutral performance indicators:

46 A data-driven method to automate the detection of traffic control systems that do not perform as intended

- Which KPI's are used in literature to evaluate a TCS?
- Which KPI's do experts use for a rough evaluation of a TCS?
- Which KPI's do experts use for a detailed evaluation of a TCS?
- Which of the policy-neutral KPI is the most important?

Phase 2 The analyse: Determine the characteristics of the KPI's:

- Is a relation/correlation between the different KPI's recognizable?
- Do the KPI's for the different TCS behave the same?
- What is the influence of several time segments/intensities on the KPI's value?
- What is the influence of the region, public transport and cyclist on the value of the performance indicator?
- Are there clusters recognizable and which TCS can be marked as an anomaly?

Phase 3 The performance: Detect anomalies and evaluate the outcome:

- What are the differences between the anomaly detection methods and time segments during the week?
- How can the method be validated?

Appendix C

Theoretical background for the case study

C.1 Intersection's geometry and labeling

C.1.1 Signal group labeling

The traffic control system (TCS) consist of multiple signal groups, every signal group consisting of one or more traffic lights. Each signal group is often labeled with its own number. This unique labeling is convenient while retrieving data off a TCS. Due to this labeling is known to which signal group detector loops belong. Furthermore, the labeling gives information about the green light direction of the signal group. The V-log data is logged according to this label number. The way of labeling the intersections'signal groups in the Netherlands is standardized and shown in figure C.1.

The labels of the signal groups are sorted by the modality, each modality has its own label and corresponding direction and this is given in table C.1.

Table C.1: Modalities and corresponding labels

Modality	Labels
Vehicles	1-12
Bicycles	21-28
Pedestrians	31-38
Public transport	41-51

In addition to the labeling mentioned above, at some intersection multiple lanes going to

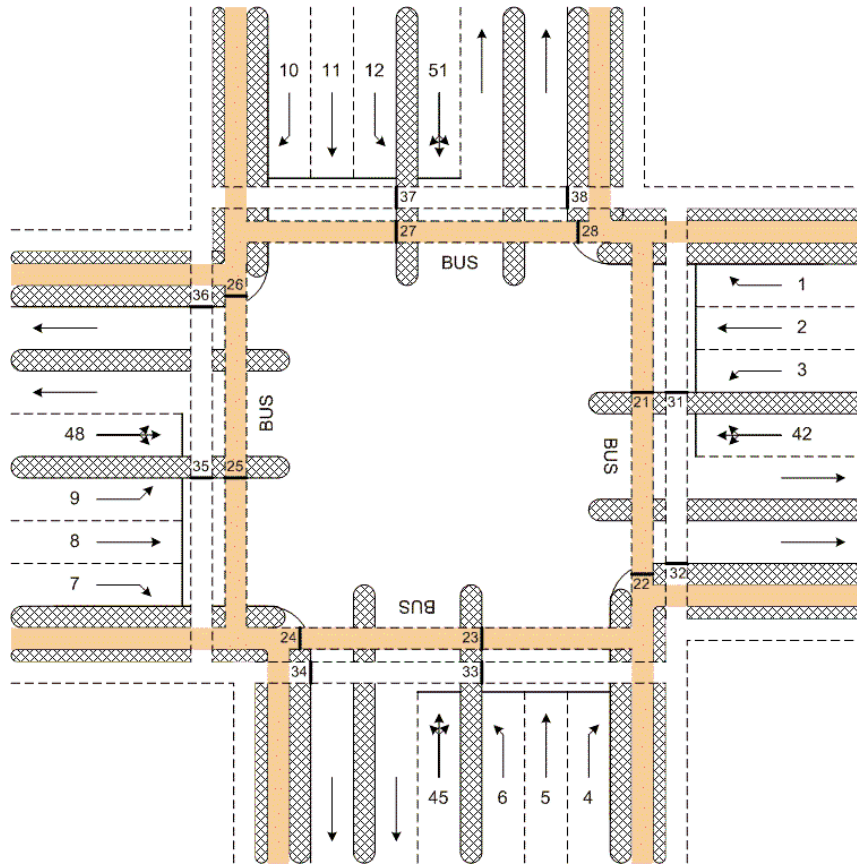


Figure C.1: Signal group labeling (Wegenwiki, 2015)

the same direction. In this situation, a signal group consists of multiple traffic lights and these lights have other labels. For example, if signal group 5 has two lanes, the corresponding traffic lights are labeled as 5.1 and 5.2. A variety in possible geometries of an intersection is possible with this labeling strategy. In this research intersection with multiple geometries will be analyzed, therefore is the labeling crucial for the algorithm in this research. With this information, a universal algorithm can be made which can be used for intersections based on this labeling.

In contrast to the U.S.A., intersections with separated infrastructure for vehicles/public transport and bicycles, as shown in figure 1, are used in many European countries(Katwijk, 2008). Safety is the most important reason for this. In this situation, the weaker road user (pedestrian or cyclist) is separated from the stronger road user (bus or car) in both space and time. In addition, priority can be given due to the separation of the modalities which provide different traffic management policies(Kaparias, Zavitsas & Tomassini, 2010).

C.1.2 Detection loop labeling

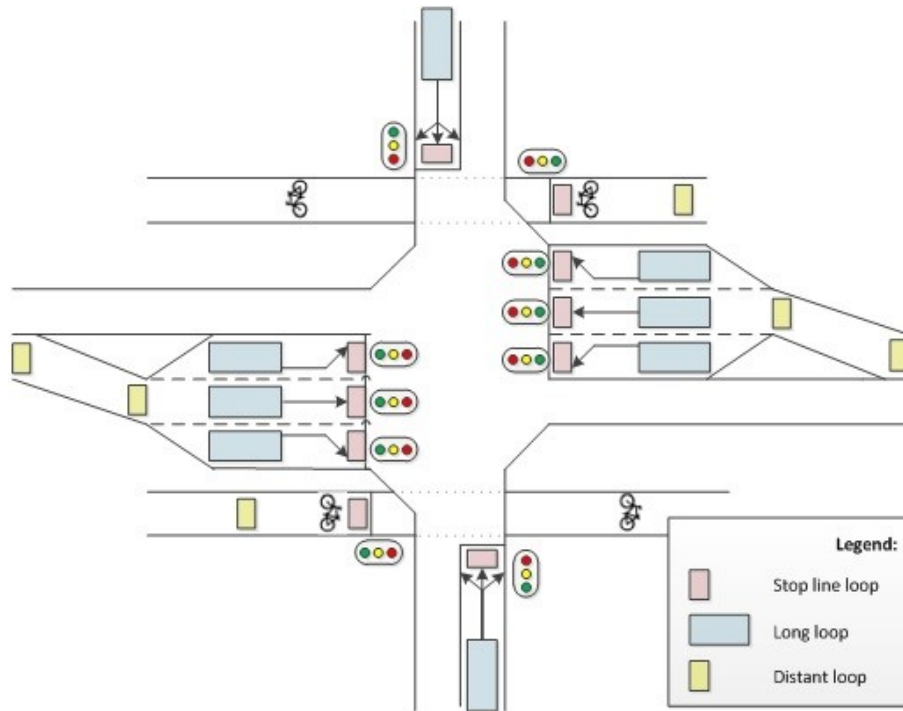


Figure C.2: Position and describing of the detector loops(Veenstra et al., 2016)

The stop line loop detector: These detector loops are placed close to the stop line of a signal group. The length of the stop line detector loop is determined so it can measure a single vehicle. Since this stop line detector loop is located nearby the stop line of the signal group, it is possible to see during which light (red, yellow or green) a vehicle enters the intersection.

Long loop detector: This long loop detector is placed 10-25 meter behind the stop line detector, this can differ for each intersection. The long loop only detects if there are oncoming vehicles, the long loop only detects if the detector is occupied or not and can not count the number of vehicles passed. Due to the length of the long loop detector, no separation between one, two or three vehicles is made.

Distant loop: Depending on the intersection, the distant loop is placed or not. If this distant loop is present, the loop is placed 40 to 60 meter behind the stop line detector loop. In some situation a fourth distant loop is present, this often places by the intersection where higher speed is allowed (Krol, Prinsen & Misdom, 2018).

C.2 The realizations structure of a TCS

The majority (91%) of the controlled intersections in the Netherlands are vehicle-actuated TCS (DTV Consultants B.V. & Willekens, 2016). Many techniques are developed to determine the best structure of vehicle-actuated controller, most of the time the control structure is based on the critical conflict group (Katwijk, 2008). Since the case study will take place in the Netherlands, the Dutch method for controlling the TCS is explained. The Dutch ministry of Transport defined a method where the several signal groups are placed in blocks. This method is called the RWSC-approach, in this method, a block is defined by a collection of signal groups that can be green simultaneously (Katwijk, 2008). The sequence of the blocks is such that the time that is lost when switching between the signal groups of the dominant conflict group is minimal (Katwijk, 2008). For the realization of the green light, three different methods can be distinguished: primary realization, early switch to green realization and alternative green realization.

The primary realization, the standard procedure of the blocks, is as follows: if a block is in the active state, the signal groups of the block can become green. If all the signal groups of a block have become green or are decided to skip a green phase of a signal group, the next block becomes active (Katwijk, 2008). The signal groups of the active block get the permission/ the turn to become green when all other signal groups of the preceding block are red, yellow or prolonging green. From this moment, the signal group procedure takes over the further course of the signal group. As soon as each signal group of the block had the turn to become green, the next block becomes active. Input for this procedure is information from detectors and from the status of the signal groups (Scheepjens, 2016).

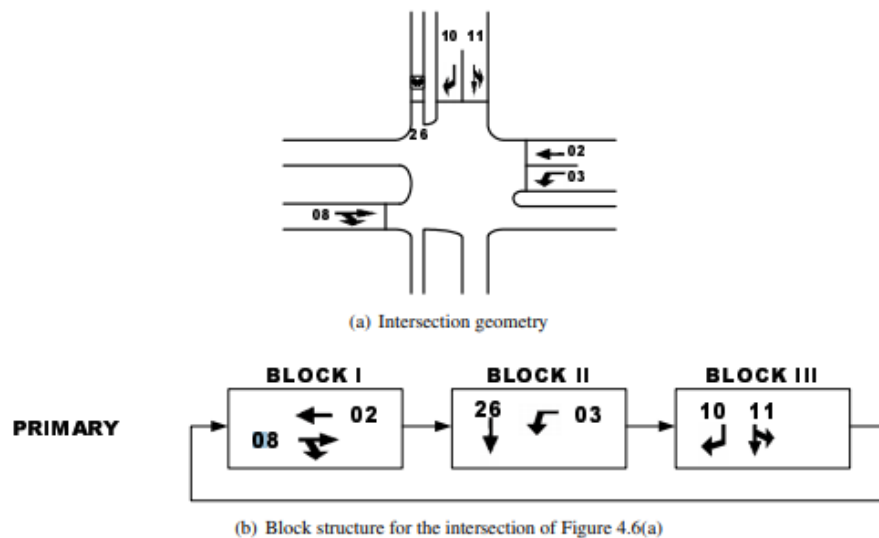


Figure C.3: Primary realization of an intersection(Katwijk, 2008)

In figure C.3 the primary realization is visualized, this is an example of RWSC-approach.

A data-driven method to automate the detection of traffic control systems that do not 51 perform as intended

To understand "the early switch to green" realization the following example is described by van Katwijk:

"During the control process the following situation may occur: suppose that all signal groups of the intersection are red and block I is active. Assume that there is no request for green for signal group 8 and that there is demand for signal groups 2 and 3. Signal group 2 and 8 are both allowed to turn green. As there is no demand for signal group 8, this signal group waives the right to turn green. Then Block II becomes active, and signal group 8 loses its right to become green. Because all conflicting signal groups of signal group 3 are red, this signal group can become green simultaneously with signal group 2 (even though they belong to different blocks). This shows that the block procedure offers the possibility for signal groups of different blocks to become green simultaneously"

This is called the early switch to green realization. In another example given by van Katwijk the alternative green realization is described:

"Suppose that the following situation occurs: Signal group 10 has a request for green, signal group 26 has no request, signal group 2 is red, block II is active and signal group 8 is green. The signal groups 3 and 26 are red; they can become green only after signal group 8 has become red. This gives an unacceptable control situation: signal groups 2 and 26 are red, there is traffic waiting for signal group 10, 8 is green and there should be the possibility to give green to signal group 10 simultaneously"

The visualization of the alternative green is given in figure C.4. The complexity of the

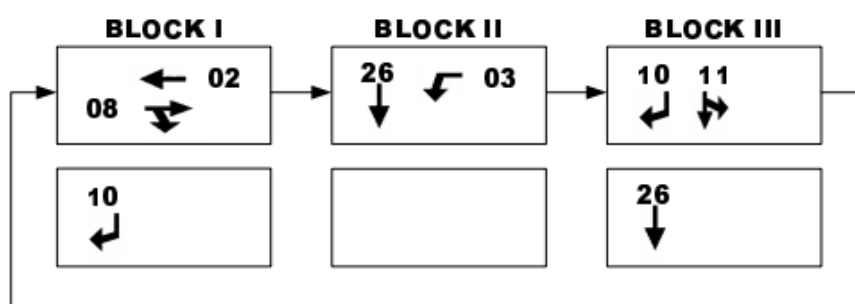


Figure C.4: Alternative realization of an intersection(Katwijk, 2008)

possibilities for alternative green and early switch to green realizations increase for intersections with a more complicated structure. To secure the intersection's safety and minimize the delay, the sequence of the blocks is important(Katwijk, 2008).

C.3 Traffic control system data: v-log data and SPaT

The main data sources for calculating the TCS'performance indicators are considered to be V-log (verkeerskundig log) and SPaT (Signal Phase and Timing data). SPaT is the internationally standardized message format for a TCS. The SPaT message is not standardized, only the data format is standardized. In the Netherlands V-log data is used, V-log data is an unstandardized data format, but is recognized as the most efficient way to log TCS data by the Dutch road authorities(Vialis, 2012). The municipality of Deventer and other municipalities stored the V-log data for every TCS for the past few years. V-Log data is logged in Discrete Event Simulation analogy and compressed to secure efficient transmission without losing performance and complexity of the data(Vialis, 2012). The V-log data is transmitted to a central database at a five-minute interval. Within this five-minute interval, data at a frequency of 1/10 of a second is transmitted. This means that the logging is approaching a real-time state updating. The V-log data contains signals regarding internal signal, detector loops, external signal and speed detection (Vialis, 2012). As a rule of thumb, the SPaT message only contains data of the external signal parameters (green, yellow and red) (Scheepjens, 2016). Internal parameters describe the transitions and the changes in different control schemes within the TCS. The detector loop data is logged binary (On or Off) and the other external/internal parameters are logged with predefined coding. The raw V-log data need to be pre-processed to make it useful for calculating TCS'performance indicators or perform a statistical/scientific analysis.

In figure C.5 a graphical representation of the raw V-log data is given. The data can be interpreted as input and output data of the TCS. The data in figure C.5 is for one signal group, the input data are the blue bars in the rows 081 to 0812. This data is logged binary; 0(no blue bar): indicates no vehicles are detected at that moment and 1 (blue bar) indicates the detection of at least one vehicle. The output data is the TCS'signal group, the legend of the different colors is given in figure C.5.

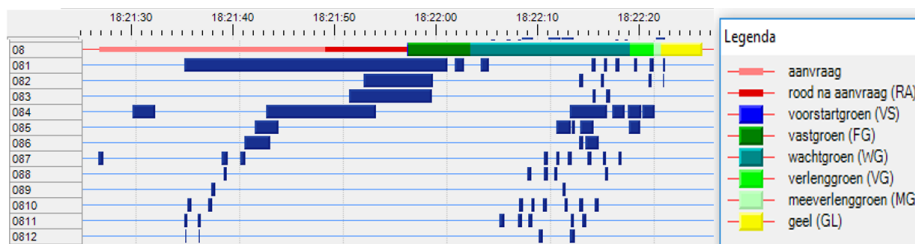


Figure C.5: V-log data visualization

C.4 Internal phase cycle of a signal group

A TCS consist of several signal groups, each signal group is a Traffic light which corresponds with a different direction. Each signal group is operating to a standard phase cycle with the eight phases mentioned in figure C.6. The meaning of these phases is important for the calculation and understanding of the policy-neutral performance indicators. The traffic controller can skip a phase by setting the lead time to zero seconds. In practice, it does not skip the phase, but the lead time is reduced to a not noticeable small interval (Scheepjens, 2016). The "voorstart groen (VS)" is not mentioned in the figure below, but this phase is always before the "vastgroen(FG)" with a duration of 0.1 seconds.

The seven phases explain the internal state changes of each signal group and determine the duration of the green and red phase. The seven phases will be shortly explained below.

RV - red before request (rood voor aanvraag) The controller is switched from yellow phase to red phase. The state of the controller is red before the request, the state will change at the moment there is traffic demand for the particle signal group. During the time there is no traffic demand the signal group's condition is red before the request.

RA - red after request (rood na aanvraag) this phase starts when the red before request end. When the detector loop detects a vehicle and sends a signal to the TCS, the TCS assigns the right to turn green. In the change from RV to RA is a small-time delay because the controller needs to calculate if the signal group indeed has the right to turn green (Scheepjens, 2016). The right to turn green can be overruled by incoming public transport on another signal group. If this group has priority realization, the RA phase will be overruled. The duration of the RA state is depending on the clearance time of the conflicting controllers. The duration of RA is not fixed because a signal group has multiple conflicts.

FG - fixed green (vastgroen) the minimal green- time for a signal group, this is usually 4 seconds. When there is just one car waiting, the driver of the car has enough time to hit the green light.

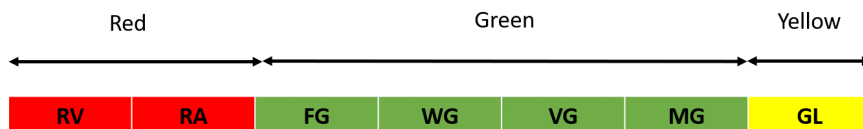


Figure C.6: Phase cycle of a signal group

WG - waiting green (wachtgroen) the waiting green phase starts when the fixed green phase ends, but the waiting green is only used when no conflicting signal groups have traffic demand. Waiting green often occurs after the extension of the green light has expired and no other conflicting signal group has traffic demand. For the TCS it is not necessary to change the green signal since there is no conflicting traffic. During peak moments the waiting green phase rarely occurs since the traffic demand of other signal groups is high.

VG extending green (verlenggroen) VG starts at the same time as the waiting green, the extending green phase causes the most variability in the duration of the total green phase. The VG phase has two functions: 1) the green time is long enough to clear the queue of vehicles waiting for the signal group. 2) the incoming vehicles can hit the green light and enter the intersection.

MG - prolonging green (meeverlenggroen) The prolonging green phase of a signal group is realized when the other signal group of the same block still have the green light and traffic demand. Change the green light to a red signal could be useless since other signal groups still had to wait for the green light of the other signal group of the same block.

GL - Yellow (geel) the duration the yellow phase, this is usually a fixed time, in this study the yellow time varies between 3 and 5 seconds.

Appendix D

Calculation of the KPIs

D.0.1 Red-light running

The first performance indicator that will be discussed now is Red-light running. Red-light runners are vehicles which pass the traffic light during the red signal. For detecting these red-light runners, the stop line loop detector is used (see paragraph 3.2). Figure D.1 shows an example of a vehicle which enters the stop line detector loop when the corresponding signal group is red and leaves the loop when the signal group is still red. Some vehicle enters the stop line loop detector during yellow and leaves during red, both will count as a red-light runner in this research.

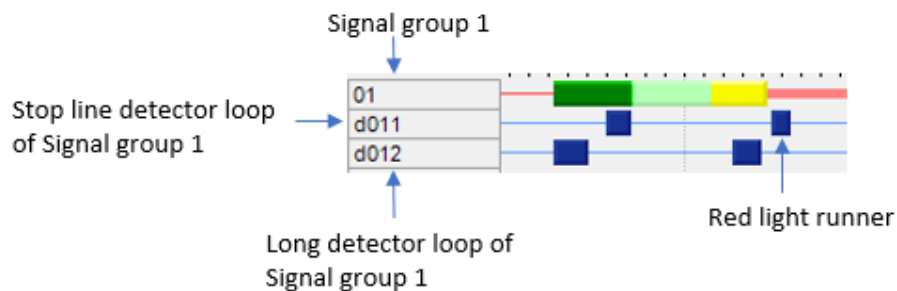


Figure D.1: Entry at red and leave during a red light

The following formula is applied for counting red light runners:

$$red\ light\ runners = \frac{\text{detection of stop line detector turns off during red}}{\text{corrected intensity} \times \text{total number of lanes}} \quad (D.1)$$

The number off red-light runners is divided by the corrected intensity to standardize the value of the performance indicator, without standardization, the values of the different TCS cannot be compared. The corrected intensity is the intensity minus the fluttering. A detection with a duration equal or lower than 0.2 seconds is counted as fluttering.

D.0.2 Unnecessary green time given during the extending green phase

Unnecessary green time can only occur in the extending green phase of a signal group (see paragraph 3.5). The duration of the fixed green time depends on the signal group and intersection, but cannot be interrupted before the end of the fixed duration. During the prolonging green time (MG) it is unnecessary for the signal group to turn red since another signal group of the same block still causes conflicts with the other signal groups, this example is given in figure D.2.

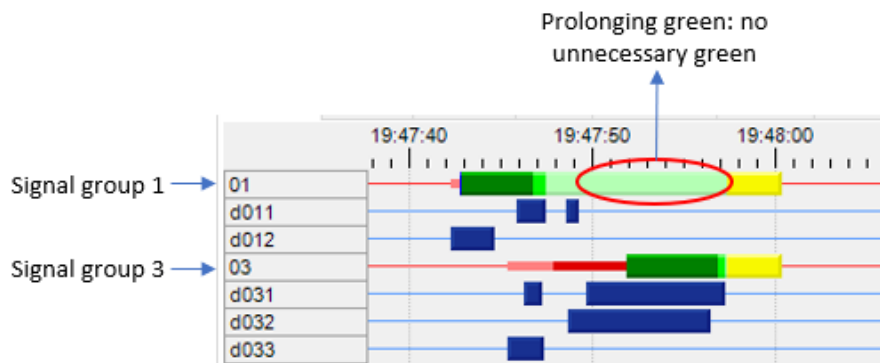


Figure D.2: Visualization why prolonging green is not unnecessary green

During the waiting green time (WG) it is unnecessary for the signal group to turn red since there is no conflicting signal group with green light demand. The situation for the MG of a TCS in figure D.2, . The constraints for unnecessary green are as follows:

- The end of the stop line loop detection is during the extending green phase.
- The next detection of the stop line detector is during the following red.

With these constraints, two possible situations for unnecessary green can occur. The first situation is when a vehicle passes the stop line loop and the TCS is still green for a few seconds before it becomes yellow. The second situation is when a vehicle leaves the stop line loop detector during the extending green phase. After that, the TCS goes to the WG or MG phase and no vehicles are detected during these or the yellow phase. Both situations are visualized in figure 3.3 and figure 3.4 respectively.

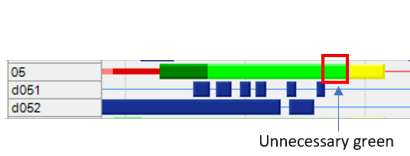


Figure D.3: Situation 1

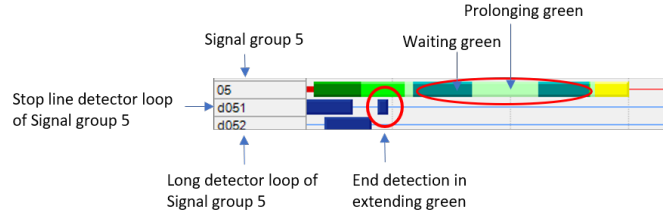


Figure D.4: Situation 2

$$Unnecessary\ green = \frac{n\ during\ prolonging\ green,\ n+1\ during\ red}{number\ of\ green\ times\ given \times total\ number\ of\ lanes} \quad (D.2)$$

The n is the number of the detection of the stop line detector loop corresponding by the signal group.

D.0.3 No use of the fixed green phase

This performance indicator counts the number of unused fixed green time. In this situation, unused means there is no detection during this phase. This performance indicator detects TCS where green is given when no vehicle is waiting for this signal group. For this performance indicator is only one constraint applied:

- The stop line detector was never”on” during the fixed green phase of the TCS.

$$No\ use\ fixed\ green = \frac{n=prolonging\ green,\ n+1= red\ or\ next\ fixed\ green}{number\ of\ green\ times\ given} \quad (D.3)$$

D.0.4 Double stop or too early stop of the green signal.

For this performance indicator, the yellow phase is important. An indicator for a double stop is when the long loop detector is occupied and at the same time, the yellow phase starts. In this situation, the long loop detector is occupied so this detector sent a green light demand to the TCS. Despite sending the green demand, the signal group turns to yellow. According to the quick data scan, this often creates a false count of a double stop or no count. The false count is caused by fluttering and the no count if there is just one car left. To avoid these false/no counts the double stop is based on early detection of stop line detector after or during yellow in this research. The following constraints are used to count double stops:

- Detection of the stop line detector during yellow or within 2 seconds of the end of the red signal.
- The duration of the detection is at least 10 seconds (to avoid counting red-light runners)

The visualization of a double stop is showed in figure D.5

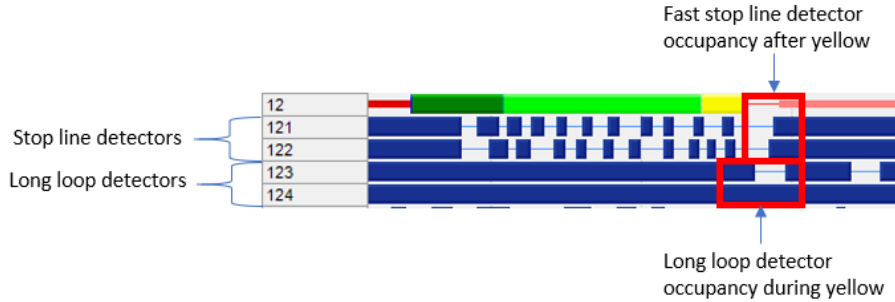


Figure D.5: Example of a double stop

The formula to calculate the standardized number of double stops:

$$Unnecessary\ green = \frac{\text{detection during yellow or within 2 seconds}}{\text{number of green times given} \times \text{total lanes}} \quad (D.4)$$

D.0.5 The use of prolonging green

In paragraph 3.3 the theory of the TCS' functioning was presented, herein the block structure and the corresponding characteristics of prolonging green is explained. The occurrence of prolonging green depends on the used block scheme of the TCS. The primary block realization of a TCS is fundamental for the performance of a TCS (Katwijk, 2008). The goal of the primary realization' design is to process all the traffic demand as good as possible (good is the way the traffic manager wants it to be processed). The alternative realization is an addition to the primary realization to improve the TCS' functioning. Prolonging green is typical for the alternative realization, so the duration of prolonging green could be an indicator of wrong/outdated design of the primary realization (block scheme) according to the expert mentioned in the introduction of this chapter.

In this research, the mean prolonging green for a signal group is calculated and taken into account as a performance indicator. For this reason is the total duration of prolonging green time divided by the number of signal groups. In addition, the number of vehicles that passed the intersection during prolonging green is calculated. This gives a better indication of the additional value of the prolonging green phase. This can be written as the following formula:

$$Use\ prolonging\ green = \frac{\text{Total given prolonging green time}}{\text{signal groups} \times \text{vehicles pass during prolonging green}} \quad (D.5)$$

D.0.6 Fluttering

Flutter behavior of detector loops is a very short false detection which can cause an unnecessary green-light request. In this research flutter behavior for both the stop line detector and the long loop detector will be extracted from the data. Detection with a duration between 0 and 0.2 seconds will be marked as a flutter behavior. Sometimes, the stop line detector cannot detect light vehicles well or has a false detection due to a passing tram or bus nearby. The intensity of TCS is measured by summing up all the detections of the stop line detectors, this value is corrected with the fluttering value.

D.0.7 Waiting time over 90 seconds

In the Netherlands, the maximum length for the cycle time is 90 seconds (Ministerie van Infrastructuur en Milieu, 2014). This performance indicator checks how often the TCS does not perform within this threshold. This performance indicator does not measure the cycle length of a TCS, but the waiting time of a vehicle before getting green. The cycle length is duration until all signal groups get green light, if the waiting time for one signal group is higher than 90 seconds, the cycle length is also higher than 90 seconds. This

performance indicator is calculated and standardized with the following formula:

$$90 \text{ seconds waiting} = \frac{\text{detection of stop line detector longer than 90 seconds}}{\text{number of green times given} \times \text{total lanes}} \quad (\text{D.6})$$

D.0.8 Early starters

Early starters are vehicles which pass the green light very early. With this are not meant the fast cars and/or people with fast reaction time, but people who start to accelerate during the red light. The normal reaction time of people is 0.2 seconds, but in traffic situations, people have to analyse the presence of other traffic at the intersection beside the change in signal of the Signal group. For this reason, an anticipation time of 1 second is normally used in traffic situations¹. In addition, there is also a certain time needed to accelerate and leave the stop line detector. For these reasons, all vehicles which pass the stop line detector within 1.5 seconds after turning green are marked as too early starters

$$\text{Early starters} = \frac{\text{vehicle pass fixed green within 1.5 seconds}}{\text{number of green times given} \times \text{total number of lanes}} \quad (\text{D.7})$$

¹<https://inask.nl/index.php/10-Onderwerpen/Beweging/40-Reactietijd>

Appendix E

KPI values during the day

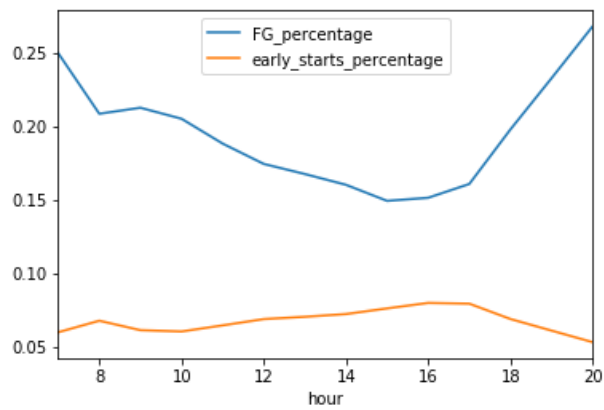


Figure E.1: mean value of a KPI during the day.

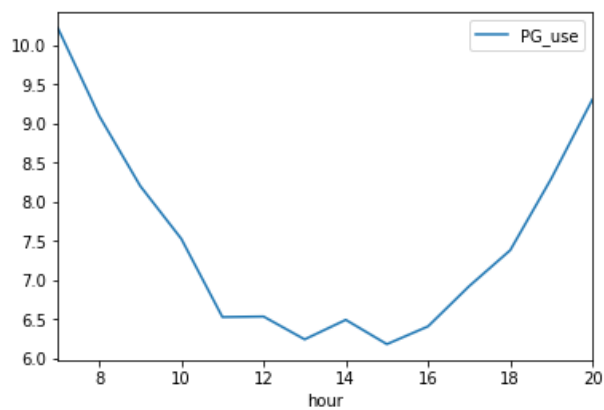


Figure E.2: mean value of a KPI during the day.

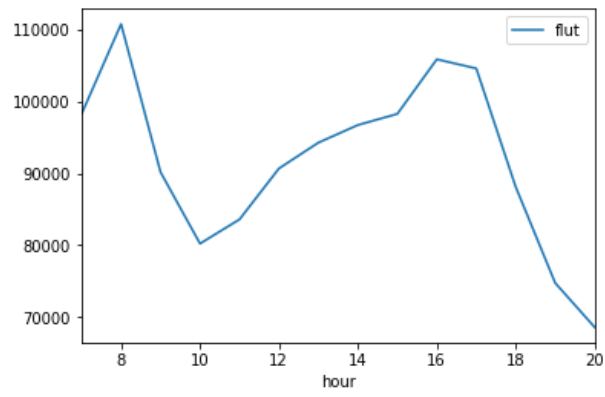


Figure E.3: mean value of a KPI during the day.