SETA

Deliverable 4.2

[Initial Development of Demand and Supply Predictors]

Grant Agreement number: 688082

Project acronym: SETA

Project title: An open, sustainable, ubiquitous data and service ecosystem for efficient, effective, safe, resilient mobility in metropolitan areas

Funding Scheme: H2020-ICT-2015

Authors

Panchamy Krishnakumari	P.K.Krishnakumari@tudelft.nl
Ding Luo	D.Luo@tudelft.nl
Tamara Djukic	tamara.djukic@aimsun.com
Oded Cats	O.Cats@tudelft.nl
Hans van Lint	J.W.C.vanLint@tudelft.nl

Internal Reviewer

Borja Alonso Oreña	borja.alonso@unican.es

State: Final

Distribution: Public

Deliverable History

Date	Author	Changes
06-02-2017	Oded Cats, TUD	Draft table of contents
12-03-2017	Hans van Lint, TUD	Update ToC
31-03-2017	Tamara Djukic, TSS	Draft sections 2, 3.2, 3.3
31-03-2017	Panchamy Krishnakumari and Ding Luo, TUD	Draft sections 3.1, 4
11-04-2017	Tamara Djukic, TSS	Update sections 2, 3.2, 3.3
11-04-2017	Panchamy Krishnakumari and Ding Luo, TUD	Update sections 3.1, 4
13-04-2017	Oded Cats, TUD	Writing summary, sections 1 and 5, editorial

Contents

Summary

This document provides an overview of a series of methods for predicting supply and demand of transport systems. These methods allow generating a wide range of predictions: local and network-wide, traffic flows and travel demand, for private vehicular traffic and public transport systems. All of the methods were developed with special consideration to their robustness and scalability including devising techniques for dimensionality reductions in light of SETA project objectives.

Following up on the extensive state-of-the-art review provided in D4.1, model-based and data-driven techniques for short-term predictions are developed based on the most promising avenues that have been identified. The data-driven and simulation-based technologies to predict the traffic state and demand in the network, require different level of transport network representation. This document details how graph and network model for the various levels of simulation models are constructed as well as a workflow to build a network model. The simulation software Aimsun is used in this project.

Both network-wide and local short-term vehicle traffic predictions are developed. The network-wide prediction uses clustering based strategies to reduce the high-dimensional nature of the network. The method is demonstrated for one of the SETA test sites (i.e. Santander, Spain). For local predictions a rule-based strategy is used which is robust and easily scalable in terms of network size and modelling complexity. In addition, a sensitivity analysis based approach is also developed to perform OD prediction along with simulation models for vehicle traffic demand predictions. In the public transport domain, methods tackling the high-dimensionality problem inherent to urban public transport systems, where passenger demand at many public transport stops needs to be forecasted simultaneously, are developed.

The methods reported in this deliverable constitute an advancement of the state-of-the-art while being devised to serve the specific objectives of the SETA project with the prospective case studies taken into consideration. The proposed vehicle traffic and demand predictions methods are applied to the three test sites as part of an on-going work. Results from these applications to all three sites and an evaluation of their performance based on a performance assessment, validation study and sensitivity analysis will be provided in the subsequent deliverable.

Glossary of Terms

DTA	*Dynamic Traffic Assignment*
GML	*Geography Markup Language*
API	*Application Programming Interface*
OSM	*OpenStreetMap*
SPSA	*Simultaneous Perturbation Stochastic Approximation*
SA	*Sensitivity Analysis*
FAST	*Fourier Amplitude Sensitivity Test*
AVL	*Automatic Vehicle Location*
APC	*Automatic Passenger Count*
AFC	*Automatic Fare Collection*
GTFS	*Generic Transit Feed Specification*
PCA	*Principal Component Analysis*
OD	*Origin-Destination*
ITS	*Intelligent Transportation Systems*
SGD	*Stochastic Gradient Descent*
GoF	*Goodness of Fit*
MAPE	*Mean Absolute Percentage Error*
RMSE	*Root Mean Squared Error*

1. Introduction

1.1. Background and Purpose

Transport operations and dynamics are inherently uncertain. Transport infrastructure and system managers rely on anticipated future system states in making traffic management and control decisions. Moreover, system users – car drivers and public transport passengers – rely on information provision in making travel decisions such as departure time and route choices. It is thus essential to generate accurate, reliable and robust short-term predictions concerning transport supply and demand. In D4.1 “Exploring prediction perspectives” the state-of-the-art of short-term prediction techniques was reviewed and the most promising avenues for developing predictions in the context of the SETA project were identified.

This document reports the “Initial development of demand and supply predictors”. The methods documented in this deliverable were developed in the context of task 4.2 in the SETA project as part of phase 1 which corresponds to the first half of the project. These methodologies relate to predicting supply quantities (speeds, travel times) and demand quantities (origin-destination (OD) flows, isolated local flows). The aim of this report as a report documenting the development of demonstrators is to provide a technical account explaining the methods with some preliminary results where ever needed or possible. Each section starts with an overview that provides a general introduction to the topic followed by a more technical description of the prediction and modelling techniques. We do not describe extensive verification or validation of the methods against data, as this will be the subject of D4.3.

1.2. Report outline

The body of this report consists of three chapters:

Chapter 2 describes how each of the SETA case study networks – Santander, Turin and Birmingham – is defined in terms of its geographical demarcation and the respective data that has been collected to construct a network model. The latter consists of a graph representation embedded with traffic and transit information as well as demand data.

Chapter 3 presents methodological advancements in predicting vehicular traffic. Novel methods for performing network-wide traffic predictions, local traffic predictions and demand predictions are detailed. All of the methods were developed with special consideration to their robustness and scalability. The methodology for generating network-wide predictions is demonstrated with the Santander network and data.

Chapter 4 turns the focus to public transport predictions. Methods for dimensionality reductions based on spatial clustering and principal component analysis have been adopted to enable the estimation and prediction of public transport flows.

Finally, Chapter 5 describes the plan for implementing the methods detailed in this deliverable to SETA case studies and discusses the outlook of traffic and demand predictions in the context of the project.

2. Network Representation and Simulation Models

2.1. Background

Due to the well-known complexity of transportation systems in our cities, together with their fundamental role in terms of environment, quality of life and economic growth, research in analysis and prediction of traffic phenomena is gaining a growing importance. This has been even more notable with the recent sensing and data processing innovations of varying nature (e.g. telecom, smart cards), globally referred to as ‘‘big data’’. We do have more data, more computing power and higher recognition of the importance of understanding traffic in our cities.

However, the problem is still very complex as it quickly reaches high dimensionality with large networks, multiple measurements, multiple data sources, several traffic control systems, and high and heterogeneous demand patterns. An approach to deal with this complexity is by using traffic simulation models. Traffic simulation models represent the mathematical modelling of transportation system through application of computer software to better support planning, design and management and control of transportation systems. In WP4, simulation of traffic plays an important role, because it can study models too complicated for analytical or numerical treatment, can be used for experimental studies, can study detailed relations that might be lost in analytical or numerical treatment and can produce attractive visual demonstrations of present and future scenarios. Within SETA, there are use cases in three different but complementary metropolitan areas in Europe, all of which have extensive and intense mobility and transport issues: Birmingham, UK; Turin, Italy; Santander, Spain. These use cases provide different social and technical challenges, as well as different mobility data sources and therefore require the representation of the transport system in simulation models as well as validation in very different situations.

The goal of this section is to give the reader an insight of the different types of data required for building a graph and network model for the various levels of simulation models as well as a workflow to build a network model. Transport geography graph involves developing abstract representation of transport networks that consists of sections and nodes and their attributes, and represents the subtask within a building process of the network model. Within WP4, we use simulation software Aimsun (TSS-Transport Simulation Systems 2015) to build the graph and network model for each use case within SETA. TSS develops Aimsun, the leading traffic modelling software environment, that stands out for the exceptionally high speed of its simulations and for integrating travel demand modelling, hybrid microscopic-mesoscopic traffic simulation and dynamic traffic assignment – all within a single software application. Aimsun also enables efficient import and export of the network graphs and models that can be used within other SETA’s work packages to ensure integrated data exchange. Note that WP4 develops data-driven and simulation-based technologies to predict the traffic state and demand in the network, and these technologies require different level of the transport network representation in their models. Therefore, in WP4 we have adopted and developed two compatible, but different at the level of detail, network representations:

- **Network graph** – corresponds to abstract representation of transport networks that requires low-level of data: location and shape of sections, nodes and turns, their parameters such as maximum speed, number of lanes, and capacity. This network representation is usually required by macroscopic simulation models and data-driven technologies.
- **Network model** – corresponds to network representation used by mesoscopic or microscopic simulation-based models, and can be seen as an extension of the

objects and attributes to represent transport network and individual vehicle behaviour. This network representation requires more detail data including, traffic control plans, pedestrian crossings, signalized nodes, intersection control type.

Once the network representation is built in Aimsun, the essential challenge in building the network model becomes the calibration of all the supply and demand parameters in order to reflect the real traffic phenomena. Different calibration requirements are expected for dynamic traffic assignment models (DTA) and for microscopic traffic simulation. For example, DTA models usually utilise mesoscopic demand and supply simulator components, that employ a mix of microscopic and macroscopic models to capture the decision of the travellers and the movement of vehicles throughout the network. They consider the (often thousands or tens of thousands) OD flows in the network as inputs that need to be calibrated. Similarly, in the supply side, segment output capacities are among the parameters that need to be calibrated, and these are easily in the order of thousands. Microscopic traffic simulator models also require OD flows as inputs, but on the supply side they require a much smaller number of parameters to be calibrated (used in the individual models, such as car-following, merging, lane-changing). For more detailed review of traffic simulation models we refer to Deliverable 4.1, Section 5.

2.1.1. Graph and network model development and utilization process

Figure 1 illustrates the general process that have been followed for the development and utilization of graph and network models for SETA in simulation software Aimsun. Typical graph and network model development and data utilization steps include:

1. **Identification of use case scope** – Identification of the use case's purpose, spatial extent, appropriate model, and level of expertise.

2. **Selection of modelling approach and simulation model** – Identification of the modelling approach and type of simulation (microscopic/ mesoscopic/ macroscopic/ hybrid) to be used.

3. **Data collection and preparation** – Collection of data required for the development of the graph and network model. This step includes collecting data from traffic monitoring systems, conducting field data collection, reviewing base maps, retrieving information from data warehouses, or requesting data from specific agencies. It also includes checking data validity, processing and reducing data to extract specific information, and formatting data for their use in data-driven and simulation-based models.

4. **Base network model development** – Creation and coding of sections, nodes and turns representing the road network geometry, definition of the geometric characteristics of each section, node and turn, insertion of traffic control elements and public transport, specification of travel demand matrices, and setting of simulation parameters.

5. **Error checking** – Checks for coding errors that can affect the execution of data-driven and simulation-based models. Refinement of the geometry to fit technologies requirements and error-checking is an important modelling step as coding errors and geometry shape carried through calibration or delivered to SETA project partners can significantly affect results. This is an iterative process with step 4, where parameters modelling network geometry, traffic demand, traffic control devices and driver behaviour are reviewed to ensure they provide valid and logical values.

6. **Network model calibration** – Adjustment of network and simulation model parameters to reproduce traveller behaviour and traffic performance. This involves the establishment of calibration targets, selection of appropriate calibration parameters to reproduce observed roadway capacities and route choice patterns, and calibration of model parameters so that its performance matches data from field observations.

Transport network graph is typically output after steps 4 (base network model development) and 5 (error checking), while the full network model requires further calibration and validation developments. Once the network graph and model development is completed, network models are delivered to the use cases leaders in WP1 for approval, before it could be used to evaluate WP4's technologies or shared with other SETA work packages. In many cases during this development process, the approval process was done iteratively with the network graph development and network model calibration.

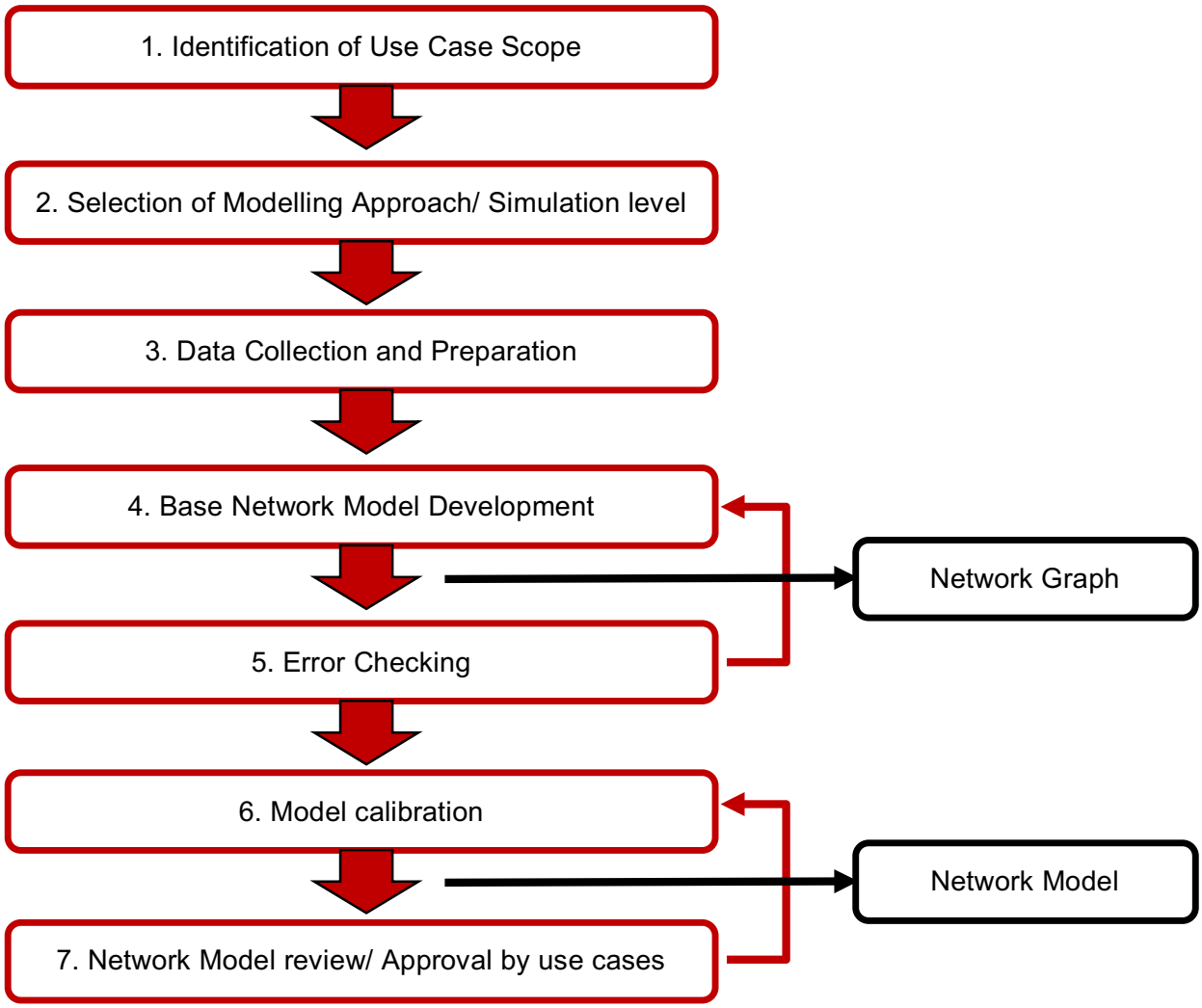

Figure 1. Workflow process for the network model development in SETA

2.1.2. Data requirements to build the graph and network model

Building a graph and network model for application in microscopic simulation models typically requires more data than other types of modelling approaches, such as macroscopic simulation models. For example, microscopic models typically require the most data due to their need to model individual vehicle behaviour in details. Mesoscopic models may require slightly fewer data depending on the simplifications made in their driver behaviour models.

Macroscopic models typically require the least amount of data, as traffic behaviour is usually only characterized by flow rates, average observed speeds, and observed link densities.

The required data to build the graph and network model for each use case in SETA, can typically be grouped into the following categories:

- **Network geometry** – Data describing various geometrical aspects of the use case area, such as the location of intersections, road widths and shapes, slopes, the number of lanes on each section, equipment used for monitoring traffic performance, etc.
- **Demand** – Data, expressed in number of trips, capturing distribution of trips between the various origin and destination nodes. Rules followed by travellers to select a path within a network may also be included in this category.
- **Traffic control** – Data characterizing the operation of traffic signals and ramp meters, priority schemes for transit vehicles at signalized intersections, etc.
- **Transit operation** – Data characterizing the operation of public transport, such as transit routes, vehicle composition, stop locations, service schedule.
- **Network traffic state and performance –** Data characterizing how traffic behaves along roadway elements, such as volumes, speeds, travel times, location of bottlenecks, etc. Data should be collected for all critical time periods being studied, e.g. AM peak, Midday peak, PM peak, event-based.

Table 1 lists data required for the development of graph and network model in SETA for technologies developments and evaluations.

Table 1. Overview of data required for building use case's network models in SETA

Data Category	Data Sub-Category	Data items
Network geometry	Road geometry elements	• **Road/section shape, length, curvature and slope** • **Road category** • **Number of lanes** • Purpose of lane (general traffic, HOV vehicles, managed lane, etc.) • **Allowed turnings directions at the node** • Lane utilization: turnings from lane to lane (through lane, left-turn lane, etc.) • Pedestrian crossings • Placement of traffic signs along roadway links • **Node/intersection layout**
	Basic Functional parameters	• **Section maximum speed** • **Section Capacity** • Section user defined costs • Turn maximum speed
	Traffic Monitoring	• **Location and type of traffic sensors**
Traffic control	Intersection control	• Type of intersection control (stop sign, yield sign, traffic signals) • Type of traffic signal control (fixed time, actuated, traffic responsive) • Signal timing plan (start time, cycle length, yellow, phases, green) • Arterial signal coordination plan (offset relative to other control plans) • Data interchange interface for actuated and adaptive control plans
	Ramp metering	• Type of ramp meter • Metering plan • Location of traffic sensors

Demand	Vehicle fleet characteristics	• Vehicle mix • Truck percentages and/or volumes • Vehicle occupancy
	Traffic zones	• Zone boundaries • Centroids and connectors
	Travel patterns	• OD flow matrices • Network entry flows, if OD matrices are not used • Mode shares (*only if for models including transit or non-vehicle modes*)
	Freeway traffic patterns	• Freeway mainline counts • Freeway ramp volumes
	Arterial traffic patterns	• Link counts along major arterial segments • Intersection turning counts
Transit operations	Public transport data	• Transit routes (ideally GPS based, GTFS file) • Stop locations • PT Service schedules and headways (including stop-time mean and deviation) • Fleet size and composition • Signal priority scheme
Network performance	Traffic state and behaviour	• Volume, speed and occupancy data from mainline loop detector stations, on-ramps, off-ramps, tube counts • Travel times along major arterial segments
	Bottlenecks	• Time bottleneck stations • Location and extent • Cause of bottleneck

Two major factors often drive data requirements: developing an accurate graph representation of the existing transport network elements and ensuring that simulated and/or predicted flows replicate observed behaviour. The modelling of network geometry in the graph form is can be seen as a relatively straightforward process since this process generally focuses on the fixed and well defined elements, that can be imported from Open Street Maps (OSM) and other GIS-based files, or from the existing network models available in traffic simulation software. Data items in bold format presented in Table 1 represent the minimum information required to build abstract transport network representation as a graph. The remaining data listed in Table 1 are used to ensure that simulated and/or predicted flows replicate observed behaviour in the network. In the reminder of this section, we present for each use case in SETA, which data from this list have been collected in Phase 1 of SETA and how we use this data to build the network graph and network model. Furthermore, using the Aimsun's embedded GIS Exporter, the network graph representation for each use case has been exported in GML (Geography Markup Language) format and shared with SETA partners to evaluate their technologies.

2.1.3. Traffic data governance tool

While the modelling of network geometry is relatively straightforward, the calibration and sensitivity analysis of traffic flows and driver behaviour usually imposes more complex data collection needs as traffic flows continuously fluctuate and driver behaviour is affected by various environmental factors. In many cases, observations from a single day and/or single data source may not be sufficient to adequately calibrate and characterize traffic flows and typical driver behaviour.

However, application of multiple data sources leads to mixed ownership of traffic data detection infrastructures and intermediary data brokers are usual sources of entropy in the traffic data management lifecycle, where it is common for the traffic modelling teams to lack knowledge about the pre-processing undergone by the traffic data, or about its reliability and consistency.

This situation makes it necessary to build a traffic data governance tool that enables consistent, efficient and traceable calibration and validation of traffic simulation models. For this task, in WP4 we have developed a new tool, data governance tool. This tool serves the role of a data hub that allows for data sources to dump data and data processing elements to consume data, as depicted in Figure 2. In this way, the different data generation and processing steps are accounted for and kept by the data governance tool.

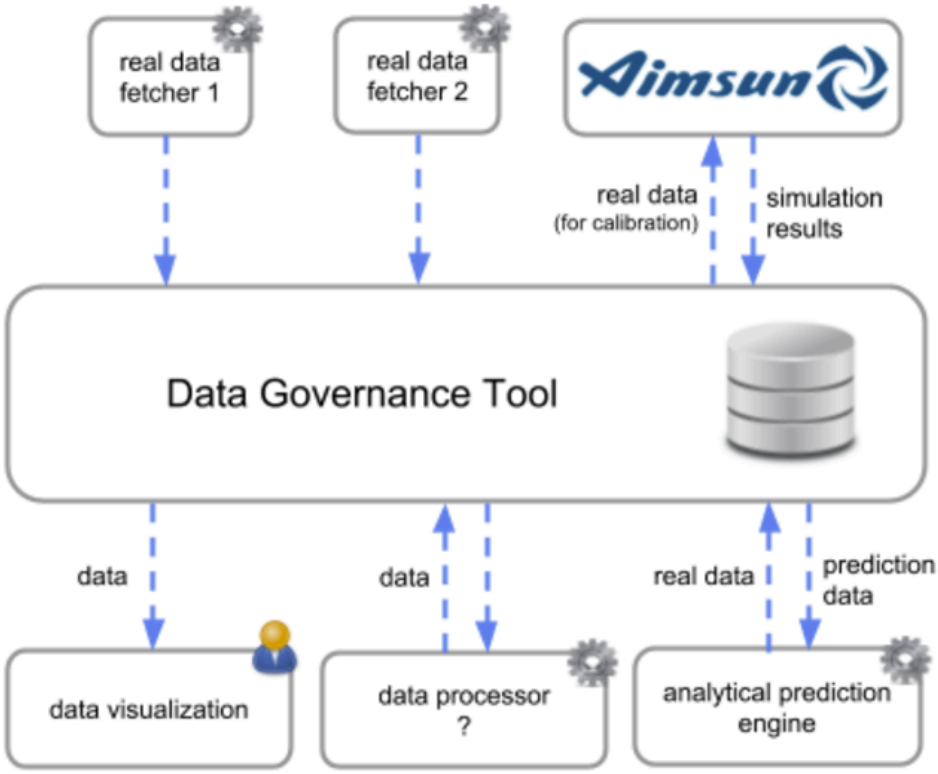

Figure 2. Data governance tool flowchart

The Data Governance Tool comprises the following components:

1. **Data ingestion interface:**

- Ingestion of traffic data coming from different sources, supporting different representation formats (csv, binary), units (international system of units, imperial units), etc.
- Support for heterogeneous traffic data sources, from classical induction loop detectors to floating car data, all exploitable through the same interface.
- Storage and management of such data, using centralised processing and storage engines that effectively enable data governance by means of a canonical version of the traffic data that avoids uncontrolled diverging copies.

2. **Data access interface**

- Data querying capabilities, offering an API (Application Programming Interface) that enables other traffic simulation models and algorithms to consume traffic data stored in the system.

- Integration with major data processing programming languages (R, Python, C++) through programmatic APIs.
- Integration with external data consumers and producers via RESTful web service (with json message payload) APIs.

3. **Data processing and analysis interface**

- Data processing pipeline, including aggregation, filtering and imputation. With full configurability and with full observability of the effects of the different processing stages, allowing the user to know the specific contribution of each stage to the final result, and the ability to perform reconfiguration of the processing stages to adapt to unforeseen situations (e.g. force allowance of traffic detector data that diverges substantially from previous measurements due to un-notified change in the number of lanes covered by the detector).
- Data consistency and completeness analysis, enabling the user to identify anomalous situations (e.g. general malfunction of all traffic detectors controlled by certain traffic authority due to a connectivity problem).
- Traffic pattern extraction, by means of machine learning techniques (e.g. clustering, linear regression) that provide the user with *insights* about traffic behaviour.

In the phase 1 of the SETA project, the data interface jointly with the data querying interface and its key features is prototyped, and their usage for model calibration and validation is exercised. These interfaces play a vital role in ensuring consistent and efficient data pre-processing for their application in calibration and validation of traffic simulation models as well as a data feed for various models developed in WP4. A visualization tool is also prototyped during phase 1, aiming at easing the navigation and exploration of data. Implementation of various traffic pattern extraction algorithms in data gathering tool will support not only calibration and validation process, but will establish a base for models evaluation as part of the Deliverable 4.3.

2.2. Santander network

2.2.1. Definition of the network model scope for Santander use case

The city of Santander is the capital of the Cantabria region in the north of Spain and has a population of 172,000 inhabitants. The urban region of Santander city has been selected for use case, as depicted in the Figure 3. The characteristics of the transport network representation that covers the urban region of Santander city are:

- Size (km): 11,00 x 6,00
- Network type: Urban
- Number of Centroids: 117 (13689 OD pairs)
- Number of Detectors: 230
- Number of sections: 4106
- Number of nodes: 1454
- Type of signal controllers: Fix
- Simulation time: AM peak (08:00-10:00h) and midday peak (13:00-15:00h)

This area was chosen based on its availability in the Aimsun simulation software, its complexity and data availability, and need to address fairly high congestion levels and main mobility critical points inside the city. By modelling this area, we can leverage prior work, support utilization of the model in other projects for Santander city and ensure its compatibility with regional travel demand model for mobility planning needs. The end result is a dynamic, lane-based network model with individual vehicle generation and an extensive

toolkit for representing traffic management operations, from local area to complete urban level, that can ensure complete consistency with Santander use cases and mobility goals.

Figure 3. Scope definition of the network model in Santander

2.2.2. Data collection and application to build the network model

Table 1 complies the data most commonly used to develop network models in simulation software, such as Aimsun. The provided and collected data for Santander city in Phase 1 are generally consistent with the requirements outlined in Table 1. This consistency was expected since data and network model have been provided by University of Cantabria that actively participated in the macroscopic model development of Santander city in Aimsun.

Major sources of data available in Santander for the network model development at mesoscopic level include the following:

- Loop detector data – Traffic observations are available for 230 individual traffic loop detectors located along major streets across Santander. For each detector on the network, traffic counts and speeds are provided with 1 minute aggregation level for an entire 24-hour period over 2016.
- Aerial photographs – Aerial photographs and photo logs were used to collect information about roadway geometry and, in some cases, to help identify bottleneck locations and the extent of queuing. Depending on the area, these analyses have relied on photographs that were already available or photographs from online sources such as Google Maps.
- Regional travel demand models – Regional travel demand model was used to obtain information about traffic flow patterns between defined origin and destination zones. OD demand matrix produced by travel demand models were used as seed matrices in Aimsun. These matrices were then subsequently manipulated to produce flows

better matching observed vehicle counts and to develop a series of 15-min OD matrices capturing the observed changes in traffic patterns across a morning and midday peak travel period.

- Signal timing plans – Signal timing plans for signalized intersections were obtained in the excel format from the agency responsible for the operation of the traffic control devices in Santander. Additional features in Aimsun have been developed to allow automatic import of the control plans in Aimsun. Manual checks had to be performed to ensure there is a perfect matching between Aimsun and provided control plans.

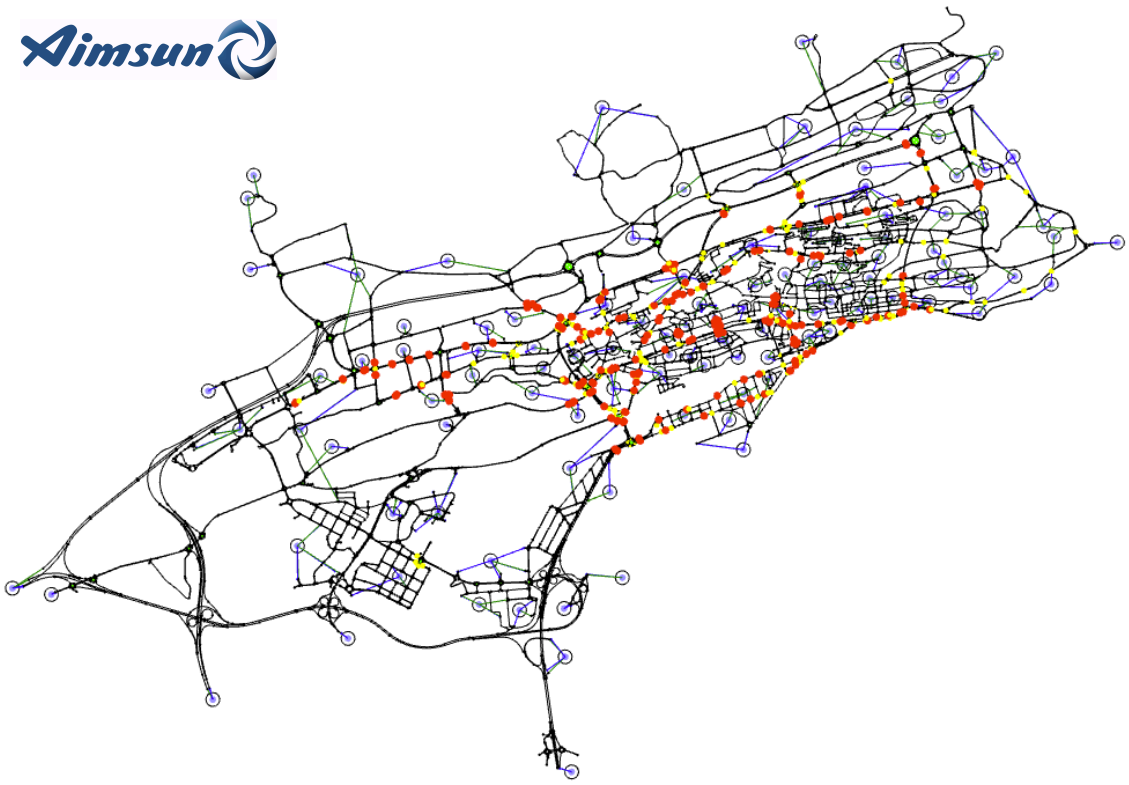

Figure 4. Location of loop detectors (red dots) and signalized intersections (yellow dots) in Santander

- Transit information – Information about the transit routes and schedules were obtained from the agencies operating the services being modelled.
- Network model – Network model with geographical representation of the network at microscopic level in Aimsun has been used to create the network graph of Santander city.

2.3. Turin network

2.3.1. Definition of the network model scope for Turin use case

The city of Turin is the capital of the Piedmonte region in the north of Italy and has a population of 2,308,000 inhabitants. The metropolitan region of Turin city has been selected for study area, as presented in the Figure 5.
The characteristics of the transport network representation that covers Turin metropolitan area are:

- Size (km): 36,00 x 46,00
- Network type: Metropolitan
- Number of centroids: 409 (167281 OD pairs)
- Number of detectors: 1334

- Number of sections: 14882
- Number of nodes: 7803
- Type of signal controllers: Fix
- Simulation time: AM peak (07:00-09:00h)

This area was chosen with support and advise by Turin City Council (TCC) and 5T, and led by network model availability in the VISUM simulation software provided by 5T. By modelling this area, we can leverage prior work, support utilization of the model in other projects and ensure its compatibility with other models available in Turin city for real-time traffic management control and mobility planning. Further network model development in Aimsun provides TCC and 5T the opportunity to work together with compatible tools and benefit from both to answer the questions at hand. Here we illustrate the opportunity of modelling in Aimsun using network model at macroscopic level in VISUM as a starting point.

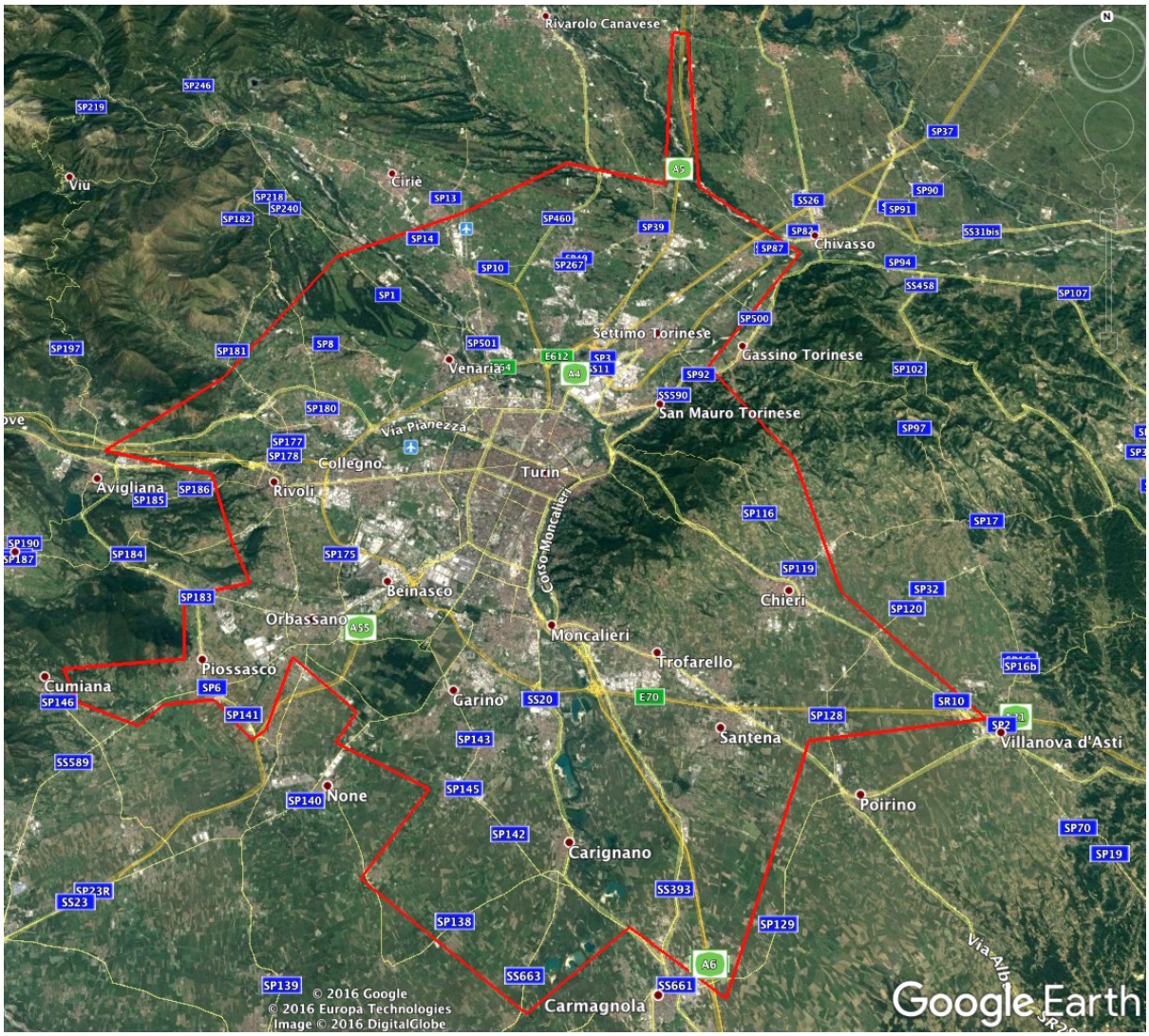

Figure 5. Scope definition of the network model covering Turin metropolitan area

2.3.2. Data collection and application to build the network model

The provided and collected data for Turin city are generally less consistent with the requirements outlined in Table 1. This slight decrease in data availability is a result of the previous practice for the network model developments in Turin city, that are based on macroscopic level of the network representation. In general, macroscopic models require

less detailed network representation, e.g., there is no information on the lane by lane turns, pedestrian crossing, public transport reserved lanes, signal control plans, etc. Following the data availability, in the phase 1 of the SETA project, the network model at the macroscopic level is developed which enables the consistent evaluation of the developments within WP4 with existing technologies available in Turin city.

Major sources of data for the network model development at macroscopic level for Turin city include the following:

- Loop detector data – Traffic observations are available for 1334 individual traffic loop detectors located along major streets and highways across metropolitan region of Turin. For each detector on the network, traffic counts and speeds are provided with 5 minute aggregation level for an entire 24-hour period over three months' period.

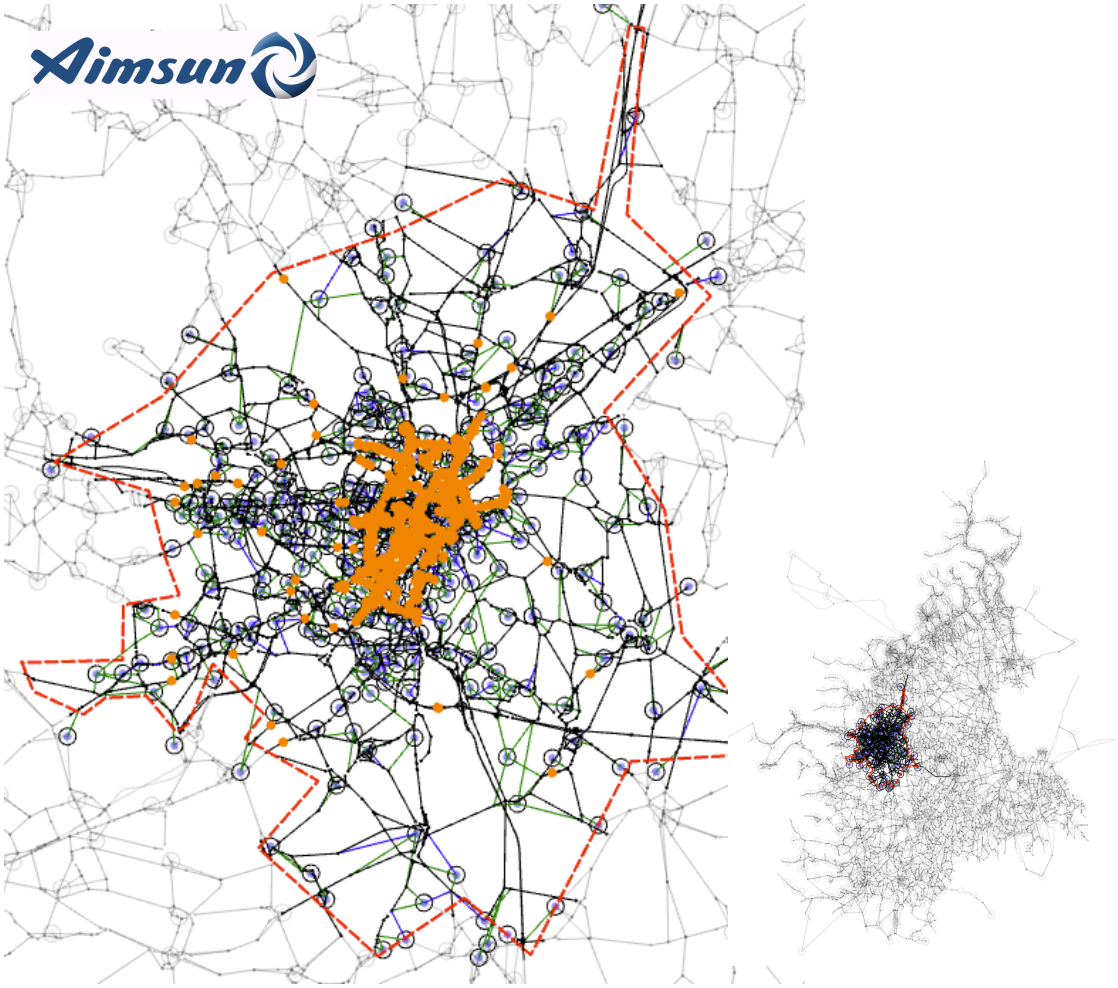

Figure 6. Location of loop detectors in Turin metropolitan area as part of the Piedmonte region

- Traffic performance – Traffic state estimation, expressed in traffic flow and speeds, is collected for all the sections in the metropolitan area over four typical working days for an entire 24-hour period. These data are used to validate consistent performance analysis of the developed model.
- Photo logs and maps – Photo logs were used to collect information about roadway geometry and, in some cases, Piedmonte mobility webpage to help identify bottleneck locations and the extent of queuing. This analysis has relied on photographs that were already available on webpage https://map.muoversinpiemonte.it/#traffic or network view from online sources such as Google Maps.

- Transit information – Information about the transit routes and schedules were obtained from the agencies operating the services being modelled.
- Network model – The entire road network in Turin metropolitan area developed in VISUM is imported using the Aimsun embedded VISUM importer, including geometric data of sections, nodes and turns and their parameters. Refinement of geometry and error-checking after the importation process has been performed to ensure they provide valid and consistent values. Further, this geographical representation of the network at macroscopic level in Aimsun has been used to create the network graph of Turin city.

2.4. Birmingham network

2.4.1. Definition of the network model scope for Birmingham use case

The city of Birmingham is the capital of the England's West Midlands region and has a population of 1,101,000 inhabitants. The part of the metropolitan region of Birmingham city has been selected for study area, as presented in the Figure 7. The characteristics of the transport network representation that covers Turin metropolitan area are:

- Size (km): 36,00 x 46,00
- Network type: Urban/Extra Urban
- Number of centroids: 287 (17030 OD pairs)
- Number of detectors: 57
- Number of sections: 22030
- Number of nodes: 8229
- Type of signal controllers: Fix
- Simulation time: AM peak (08:00 - 09:00)

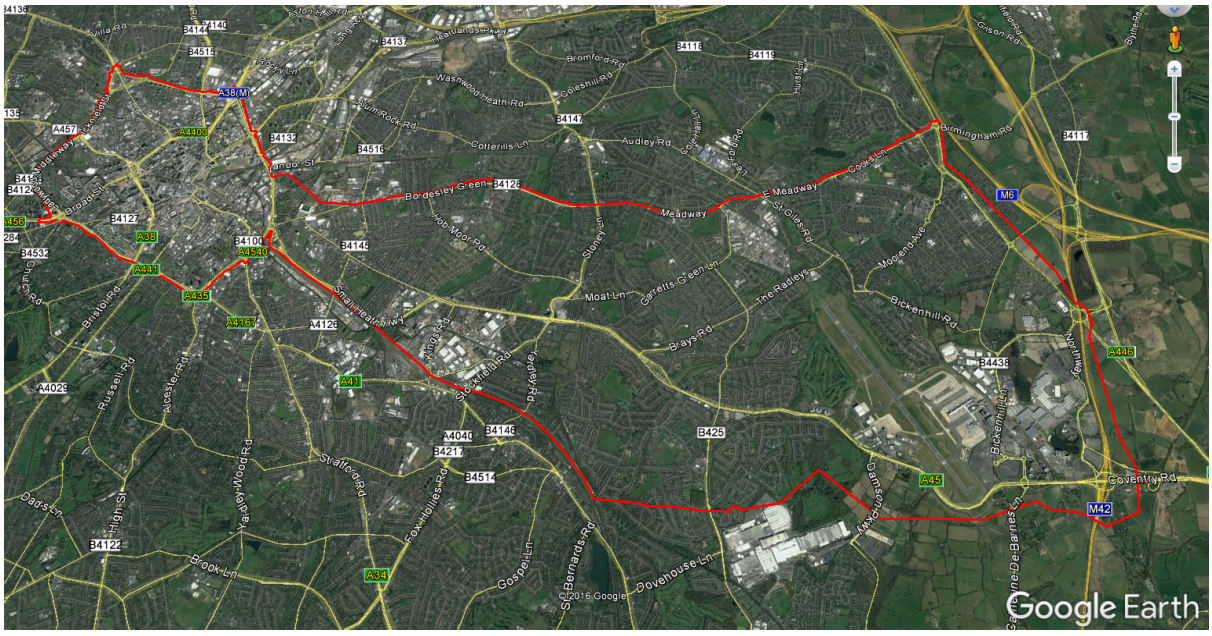

Figure 7. Scope of the network model in Birmingham

This area was chosen based on its availability and high quality representation in Open Street Maps, its complexity and need to address fairly high congestion levels and main mobility critical points in the city, such as Birmingham Airport and National Exhibition Centre. The area encompasses to the west the A4540 ring road plus enclosed city centre area, then to the north the Bordesley Green/Meadway/Cooks Lane, to the East by the A452 and M42,

then to the south by the A45 and Olton railway line. The area was selected with support and advice by Birmingham City Council (BCC), Birmingham use case leader. By modelling this area, we can leverage prior work, support utilization of the model in other projects and ensure its compatibility with other models available in Birmingham city for dynamic traffic management control and mobility planning.

2.4.2. Data collection and application to build the network model

The provided and collected data for Birmingham city are generally less consistent with the requirements outlined in Table 1. This slight decrease in data consistency is a result of the previous practice for the network model developments in Birmingham city, that are based on regional transport models and other models with macroscopic level of the network representation. Following the data availability, in the Phase 1 of the SETA project, the network graph is developed based on high quality of network representation in OpenStreetMap (OSM) and other open data sources provided by BCC.

Major sources of data for the network model development at mesoscopic level for Birmingham city include the following:

- Loop detector data – Traffic observations are available for 57 individual traffic loop detectors located along major streets and highways across metropolitan region of Birmingham which fall within the model area. In the Figure 8, the location of the historic detection information supplied by BCC is shown by black dots. As can be seen there is coverage for the city centre, and decreasing coverage as we move out along the A45 corridor, and while there is some coverage along the B4128, the north-east of the model is not covered. However sufficient coverage looks to be present to investigate any diversion for incoming traffic moving up the A4040. For each detector on the network, flow data has been made available for each day throughout 2016. For initial model build this data has been aggregated to 15 minute intervals and averaged to a typical day.

Figure 8. The loop detector location within use case area

- Floating car data – FLOOW was the primary source of data for majority of links. The FLOOW system uses the floating car data and process them with various built-in capabilities to derive estimates of the average daily traffic flow within 24-hours.

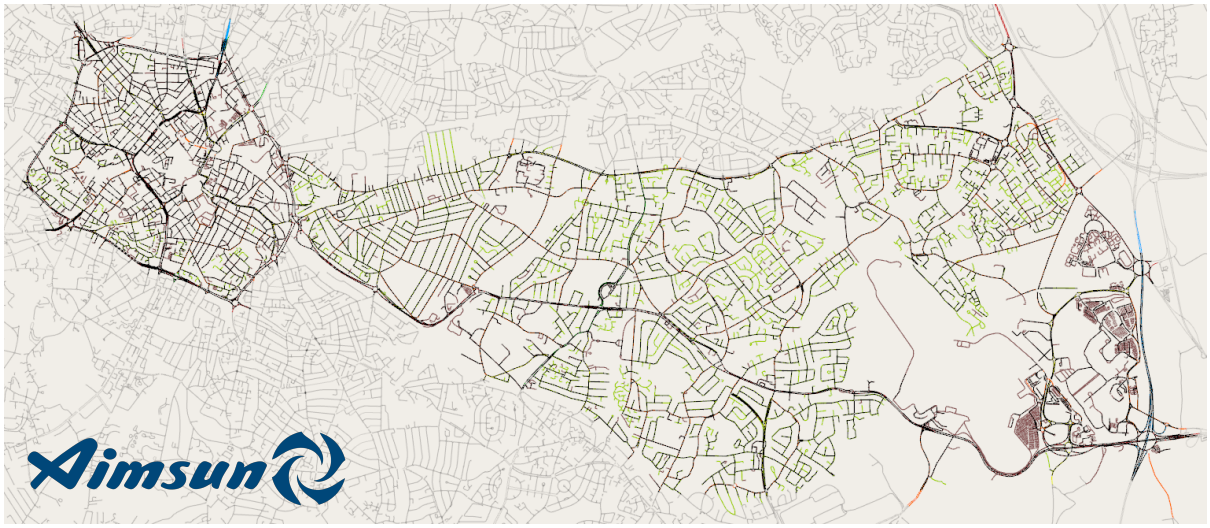

Figure 9. The spatial coverage of FLOOW data

- Photo logs and maps – Photo logs were used to collect information about roadway geometry to help identify bottleneck locations and the extent of queuing. These analyses have relied on photographs that were already available from online sources such as Google Maps.
- Signal timing plans – Signal timing plans for signalized intersections were obtained in spreadsheet and pdf format from the agency responsible for the operation of the traffic control devices in Birmingham. Additional features in Aimsun have been developed to allow automatic import of the control plans into Aimsun. However, limited availability of control plans in electronic version required manual editing and checks had to be performed to ensure there is a reasonable match between Aimsun and provided control plans.

Figure 10. The location of the signalized intersection in use case area

- Transit information – Information about the transit routes and schedules were obtained from Traveline employing information in TransXChange and NaPTAN standard formats, which were then read into the network.
- Network model – Network model with geographical representation of the network at macroscopic model developed in another traffic simulation software called SATURN has been imported and reviewed in Aimsun. This network model has been used for the graph validation developed based on OSM. Further, this geographical representation of the network at macroscopic level in Aimsun has been used to

create the network graph of Birmingham city. This imported network model was used to obtain a peak hour traversal demand configuration suitable for connection to and initial testing of the OSM based network.

3. Vehicle Traffic Prediction Methodology

This section is focused on short-term traffic predictions and demand predictions of vehicles. There is numerous research on short-term predictions using model-based and data-driven techniques as discussed in the deliverable D4.1. Given the data availability within SETA, we have developed robust and scalable techniques for data-driven short-term predictions. We have explored both network-wide and local short-term traffic predictions and they are discussed in this section. The network-wide prediction (Section 3.1) uses clustering based strategies to reduce the high-dimensional nature of the network and the local predictions (Section 3.2) uses rule-based strategy which is robust and easily scalable in terms of network size and modelling complexity. A sensitivity analysis based approach is also developed to perform OD prediction along with simulation models (Section 3.3).

3.1. Network-wide Traffic Prediction

Nowadays, the deployment of sensing technology permits to collect massive spatio-temporal data in urban cities. These data can provide comprehensive traffic state conditions for an urban network and for a particular day. However, they are often too numerous and too detailed to be of direct use, particularly for applications like delivery tour planning, trip advisors and dynamic route guidance. A rough estimation of travel times and their variability may be sufficient if the information is available at the full city scale. The concept of spatio-temporal speed cluster map is a promising avenue for these applications. Instead of modelling each link of the network, the dimensionality is reduced by modelling each cluster/zone of the network. In this section, we introduce generic methodologies for coarsening the network for reducing the network complexity at the city scale and also naïve estimation of the speed for the missing links. The pre-processed data is used to build the spatiotemporal speed cluster using Growing Neural Gas (GNG). A post-treatment methodology is introduced for GNG, which are based on data point clustering, to generate connected zones. An evaluation of the GNG for generating zones is based on the internal variance, inter-cluster dissimilarity and the computation time. The 3D zones are clustered into different classes for dimensionality reduction of the prediction using consensus learning, which is mainly used for clustering of clusters. These clusters define the daily pattern classes. These classes are used to build the models for the prediction. As a benchmark, the speed profiles are used to build the same number of classes as the consensus learning, but using Gaussian Mixture Models. The prediction from the speed profiles using GMM and the zone profiles using consensus are compared to estimate the prediction accuracy. We demonstrate the methodology with the case study example of Santander using loop detector data. An overview of the process is given in Figure 11.

The three main steps are:

- Constructing time-dependent graphs to describe the link-based traffic states.
- Constructing time-dependent graphs to describe the zone-based traffic states.
- Pattern recognition techniques for short term predictions of the traffic states.

These steps are detailed in the sections below.

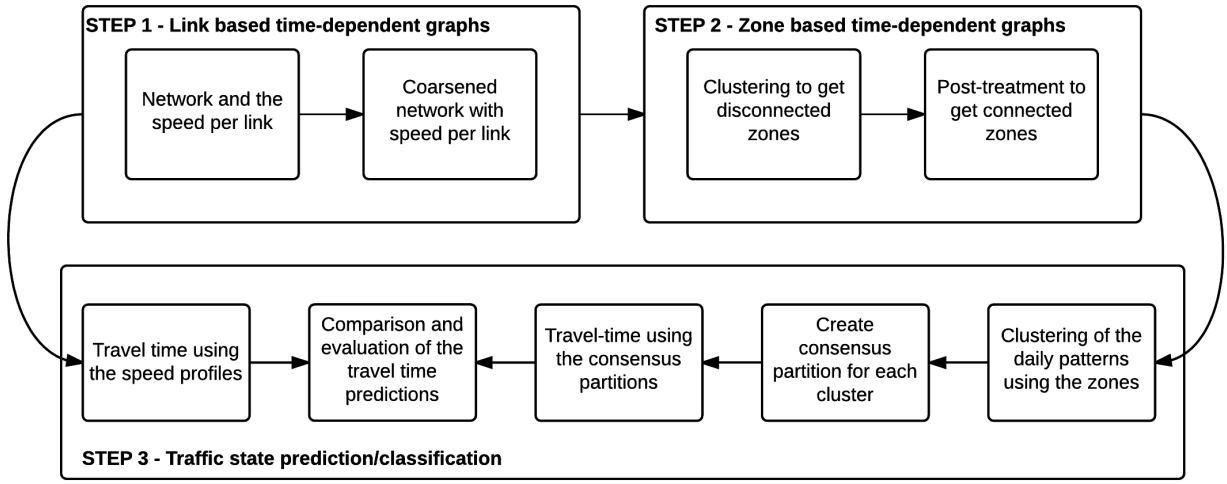

Figure 11: Overview of the network-wide traffic prediction

3.1.1. Constructing time-dependent graphs describing link-based traffic states

Network-wide predictions are usually done using time series which does not contain information about the spatial correlations of the whole network. In our methodology, we construct time-dependent graphs to consider the traffic dynamics in both space and time. Instead of time series, we use 3D spatio-temporal speed maps for network traffic predictions. For this, we need a network representation of the area of interest and the speed per link of the network. The network provided by AIMSUN is enriched with data from WP3 which can be from multiple data sources such as loop detector, floating detector data, etc. This is the input for the traffic predictions. For example, the Santander network with the link speed estimated from loop detector data using the naïve nearest detector method, which is explained in D3.2, is shown in Figure 12(a) and Figure 12(b) shows the 3D Santander map where each slice represents a aggregation in time. Figure 12(b) represents 10 time slices with 15 minutes aggregation.

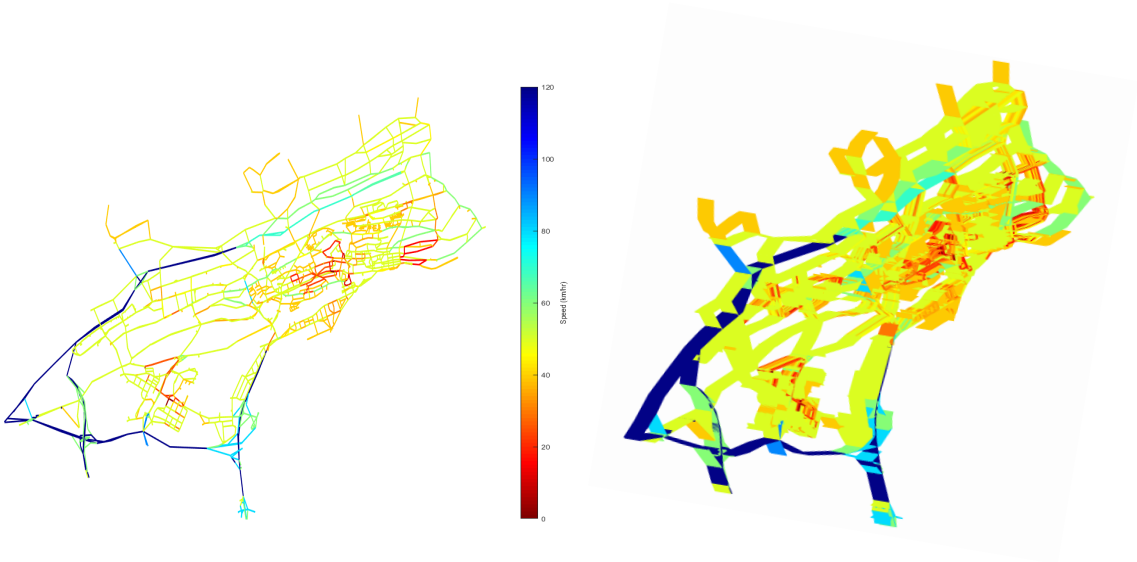

Figure 12: Santander network with 4106 links with their corresponding speed (a) For a given time slice (b) For 10 consecutive time slices for a given day

Depending on the complexity of the network, it is necessary to coarsen the network for faster computations as the network is replicated for each time slice for the prediction. Using the coarsening methodology, we remove nodes that satisfy certain criteria. The matching criteria can differ according to the application. In this work, the matching criterion relates to the differences in link weights which represents the speed of each links. The general idea is that if the links have the same weight, the node that connects the links will be collapsed/removed. The construction of the coarsened graph is based on three steps: (a) the nodes are prioritized or ranked for contraction, i.e. a node with a lower rank will be removed before a node with a higher rank, (b) the contraction rules are determined based on the link difference, and (c) the new weights of the link for the coarse graph are calculated. These steps are detailed in Lopez et al. (2017)

The speed of the link is used for the coarsening. Just using one time slice of data for this might smoothen or remove the network traffic dynamics. In order to maintain the traffic dynamics, the speed of each link averaged over the whole year is used as the weight of the link. This is plausible if only a small percentage of the network have dynamic data. So, most of the links with the static data can be collapsed unless they are connected to the links with dynamic data. Thus, most of the traffic dynamics can be captured with the coarsened network using these averaged speeds.

Figure 13 shows the result of the coarsening on the Santander network. The network complexity is reduced by 30% while keeping the topology of the network intact and capturing the maximum dynamics within the data. In the Santander network, the detector covers only 25% of the network. The rest of the network uses the speed limit of the links as the static data.

Figure 13: Coarsened Santander network with 2879 links

3.1.2. Constructing time-dependent graphs describing zone-based traffic states

Even though the network is coarsened, the prediction complexity is still O(NlogN) for the whole network where N is the number of links. Thus, for network-wide traffic state estimation and predictions, approximations are essential for dimensionality reduction. Instead of modelling each link, we can partition the heterogeneous network, such as an urban network, into homogeneous zones to capture the network-wide congestion dynamics. This can be extremely useful for many applications, such as the dynamic traffic management control and monitoring, route guidance refinement, tour planning and trip advisors.

For partitioning the network, there are various existing methods. Lopez et al. (2017) compares three partitioning methods that belong to two inherently different family of clustering. The normalized cut method based on graph theory and two data point clustering methods - GNG and DBSCAN. A preliminary cross comparison of the clustering techniques in the paper showed that the GNG performs best in generating zones with minimum internal variance, Normalized Cut computes 3D zones with the best inter-cluster dissimilarity and GNG has the faster computation time. Therefore, in this work, we have used GNG for partitioning the network to generate the zones.

GNG is an Artificial Neural Network variant of Neural Gas (Martinetz et al., 1991). GNG begins with two neurons and the network grows during the execution of the algorithm. GNG has been adapted for clustering through a two-step process: running GNG and reconstructing data point clusters based on GNG centroids. The user-specified parameters are the number of centroids N, the maximum number of iteration m, L, the adaptation threshold ε_b, ε_n, the neighborhood size α, δ, the time T, which have been set as $N = 10$, $m = 20$, $L = 50$, $\varepsilon_b = 0.2$, $\varepsilon_n = 0.005$, $\alpha = 0.5$, $\delta = 0.995$, $T = 50$. We represented the 3D network into a data set containing four variables, link coordinates with their corresponding speed and time measurement (x, y, t, s). The four quantitative variables have been normalized. After normalization, we multiply the speed column by a fixed coefficient equal to 3 to be sure that speed is the predominant variable over spatial and temporal coordinates during clustering.

An example of the clustering results are shown in Figure 14(b). However, these clusters are not connected. The homogeneous zone partition needs to be a single connected cluster. Therefore, post-treatment is needed on the clusters that is obtained from data point clustering to obtain connected clusters with minimum intra-cluster speed variance. There are three steps for the post-treatment algorithm and these are:

1. Identifying the connected clusters(CCs) in each of the cluster
2. Assigning the biggest CCs as the initial clusters
3. Assigning all the other CCs to the initial clusters by minimizing the intra-cluster speed variance

These steps are detailed in Lopez et al. (2017)

The data preparation process - the coarsening methodology and the speed estimation of the link - considers a weighted directed network. However, since the partitioning method requires a strongly connected graph, i.e. a directed path exists for every pair of vertices. A real network is strongly connected when a vehicle can reach any link from any starting point. For most of the networks, this constraint is not true. Thus, direction is not a convenient attribute to partition the network. Therefore, for the post-treatment, we assume the network to be undirected.

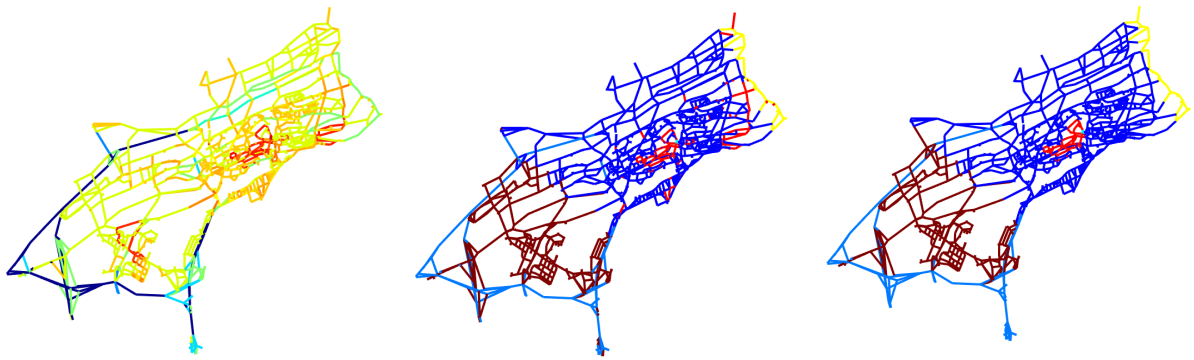

Figure 14: Constructing the connected zones in 2D (a) Network with speed per link (b) Network with unconnected zones after GNG clustering (c) Network with connected clusters after post-treatment

Figure 14 and Figure 15 show an example of post-treatment results in 2D and 3D respectively. Figure 14(a) shows the estimated speed per link which is the raw input for the predictions. Figure 14(b) shows the cluster results from GNG before post-treatment and Figure 14(c) shows the post-treatment results with the same number of clusters as the input for a single time slice. Figure 15 shows the same but in 3D with the result of post-treatment with same number of clusters as the input which is shown in Figure 15(b). There is no difference in post-treatment methodology between 2D and 3D. The only difference is in calculating the 3D adjacency matrix for finding the 3D CCs and the connectivity. The 3D adjacency matrix is defined by creating bi-directional links between the time slices.

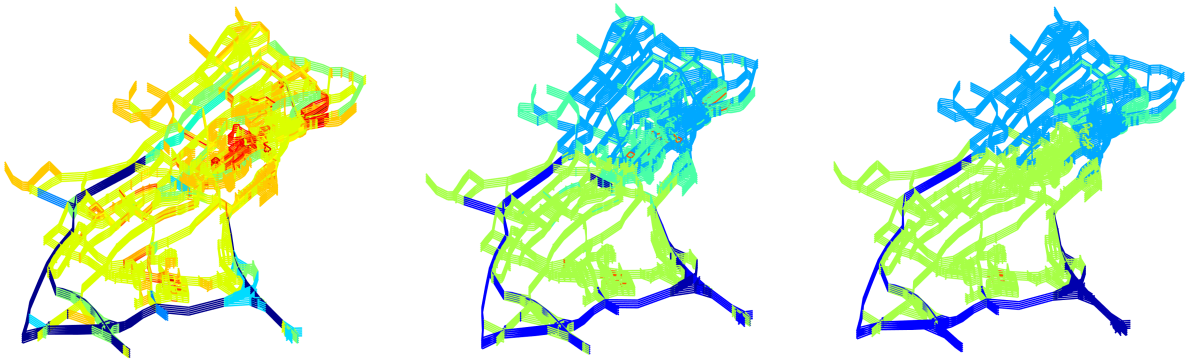

Figure 15: Constructing the connected zones in 3D (a) Network with speed per link for 5 time slices (b) Network with unconnected zones (c) Network with connected clusters after post-treatment

Evaluation Metrics for the Clusters

The zones that are generated are evaluated using three indicators: (i) Total Variance normalized (TVn), (ii) Connected Clusters Dissimilarity (CCD), and (iii) time computation (Lopez et al., 2017). (i) The normalised TV is defined as:

$$TVn = \frac{1}{N} \frac{\sum_{A \in C} N_A * Var(A)}{S^2} \quad (1)$$

This indicator is based on the assumption that a given cluster is composed of links characterized by similar speeds. The speed variance is highlighted. (ii) The second metric used is the CCD. The criterion is the dissimilarity between a given cluster and its neighbouring cluster, i.e. clusters touching the given cluster. CDD is defined as follows:

$$CCD = \frac{\sum_{i=1}^{n} \sum_{k=1+i}^{n} \delta_{ik} |\bar{x}_i - \bar{x}_k|}{\sum_{i=1}^{n} \sum_{k=1+i}^{n} \delta_{ik}} \quad (2)$$

$$\delta_{ik} \begin{cases} 1 & if\ k\ and\ i\ are\ connected\ clusters \\ 0 & otherwise \end{cases} \quad (3)$$

(iii) The time computation indicator evaluates the computational cost of the algorithms. The complexity has to be considered for another size of network and number of time slices. The basic data point clustering methods are faster but a post-treatment process is required. The computational cost of the post-treatment is heavy because it checks the connectivity of the previous results and iteratively updates the clusters. The time computation evaluation includes both partitioning methods and the internal processes for a single day.

TVn and CCD measure the quality of clusters representing the homogeneous zones and the inter-cluster dissimilarity respectively. The time computation quantifies the computational cost. A combined metric is also estimated using these three indicators. An optimal number cluster is directly proportional to CCD (dissimilarity value should be high) and inversely proportional to TVn (low variance within the cluster) and the computational time (lower the computational time, the better). Thus, a normalized combined metric defined in Eq.4 is maximized to estimate the optimal number of clusters.

$$CMn = \frac{CCD}{TVn * computation\ time} \quad (4)$$

For example, for the Santander network, it was found that 10 is the optimal number of clusters based on these evaluation metrics. A thorough sensitivity analysis based on these metrics for the use cases will be provided in D4.3.

3.1.3. Pattern recognition techniques for short-term predictions of traffic states

These connected 3D zones are used for predictions instead of the time series of each links to reduce the dimensionality. However, given a new dataset, it is computationally expensive to match all the historical data to the new data. Therefore, the 3D zones are clustered to form n number of classes. Given a new dataset, this will be matched to each classes instead of matching it to each individual 3D daily zones. The predictions from the 3D speed profiles are used as the benchmark for comparing the predictions from the 3D zones.

The 3D homogeneous regions are clustered to create a daily 3D model for each class. These models are used to match the current traffic state to the historical days. Once, the traffic states are predicted for the next hour or so, it is important to map a route that traverses through the 3D map in terms of travel time rather than traversing a 2D map. In this section, the methodology for classification of the 3D speed profiles and the 3D zones are described. The evaluation metrics is based on the trajectory travel time traversed through the 3D.

Clustering of the daily patterns based on the zone profiles

The 3D zones classes are generated using Normalized cut which is a clustering technique based on similarity matrix. A similarity matrix W is computed between the 3D zones of all the days for clustering the days into different classes. In this work, we used the normalized mutual information(NMI) of the two clusters as the W. NMI is proposed in Cover and Thomas (1991). This metric has been used by Wenjun (2002) to measure the quality of clustering as well. W is defined as follows:

$$W = \frac{I(x,y)}{\sqrt{H(x) * H(y)}} \quad (5)$$

Where x,y are the clustered zones, I(x,y) is the mutual information and H(x) and H(y) are the marginal entropies (Cover and Thomas, 1991).

Normalized cut(NCut) is used to cluster the days based on the similarity matrix W, as the data point clustering cannot be used here. Here, NCut is the spectral clustering using the eigen values of the similarity matrix W. It has been proven that using the spectral clustering is equal to solving the normalized cut (Shi & Malik, 2000). A few examples of the clustering results of Santander into 3 classes is shown in Figure 16.

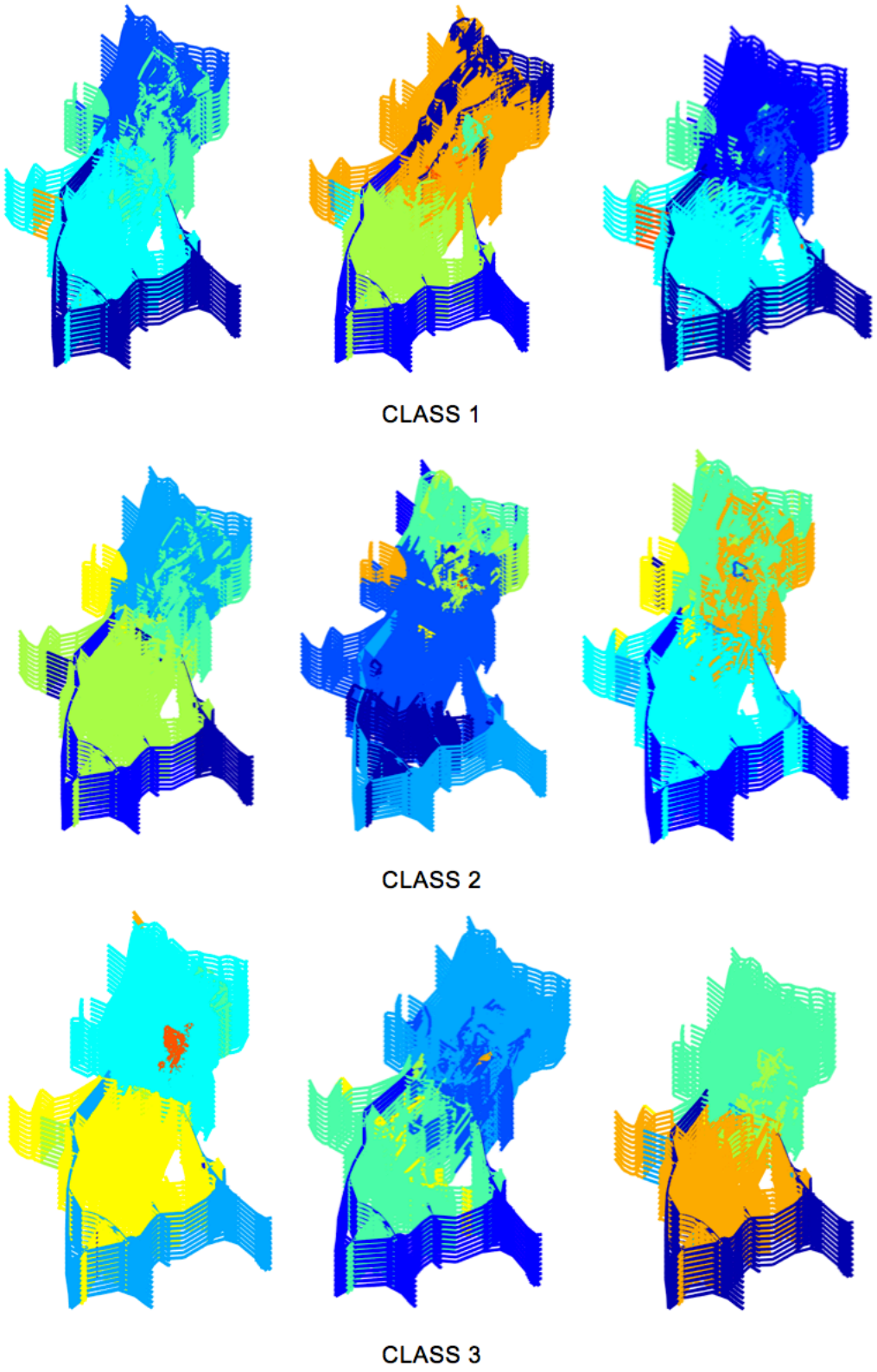

Figure 16: Examples of partitions in each class for Santander use case

Real-time travel time prediction using the clustering

Now there are N number of classes or clusters, we build a partition for each class that is representative of that class. This is build using consensus learning. We call the

representative partition, the consensus median partition. The median partition for each class is initialized by first assigning the partition of the day that maximizes the similarity of all the partitions of that class. This is the partition of 3D zone that is most representative of that class. The initialized median partition is updated by randomly changing one element of the partition and re-computing the similarity matrix again, known as the one element move method proposed by Filkov & Skiena (2004). If the similarity matrix didn't improve after the element move in an iteration, the consensus partition is updated as the partition in the previous iteration. The convergence criteria to get the optimal median partition is when the change in the similarity matrix is not statistically significant after each iteration. The similarity matrix, S, is the sum of all similarity between every partition in a class. Thus, we have a consensus median partition for each class. The consensus partitions of Santander for the 3 classes from Figure 16 are shown in

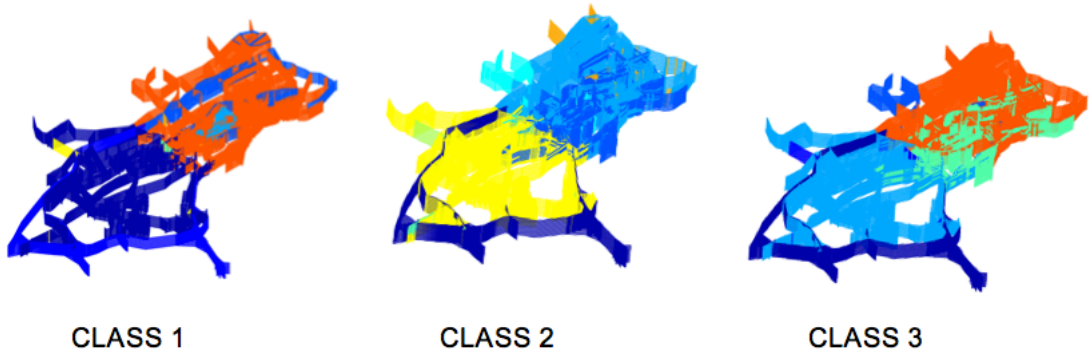

Figure 17.

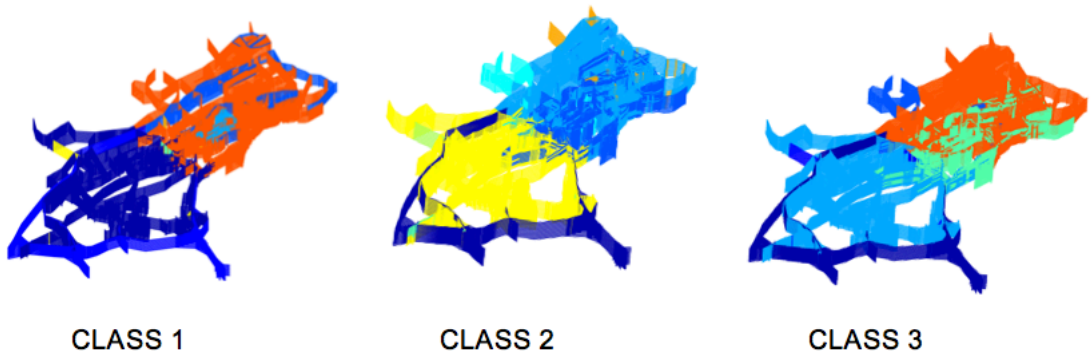

Figure 17: Consensus partition of classes for Santander use case

For the real-time prediction of the travel time from these consensus partitions, there are two main steps – find the class that fits the new data and estimate travel time through the median partition of that class. For fitting the new speed data to the class, depending on the size of the available real-time speed data, the consensus partition is truncated and a similarity matrix is computed between the real-time speed and truncated consensus partition of each class. The consensus partition that maximizes the similarity is chosen as the predicted partition of the new data.

Evaluation Metrics for the Predictions

For the evaluation, we are doing leave-one-out validation. For a given day, the following steps are done without considering that day:

- clustering the days into *n* classes
- create the consensus partition of these classes
- fit the speed profile of the given day to the consensus partitions
- the fitted partition is used to generate travel time for *m* pre-defined routes

These steps are done for each day in the dataset. The prediction accuracy is evaluated by computing the travel time of the pre-defined routes through the fitted consensus partition and the speed profiles and compare the results. The travel time of the pre-defined routes is computed every *t* minutes in the 3D map so as to get a more representative sample. An exhaustive analysis of the travel time error is done to evaluate the prediction using basic performance indicators such as Mean Absolute Percentage Error (MAPE) and the Root Mean Squared Error (RMSE).

3.2. Local Traffic Prediction

Nowadays the trend for short-term traffic forecasting relies on data-driven empirical approaches, given the growing data availability, known as a big data. This creates the necessity to handle both structured and non-structured data as well as to take advantage from contextual information and data coming from multiple sources and observation technologies. Additionally, the short-term traffic forecasting task is inherently a real-time task that must deal itself with the common challenges found in this field, namely high-dimensionality and non-linearity, noisy data from the measurement devices, missing data from faulty or disabled ones, volatility, and adaptation to change in the traffic demand and the traffic supply characteristics. For these reasons, it is widely accepted that a non-parametric approach is usually required to manage the growing complexities as new data is collected.

To deal with some of these difficulties, shallow neural networks and, more recently, deeper architectures have been applied extensively in the short-term traffic forecasting field as they are considered well suited to problems where (i) the input–output data are noisy; (ii) the relationships between these variables are multivariate and highly nonlinear; and (iii) the mapping or relationship is poorly understood (Van Lint and Van Hinsbergen, 2012). In addition, they are well suited for online learning with new incoming data as there are very well studied optimization techniques such as stochastic gradient descent (SGD) which can be applied to tune the parameters over time (learning), in fact the online optimization field is also an active research area nowadays. Besides the usual huge time required to train deeper architectures, the main drawback of this approach is the lack of interpretability and causality in the results, because often the traffic manager agent does not only care about the final accuracy results, but also about understanding the factors that mostly influenced such results. This is usually not possible with neural networks as they work as a black-box approach. In addition, other works reviewed in the literature simply disregard this kind of problematics and set up experiments with cleaned, imputed and even sometimes dropping out anomalous samples (e.g. holidays) from the testing datasets; these scenarios are far away from the expect in a real-time operating setting.

In WP4 we develop a framework built from different machine learning and data analysis components whose predictive system is robust to outliers, irrelevant features and missing data. Developed framework is scalable in terms of network size and can handle growing modelling complexity with new data arrival and adapt to changes in traffic conditions through concept drift detection. The framework is inspired by the works of (Gama, 2010) applied to data streaming scenarios, but tailored to the requirements for this application. In the following sections, the different components of developed framework are presented.

3.2.1. Rule-based prediction framework for automatic knowledge discovery

The framework works in a supervised manner, meaning that for each desired prediction target, i.e. different network locations or forecasting horizons, it is going to discover or unveil a set of rules to gain knowledge about the supervised task, having past observations with

their correct prediction. Then, each rule R contained within each ruleset $\mathfrak{R}$ is composed of an antecedent A and a consequent C with the logical form: $A \Longrightarrow C$. The rule antecedent can be composed of several literals L, where a literal L is a single condition over a specific attribute with a specific split-point v; with the form $(x_j > v), (x_j \leq v)$ if it is numerical, or $(x_j = v)$ if it is categorical. $L(x_i)$ returns *True* if x_i satisfies L, and *False* otherwise. The antecedent is interpreted as a conjunction. In this way, a rule R is said to *trigger*, or to cover, an example x_i if all its literals (the antecedent) are evaluated to *True* on the example.

The consequent (of a rule) is composed of an adaptive output using the multiple rule predictors that the rule may hold (e.g. constant, weighted mean, linear model, or any other functional form). The individual outputs are built at prediction time from the examples gathered in the scope of that rule, then the adaptive output is generated from that population of individual outputs (also could be called experts, following an expert advice schema) weighted by their respective online errors. In addition to the prediction point estimate, an uncertainty interval is given based on the error seen which approximates the real one as the uncertainty associated with covariates is neglected. Finally, each rule R has an associated data structure $\mathcal{L}$ which contains updated statistics from the observed streams (attributes, targets and errors) for those observations gathered by the rule. These statistics are later used for multiple aspects: making predictions, detecting distributional changes and anomalies, evaluating the expansion of a rule, etc. The framework has been designed and implemented based on a modular architecture as presented in Figure 18 such that each unit can be separately replaced or improved. Each component of this framework is described in this section.

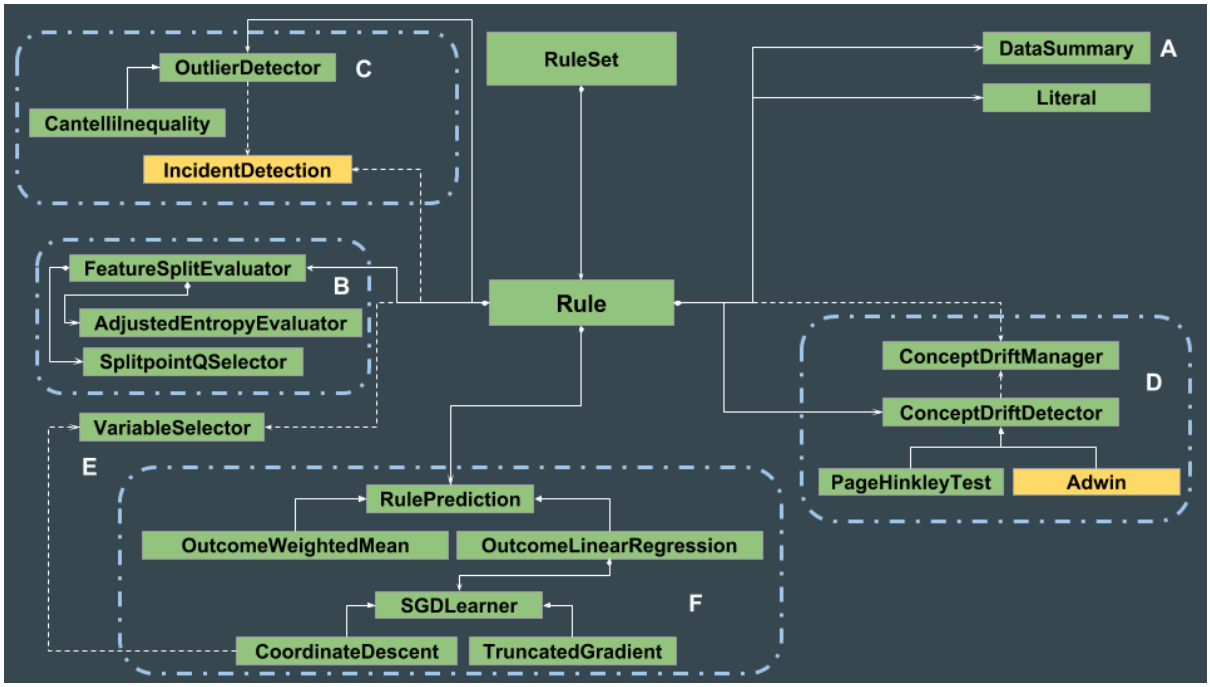

Figure 18. Framework units' graphical schema. Units in green are implemented and in production. Units in yellow are considered for future integration

The traffic prediction framework consists of the following components: the variable selector (E), the anomaly (outlier) detection (C), the change detection (D), and the handling of missing data and winsorizing which take place inside the data summary structure (A).

1. **Component E: Variable selector to manage prior knowledge about the road-network**

The variable selector is the unit aimed to handle the prior information usually put as expert knowledge. In our case, it just set a normalized attractiveness value to each feature separated in different categories (e.g. count, occupancy, speed, time, weather…). Thus, they have an associated probability, that could be updated with new gathered evidence, that is used to select features stochastically as will be explained later in the rule expansion process and in the online learning procedure. Therefore, features associated with detectors in the road network have a normalized attractiveness based on their distance to the point to be predicted. More specifically, the attractiveness is set by the function $1/d$, where d is the orthodromic distance which could be easily replaced by using travel times coming from a transport network model. On the other hand, discrete attributes (e.g. time, weekday, weather…) have a uniform probability in the scope of its own category to reduce the computation time stochastically. Anyway, all this kind of prior knowledge can be adjusted manually beforehand, or a function can be set to adjust these probabilities in runtime.

2. **Component C: Anomaly detection**

Detection of outliers or anomalous examples is very important in on-line learning because of its potential negative impact in the performance of the learning process. For this reason, incoming samples are analysed to detect anomalous samples and to avoid its learning.
Considering the probability $P(X_j = x_{ij}|\mathcal{L})$ of observing a certain value x_{ij} in a rule R given the observed statistics in $\mathcal{L}$, we can compute a score representing its anomaly for X_j:

$$U_j = 1 - P\left(X_j = x_{ij} \,\middle|\mathcal{L}\right) \quad (6)$$

To calculate this probability, we have used Cantelli's inequality (Bhattacharyya 1987), which is a generalization of Chebyshev's inequality in the case of a single tail. Then, for any real number $b > 0$,

$$P\left(\left|x_{ij} - \overline{X_j}\right| \geq b\right) \leq \frac{\sigma_j^2}{\sigma_j^2 + b^2} \quad (7)$$

And then, replacing, the score U_j can be calculated:

$$U_j = 1 - \frac{\sigma_j^2}{\sigma_j^2 + \left|x_{ij} - \overline{X_j}\right|^2} \quad (8)$$

When this score U_j is greater than a specified threshold λ_U (0.9 by default), x_{ij} is considered an anomaly in the context of R. This is an univariate score; then assuming that the attributes are independent, the joint degree of anomaly is computed over all attributes whose univariate score is higher than λ_U:

$$\prod_{j:U_j > \lambda_U} U_j \quad (9)$$

To normalize the degree of anomaly into the interval [0, 1] (where 1 corresponds to all attributes being anomalous and 0 means none of the attributes are anomalous), the following ratio is applied (logarithm functions are applied to avoid numerical instabilities):

$$Ascore = \frac{\sum_{j:U_j > \lambda_U} log\ (U_j)}{\sum_{j=1}^{d} log\ (U_j)} \quad (10)$$

Finally, when $Ascore$ is higher than a specified threshold λ_M (0.99 by default) the observation is considered anomalous and discarded from learning.

3. **Component D: Change detection**

Change detection, also known as *concept drift detection* in the machine learning community (Gama et al. 2014), is a critical component for modelling non-stationary processes as it is our case. For this purpose, each rule has associated a change detector which monitors their error. The idea is that, after a rule has been expanded and, thus, two new rules are created: their individual rule predictors are trained in their respective 'batch' mode with their corresponding gathered observations. From this moment, their residual mean error should be located at zero and it is started to be monitored for changes. When a change is detected (i.e. a significant increase in the error), a signal is sent to the concept drift handler and the rule is removed from the ruleset. The current implemented approach for detecting a change is based on the Page-Hinkley (PH) test (Page 1954), although other approaches are being considered (Bifet & Gavaldà 2009; Bifet & Gavaldà 2007).

The PH test is used to monitor the evolution of a random variable, in our case the on-line error e_i of a rule. PH test updates a cumulative variable m_n which is defined as the accumulated difference between the observed values e_i and their mean $\bar{e}_n$ at the current moment

$$m_n = \sum_{i=1}^{n} e_i - \bar{e}_n - \gamma, \qquad \bar{e}_n = \frac{1}{n}\sum_{i=1}^{n} e_i \tag{11}$$

where γ (0.005 by default) corresponds to the magnitude of changes that are allowed. The minimum value of m_n at the current moment is also maintained: $M_n = min_{i=1}^{n} m_i$. When the difference $(m_n - M_n)$ is greater than a given threshold λ (50 by default), a change is detected and a rupture point is signalled.

4. **Component A1: Handling missing data**

The framework gathers online statistics for each attribute in the context of each rule which corresponds to specific road conditions. So, in the long term, with enough sample size each rule has a good view of their data distribution for each recognized road condition. Thus, for each missing attribute, the framework reconstructs a normal distribution with the gathered mean and dispersion, but limiting the probability density at zero at the current minimum and maximum values in order to avoid extrapolation in the covariates. Finally, missing values can be replaced with samples gathered from this distribution.

5. **Component A2: Winsorizing for extreme values**

When extreme values (outliers) are received, i.e. those whose probability is extremely low in the scope of a specific rule, it is often better to filter them, or else replace them using the handler for missing data described above. Again, assuming a Gaussian distribution, this means considering outliers those values beyond or above approximately 3 standard deviations from the mean.

3.2.2. Rule expansion

Rules could be viewed as high-level features discovered in the road network with the aim of reducing the uncertainty around the prediction target using a specific goodness of fit function. For this purpose, existing rules have a chance to run a rule expansion evaluation process (component B). If the evaluation process is favourable, the current rule disappears and it is specialized into two new rules with their respective observations and statistics. The frequency of this evaluation, which takes place for each rule separately, is crucial as a low frequency can lead to a slow learning of the high-level features while a high frequency can

make the process too sensitive to transient noise. The parameter N_{min} dictates the minimum amount of observations which must be seen, separately on each rule scope, to proceed with a rule expansion evaluation. This threshold N_{min} is preset to an initial value N_{min0}, that is later dynamically adjusted, but never increasing, based on the dispersion of the rule error in a logarithmic scale. More specifically:

$$N_{min} = N_{min0} \frac{1}{ln(e + \sigma_{error})} \quad (12)$$

This dynamic adjustment aims at relaxing the trade-off between prompt but expensive checks and slow but inefficient checks. A high initial value can be set because, afterwards, it is going to be adjusted automatically on the basis of the dispersion of the error rule, which means that if the rule is having a narrow error then it is not necessary to try to specialize it so often.

The rule expansion evaluation process searches to determine the attribute and split-point best scored based on a specific goodness of fit function using the examples seen so far. After selected combinations of features and split-points have been scored, the success of the rule expansion evaluation process is determined by using the ratio of the two best scores and a predetermined confidence-level on the split must be guaranteed so that it can be expanded, by means of the Hoeffding bound (Hoeffding 1963), as used in (Duarte & Gama 2015; Gama 2010). This probabilistic inequality gives the number of examples n required to expand a rule:

$$\varepsilon = \sqrt{\frac{R^2 \, ln\,(1/\delta)}{2n}} \quad (13)$$

It guarantees that the true mean of a random variable r, with range R, will not differ from the sample mean more than ε with probability $1 - \delta$ (0.01 by default). The best two potential splits are compared, dividing the second-best scoring by the best one to generate a ratio r in the range 0 to 1. To decide if the rule is expanded or not, the upper bound of the ratio of the sample average ($r^+ = r + \varepsilon$) is checked to be below 1, $r^+ < 1$, and if that happens then the true mean is also below 1 meaning that with confidence $1 - \varepsilon$ the best attribute and split-point of the data are the ones being tested. Nevertheless, the measurements of the two best splits are often extremely similar and, despite ε decreases considerably as more examples are seen, it is not possible to select which one is better with certainty. In this case, a threshold τ (0.05 by default) is used and if $\varepsilon < \tau$ the split option with higher scoring is chosen to expand the rule.

In the end, if the rule expansion evaluation process is successful, expanding a rule R consists of creating two new separate rules (R_{left}, R_{right}) with their respective observations by adding the new literal created with the corresponding attribute and split-point to the sets of antecedents.

There are two steps in the rule expansion evaluation process, namely: (1) the searching step to find which attributes along with their corresponding split points are going to be evaluated, and (2) the scoring process to rank those selected combinations.

1. **Reducing the search for rule expansion**

When it is time to run the rule expansion evaluation process, it is needed to decide which attributes and split points are going to be measured. Perhaps the intuitive idea is simply to evaluate all the existing features, but in the current high-dimensional problem this can lead to time-consumption problems especially if the threshold N_{min} is low. Not only that,

overfitting may occur if, for instance, detectors that are very far away are selected as antecedents. Therefore, the candidates to be evaluated are selected probabilistically based on their distance using the variable selector.

The split points to be evaluated for each selected continuous attribute, are selected using the cumulative probabilities, or quantile functions, to represent the whole distribution of the gathered observations. While in the case of discrete attributes the selection is based on the generation of multiple continuous intervals.

Continuous attributes considered include the traffic count, occupancy and speed from the whole road network. Discrete attributes considered include the time of the day, weekday and weather information.

2. **Scoring the candidates for rules' literals**

The goodness of fit used to evaluate the different combinations of features and split-points is based on entropy minimization, process which is also known as information gain. From an information theory perspective, entropy $H(X)$ measure the randomness of the information in the random variable X. The entropy is maximized if the distribution is vague (i.e. uniform with equal probability in the whole space), this is the situation of maximum uncertainty as it is most difficult to predict the outcome. When there is less uncertainty, i.e. when the outcome is peaked around certain location values, there is a lower entropy quantity. At the extreme case, when there is no uncertainty because we are sure about the outcome the entropy is zero (MacKay 2003).

When scoring a proposed splitting, entropy is used as information gain score. This means that we score the entropy of the current rule before splitting versus the entropy of the proposed new rules weighted by their respective new sample sizes. If entropy is reduced with the new splits, that means we have gained certainty about the outcome.
In addition, the goodness of fit function considers the missing data ratios of the feature candidates, penalizing those whose missing data ratio is higher considering these as untrustworthy candidates.

3.2.3. Rule prediction

Currently, there are two proposed strategies to forecast within the rules' scope, and a strategy to combine these forecasts into a single point-estimate prediction.

1. **Weighted mean**

This forecaster is simply the weighted historical mean of the true target of the past examples covered by the rule. This is equivalent to a naïve predictor, which is good to maintain among the forecasters population as it has no direct dependencies on external states.

2. **Penalized linear regression**

A linear regression model is built using the examples covered by the rule. Although short-term traffic prediction is a highly non-linear problem, we use the rules to discover the nonlinearities and combine a population of lower-level, specialized linear models.

Concerning the learning procedure an incremental approach based on Stochastic Gradient Descent (SGD) has been adopted instead of using a closed-form solution to take advantage of continuously data incoming in real-time. Bayesian approaches usually scale poorly and have been discarded due to the high dimensionality and real-time operation. The loss in the cost function is the residual sum of squares, but in addition to the sum of squared error loss, a penalty term (L1 norm) has been included in the minimization problem in the search of shrinkage and sparse solutions. This approach is also known as LASSO (Hastie et al. 2015; Tibshirani 1996). Coordinate-wise gradient descent has been used to obtain the parameter

estimates because it applies well to our case where $n \ll p$, and it has been successfully applied to this kind of high-dimensional problems (Friedman et al. 2010), and it has been demonstrated to be efficient in large problems (Nesterov 2012). The minimization problem to solve has the following (Lagrangian) form:

$$\min_{\beta \in \mathbb{R}^p} \left\{ \frac{1}{N} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1 \right\} \quad (14)$$

When training in batch mode, a regularization path is obtained efficiently using warm-up starts for different penalty λ values, which means that multiple solutions exist ranging from the least penalized (i.e. ordinary least squares solution) to more penalized and sparse solutions. This is ideal for our problem as we cannot select the best penalty value λ through cross-validation in real-time operation, and thus we let a set of multiple solutions or experts coexist while combining them adaptively. Furthermore, there is an additional advantage as we can use the sparsest solutions to calculate predictions even when less penalized solutions cannot due to missing data from some data dependencies.

In online learning, the coordinate descent is also applicable using mini-batches of incoming data, a small learning rate and the soft-thresholding technique (Shalev-Shwartz & Tewari 2011). Additionally, instead of updating all the coordinates in each call, it is possible to rely on the variable selector to select probabilistically a subset of the features. Another implemented and tested approach for online learning with sparse solutions is called Truncated Gradient based on the work by (Langford et al. 2009), however in the end we have decided to rely on the online coordinate-wise gradient descent due to a more direct mathematical relationship with the batch version. The truncated gradient has other hyper-parameters with no direct relationship with the penalty value λ in batch. Anyway, more testing would be needed, perhaps including other stabilized versions of it (Ma & Zheng 2016).

In addition to the individual models which form the experts' population for each rule, there is an adaptive strategy to combine them.

3. **Adaptive strategy**

Finally, an adaptive strategy combines the forecasters population derived from the previous two strategies that exist within a rule, namely: the weighted mean and the different penalized linear regressions. This adaptive strategy is based on the on-line estimation of the mean absolute error (MAE), where the contribution from each forecaster to the final point-estimate prediction is determined inversely proportional to their current online estimation of the error. This on-line estimation of the weighted error follows a fading factor strategy. To do so, the total sum of absolute deviations T is monitored. When new examples arrive, T is updated as follows:

$$T = \alpha T + |\hat{y}_i - y_i| \quad (15)$$

Where $0 < \alpha < 1$ (0.95 by default) is a parameter that controls the importance of the oldest and newest examples.

A similar strategy is followed when multiple rules cover a single observation and an adaptive response is given in addition to the individual rule responses. Finally, in addition to the prediction point estimate, an uncertainty interval is given based on the error seen which approximates the real uncertainty because the uncertainty associated with the input information is neglected. But, for our case, it is an inexpensive approach that give us a good approximation about the uncertainty.

3.2.2.1 Regularization to avoid overfitting in prediction

It is especially important to control the complexity growth to not incur in overfitting and thus losing the ability to generalize with new observations, which is especially sensitive in a non-parametric approach. For example, in traditional rule sets and decision trees, a post-pruning schema is adopted after the model has been built using the whole data set in batch mode. However, it is not trivial to control the trade-off in a continuously evolving and non-stationary scenario with new incoming data where it is unexpected how complex the road-network modelling will grow. In this way, the following regularization procedures have been adopted to help in preventing overfitting.

Regularization at rule set level

Different ways of regularization have been adopted within the adaptive framework (ruleset), namely:

1. **Timing for rule expansion evaluation process**

As commented previously, rules are candidates to be expanded (i.e. specialized) after observing a given number of examples (N_{min}). Thus, using the dynamic adjustment previously explained, the idea is to have a relaxed high initial value that will decrease if the standard error dispersion raise. What this means is that if we have a rule with a narrow residual distribution, we can let the rule as-is during more time before a rule expansion check. On the other hand, if the rule has a wider error dispersion, then it is going to be checked earlier for a rule expansion check in order to get specialized.

2. **Deciding the (truly) best split**

We rely on the application of the Hoeffding's bound to be truly confident that the greatest score from a combined feature and split-point is the best one given the current sample size. This means that the success of the rule expansion evaluation process is determined by using the ratio of the two best scores, and a predetermined confidence-level on the split must be guaranteed so that it can be expanded.

3. **Minimum number of observations for splitting**

The splitting evaluators require a minimum number of observations in order to consider the evaluation of a given split. This aims at avoiding rules too specialized with very few cases.

4. **Penalizing untrustworthy attributes for splitting**

It may happen that sometimes a specific combined feature and split-point gives us an extremely good predictive value, thus being very well scored, but unfortunately it is also an untrustworthy feature which means that it is missing most of time. In this sense, a penalty term is added to the splitting evaluator cost-function penalizing the score of those attributes with more missing data ratio.

5. **Drift detection**

The main goal of concept drift detection mechanisms is to deal with non-stationary processes. In our case, the mechanism monitors the mean error of each rule. Therefore, it can also be used to detect that the rule is losing its initial accuracy, which suggests that its generalizing ability is poor and, thus, over fitted. This is a reason to remove it from the rule set.

Regularization within each rule

1. **Shrinkage and sparsity**

For the linear regression models, the penalty term (L1 norm) added to the learning procedure helps in obtaining shrinkage coefficients to avoid high-variance models and overfitting. This especially applies to our high-dimensionality problem. The sparsity achieved also helps in getting more parsimonious model and thus less data dependencies, which mean more robust models when facing missing data.

3.2.2.2 Relevance of framework for real-time prediction

Updating statistics can be done efficiently in streaming, either in real-time or using small mini-batches of samples. Online learning usually takes place using mini-batches of data so it is not done after each observation is received. The rule expansion evaluation process, which is the most expensive step, is only checked after the rule has seen a customizable, usually large, number of observations which can be adjusted automatically if the dispersion of the rule error starts to raise. Additionally, as all the rulesets for all sensors are usually fed from the same data source (i.e. the road network database), the approach benefits from concurrent reading access and avoids memory redundancy. Finally, since the approach reads a single input data source but creates the rules independently for each location, it can be easily parallelized.

3.2.4. Application relevance in SETA use cases

The current adaptive rule-based framework for traffic prediction will be applied to the short-term traffic prediction for the traffic volume up to 1 hour ahead, using the cases of Santander, Turin and Birmingham as a high-dimensional case. This also covers the results illustration and report including basic performance indicators which include the Mean Absolute Percentage Error (MAPE) and the Root Mean Squared Error (RMSE).

Conceptually, the goal is to build an analytical model or mapping function, as described in this Section, which unveils the dependencies for traffic state variables short-term future trend. This mathematical structure (along with its learned parameter values) has the form:

$$Y(t+h)_{D,k} = f(X(t)), \quad (16)$$

Where:

$Y(t+h)_{D,k}$ - represents the k predicted variable for detector D for time $(t+h)$, being h the forecasting horizon.

f represents the analytical model.

$X(t)$ represents the input information at time t.

In the phase 1 of SETA, the model input information is composed of the traffic measurement from real loop detectors in the whole network, i.e. those recorded by detectors and stations placed along the road network usually in the form of single or dual loops. However, the model developed in this phase is not limited to application of traffic measurements from loop detectors but can incorporate measurements from other mobility data sources that may act as virtual detectors, e.g. traffic cameras, floating car data or aggregated GPS and phone data.

The predicted traffic state is comprised of the macroscopic traffic variables, i.e., number of vehicles or traffic flow, occupancy and speed in the network. From the perspective of the SETA project innovative technologies, this subtask is twofold. From the perspective of personal traffic information, traffic state prediction may assist in making better route choice and departure time decisions. For professionals, such traffic state information will provide criteria with which to better manage and control traffic to reduce congestion.

Highlighting the main contributions that this work intends to do:

- Non-parametric approach: complexity grows and evolves with more data
- Adaptation to change (concept drift) over time in the data streaming scenario
- Automatic network relationships (rules) discovery
- Automatic feature selection
- Expressiveness and interpretability

- Robustness to outliers and irrelevant features
- Ease of including new input data sources (weather, incidents reports)
- Handling of missing data
- Scalability

Another advantage of the model developed in this phase is to predict traffic state in the network when the amount of historical data is limited. In the evaluation phase 4.3 and 4.5 it is expected to demonstrate the performance and improvement of traffic state prediction accuracy by combining various data sources.

3.3. Vehicle Traffic Demand Prediction

Transport authorities and practitioners have long been concerned about the unavailability of reliable dynamic OD demand estimates which limits the potential for dynamic traffic assignment (DTA) deployments to analyse and alleviate traffic congestion as part of the Intelligent Transportation Systems (ITS). In congested networks, changes in the demand affect travel times. In turn, travel times affect the route choices and travel times from trip origin nodes to traffic observation detectors that determine the assignment fractions or the relationship between OD flows and traffic observations. Modelling of non-linear relationship between traffic observations and OD flows, and its dependency on variations in OD flows has been identified by many researchers as a key challenge in the estimation and prediction of a high-quality OD matrices. For example, the dependence of link-flow proportions on the demand flows in assignment matrix should be explicitly included in the DTA process. Finding derivatives of link-flow proportions and traffic observation with respect to demand flows can be cumbersome task, often judged not feasible in terms of computation time.

From a modelling point of view, the most distinguishing difference between the OD demand estimation approaches, is how the relationship between state variables (e.g., OD flows, OD proportions) and any available traffic data (e.g., link traffic counts, speeds) is defined, calculated and re-calculated throughout the estimation process. An accurate description of this relationship leads to an accurate description of traffic state reality in the network, but to more complexity as well. In the past decades, a rich body of literature, stressed the need of relaxing the fixed relationship assumptions in mapping demand flows to traffic observations through estimation process when the congestion occurs in the network. Researchers have been devoted to development of the methods to capture the impact of demand variation on traffic observations, that can be categorized in analytical derivation, simulation-, numerical- and heuristic-based approximation methods.

Typically, to calculate the weights between OD flows and link traffic counts (usually measured by loop detectors in the form of sensor counts), dynamic assignment matrices are commonly used. Theoretically, these assignment matrices can be analytically derived using network topology, path choice set, current route choice model and equilibrium travel times (Ashok & Ben-Akiva 2002). However, it is recognised that the complexity of the problem at hand can quickly lead to intractable situations (Ashok & Ben-Akiva 2002). Further sophisticated analytical derivations are required to capture the relationship between parameters with less direct impact and non-linear relationship.

The simulation-based approximation of the relationship between demand flows and traffic observations uses traffic simulator to uncover this relationship without the direct derivation of the assignment matrix. The most studied assignment matrix-free method is the Simultaneous Perturbation Stochastic Approximation (SPSA) method (Balakrishna et al. n.d.; Cipriani et al. 2011; Cipriani et al. 2013; Cantelmo et al. 2014; Antoniou et al. 2015) which allows one to approximate a descent gradient direction with significantly lower computational resources than through the explicit calculation. (Antoniou et al. 2015)

proposed the Weighted-SPSA (W-SPSA) algorithm to overcome the deteriorated performance of the gradient calculated by SPSA algorithm. Although, the main advantage of such method is that the complex relationship between demand flows and traffic observations, not only traffic counts, is estimated by simulation model, the high number of simulation runs for large-scale networks where the DTA is computationally intensive still remains to be resolved. For example, due to stochasticity of the simulation model, for each perturbation of the SPSA or W-SPSA where gradient needs to be determined, the DTA has to be replicated R times, leading to 2R runs (Antoniou et al. 2015).

Recent studies by (Toledo & Kolechkina 2013; Frederix et al. 2013; Shafiei et al. 2017) rely on linear approximation of the assignment matrix with non-separable response in every iteration, which relaxes the assumption of constant assignment proportions and explicitly accounts the congestion effects. This definition requires the computation of the marginal effects of demand flow change on the assignment proportions at the current solution of each iteration. It is possible to use the finite differences approach to numerically approximate the Jacobian matrix by using traffic simulator, but it would require in every iteration of the gradient solution to perturb each element in the OD demand vector, one at a time, leading to 2RDK runs, where D the number of OD pairs in the network and K the number of time intervals for the simulation period. To overcome computational overhead, authors proposed heuristic-based approaches. (Toledo & Kolechkina 2013) neglected the effect of changes in one OD pair over the other OD pairs in the assignment matrix, (Frederix et al. 2013) implemented space decomposition of the network in the congested and non-congested sub-networks, where derivatives were computed only for congested area. (Shafiei et al. 2017) reduces computation costs through iterations progress and computing derivatives on OD pairs whose flows have higher tendency to vary during dynamic OD demand estimation process. However, all these approaches rely on strong heuristic assumptions such as ignoring the effect of OD demand changes outside of congested area or have been tested on relatively medium sized networks. Further research is necessary to develop solution approach for nonlinear OD estimation problem that will guaranty its reliability and applicability in large scale networks.

Here we choose a very different method for exploring the relationship between OD flows and traffic observations, and impact of demand flow changes on traffic conditions in the network. In other fields where exact analytical expression between input and output data is almost impossible to derive, sensitivity analysis (SA) is a commonly used method to reveal the impact of the input parameters on the model predictions. The obvious choice for a SA approach is to use some sort of affordable model-free computational method to estimate the first-order sensitivity Sobol indices, which does not scale with input dimensionality. In this paper we perform sensitivity analysis based on Random Balance Design FAST (RBD-FAST) technique proposed by (Tarantola et al. 2006) to identify the OD pairs whose demand variation has a significant impact on the traffic observations without explicitly relying on the assignment matrix. RBD-FAST technique belongs to group of frequency-based SA methods, which is computationally cheaper than Monte Carlo-based techniques. In the sensitivity analysis, mesoscopic traffic simulation model in Aimsun is employed as a black-box which realistically captures the congestion phenomena that is more adequate in developing dynamic OD estimation algorithms.

3.3.1. Problem formulation

This section describes the most critical issue in OD matrix estimation, whether static or dynamic, the relationship (mapping) of the observed traffic condition data with unobserved OD flows. From a modelling point of view, the most distinguishing difference between the OD demand estimation approaches presented in the literature, is how the relationship between OD flows and any available traffic data (e.g. link traffic counts, speeds, densities, etc.) is defined, calculated and re-calculated throughout the estimation process. This

relationship is often accomplished by means of an assignment matrix. In the dynamic problem, the assignment matrix depends on link and path travel times and traveller route choice fractions - all of which are time-varying, and the result of dynamic network loading models and route choice models. These dynamics are reflected in travel times between each origin and destination trips on a network, influenced by traffic link flow. While a vast body of literature has been developed in this area over the past two decades, this section focuses on some of the efforts that highlight the basic problem dimensions.

The general OD estimation problem is to find an estimate of OD demand matrix by effectively utilizing traffic and demand observations. Let $\Omega \subseteq U \times U$ be set of all d OD pairs in the network, and $\hat{L} \subseteq L$ be the set of r links where traffic data observations are available. The time horizon under consideration is discretized into K time intervals of equal duration, indexed by $k = 1, 2 \ldots, K$. If $\mathrm{x} \in \mathrm{R}^n$ represents the OD demand for each OD pair in Ω, the x_k represents the OD demand at departure time interval $k_i, i = 1, \ldots, K$. Here the dynamic OD demand is represented by a vector, rather than a matrix. It is also important to define κ, the maximum number of time intervals needed to travel between any OD pair in the network. For instance, in dynamic context, depending on the size of the network and its complexity (travel times and distance from the origin o to the destination d), some vehicles could need more than one-time interval to reach their destination d or pass traffic sensor at link l. The vector $\mathrm{y}_{k,\hat{L}} = A(\mathrm{x}_h) \in \mathrm{R}^r$, for time interval $h = \{k, k-1, \ldots, k-\kappa\}$, represents the observed link traffic data at time interval k (e.g. link traffic counts) for each link in $\hat{L}$.

Given a vector of observed traffic data at time interval k, $\mathrm{y}_k \in \mathrm{R}^r$, the dynamic OD estimation problem consists of finding an OD demand for departure time k, x_k, $\hat{y}_{k\,\hat{L}}(\mathrm{x}_h)$ such that is as close as possible to observed values y_k. Therefore, the dynamic OD estimation problem is formulated as:

$$\hat{x}_k = \underset{x \geq 0}{argmin}[\alpha f(x_k, \hat{x}_k) + (1-\alpha) f(\hat{y}_k, y_k)] \quad (17)$$

Regardless of the function f used, the purpose is to obtain an OD matrix that yields OD flows and traffic data as closely as possible to their observed values. When solving the OD problem in Eq.(17) the relationship between traffic observations and OD demand has to be defined, implicitly or explicitly. Most dynamic OD demand estimation methods, define this relationship implicitly by the assignment matrix that can be expressed as:

$$\hat{y}_k = \sum_{h=k-\kappa}^{k} A_k^h x_k \quad (18)$$

There are two main drawbacks of relationship defined in Eg. (18):

1. Separability of traffic count observations: it assumes that the traffic flow observed at the link l during time interval k can always and only be changed by changing one of the OD flows that passes link l in time interval h when x_k is assigned. This assumption of separability is incompatible with some typical phenomena in congested networks, such as congestion spillback between links and time lags due to the delay during congestion. In these cases, it is very likely that increasing an OD flow will cause delays to other flows that do not pass that time-space interval, hereby altering the amount of flow passing the link in the considered time interval. This issue has been addressed in past studies (Yang & Zhou 1998; Tavana & Mahmassani 2001; Lundgren & Peterson 2008). (Frederix et al. 2013) suggested using the Taylor

approximation to specify the linear approximation of Eg.(18) using non-separable response function, given by

$$\hat{y}_k = \sum_{h=k-\kappa}^{k} A_k^h(x_0)x_k + \sum_{h=k-\kappa'}^{k} (x_h - x_0)\left[\sum_{h'=k-\kappa'}^{k} \frac{d(A_k^{h'}(x_{h'}))}{dx_{h'}}\, x_{h'}\right] \quad (19)$$

2. Limited only to one data source: formulation of relationship by assignment matrix in Eg. (18) and Eg. (19) restricts dynamic OD demand estimation problem to use of traffic count data only, which can potentially over-fit to counts at the expense of traffic dynamics. Relationship between traffic condition data, such as speeds and densities, and OD flows are expected to be non-linear and approximations similar to the assignment matrix cannot be justify (Balakrishna & Koutsopoulos 2008). This issue has been addressed in the past studies (Balakrishna & Koutsopoulos 2008; Cipriani et al. 2011; Cantelmo et al. 2014; Antoniou et al. 2015) who proposed use of traffic simulation models to capture the nonlinear relationship between OD flows and traffic observations instead of assignment matrix.

Although presented solutions significantly contributed to quality improvement of dynamic OD demand estimates, they still share a common challenge to overcome high computational costs. A complicating factor in utilizing these methods for estimation or prediction purposes, is that OD matrices are very large data structures, that grows rapidly in large networks. Even in case such high-dimensional OD flows can be reduced (see e.g. (Djukic et al. 2012) and this is not entirely unlikely, there are serious methodological difficulties in finding optimal solutions (e.g. getting stuck in local minima, slow convergence, high number of simulation runs, etc.), aside from the computational and memory requirements for such a procedure on the basis of thousands (to millions) of traffic observations. For example, computing the exact Jacobian vector in the second term of Eg. (19) with respect to changes in OD flows for each OD pair remains intractable even when efficient, well calibrated, DTA model is used.

In the next section, we propose a different method for exploring the relationship between OD flows and traffic observations by applying sensitivity analysis. We relax assignment matrix dependence of Eg. (19) to evaluate marginal effect of demand flow changes on traffic conditions in the network.

3.3.2. The concept of sensitivity analysis based on RBC-FAST method

One way to explore the marginal impact of the changes in the OD demand on traffic observations, is to identify the most sensitive input parameters, i.e., OD pairs, through sensitivity analysis. Sensitivity analysis (SA) of model output investigates how the predictions of a model are related to its input parameters. In doing so, the traffic simulation software is considered a black box in function form $Y = f(X)$, providing a certain outcome Y (traffic observations) given certain input (OD demand), X. The obvious choice for a SA approach is to use some sort of affordable model-free computational method to estimate the first-order sensitivity Sobol indices, which does not scale with input dimensionality. There are various solutions to limit the large number of samples needed without fundamentally changing the overall idea. In this sensitivity analysis, the solution strategy is to use so-called RBD-FAST (Random balance design) technique from frequency-based SA methods, which is computationally cheaper than Monte Carlo-based.

The original formulation of FAST (Fourier amplitude sensitivity test) technique is introduced by (Cukier et al. 1978). Sample points of input parameter space are chosen such that the indices can be interpreted as amplitudes obtained by Fourier analysis of the function. Further extension of this method provided by (Tarantola et al. 2006) to avoid the problem of interferences, is RBD-FAST, where the RBD stands for random balanced design. RBD-

FAST is a group of modifications of FAST technique, which use random permutations of design points to avoid interferences. The original idea by (Tarantola et al. 2006) is to assign the same frequency to all input variables, but to randomize their values independently before evaluating f. The first-order Sobol index of X_j, for $j = 1,\ldots,N$ can then be estimated by reordering the evaluations in the way X_j was permuted, so that the amplitude at the frequency returns the sensitivity of X_j only.

3.3.3. Sensitivity analysis with mesoscopic simulation model

This section describes how the developed models described in Section 2 of this report will be set up for the evaluation experiments of this technology and how these models are organized for sensitivity analysis with mesoscopic simulation model in Aimsun.

In the evaluation experiments, an advanced traffic simulation model Aimsun, will be used as a black-box to perform the proposed sensitivity analysis. In our experiments, we will use the mesoscopic event-based demand and supply models in Aimsun, each synthesizing microscopic and macroscopic modelling concepts. They couple the detailed behaviour of individual drivers' route choice behaviours with more macroscopic models of traffic dynamics. The travel demand in Aimsun is represented by dynamic OD demand matrices. Vehicle generation is done for each OD pair separately with arrival times that follow an exponential distribution. The iterative interaction between demand and supply models allows the system to update the set of routes and the travel times after each iteration leading to robust estimation and prediction of traffic conditions in the network.

For this study, a route choice set will be pre-computed in Aimsun and used as fixed for all the simulation runs in sensitivity analysis. In this way, dependence of re-routing effects on the changes in the OD demand is ignored. Here we focus to investigate effects of travel time variation and congestion spill-back on traffic observations in the network.

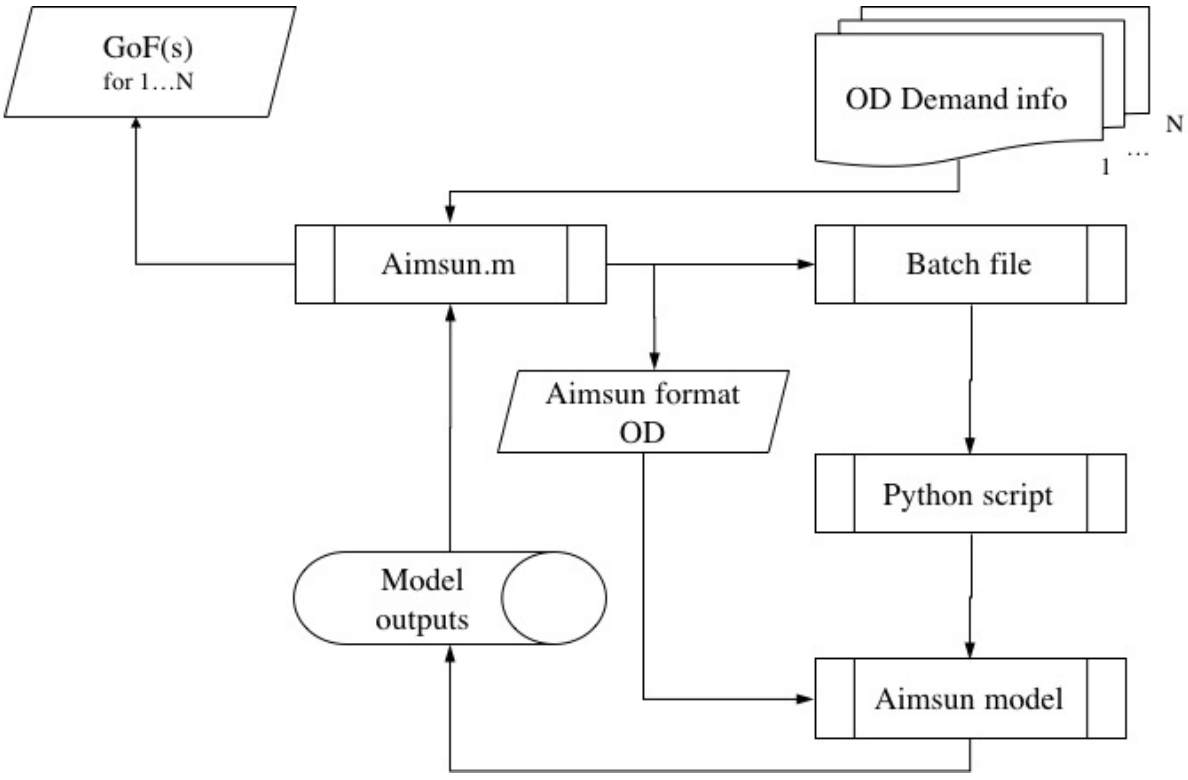

Figure 19. Sensitivity analysis flowchart with the main elements.

Figure 19 presents a sensitivity analyses flowchart with the main elements implemented in Aimsun. For each OD demand vector generated by RBD-FAST technique, the Aimsun call function (AIMSUN.m) is initiated. This function converts the demand to be simulated to the Aimsun format, creates the batch file to execute the requested simulations, generates the Python file with the Aimsun run flags and finally calls and executes mesoscopic traffic simulation in Aimsun with these inputs and fixed paths. After the simulation runs have been completed, it imports the observed traffic data and the simulation outputs and calculates the Goodness of Fit (GoF) measures that were defined within the algorithm and assignment matrices. Since the possible measures of performance are all the time series of counts at the existing detectors (see Figure 4 Figure 6 Figure 8) a strategy to aggregate them in a single measure needs to be put in place. In order to assess the dependence from the GoF measure selected, we defined three of them, namely, the root-mean-square error, the mean absolute error and GEH statistics.

3.3.4. Application relevance in SETA use cases

Dynamic OD demand estimation and prediction methods consist in predicting unknown dynamic OD matrices based on past known high dimensional historical OD matrices data and traffic observations. Solution algorithms proposed in literature to solve the problem given in Eq. (17) incorporate computing the marginal effects of demand changes on traffic observations that lead to high computational costs for medium- or large-scale networks. In this situation, dimensionality reduction of simulation runs required to capture these marginal effects is necessary leading to improve prediction computational performance.

In order to overcome problem related to dimensionality of OD demand we propose the application of sensitivity analysis based on RBD-FAST technique as a pre-processing tool with historical OD demand data before estimation and prediction process. As we shown in previous sections, RBD-FAST technique provides tools for identification and selection of the OD pairs whose changes significantly dominate changes in traffic conditions and observations in the network. Application of proposed methodology on the use cases in SETA is twofold:

1. Proposed heuristic solution approach can be employed to reduce the computation time for the dynamic OD demand estimation and prediction
2. Proposed sensitivity analysis can reveal the most significant OD flows that cause the congestion in the network and to indicate whether these OD flows are observed with sufficient number of detectors.

Here we briefly summarize a heuristic approach that uses RBD-FAST technique in the following steps:

1. **Identification step**: first in this step we run RBD-FAST technique to explore impact of the variability patterns in OD demand matrices on traffic observations using traffic simulation model as a black-box, as we described in previous sections. In this manner, we can identify the OD pairs that significantly dominate changes in traffic conditions in the network.
2. **Reduction step**: in this step we propose the selection criteria of OD pairs based on sensitivity indices magnitudes. For example, only the OD pairs having a sensitivity index above the specified threshold value will be selected.
3. **Estimation step**: finally, OD demand estimation and prediction methods are applied such that numerical derivatives of changes in OD demand on traffic observations are computed only for the chosen subset of OD pairs to estimate OD demand. Advantage of this approach is that it is not limited to particular OD estimation method, rather it can be combined with multiple existing approaches.

This approach will be evaluated for the use cases defined in Section 2.1 and 2.3 in deliverable 4.3.

4. Public Transport Traffic Prediction Methodology

4.1. Prediction of public transport within SETA

Generally speaking, the prediction of public transport consists of two components, the demand and supply. Differing from the road traffic, the supply of public transport is usually predetermined by the scheduled timetable, and public transport users, i.e., the demand part, would decide whether they should coordinate their travel plans depending on the frequency of public transport services. The prediction of public transport, especially the short-term prediction, used to be quite difficult mainly due to the lack of data which can help to monitor and quantify the dynamics of such complex systems in terms of both passengers (demand) and vehicles (supply). Nowadays, with the emergence of advanced public transport data, such as Automatic Vehicle Location (AVL) data, Automatic Passenger Count (APC) data, and Automatic Fare Collection (AFC) data, performing such predictions becomes a promising task, yet a significant amount of research effort is still required.

During the first phase of SETA, while data collection is still ongoing in all three use case cities, two preliminary studies that are closely related to public transport predictions have been conducted. Both of these studies intend to address the high-dimensionality problem adhered to most public transport systems where passenger demand at a large number of public transport stops needs to be forecasted simultaneously. Moreover, a more useful representation form of demand, i.e., origin-destination (OD) matrix, even makes this problem more difficult as the dimension (number of stops) surges. Consequently, we first attempt to perform spatial clustering of public transport stops (Section 4.2) and then apply Principal Component Analysis (PCA) to reduce the dimensionality (Section 4.3). The clustering of public transport stops makes sense not only for the purpose of dimensionality reduction, but also for the modelling and prediction of public transport users' behaviour. This is because in many cases travellers can always board and alight in the neighbouring stops of their origin and destination, considering a reasonable walking or biking distance. The application of PCA, from another perspective, shows the possibility of integrating dimensionality reduction technique into the whole prediction methodology. These developed methodologies lay a foundation for further development of public transport prediction while more data are being collected.

4.2. Spatial clustering of public transport stops

We developed a method to quantitatively determine the clustering of public transport stops by considering both flow and spatial distance information (Luo et al., 2017). Differing from the traditional way of grouping stops based on Traffic Analysis Zones, the proposed data-driven method is based on the k-means clustering algorithm with four steps (Figure 20), and provides another effective and efficient solution to those applications involving transit demand aggregation based on directly observed flows rather than their proxies.

Given a set of *n* observations $(\mathbf{x_1}, \mathbf{x_2}, \dots, \mathbf{x_n})$, each of which is a *d*-dimensional real vector, this clustering algorithm aims to partition the *n* observations into K $(\leq n)$ mutually exclusive and collectively exhaustive clusters $\mathbf{C} = \{C_1, C_2, \dots, C_K\}$. It iteratively determines the centre $\boldsymbol{\mu_i}$ for each cluster C_i and assigns each observation to a cluster whose centre is closest to the observation. This iterative clustering process terminates when the assignments no longer change, which can be described as to minimize the within-cluster sum of squares (sum of distance functions of each observation in the cluster C_i to the centre $\boldsymbol{\mu_i}$):

$$\underset{\mathbf{C}}{\operatorname{argmin}} \sum_{i=1}^{K} \sum_{\mathbf{x} \epsilon C_i} \|\mathbf{x} - \boldsymbol{\mu_i}\|^2 \qquad (20)$$

Its main disadvantage is that the number of clusters, *K*, must be supplied as a parameter. In this study, a four-step *k*-means-based station aggregation method is proposed, in which a quantitative way to determine the optimal *K* is incorporated as described in Figure 20.

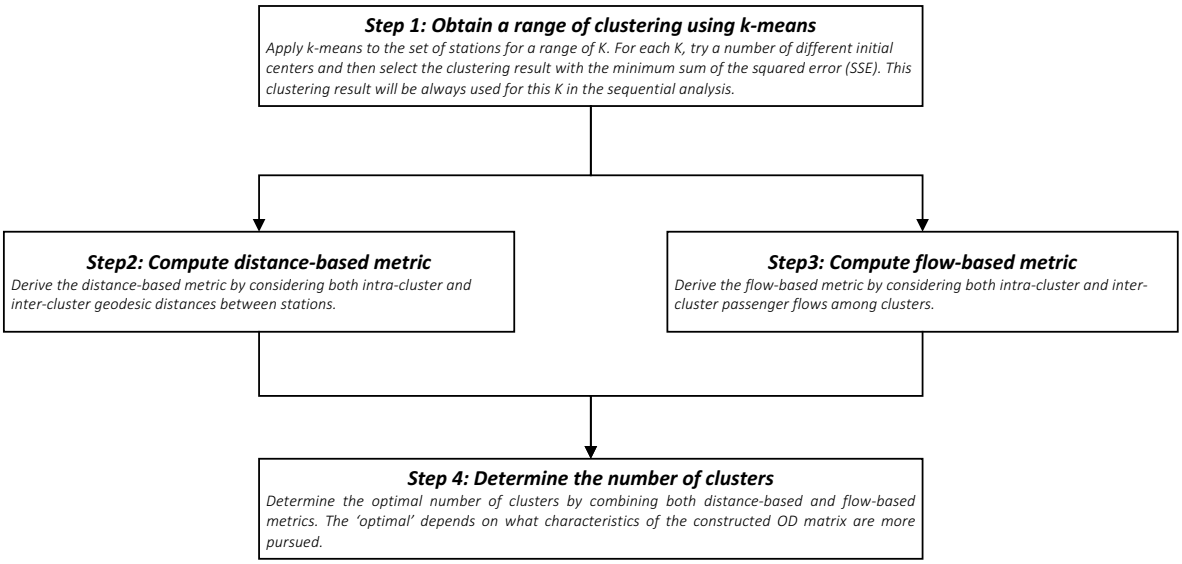

Figure 20 Implementation steps of the proposed k-means-based station aggregation method.

The method starts with finding the best clustering based on a chosen measure for each *K*, and then continues with the computation of two metrics related to spatial distance and passenger flow respectively. In the final step, two metrics are integrated for the determination of the optimal number of clusters K^*. Such a method is flexible as it can accommodate different formulations of both metrics and final integration function in order to cater different purposes pertaining to the construction of transit O-D matrix. The essential idea, however, is to maximize either the intra-cluster or the inter-cluster flow while maintaining the spatial compactness of all clusters simultaneously. More details are discussed in the following subsections.

K-means-based Clustering

Given that the clusters of transit stations should be spatially compact, the geodesic distance between points, which can be calculated based on their coordinates, was used as the only feature in the *k*-means clustering. While implementing the *k*-means algorithm, a set of *K* points were input as the initial cluster centres so that the algorithm could proceed with iterations itself. Since the result of the *k*-means algorithm can vary given different initial centres, a common way to obtain better and reproducible results is to perform the algorithm a number of times with different initial centres and select the initial centres which produces the optimal clustering in terms of the adopted measure. In this study, a measure called sum of the squared error (SSE) was employed to help select the initial centres because it can reflect the quality of a clustering. The lower SSE is, the better the clustering. The SSE is defined as follow in the current case.

$$SSE(K) = \sum_{i=1}^{K} \sum_{x \in C_i} d_{\mu_i,x}^2 \qquad (21)$$

where $d_{\mu_i,x}$ denotes the geodesic distance between a station and the cluster centre to which it belongs.

Distance-based Metric

The construction of the distance-based metric examines the spatial compactness of a clustering by taking into consideration both intra-cluster and inter-cluster distance measures. The former one computes the square of distance between a point and its cluster center, and then takes the average of all of them, denoted by D^{intra}:

$$D^{intra} = \frac{1}{N}\sum_{i=1}^{K}\sum_{x \epsilon C_i} d_{\mu_i,x}^2 \quad (22)$$

N is the number of stations in the study area.

The inter-cluster distance measure, D^{inter}, on the other hand, only takes the square of minimum distance between cluster centres because as long as the minimum of such distance is maximized, the others will by definition be larger than it. This measure is defined as follow:

$$D^{inter} = \min d_{\mu_i,\mu_j}^2, \forall j \neq i \quad (23)$$

The two measures are then combined by taking the ratio as follows:

$$\tau = \frac{D^{intra}}{D^{inter}} \quad (24)$$

where τ denotes the final distance-based metric. To obtain the optimal number of clusters in terms of spatial compactness, τ is minimized since the intra-cluster distance measure D^{intra} in the numerator should be minimized while the inter-cluster distance measure D^{inter} in the denominator should be maximized.

Flow-based Metric

The passenger flow at a station level can be first derived from the original dataset and then be aggregated based on a specific clustering. The flow-based metric provides additional information that can be utilized to determine the optimal number of clusters. Intuitively, total intra-cluster flow decreases as the number of clusters grows given the constant total flow over the entire study period. More flow is naturally assigned to the inter-cluster one.

When considering the flow information, we can either seek to maximize the total inter-cluster flow over total intra-cluster one or vice-versa depending on the application and the analysis objectives. An argument in favour of the former case is that it leads to more flow being assigned as inter-cluster (non-diagonal) elements in the O-D matrix. In contrast, by making the intra-cluster flow more significant, most self-contained and coherent clusters in terms of travel demand (diagonal elements) can be obtained, which is more desirable from a planning perspective. In the current case of Haaglanden, the Netherlands, the second option was eventually adopted and the following two flow measures are proposed:

$$F^{intra} = \frac{1}{K}\sum_{i=1}^{K}\sum_{x_m,x_n \epsilon C_i} f_{x_m,x_n} \quad (25)$$

$$F^{inter} = \frac{1}{K^2 - K}\sum_{i=1}^{K}\sum_{j=i}^{K}\sum_{x_m \epsilon C_i, x_n \epsilon C_j, \forall i \neq j} f_{x_m,x_n} \quad (26)$$

where f_{x_m,x_n} denotes the passenger flow from station x_m to station x_n and *K* denotes the number of clusters. F^{intra} and F^{inter}, essentially, represent the average intra-cluster and average inter-cluster flow, respectively. To combine two measures, the ratio of F^{intra} to F^{inter} is adopted and defined as follow:

$$\delta = \frac{F^{intra}}{F^{inter}} \quad (27)$$

where δ denotes the flow-based metric. To obtain most self-contained clusters, δ should be maximized so that the average intra-cluster flow is as significant as possible.

Determination of the Number of Clusters

To determine the optimal number of cluster with both distance-based and flow-based metrics, different objective functions can be formulated. Since in the current case we aim to (1) obtain clusters that are as spatially compact as possible, which can be achieved by minimizing τ; (2) attain an intra-cluster flow as strong as possible, which can be achieved by maximizing δ, a straightforward way that takes the ratio of δ to τ is adopted. A scaling procedure is applied to both metrics before taking the ratio so that their magnitudes are comparable.

$$X' = \frac{X}{X_{max}} \quad (28)$$

After applying the scaling procedure, the optimal number of clusters K^* is attained:

$$\underset{K \in [K_{min}, K_{max}]}{\arg\max} \frac{\delta'_K}{\tau'_K} \quad (29)$$

where δ'_K and τ'_K denote the scaled flow-based and distance-based metrics for the *K* clustering, respectively.

4.3. Passenger flow analysis based on Principal Component Analysis

We conducted a multivariate analysis of transit passenger flows based on a well-known dimensionality reduction technique, Principal Component Analysis (PCA). It contributes to the development of multivariate analysis on transit passenger flows, and shows the potential of incorporating PCA into short-term forecasting.

Background

PCA was initially proposed to describe the variation of a set of uncorrelated variables in a multivariate data set. So far it has been extensively used as a technique to perform various tasks, such as dimensionality reduction, factor analysis, feature extraction, and lossy data compression. In the field of traffic and transportation, for example, PCA was utilized to compress traffic network flow data, and was integrated into dynamic origin-destination (O-D) estimation and prediction in order to overcome the computational problem caused by high-dimensional O-D matrix data.

The central idea of PCA is to reduce the dimensionality of a data set consisting of a large number of interrelated variables, while retaining as much as possible of the variation present in the data set. PCA achieves this target by projecting the observations onto a new set of axes which are called the PCs. Each PC has the property that it points in the direction of maximum variance remaining in the data, given the variance already accounted for in the preceding components. As such, the first PC captures the total energy of the original data to the maximal degree possible on a single axis. The following PCs then capture the maximum

residual energy among the remaining orthogonal directions. In this sense, the PCs are ordered by the amount of energy in the data they capture.

Methodology

By performing PCA on the flow data, a smaller number of dimensions can be found and leveraged to well approximate original high-dimensional data. Let *X* denote a matrix of multivariate flow time series as the equation below shows. Each column i of *X* denotes a single flow variable, while each row j represents an observation of all flow variables at time j. This yields a $t \times p$ matrix X, where t represents the total number of time instances and p represents the total number of flow variables.

$$X = \begin{pmatrix} x_1(1) & \cdots & x_p(1) \\ \vdots & \ddots & \vdots \\ x_1(t) & \cdots & x_p(t) \end{pmatrix} \quad (30)$$

As shown in the equation below, obtaining all the PCs of *X* is actually equivalent to calculating the eigenvectors of $X^T X$ which is a measure of the covariance between flows.

$$X^T X v_i = \lambda_i v_i \quad (31)$$

where λ_i is the eigenvalue corresponding to eigenvector v_i $(p \times 1)$ and the number of eigenvalues/eigenvectors is equal to the number of variables p. In fact, the eigenvalue λ_i indicates how much variance of the original data is explained by the dimension i specified by eigenvector v_i.

Arranging all the eigenvalues in a descending order, the first PC is thus the eigenvector which corresponds to the largest eigenvalue since it accounts for the greatest variance in the entire data. By mapping the original data onto the derived principal component space, it can be seen that the contribution of dimension i (the i-th PC) as a function of time is given by Xv_i. Normalizing this vector to unit length as shown in equation below, we obtain a $t \times 1$ vector u_i which contains the information of temporal variation along the i-th PC. As a matter of fact, the vector u_i captures the temporal variation common to all flows along this dimension (PC). The set of vectors $[u_1, u_2, \ldots, u_p]$, which are perpendicular, can thus be referred to as the eigen-flows of X.

$$u_i = \frac{Xv_i}{\sqrt{\lambda_i}} \quad i = 1,2,\ldots,p \quad (32)$$

Let *V* denote a $p \times p$ matrix consisting of all the PCs $[v_1, v_2, \ldots, v_p]$ which are arranged in order. The first column v_1 refers to the first PC, and so on. Let *U* denote a $t \times p$ matrix of which column i is u_i. Consequently, each individual flow X_i can be written as:

$$\frac{X_i}{\sqrt{\lambda_i}} = U(V^T)_i \quad i = 1,2,\ldots,p \quad (33)$$

where X_i is the time series of *i*-th flow and $(V^T)_i$ is the *i*-th row of *V*. This equation indicates that each flow X_i is essentially a linear combination of the eigen-flows with weights specified by $(V^T)_i$. By selecting the first $r\,(r \leq p)$ eigenvectors with largest eigenvalues, the information contained in original data *X* can then be effectively transformed onto a r-dimensional subspace. It is shown in the equation below how the approximation can be done.

$$\hat{X} \approx \sum_{i=1}^{r} \sqrt{\lambda_i} u_i v_i \quad (34)$$

Result demonstration

The proposed PCA approach is demonstrated by using the smart card data from Shenzhen metro system. The scree plot in Figure 21a provides the readers with a chance to conduct visual examination of PCA. It can be seen through the sharp knee of the curve that the majority of variance contained in the data is virtually contributed by the first few eigen-flows, namely the temporal variability on the first few PCs. Figure 21b further explicitly displays that 8 and 29 PCs, respectively, can account for over 90% and over 95% variance in the data.

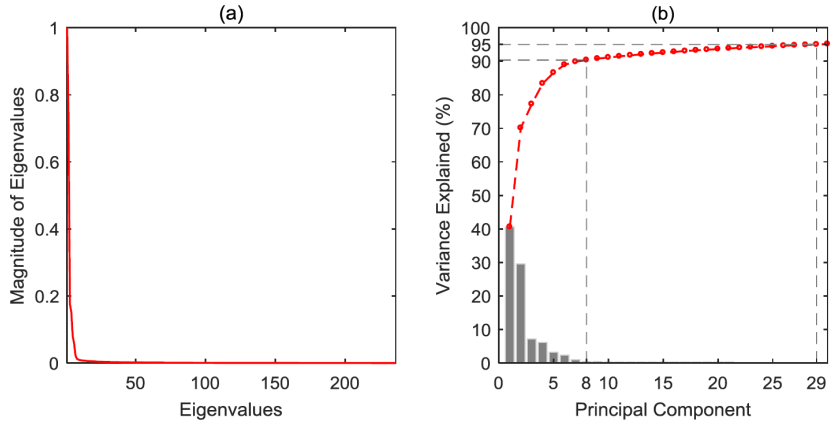

Figure 21. Demonstration of the low dimensionality of entry and exit flows. (a) Scree plot of eigenvalues; (b) Cumulative percentage of the total variance explained by PCs (principal components). Over 90% variance can be explained by only 8 PCs, while over 95% can be explained by 29 PCs.

Based on PCA, the original flows can be approximated using a set of selected PCs. Essentially, such approximation is realized by forming a linear combination of eigen-flows. Figure 22a demonstrates three typical examples of reconstructed flow time series using both 8 (90% total variance explained) and 29 PCs (95% total variance explained). Specifically, the left column displays the overall time series for three weeks, while the right one zooms into a more detailed level with only one day illustrated.

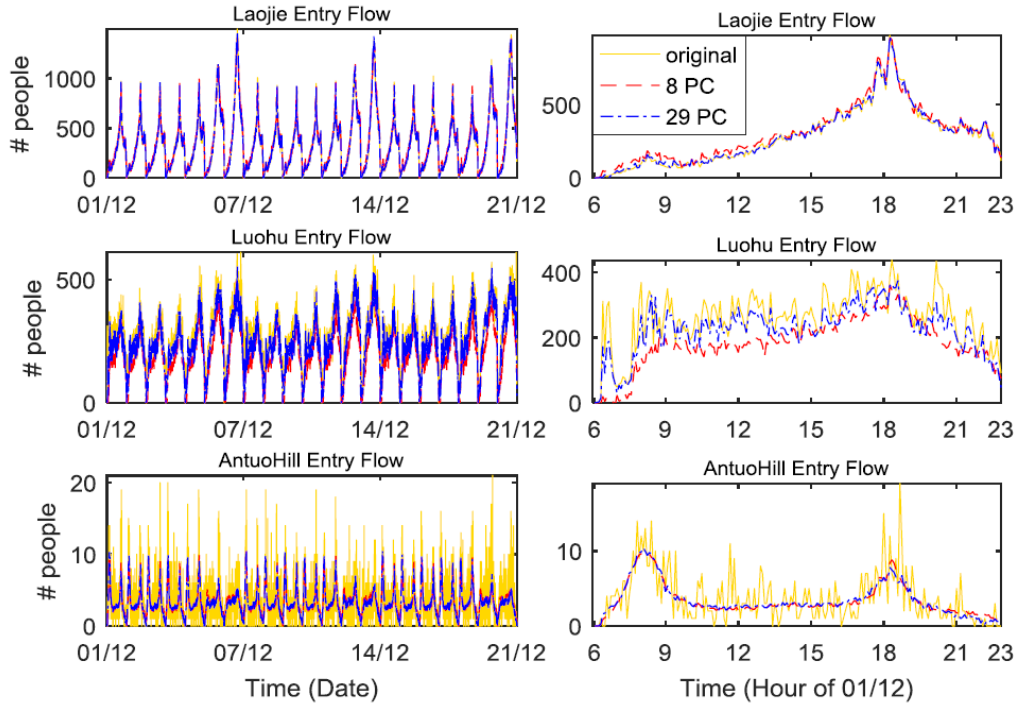

Figure 22. Examples of approximating original flows using different number of PCs. The left column illustrates the results of the entire period covered by the training data, while the right column shows the zoom-in plots of the first day (December 1, 2014).

5. Outlook

This report details the development of methods for generating a wide range of predictions: local and network-wide, traffic flows and travel demand, for private vehicular traffic and public transport systems. In addition, the construction of the underlying networks for the three test sites involved in the SETA project is detailed.

The proposed vehicle traffic and demand predictions methods are applied to the three test sites as part of an on-going work. Results from one of the cases, Santander, were presented as part of the network-wide traffic prediction methodology section for demonstration sake. Results from all three sites and an evaluation of their performance based on a performance assessment, validation study and sensitivity analysis will be provided in D4.3 “Initial evaluation of predictors for smart mobility”. The public transport traffic predictions were implemented and tested for selected networks and will be applied to SETA test sites in Phase 2 of the project when suitable data is expected to become available.

The methods developed insofar and reported in this deliverable will be extended and refined during the course of the second phase of the SETA project. The prediction methods will be extended to cope with big data streams. A learning mechanism for updating the prediction techniques based on data streams will be developed to account for recurrent patterns. The data used for generating and validating predictions will be enriched with data collected using new social and physical passive, participatory and opportunistic sensing available from WP2 and fused to generate state estimations in WP3. The predictions generated in this WP4 will feed into the large-scale visual analytics and decision making developed in WP5.

References

Antoniou, C. et al., 2015. W-SPSA in practice: Approximation of weight matrices and calibration of traffic simulation models. *Transportation Research Part C: Emerging Technologies*, 59, pp.129–146.

Ashok, K. & Ben-Akiva, M.E., 2002. Estimation and Prediction of Time-Dependent Origin-Destination Flows with a Stochastic Mapping to Path Flows and Link Flows. *TRANSPORTATION SCIENCE*, 36(2), pp.184–198.

Balakrishna, R., Ben-Akiva, M. & Koutsopoulos, H., Off-line Calibration of Dynamic Traffic Assignment: Simultaneous Demand and Supply Estimation. *Transportation Research Record*.

Balakrishna, R. & Koutsopoulos, H., 2008. Incorporating Within-Day Transitions in the Simultaneous Off-line Estimation of Dynamic Origin-Destination Flows without Assignment Matrices. *Transportation Research Record*.

Bhattacharyya, B., 1987. One sided Chebyshev inequality when the first four moments are known. *Communications in Statistics-Theory and Methods*, 16(9), pp.2789–2791.

Bifet, A. & Gavaldà, R., 2009. Adaptive Learning from Evolving Data Streams. In *Proceedings of the 8th International Symposium on Intelligent Data Analysis: Advances in Intelligent Data Analysis VIII*. IDA ’09. Berlin, Heidelberg: Springer-Verlag, pp. 249–260.

Bifet, A. & Gavaldà, R., 2007. Learning from Time-Changing Data with Adaptive Windowing. In *Proceedings of the 2007 SIAM International Conference on Data Mining*.

Proceedings. Society for Industrial and Applied Mathematics, pp. 443–448.

Cantelmo, G. et al., 2014. An Adaptive Bi-Level Gradient Procedure for the Estimation of Dynamic Traffic Demand. *Intelligent Transportation Systems, IEEE Transactions on*, 15(3), pp.1348–1361.

Cipriani, E. et al., 2011. A gradient approximation approach for adjusting temporal origin–destination matrices. *Transportation Research Part C: Emerging Technologies*, 19(2), pp.270–282.

Cipriani, E., Gemma, A. & Nigro, M., 2013. A bi-level gradient approximation method for dynamic traffic demand estimation: Sensitivity analysis and adaptive approach. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. IEEE, pp. 2100–2105.

Cover, T.M., Thomas, J.A., 1991. Information theory and statistics. Elements of information theory 279335.

Cukier, R.I., Levine, H.B. & Shuler, K.E., 1978. Nonlinear sensitivity analysis of multiparameter model systems. *Journal of Computational Physics*, 26(1), pp.1–42.

Djukic, T. et al., 2012. Efficient real time OD matrix estimation based on Principal Component Analysis. In *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE*. pp. 115–121.

Duarte, J. & Gama, J., 2015. Multi-target regression from high-speed data streams with adaptive model rules. In *Data Science and Advanced Analytics (DSAA), 2015. 36678 2015. IEEE International Conference on*. IEEE, pp. 1–10.

Filkov, Vladimir, and Steven Skiena. "Integrating microarray data by consensus clustering." *International Journal on Artificial Intelligence Tools*13.04 (2004): 863-880.

Frederix, R., Viti, F. & Tampére, C.M.J., 2013. Dynamic origin-destination estimation in congested networks: theoretical findings and implications in practice. *Transportmetrica A: Transport Science*, 9(6), pp.494–513.

Friedman, J., Hastie, T. & Tibshirani, R., 2010. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1), p.1.

Gama, J. et al., 2014. A Survey on Concept Drift Adaptation. *ACM Comput. Surv.*, 46(4), p.44:1–44:37.

Gama, J., 2010. *Knowledge discovery from data streams*, CRC Press.

Hastie, T., Tibshirani, R. & Wainwright, M., 2015. *Statistical Learning with Sparsity: The Lasso and Generalizations*, Chapman and Hall/CRC.

Hoeffding, W., 1963. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301), pp.13–30.

Langford, J., Li, L. & Zhang, T., 2009. Sparse online learning via truncated gradient. *Journal of Machine Learning Research*, 10(Mar), pp.777–801.

Van Lint, J., Van Hinsbergen, C., 2012. Short-term traffic and travel time prediction models. Artificial Intelligence Applications to Critical Transportation Issues 22, 22–41.

Lopez, C., Krishnakumari, P., Leclercq, L., Chiabaut, N., Van Lint, H., 2017. Spatio-Temporal Partitioning of Transportation Network Using Travel Time Data. *Proceedings of Transportation Research Board - 96th Annual Meeting*, Washington D.C.

Luo, D., Cats, O., Lint, H. van, 2017. Constructing Transit Origin-Destination Matrices Using Spatial Clustering. Transportation Research Record: Journal of the Transportation Research Board In press.

Lundgren, J.T. & Peterson, A., 2008. A heuristic for the bilevel origin?destination matrix estimation problem. *Transportation Research Part B: Methodological*, 42(4), pp.339–354.

Ma, Y. & Zheng, T., 2016. Stabilized Sparse Online Learning for Sparse Data. *arXiv preprint arXiv:1604.06498*.

MacKay, D.J., 2003. *Information theory, inference and learning algorithms*, Cambridge university press.

Martinetz, T., Schulten, K., others, 1991. A“ neural-gas” network learns topologies.

Nesterov, Y., 2012. Efficiency of Coordinate Descent Methods on Huge-Scale Optimization Problems. *SIAM Journal on Optimization*, 22(2), pp.341–362.

Page, E., 1954. Continuous inspection schemes. *Biometrika*, 41(1/2), pp.100–115.

Shafiei, S. et al., 2017. A Sensitivity-Based Linear Approximation Method to Estimate Time-Dependent Origin-Destination Demand in Congested Networks. *Proceedings of Transportation Research Board - 96th Annual Meeting*, (Washington D.C.), pp.1–16.

Shalev-Shwartz, S. & Tewari, A., 2011. Stochastic methods for l1-regularized loss minimization. *Journal of Machine Learning Research*, 12(Jun), pp.1865–1892.

Shi, Jianbo, and Jitendra Malik. "Normalized cuts and image segmentation." IEEE Transactions on pattern analysis and machine intelligence 22.8 (2000): 888-905.

Tarantola, S., Gatelli, D. & Mara, T.A., 2006. Random balance designs for the estimation of first order global sensitivity indices. *Reliability Engineering & System Safety*, 91(6), pp.717–727.

Tavana, H. & Mahmassani, H.S., 2001. Estimation of Dynamic Origin-Destination Flows from Sensor Data using Bi-level Optimization Method. *Presented at the 80th annual meeting of the Transportation Reasearch Board, Washington DC, USA*.

Tibshirani, R., 1996. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1), pp.267–288.

Toledo, T. & Kolechkina, T., 2013. Estimation of Dynamic Origin-Destination Matrices Using Linear Assignment Matrix Approximations. *IEEE Transactions on Intelligent Transportation Systems*, 14(2), pp.618–626.

TSS-Transport Simulation Systems, 2015. Aimsun Dynamic Simulators Users Manual v8.

Wenjun, Y.X.G.D., 2002. A New Approach to Target Recognition Based on Image NMI Feature. *Computer Engineering* 6, 062.

Yang, H. & Zhou, J., 1998. Optimal traffic counting locations for origin-destination matrix estimation. *Transportation Research Part B: Methodological*, 32(2), pp.109–126.