UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

PhD program in Computing

An adaptive, fault-tolerant system for road network traffic prediction using machine learning

Doctoral thesis by:

Rafael Mena-Yedra

Thesis advisors:

Ricard Gavaldà Mestre

Jordi Casas Vilaró

Department of Computer Science

Barcelona, January 2020

Contents

Abstract

This thesis has addressed the design and development of an integrated system for real-time traffic forecasting based on machine learning methods. Although traffic prediction has been the driving motivation for the thesis development, a great part of the proposed ideas and scientific contributions in this thesis are generic enough to be applied in any other problem where, ideally, their definition is that of the flow of information in a graph-like structure. Such application is of special interest in environments susceptible to changes in the underlying data generation process. Moreover, the modular architecture of the proposed solution facilitates the adoption of small changes to the components that allow it to be adapted to a broader range of problems. On the other hand, certain specific parts of this thesis are strongly tied to the traffic flow theory.

This thesis has been developed within the context of Aimsun Live —a simulation-based traffic forecasting solution, developed and marketed by Aimsun—, being the main aim to improve Aimsun Live products as well as to cooperate in a mutually beneficial relationship between simulation-based and data-driven forecasting solutions.

The focus in this thesis is on a macroscopic perspective of the traffic flow where the individual road traffic flows are correlated to the underlying traffic demand. These short-term forecasts include the road network characterization in terms of the corresponding traffic measurements —traffic flow, density and/or speed—, the traffic state —whether a road is congested or not, and its severity—, and anomalous road conditions —incidents or other non-recurrent events—. The main traffic data used in this thesis is aggregated data coming from inductive-loop detectors installed along the road networks. Nevertheless, other kinds of traffic data sources could be equally suitable with the appropriate data preprocessing.

The simulation-based platform —where multiple traffic models are implemented— is Aimsun Next and the corresponding real-time version for traffic management is Aimsun Live. Furthermore, the proposed data-driven forecasting system is planned to be linked to the simulation-based traffic model in a mutually beneficial relationship where they cooperate and assist each other. An example is when an incident or non-recurrent event is detected with the proposed methods in this thesis,

then the simulation-based forecasting module can simulate different strategies to measure their impact.

Part of this thesis has been also developed in the context of the EU research project "SETA" (H2020-ICT-2015) whose aim is the creation of a ubiquitous data and service ecosystem for a better metropolitan mobility and the analysis of how short-term prediction can be improved through the use of multiple, highly diverse sources.

The main motivation that has guided the development of this thesis is enhancing those weak points and limitations previously identified in Aimsun Live, and whose research found in literature has not been especially extensive. These include:

1. Autonomy, both in the preparation and real-time stages.
2. Adaptation, to gradual or abrupt changes in traffic demand or supply.
3. Informativeness, about anomalous road conditions.
4. Forecasting accuracy improved with respect to previous methodology at Aimsun and a typical forecasting baseline.
5. Robustness, to deal with faulty or missing data in real-time.
6. Interpretability, adopting modelling choices towards a more transparent reasoning and understanding of the underlying data-driven decisions.
7. Scalable, using a modular architecture with emphasis on a parallelizable exploitation of large amounts of data.

The result of this thesis is an integrated system —*Adarules*— for real-time forecasting which has the ability to make the best of the available historical data, while at the same time it also leverages the theoretical unbounded size of data in a continuously streaming scenario. This is achieved through the online learning and change detection features of the system along with the automatic finding and maintenance of patterns in the network graph. In addition to the Adarules system, another result is a probabilistic model that characterizes a set of interpretable latent variables related to the traffic state based on the traffic data provided by the sensors along with optional prior knowledge provided by the traffic expert following a Bayesian approach. On top of this traffic state model, it is built the probabilistic spatiotemporal model that learns the dynamics of the transition of traffic states in the network, and whose objectives include the automatic incident detection.

The obtained results conclude:

- Better accuracy than current solution at Aimsun Live.

- Better adaption to changes.
- Efficient and robust to missing values.
- Anomaly detection.
- Informativeness.
- Scalable.

Acknowledgements

Undertaking a doctorate is an experience that shapes the doctoral candidate's skills by developing both work and planning skills as well as personal skills. The completion of a doctorate usually requires a great investment of time, patience and motivation. However, this has not been the only thing necessary and, in these lines, I would like to express my gratitude to all those people who in one way or another have contributed to making it possible for me to achieve it.

First of all, I would like to thank the support and guidance of my thesis directors, Ricard Gavaldà and Jordi Casas, who have made this thesis possible. Ricard's experience and advice helped me to move forward in those moments when I needed it most. Likewise, Jordi's knowledge has been a fundamental contribution to the thesis, as well as the encouragement.

I want to thank Aimsun as an organization for having trusted me to develop the thesis in their environment during these years, as well as for having provided me with the data used in this thesis and the tools for the good development of this thesis. In the same way, I would like to thank the funding agencies that have promoted the Industrial Doctorate program: both the Generalitat de Catalunya and the Agència de Gestió d'Ajuts Universitaris i de Recerca. As well as the Universitat Politècnica de Catalunya for providing material during the course of the doctorate.

This thesis was partially supported and/or developed within the framework of the following projects: Pla de Doctorats Industrials (DI-2014) of AGAUR, the EU research project SETA (H2020-ICT-2015), the AGAUR Projects 2014 SGR-890 (MACDA) and 2017 SGR-86 (MACDA) and by MINECO Projects TIN2014-57226-P (APCOM) and TIN2017-89244-R (MACDA).

During my doctorate I spent three months in the DiTTlab at the Technical University of Delft under the supervision of Professor Hans van Lint. The experience in that country and its people was really good. And I would like to thank Hans for the good reception, as well as my colleagues Panchamy, Ding, Tin, Simeon, and all the rest with whom I shared time.

To all the colleagues in Aimsun with whom I have shared a wonderful time, and learning as well. Especially with all those I have been collaborating with, and whose work has also enriched this thesis.

Personally, I would also like to thank the support that friends have given me during the course of my doctorate. Not only for their encouragement but also because I have learned from them. I would like to mention Ariel for having met him during the course of his doctorate at the UPC also as a doctoral student, and whose support has been of great value.

I am grateful for the unconditional support of my mother and family, which has been crucial. I would also like to thank, apart from the course of this doctorate itself, all those people who have contributed to my progress along this path by learning and enjoying the research up to this day. I want to remember colleagues from the TEP-197 research group at the University of Almeria like María del Mar, Domingo, Ricardo… with whom I started learning in research. And of course other members of the group like Francisco, José Luis, Manolo… who trusted me and motivated my desire to continue. I also show my gratitude for those teachers who exceptionally perform their work making the motivation greater: I remember especially Antonio, José Antonio, but there are many more.

Finally, I would like to express my gratitude and highlight the great value and importance of public education in our society. It is a common good to protect and to be proud of both for its quality and for the objective of making knowledge and equal opportunities more accessible and universal.

Emprender un doctorado es una experiencia que moldea las aptitudes del doctorando desarrollando sus habilidades tanto de trabajo y planificación como personales. La culminación del doctorado requiere una gran inversión de tiempo, paciencia y motivación. Sin embargo, no ha sido lo único y, en estas líneas, quiero expresar mi gratitud hacia todas aquellas personas que de una manera u otra han contribuido a que haya podido alcanzarlo.

En primer lugar quiero agradecer el apoyo y guía de mis directores de tesis, Ricard Gavaldà y Jordi Casas, que han hecho que esta tesis sea realizable. La experiencia de Ricard y su consejo me han servido para seguir hacia delante en aquellos momentos que más lo necesitaba. Asimismo, el conocimiento de Jordi ha sido una contribución fundamental para la tesis, así como su apoyo.

Quiero agradecer a Aimsun como organización por haber confiado en mí para desarrollar la tesis en su entorno durante estos años, así como haberme proveído de los datos utilizados en esta tesis y las herramientas para el buen desarrollo de esta tesis. De la misma manera quiero mostrar mi agradecimiento a las agencias de financiación que han promovido la iniciativa del Doctorat Industrial: tanto la Generalitat de Catalunya como la Agència de Gestió d'Ajuts Universitaris i de Recerca. Así como la Universitat Politècnica de Catalunya por proporcionar material durante el transcurso del doctorado.

Esta tesis fue parcialmente desarrollada en el marco de los siguientes proyectos: Pla de Doctorats Industrials (DI-2014) de la AGAUR, el proyecto de investigación de la UE SETA (H2020-ICT-2015), los proyectos de la AGAUR 2014 SGR-890 (MACDA) y 2017 SGR-86 (MACDA) y los proyectos MINECO TIN2014-57226-P (APCOM) y TIN2017-89244-R (MACDA).

Durante el doctorado realicé una estancia de tres meses en el laboratorio DiTTlab de la Universidad Técnica de Delft bajo la supervisión del profesor Hans van Lint. La experiencia en aquel país y su gente fue realmente buena. Y quiero agradecer la buena acogida de Hans, así como los compañeros Panchamy, Ding, Tin, Simeon, y todo el resto con el que compartí tiempo.

A todos los compañeros de Aimsun con los que he compartido un magnífico tiempo, y aprendiendo también. En especial con todos aquellos que he colaborado y que su labor también ha enriquecido esta tesis.

Personalmente también quiero agradecer el apoyo que me han brindado amigos durante el transcurso del doctorado. No solamente por haber estado ahí, también por haber aprendido de ellos. Quiero hacer mención de Ariel por haberle conocido durante el transcurso del doctorado en la UPC también como doctorando y haber sido un gran apoyo.

Agradezco el apoyo incondicional a mi madre y familia, el cual ha sido fundamental.

También quisiera agradecer, al margen del transcurso de este doctorado, a todas aquellas personas que han contribuido a que hoy esté aquí siguiendo aprendiendo y disfrutando de la investigación. Me quiero acordar de compañeros del grupo de investigación TEP-197 de la Universidad de Almería como María del Mar, Domingo, Ricardo… con los que comencé aprendiendo en investigación. Y por supuesto de otros miembros del grupo como Francisco, José Luis, Manolo… que confiaron en mí y motivaron mi inquietud por continuar. Asimismo, muestro mi gratitud por aquellos docentes que ejercen excepcionalmente su labor haciendo que la motivación fuese mayor: me acuerdo especialmente de Antonio, José Antonio, pero son muchos más.

Por último, quiero agradecer así como resaltar el gran valor e importancia de la educación pública en nuestra sociedad. Es un bien común a proteger y del que sentir orgullo tanto por su calidad como por el cometido de hacer más accesible y universal el conocimiento y la igualdad de oportunidades.

List of Figures

List of Tables

List of Abbreviations

Acronym	Term
AIC	Akaike information criterion
BIC	Bayesian information criterion
CD	Coordinate descent
CDF	Empirical cumulative distribution function
DAG	Directed acylic graph
DPP	Dual polytope projection
DTA	Dynamic traffic assignment
DUE	Deterministic user equilibrium
ERM	Empirical risk minimization
EU	European Union
GHG	Greenhouse gas
i.i.d.	Independent and identically distributed
IQR	Interquartile range
ITS	Intelligent transport system
K-S	Kolmogorov-Smirnov test
KKT	Karush-Kuhn-Tucker
lasso	Least absolute shrinkage and selection operator
MAP	Maximum a posteriori
ML	Machine learning
MSE	Mean squared error
$MtCO_2e$	Million tonnes of CO_2 equivalent
MTL	Multi-task learning
OLS	Ordinary least squares
PGM	Probabilistic graphical model
RMSE	Root mean squared error
STA	Static traffic assignment

Acronym	Term
STL	Single-task learning
SUE	Stochastic user equilibrium
SVD	Singular Value Decomposition

1 Introduction

1.1 The need for more efficient traffic management

In modern society, human mobility is characterized by an apparently never-ending growth. This has motivated a broad study in the field including aspects of its measurement and modelling [26]. Understanding the process by placing the correct modelling assumptions is a crucial step to unveil the underlying patterns that can explain the process [108]. Some empirical studies, e.g. [45, 210], have found reasonable results in validating human mobility patterns. This finding of patterns is fundamental to know to which extent such process is predictable [228] and, actually, it does follow some recognizable patterns [217] that can be used to approach the problem.

This continuous and rapid growth of the transportation systems to meet the users' mobility demand has been especially prominent in regards to the road transportation system with the private car as the main protagonist, which has lead to severe problems such as traffic congestion and environmental pollution in the large urban areas. However, despite the fact that the international community is gradually getting sensitized about the serious environmental problem, there is still a long way to go. There is scientific evidence about the human activities impact into the climate due to greenhouse gases (GHG) emission and the possibility to reach a climate tipping point [132].

In the last report [83] from the European Environment Agency (EEA) it is analyzed how GHG emissions in the EU were cut down by 22.4% between 1990 and 2016 in compliance with the international treaties of 1992 United Nations Framework Convention on Climate Change (UNFCCC) and the 1997 Kyoto Protocol, and which is on track to achieve the EU reduction commitment of 20% GHG by 2020 compared with 1990. In spite of it, looking deeper into the report it can be seen that GHG emissions from road transport —which was responsible for almost 73% of total GHG emissions from transport including aviation and international shipping during 2015— constitute the second largest key source category in terms of CO_2 emissions share during 2016 in the EU as can be seen in Figure 1.1. Even more, it can be also observed how all the large key source categories have decreased the CO_2 emissions from 1990 to 2016 —in terms of absolute change of CO_2 emissions exclusively—, with the exception of the road transportation sector which has raised the

CO_2 emissions more than 23%. In this sense, these emissions coming from the road transportation sector will need to fall by 68% by 2050 in order to meet the 60% GHG emission reduction target of the 2011 Transport White Paper [82].

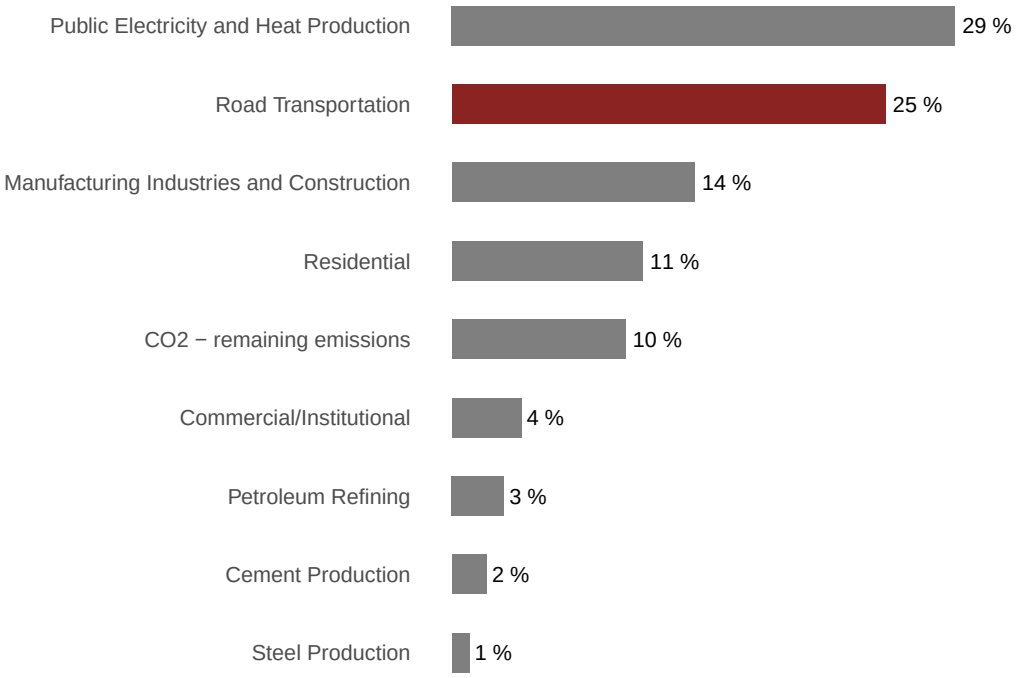
Another major problem derived from the rapid and constant growth in transport demand is traffic congestion, which not only has consequences at the individual level by increasing travel times as the infrastructure efficiency is reduced, but it also rises fuel and energy consumption. In the end, it further worsens the situation with a significant increase in the environmental pollution and its impact on public health and quality of life [47, 158]. Furthermore, in 2018, an estimated 55% of the total world's population (7.46 billion) is residing in urban areas, whereas the United Nations estimation for 2050 is that 68% of the estimated total world's population (9 billion) is projected to be urban [249]. How this affects the mentioned environmental problems is subject of ongoing research [88]. Thus, the aforementioned problems can only be exacerbated in the long-term if no appropriate measures are taken in order to reach a sustainable transportation where the role of the private car is still somewhat uncertain [101, 193].

Among such measures, a distinction could be made between behavioral and technological changes. In general, behavioural changes are aimed to reduce the transport demand or promoting a shift to less polluting modes of transport, while technological solutions are aimed at reducing the negative impact of passenger-kilometer (pkm) which include a shift away from transport based on fossil fuels, and the implementation of smart traffic management strategies. The final goal of these strategies is to improve environmental quality, urban quality of life and destination accessibility [231].

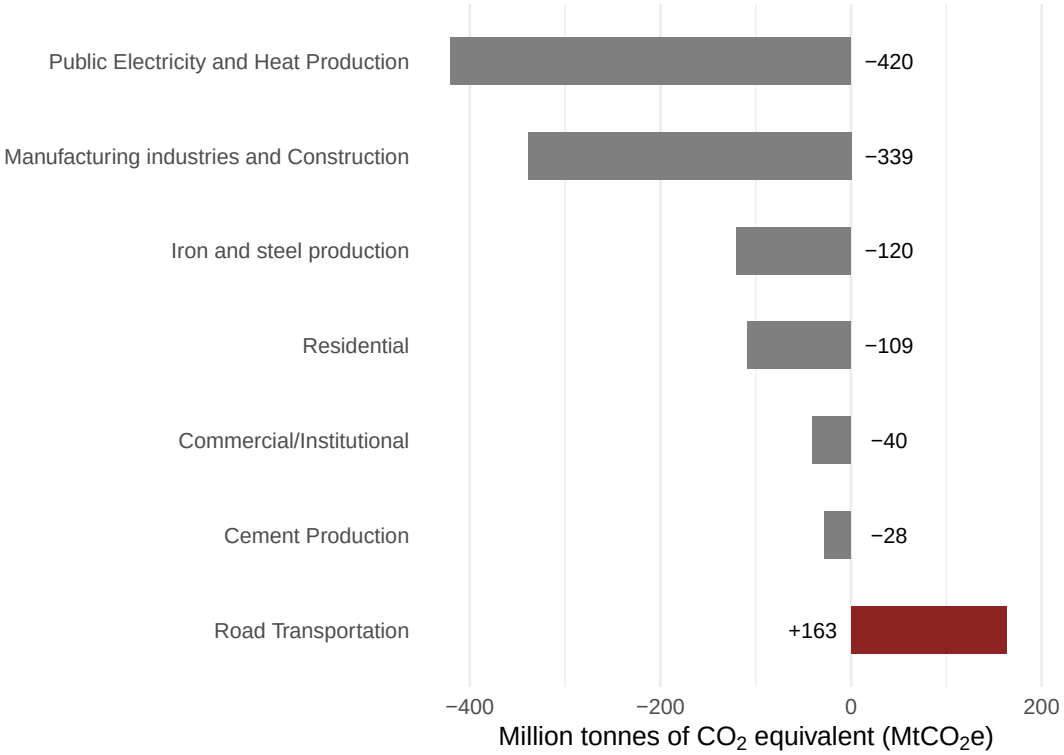
The required traffic management to address this challenge must be responsive and adaptive and cannot be limited only to classical measures such as a continuous increase of the infrastructure supply. Instead, technology and solutions based on research, development and innovation must play an important role. Certainly, the EU in its Transport White Paper [82] highlights action points with the aim to reach a reduction of at least 60% of GHG emissions in the transport sector by 2050 with respect to 1990 and summarizes technology-based solutions as follows:

> Technological innovation can achieve a faster and cheaper transition to a more efficient and sustainable European transport system by acting on three main factors: vehicles' efficiency through new engines, materials and design; cleaner energy use through new fuels and propulsion systems; better use of network and safer and more secure operations through information and communication systems.

It is therefore clear that utilization of information and data has a leading role. For instance, recently the European Parliament and the Council of the European Union have issued a directive



(a) Share of key source categories and all remaining categories in 2016.



(b) Absolute change of CO_2 emissions by large key source categories from 1990 to 2016.

Figure 1.1: CO_2 emissions for EU-28 and Iceland. Data source: [83].

[240] urging member states to establish a new legal framework for the deployment of Intelligent Transport Systems (ITS) in the field of road transport and for interfaces with other modes of transport. In this framework, the concept of ITS is described as:

> Advanced applications which without embodying intelligence as such aim to provide innovative services relating to different modes of transport and traffic management and enable various users to be better informed and make safer, more and 'smarter' use of transport networks. [They] integrate telecommunications, electronics and information technologies with transport engineering in order to plan, design, operate, maintain and manage transport systems.

In this way, traffic management can be seen as a combination of measures implemented with the aim of preserving traffic capacity and improve the security, safety and reliability of the overall road transport system. Therefore, ITS play a very important role in the reduction of road fatalities [265]. In addition, ITS also contribute to environmental and climate change objectives. Their main applications, as stated in [59], are:

- Transport demand / Mode choice
  - (multi-modal) travel information, journey planning
  - car-sharing, ride-sharing
  - road charging, integrated ticketing
  - access management
  - logistics, fleet management (avoidance of empty runs)
- Efficiency of traffic
  - traffic management
  - travel information, navigation
  - public transport priority
- Driver behaviour
  - eco-driving support, navigation

A specific example can be found in the EU EasyWay project [79], which pursues to contribute to the objectives of the European Commission's ITS Action Plan and the aforementioned ITS Directive in the way to achieve a sustainable road transport system. In this regard, it identifies a set of necessary ITS European services to deploy: Traveller Information, Traffic Management and Freight and Logistic Services.

In particular, Traveller Information Services is a key element of the ITS ecosystem as they aim to provide with comprehensive real-time and predictive traffic information, allowing thus to perform suitable travel decisions before the departure time (pre-trip information) and during the journey (on-trip information). It has been demonstrated that the use of this kind of information on drivers' decisions has a beneficial impact on network performance, as can be seen in some studies with real data. For instance, in [137] it is concluded for a case study in Amsterdam that if drivers are provided with descriptive pre-trip information and this information is targeted such that the drivers are given opportunities to change their departure time, routes and/or their entire trips then this kind of behavior would result in improved network performance in terms of throughput, congestion length, and average network speeds. Similar conclusions are found in other studies [3, 34, 221], with the additional finding that there is a positive correlation between the tendency to choose riskier routes —i.e. those characterised by a lower average but greater variance of travel time— and the experience of the road user with such routes; which means that travelers in the study were found to be more reluctant to be influenced by pre-trip information if they had more experience with the routes. On-trip information and traffic forecasts are also advantageous for the road user, for instance, intelligent devices are able to calculate more efficient routes and reduce the travel time. Furthermore, it could be a very valuable information for emerging V2X-based traffic control systems [134], which could also serve for the route planning of self-driving cars.

This generated information is not only useful for the driver agent, but also for the traffic control agent in the framework of an Advanced Traffic Management System (ATMS) within an ITS. Traffic Management is defined as an overall plan of strategies and tactical actions for accommodating traffic flow in an efficient, effective and safe manner during recurrent and non-recurrent events on the transportation network, and a subset of them could comprehend:

- Lane / line control,
- Speed control,
- Ramp metering,
- Hard shoulder running,
- Incident warning and management.

Future (road) transport systems and mobility patterns are probably subject to substantial changes. These changes will also depend on the advent and eventual integration of connected (V2X) and autonomous vehicles (CAVs), and if car-pooling will be widely adopted [229]. There are already studies that analyze what such impact will be [21, 238]. The future is uncertain, but we are increasingly able to measure more things, as well as the data availability is vastly increasing. This will probably lead to data-driven and evidence-based procedures to take a more significant role into

the field, and more especially favoring those with the ability to detect, react and being adaptable to changes and new scenarios.

1.2 Real-time traffic forecasting

As previously stated, Advanced Traveler Information System (ATIS) and Advanced Traffic Management System (ATMS) are key components within the ITS ecosystem. In order to serve properly to this end, accurate and real-time traffic forecasts are required. These short-term forecasts include the road network characterization in terms of the corresponding traffic measurements —traffic flow, density and/or speed—, the traffic state —whether a road is congested or not—, and anomalous road conditions —incidents, unexpected roadworks, and other non-recurrent events—.

This forecast information has a great value by itself to travelers and traffic managers, but its value can be further enhanced if the information —network state predictions, incidents…— are employed to feed a simulation-based traffic model with the aim of reproducing multiple traffic management strategies and assessing quantitatively their ability to mitigate congestion before their implementation in the field, all this in the realm of a complete decision support system [244].

Simulation-based traffic models are explicitly programmed to mimic the causality effects over the entire traffic network. Although they need comprehensive knowledge about the traffic network and tune a wide amount of parameters, and spite of their high computational cost, they are extremely useful tools for the traffic management as they allow to test "what-if" scenarios and produce measures of effectiveness associated with different traffic control strategies.

The issue with simulation-based traffic models for short-term traffic forecasting is that they demand large amounts of computational resources which can be challenging for real-time operation when either the traffic network is large-scale or the forecasting horizon is large —e.g. up to one hour—. Moreover, short-term traffic forecasting is a complex process where traffic demand interplays with the available traffic supply and it is subject to different kind of changes and influenced by external factors. For such cases simulation-based models may not perform as well as expected, and it is the context where a machine learning (ML) approach may perfom a good job in a effective and efficient manner. However, the strength of simulation-based forecasting resides in their ability to provide a full picture of the traffic network state with a high level of detail when all required information is known and modelling assumptions are reasonably met —e.g. behavioural parameters—. This makes simulation-based solutions potentially more suitable to handle non-recurrent traffic conditions when similar historical data is hardly available for a ML solution, and also having the possibility to simulate different response plans to measure their impact on the network.

1.3 Thesis objectives

This thesis addresses real-time short-term traffic forecasting with the support of machine learning methods, and focuses on an aggregated macroscopic view where the road traffic flows are correlated to the underlying traffic demand. These short-term forecasts include the road network characterization in terms of the corresponding traffic measurements —traffic flow, density and/or speed—, the traffic state —whether a road is congested or not—, and anomalous road conditions —incidents or other non-recurrent events—. The main traffic data used in this thesis is aggregated data coming from inductive-loop detectors installed along the road networks. In spite of their pitfalls, the main reason is that they have become the most widely used sensor in traffic management systems since their introduction in the early 1960s and such the large source of available data. Nevertheless, other kinds of traffic data sources could be equally suitable with the appropriate data preprocessing.

Furthermore, this approach will be linked to simulation-based traffic models in a mutually beneficial relationship where they cooperate and improve each other. The simulation-based integrated platform with multi-tier traffic models —macroscopic, mesoscopic and microscopic— is Aimsun Next and the corresponding real-time version for traffic management is Aimsun Live, which are developed by Aimsun, a Siemens company [5]. It must be noted that even though contributions to improve Aimsun products have been a driving motivation, we propose ideas and scientific contributions that are generic enough to be applicable to other complex monitoring and control environments, not necessarily traffic-related. Part of this thesis has been also developed in the context of the EU research project "SETA" (H2020-ICT-2015) [61] which is creating a ubiquitous data and service ecosystem for a better metropolitan mobility and the analysis of how short-term prediction can be improved through the use of multiple, highly diverse sources.

In order to achieve the goal of a self-adaptive, fault-tolerant system for road network traffic prediction using machine learning, we need to present context and current shortcomings.

1.3.1 The Aimsun context

The Aimsun transport modelling software was originally the focus of a multi-year research project at the Technical University of Catalonia —*Universitat Politècnica de Catalunya* (UPC)—, and currently it is in its 8th commercial major version. Aimsun has grown from the stated aim of the original acronym "**A**dvanced **I**nteractive **M**icroscopic **S**imulator for **U**rban and **N**on-urban **N**etworks" [87] to a fully integrated application that fuses different traffic models with multiple levels of detail —travel demand modelling, macroscopic functionalities and the mesoscopic-microscopic hybrid

simulator—. They are currently known as Aimsun Next —the offline traffic modeling software— , and Aimsun Live —the decision support system for real-time traffic management based on the former—. The Aimsun company was recently acquired by Siemens with the aim of being integrated into a wide ITS ecosystem.

As stated in [53], the first versions of Aimsun relied only on simulation-based traffic models within Aimsun Live. These were considered to deal better with non-recurrent events because fluctuations in supply could be explicitly factored in and their impact under different scenarios could be quantified. These scenarios could be comprised of different actions —-e.g. a lane closure, rerouting with variable-message signs (VMS) or speed limit variation—- and they could be activated manually or automatically based on rules which constantly process detection data. On the other hand, recurrent or predictable incidents could be managed using according scenarios picked from a scenario catalogue and already implemented in the simulation model. In the end, different measures of effectiveness (safety, environmental, economic, operational or a combination of these) could be used to compare the response strategies and anticipating the consequences of those actions. These results ultimately allow the operator to quickly see, first, if any traffic control strategies improve the situation compared to the "do-nothing" case and if yes, which ones offer the best performance.

The current scene has moved towards a combination of traffic simulation along with machine learning methods. This combination of methods is called Network Prediction System (NPS) in the Aimsun context, and it provides traffic forecasts for the full network including traffic flows, speeds, delays and travel times among others. A schematic view of the functional structure is shown in Figure 1.2. In this way, the analytical prediction subsystem based on machine learning methods can enhance the simulation-based subsystem by providing accurate real-time forecasts —in a process called dynamic demand adjustment— which are then extended to the entire network. Incident detection and anomalous —i.e. non-recurrent— events identified by the ML-based subsystem are also extremely valuable for the simulation-based subsystem in order to properly reproduce the scenarios and the different traffic management strategies.

1.3.2 Identified limitations

The Aimsun Live architecture has been deployed successfully in multiple real projects as in San Diego [9], Lyon [10], Madrid [7], Gold Coast [6], Sydney [8] or Leicester [11]. Nevertheless, several challenges and problems have been identified during the deployment, and also during the lifetime and maintenance of these traffic management applications. Namely:

Issue 1. Lack of consideration of the underlying traffic dynamics: The procedure for

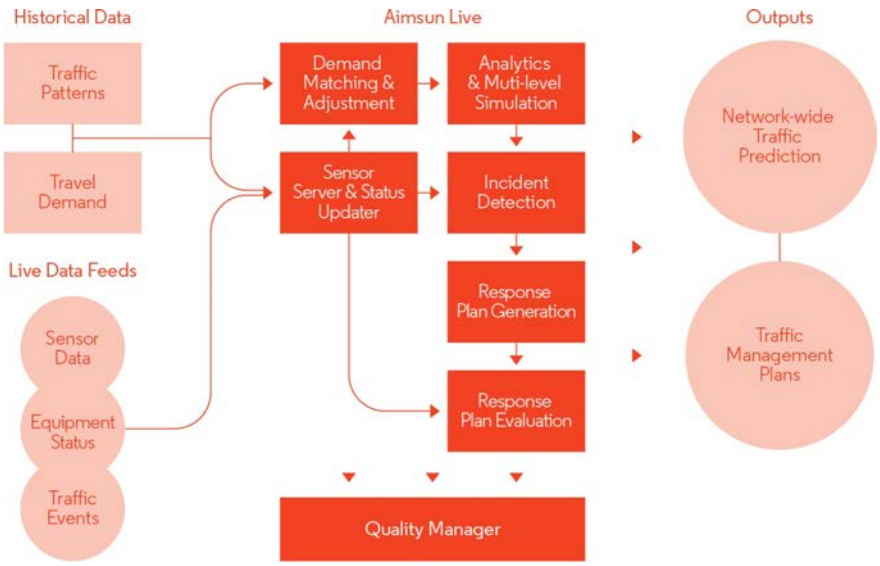


Figure 1.2: Functional structure of Aimsun Live Network Prediction System (NPS). Source: [5].

fitting the predictive ML models was too fixed and it did not fully consider the traffic dynamics of the transport network at hand. The design criteria was based on building a predictive ML model for every location with detection data, every forecasting horizon and every time of the day. For this reason, it could happen that a vast amount of ML models needed to be built depending on the number of detection locations, forecasting horizons —it could be assummed to be 15, 30, 45 and 60 minutes in direct forecasting—, and the number of time steps —if time is discretized in 5 minute slots, there would be 288 per day—. This could lead not only to computational inefficiency, but also to bad performance for the learning stage of ML models if not enough data was presented —note that, within this approach, one year of data contains only 365 observations per ML model (detection location, forecasting horizon, time slot)— thus leading to spurious patterns in such small dataset.

Issue 2. Lack of online learning and adaptation to change: Moreover, this set of predictive ML models was built offline in batch mode with historical data, and thus there was no more learning with new incoming data neglecting any opportunity to react and to be adapted to changes. These changes may have different forms and impact on the transport network. For instance, such changes can comprehend the advent of a new hot-spot in the network that pulls travel demand —e.g. a new commercial center—, or changes in the road network geometry that can also lead to new traffic detectors or the deletion of some of them, or even changes in the travellers' behavioural patterns. All of these changes —whose impact on the network is either with local or global scope— have effects on the underlying travel demand. For this reason, adaptation to change has been one of the main motivations that has driven this thesis.

Issue 3. Modelling assumptions strongly dependent on available historical data: At the

same time, other kinds of recurrent effects such as seasonality, special days or weather conditions among other contextual factors which may influence traffic conditions, can only be included in the predictive modelling if they are available at the modelling time. Although this may seem obvious, it is worth to note that sometimes long records of historical data are not available for all of the projects at their starting time or, even if so, sometimes these effects are only noticed after a time, when more data have been collected.

Issue 4. Limited treatment of non-recurrent effects: On the other hand, non-recurrent effects —e.g. traffic incidents— are transient and sudden events that depict a challenging situation for short-term traffic forecasting and traffic state identification. Forecasting under such circumstances is, by definition, extremely challenging especially for ML models with limited causal inference. However, the prompt detection and reporting of such events are crucial for a efficient real-time traffic management including their feed to simulation-based traffic models for selecting the best traffic control strategy. In the same way, detection and reporting of anomalous road conditions —e.g. whether the traffic is significantly below or beyond the expected— may require a further diagnosis by traffic managers. The incident and anomaly detection found is very limited and mostly aimed at freeway transport networks, neglecting urban networks.

Issue 5. Little robustness to missing and faulty data: The resilience and robustness of the method coping with missing data was very limited, with the consequent impact on the forecasting performance. This is especially relevant as malfunctioning traffic measurement devices are not uncommon, such leading to missing data during real-time operating.

Issue 6. Lack of automated solution: The solution was not entirely automated because some decisions were to be made by the modelling analysts at the beginning of the projects —e.g. what amount of data to use—, and also during the projects' lifetime —such as the maintenance timings to update the ML models with new data—. This made it hard in terms of scalability to face new projects.

Issue 7. Limited interpretability: Finally, one critical issue with the previously established approach is the interpretability. Traffic engineers and managers do not only wish high forecasting accuracy, but they also value the possibility to interpret the model and assess which factors have lead to such forecasting decision. Furthermore, there was scarce information visualization and reporting.

1.3.3 Goals

Being this thesis proposal mainly a data-driven approach, the number and quality of the available data sources along the transport network is totally crucial for good performance. Despite the fact that more and more data is available, it is known that some kinds of detector installations tend to be expensive to extend or modify because of the cost of installation and maintenance. For such reason, the approach presented aims at making the most of the available data, even when these are scarce.

Thus, the major contributions of this thesis are:

1. To provide the required autonomy and adaptation in order to achieve an automatic operational level in the real-time traffic forecasting task. Through the automation of the process the required human intervention in the process operation should be reduced and affordable to traffic engineers.
2. Additionally, the system must be 'intelligent' enough to evolve over time reducing the required human intervention in the maintenance stage to a minimum. ML-based predictive models need to be updated in real-time and adapted to detected changes in the data generation process.
3. At the same time, forecasting accuracy must be appropriate and useful for traffic management purposes.
4. The approach should be fault-tolerant with missing data such as malfunctioning data devices during real-time.
5. Robustness to handle properly unexpected mobility patterns that do not match with the expected pattern behavior.
6. Sudden and transient changes in the data distribution —which includes non-recurrent events such as incidents— must be detected, identified and at worst case at least handled in order to avoid a degradation in the short-term traffic forecasting performance.
7. Selection and development of statistical and machine learning models must be aimed to not sacrifice interpretability for traffic engineers.

Therefore, all identified shortcomings in the previous subsection will be satisfied by the achievement of these goals.

1.4 Summary of results

For the validation of the different methods proposed in this thesis, two datasets with different characteristics have been used: the M4 and M7 motorways in Sydney, Australia; and the urban network of Santander City, Spain. The main results of this thesis can be classified according to their scope and the performance indicator used:

- Forecasting
  - Accuracy
  - Complexity
  - Interpretability and informativeness
- Traffic states model
  - Interpretability
- Spatiotemporal model for traffic states dynamics
  - Interpretability and informativeness

1.4.1 Forecasting

The forecasting evaluation is performed in the context of the Adarules algorithm —described in Chapter 5— whose validation is shown in Chapter 6. This algorithm [181] is designed as a self-contained system for real-time forecasting especially suited for the streaming scenario as it contains modules for change adaption and online pattern mining. The algorithm motivation arises from the work by Gama [94].

Within the Adarules system, we propose two methods for the aim of pattern mining —i.e. the seek of graph patterns described as *rules*—. The two approaches differ in how Adarules performs the graph pattern mining depending on the spatial scope of such pattern mining: looking only specific individual points in the road network —single-task mining (STM)— or looking the entire network as a whole —multi-task mining (MTM)—. The conclusion is clear and identical for both networks' datasets (M4/M7 and Santander): the resulting complexity is much lower in the case of MTM. In terms of forecasting accuracy, the MTM approach is also superior on average as shown in Table 1.1 using the normalized root mean squared error (nRMSE) described in Section 6.1.

The second evaluation concerns the comparison of Adarules building the underlying forecasting models —i.e. the sparse model for the spatiotemporal correlations— using either single-task learning (STL) or the multi-task learning approach regarding the number of forecasting dimensions

Table 1.1: Summary of average forecasting accuracy related to the pattern mining procedure: single-task mining (STM) vs multi-task mining (MTM). KPI is the normalized RMSE.

	M4/M7		Santander	
Var-forecast	STM	MTM	STM	MTM
flow-15m	1.91	1.48	1.89	1.69
flow-60m	5.56	4.31	5.05	4.11
occ-15m	2.49	2.20	3.67	2.85
occ-60m	3.40	3.06	4.16	3.95
speed-15m	6.26	5.78		
speed-60m	7.11	6.63		

Table 1.2: Summary of average forecasting accuracy related to the forecasting models' learning procedure: single-task learning (STL) vs multi-task learning (MTL). KPI is the normalized RMSE.

	M4/M7		Santander	
Var-forecast	STL	MTL	STL	MTL
flow-15m	1.49	1.52	1.73	1.77
flow-60m	4.31	3.65	5.02	4.22
occ-15m	2.20	2.22	3.20	3.11
occ-60m	3.03	2.99	3.73	3.57
speed-15m	5.53	5.51		
speed-60m	6.95	6.52		

jointly learned. In this case, the forecasting accuracy for the very short-term —15 minutes— is practically the same for both approaches. However, for longer forecasting horizons —60 minutes— the difference is more noticeable since the MTL approach achieves a lower forecasting error on average.

The following evaluations have been aimed in comparing the forecasting accuracy and adaptation ability of Adarules —configured using MTM and MTL— against the previous methodology at Aimsun —*ANA*— and a seasonal historical average —*HA*— as baselines. Moreover, as *ANA* and *HA* are static approaches that do not consider the procedure of incremental learning, different yearly (Y), quarterly (Q) and monthly (M) update schedules have been evaluated. The evaluations have been performed using different change scenarios in both network datasets.

The first scenario was simply using the real data over the two-year period in both datasets. The resulting complexity in Adarules has been the identification of 25 rules in the case of the M4/M7 network after the two-year period, and 37 rules in the case of the Santander network. The rules' antecedents are shown and the interpretability of such rules has been verified with traffic engineers at Aimsun who have confirmed their agreement and the appropriateness of the rules given the prior

Table 1.3: Summary of average forecasting accuracy in real data experiment. Comparison between the thesis' proposal (Adarules) versus baselines (ANA is the current approach at Aimsun Live, and HA is a historical average or seasonal naïve forecast) with different model updating schedules (yearly, quarterly or monthly). KPI is the normalized RMSE.

Var-dim	Adarules	ANAy	ANAq	ANAm	HAy	HAq	HAm
M4/M7							
flow-15m	1.53	3.05	4.63	6.54	5.16	5.03	4.94
flow-60m	3.55	6.10	7.20	8.14	5.21	5.10	5.00
occ-15m	2.23	2.79	3.08	3.51	3.12	3.31	3.01
occ-60m	2.94	3.48	3.79	3.90	3.19	3.37	3.09
speed-15m	5.74	6.30	6.77	7.55	7.35	7.49	7.08
speed-60m	6.51	7.13	7.45	7.88	7.31	7.45	7.01
Santander							
flow-15m	1.71	2.25	2.89	4.17	5.10	5.41	4.93
flow-60m	4.02	5.36	5.91	6.37	5.05	5.37	4.88
occ-15m	3.01	3.72	3.88	4.35	3.76	3.91	3.65
occ-60m	3.44	4.15	4.30	4.71	3.82	3.99	3.71

knowledge —in the form of qualitative variables— used by Adarules in the experiments. Regarding the evaluation in terms of forecasting accuracy using the nRMSE as shown in the Table 1.3, it can be observed that, on average, Adarules is always superior as it achieves a lower forecasting error —both in the 15 and 60 minutes forecasting horizon— compared to the ANA forecast methodology as well as the HA baseline.

This pattern of results is repeated for the rest of experiments aimed to test the adaptive ability of Adarules in different change scenarios —from gradual to more extreme—. More specifically:

1. A scenario with explicit no-drift during the second year. This means that the first year on each dataset is the real one, but the subsequent year data is fake by using a specific month of the second year —e.g. May— and replicating it for twelve times. The goal of such experiment is to test the performance in a absolutely no-change scenario, i.e. how the forecasting accuracy and the model complexity evolves over time.
2. A first year with the real data, and then starting from the second year and every two months (January, March, May, July, September, November) a fake change is introduced over all the network by selecting 200 detectors at random where the traffic variables —flow, occupancy, and speed— from 100 of these detectors are incremented by a 4% while the others 100 detectors experience a 4% decrease in the traffic variable. This is maintained until, two months later, another round of smooth changes takes place while accumulating the one from the previous swapping. The goal is to determine the Adarules ability to react and adapt to these gradual changes.

3. A first year with the real data, and then starting from the second year and every two months a fake change is introduced over all the network by swapping the AM and PM periods. This means that every two months (January, March, May, July, September, November) the traffic is swapped and, thus, traffic during the night takes place during the day and vice versa. This is maintained for two months until the next swap takes place. The goal is to determine the Adarules ability to react and adapt to these abrupt changes.
4. A first year with the real data, and then starting from the second year and every two months a fake change is introduced over all the network by swapping 100 detectors identifiers selected at random. This means that every two months (January, March, May, July, September, November) the traffic from these detectors is swapped with others in the network. This is maintained until, two months later, another swapping takes place while accumulating the one from the previous swapping. The goal is to determine the Adarules ability to react and adapt to these extreme abrupt changes.

In all of these experiments, the performance of Adarules is clearly superior to the baselines compared. The superiority is not only on the average but also achieving a much less harmful 'worst case' as shown in the different figures and tables in the Chapter 6. These results have been verified by traffic engineers at Aimsun confirming their suitability for the purpose of real-time forecasting.

1.4.2 Traffic states model

The results related to the traffic states model are presented in Chapter 7. They are mainly *sanity checks* in order to test the validity of the model because in this case there is not forecasting involved. The results are shown for both networks —M4/M7 and Santander—, and they are consistent with traffic flow theory as verified by traffic engineers in Aimsun.

1.4.3 Spatiotemporal model for traffic state dynamics

The results concerning the spatiotemporal model for traffic states dynamics are presented in Chapter 8. On one hand, they show a use-case to quantify the existing outlierness in a network graph. This outlierness is measured both from the point of view of Adarules using its identified graph patterns —rules— and from the point of view of the transition of traffic states in the time-space. On the other hand, results from the incident detection are shown for some real cases in the M4/M7 network as well as from some synthetic data coming from simulations performed in the Bristol urban network. These results are shown to be informative and useful from the perspective of traffic engineers at Aimsun.

1.5 Thesis outline

Chapter 1 describes the underlying motivation for the development of this thesis in the context of pursuing a more intelligent and efficient traffic management. The main goals within the Aimsun context showing the identified existing limitations are also shown in this chapter, as well as a summary of the results obtained through the thesis.

Chapter 2 contains a comprehensive background describing the basics about traffic data, traffic modelling and data modelling, with a special emphasis on the methods used through this thesis, but giving an overall framework to the reader about traffic and data modelling.

Chapter 3 reviews the literature in the research fields of short-term traffic prediction, traffic state identification and incident detection, describing what is the current state of the art.

Chapter 4 shows a first overall representation of the proposed solution for short-term traffic forecasting. The overview describes the different components in the integrated framework and their aim.

Chapter 5 presents the Adarules framework for real-time forecasting coupled with the process of graph pattern mining. The chapter includes the algorithms' pseudocodes for every module within Adarules.

Chapter 6 shows an exhaustive set of experiments to validate the performance of Adarules in terms of forecasting accuracy, adaptation to change and modelling complexity.

Chapter 7 presents the probabilistic graphical model to identify the underlying latent variable within the fundamental diagram of traffic flow. Some sanity checks are included to show the expected output.

Chapter 8 presents and describes the spatiotemporal probabilistic model for learning the traffic states dynamics including the showcase of several incidents detection in the M4/M7 network and a synthethic dataset from simulations using the Bristol urban network.

Finally, in Chapter 10 conclusions are presented based of the thesis goals and achievements, and some future work is proposed in order to continue the work.

1.6 Publications

Parts of the developed work in this thesis have been presented in the following publications:

- Rafael Mena-Yedra, Ricard Gavaldà, and Jordi Casas. «Adarules: Learning rules for realtime road-traffic prediction». In: Transportation Research Procedia 27 (2017), pp. 11–18. ISSN: 2352-1465. DOI: https://doi.org/10.1016/j.trpro.2017.12.106 URL: http://www.sciencedirect.com/science/article/pii/S2352146517310037

  – Preliminary version in: Rafael Mena-Yedra, Ricard Gavaldà, and Jordi Casas. «Adarules: Learning rules for real-time road-traffic prediction». In: Proceedings of the 20th EURO Working Group on Transportation Meeting (EWGT 2017). Budapest, Hungary, September 2017.

- Rafael Mena-Yedra, Jordi Casas, and Ricard Gavaldà. «Assessing spatiotemporal correlations from data for short-term traffic prediction using multi-task learning». In: Transportation Research Procedia 34 (2018), pp. 155–162. ISSN: 2352-1465. DOI: https://doi.org/10.1016/j.trpro.2018.11.027 URL: http://www.sciencedirect.com/science/article/pii/S2352146518303168

  – Preliminary version in: Rafael Mena-Yedra, Jordi Casas, and Ricard Gavaldà. «Assessing spatiotemporal correlations from data for short-term traffic prediction using multi-task learning» In: Proceedings of the 6th International Symposium of Transport Simulation (ISTS'18) and the 5th International Workshop on Traffic Data Collection and Its Standardization (IWTDCS'18). Matsuyama, Japan, August 2018.

- Rafael Mena-Yedra, Jordi Casas and Ricard Gavaldà, «Probabilistic model for robust traffic state identification in urban networks». 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, October 2019, pp. 1934-1940. DOI: https://doi.org/10.1109/ITSC.2019.8917259 URL: https://ieeexplore.ieee.org/document/8917259

I have also contributed to the related publications by the Aimsun group:

- Yaroslav Hernandez-Potiomkin, Mohammad Saifuzzaman, Emmanuel Bert, Rafael Mena-Yedra, Tamara Djukic, and Jordi Casas. «Unsupervised incident detection model in urban and freeway networks». 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, November 2018, pp. 1763-1769. DOI: https://doi.org/10.1109/ITSC.2018.8569642 URL: https://ieeexplore.ieee.org/document/8569642
- Mohammad Saifuzzaman, Tamara Djukic, Yaroslav Hernandez-Potiomkin, Rafael Mena-Yedra, Emmanuel Bert, and Jordi Casas. «Understanding incident impact on traffic variables to reduce false incident detection». In: Proceedings of the 40th Australasian Transport Research Forum (ATRF). Darwin, Northern Territory, Australia, November 2018.

2 Background

2.1 Traffic data

> "Measure what is measurable, and make measurable what is not so"
>
> — Galileo Galilei

The quote attributed to Galileo Galilei —with certain controversy on authorship [152]— is a clear statement that characterizes the methods of modern science and makes even more sense in the more recent data science. In the specific case of traffic flow, different aspects of its dynamics are captured by different measurement methods [247]. From a global perspective, an analogy can be made with field theory where there exists two specifications of the flow field: the Lagrangian and Eulerian specifications of the field. The Lagrangian specification is the way of looking at the fluid motion as it moves through space and time, thus giving its path. While the Eulerian specification is the way of looking at fluid motion that focuses on specific locations in the space through which the fluid flows as time passes.

Analogously, traffic flow data can be distinguished by how it is measured, i.e. by observing the flow either with a Lagrangian or a Eulerian specification, although methods of traffic data collection are diverse as a result of the need to make measurable more traffic aspects and in a more accurate way. Thus, some of them can be considered from both perspectives depending on the post-processing performed. For instance, data recorded from traffic cameras or drones can be processed to derive trajectory data which provides an unbiased estimate of traffic density and lane changes in spite of the involved technical difficulties. Moreover, equipped vehicles can provide floating-car data with a high level of microscopic detail, although this data can be biased according to the type of supplier vehicles. This data can be combined with other floating data such as mobile data or connected vehicle data. Toll data could be seen as a special type of trajectory data. Finally, cross-sectional data is captured by stationary induction loops, radar, or infrared sensors which are placed on a fixed location within the road traffic network. Deriving data from a Lagrangian specification —trajectory or floating data— to an Eulerian specification —i.e. cross-sectional data— is easier than

the opposite conversion, which maybe simply not possible —e.g. estimating trajectories or origin-destination paths from isolated cross-sectional data—. Obviously, each conversion will have its own error depending on the assumptions. Each of these data sources provides different measurement with different level of microscopic detail that can be post-processed to derive the desired levels of aggregation up to a macroscopic detail. Measurement devices and processes have evolved to provide more accurate and reliable measurements over time, but there is still room for improvement as noisy and false measurements are significantly present in traffic data measurements. All said, every data source has its own advantage and the best result is obtained by fusing them [85], as this results in a better observability of the network.

As previously stated in the thesis objectives, the aim of this thesis is not to deal with microscopic details of traffic dynamics but from a more aggregated macroscopic level of detail. Besides, the data used in this thesis is collected from stationary inductive-loop traffic detectors given that they are historically the most common traffic data collection method, being almost ubiquitous. This does not prevent, however, using other kinds of data sources that are becoming more extended, such as mobile data or connected vehicle data. Our proposals are flexible enough to incorporate this info if available.

Collected data used in this thesis —provided by stationary inductive-loop traffic detectors placed through the road network— include measurements, recorded as time series, of:

- Traffic flow usually given as traffic counts —i.e. the number of counted vehicles during the measurement interval Δt— which are converted to traffic flow Q given in vehicles per hour. It is defined as the number of vehicles ΔN passing through a specific location x within a time interval Δt:

$$Q(x,t) = \frac{\Delta N}{\Delta t}$$

- Occupancy $O(x,t)$ is the fraction of the time interval Δt during which the location x is occupied by a vehicle:

$$O(x,t) = \frac{1}{\Delta t} \sum_{\alpha=\alpha_0}^{\alpha_0+\Delta N-1} (t_\alpha^1 - t_\alpha^0)$$

  where α represents each individual vehicle, t_α^0 the instant when the α^{th}-vehicle's front passes the detector and t_α^1 the instant when the α^{th}-vehicle's rear end passes the detector. Sometimes,

the spatial measurement traffic density $\rho(x,t)$ is derived from the temporal measurement occupancy with certain assumptions. However, this derivation is not used in this thesis.

- Speed $V(x,t)$, namely arithmetic mean speed or time mean speed, is the average speed of the ΔN vehicles passing the cross-section x during the aggregation interval Δt:

$$V(x,t) = \frac{1}{\Delta N} \sum_{\alpha=\alpha_0}^{\alpha_0+\Delta N-1} v_\alpha$$

where v_α is the microscopic speed of single vehicles. It is important to remark that speed, if measured using stationary inductive-loop traffic detectors, can only be directly observed when these devices are composed of more than one loop. Otherwise, it is impossible for single-loop detectors to measure vehicle speed but being estimated instead using certain assumptions. For this reason, we have only used speed in one —M4/M7 Western Motorway in Sydney— of the two datasets used in this thesis because in the case of Santander data is retrieved only from single-loop detectors.

Sometimes this data is given per road lane detector, but most of the time it is retrieved as aggregated multilane data —i.e. stations—. Thus, already aggregated data along multiple lanes in the road pointing to the same direction is used in this thesis. Lastly, the aggregation interval Δt used in this thesis is $\Delta t = 15$ minutes as it represents a proper timing for real-time traffic management and also in agreement with the rest of performed operations in Aimsun Live.

There exists a correlation among these macroscopic measurements (flow, occupancy or density, and speed). More specifically, a pairwise non-linear dependence can be observed between them which underlies the existence of the concepts of capacity and congestion by relating the traffic demand and supply. These relationships were empirically observed by Greenshields [110] almost 100 years ago and captured in the form of a fundamental diagram of traffic flow for each pair of macroscopic measurements. From this fundamental diagram of traffic flow —which will be put in context in the *Traffic state identification* section within Chapter 3—, it can be observed the different underlying traffic states such as free-flow or traffic congestion which relate both sides of transport: traffic demand and traffic supply.

The concept of the fundamental diagram of traffic flow as shown in Figure 2.1 is observed at a macroscopic scale of the traffic flow, and it characterizes the traffic state. A large volume of literature exists on the description of the traffic state [41, 246, 143]. The typical situation is free-flowing conditions when the demand flows are below the capacity of the road network. Here, speeds tend to be near the speed limit, the occupancies are low, and vehicle headways are comfortable. In

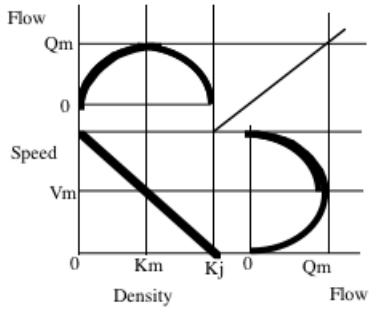


Figure 2.1: Traditional fundamental diagram of traffic: Flow-density and speed-concentration curves. Source: [164].

congested conditions, the actual flows reduce, but the demand flows remain high; the vehicles slow down, the occupancies increase, and vehicles pack more closely together. During congestion the road system is operating in an inefficient manner, with increased vehicle delays, driver frustration, and greater potential for accidents. In addition to these two states, there exist two distinct transition states, where the traffic state changes from free-flowing to congested and from congested to free-flow conditions. These two states may be different from each other in their characteristics. Some properties of the traditional fundamental diagram are the following:

- The variables of flow, density, and space mean speed are related by the definition: $q = k\bar{v_s}$.
- When density on the highway is zero, the flow is also zero because there are no vehicles on the highway.
- As density increases, flow increases.
- When the density reaches a maximum jam density (k_j), flow must be zero because vehicles will line up end to end.
- Flow will also increase to a maximum value (q_m), increases in density beyond that point result in reductions of flow.
- When density is zero, speed is freeflow (v_f). The upper half of the flow curve is uncongested and the lower half is congested.
- The slope of the flow density curve gives speed.

For a comprehensive taxonomy of traffic modelling according to the different scopes of travel demand modelling and traffic flow dynamics, the reader is referred to the Appendix I: Taxonomy of traffic modelling, and those interested readers in knowing more are referred to [28, 100, 125, 174, 247].

2.2 Data modelling

Given that traffic flow in the road transportation systems is a highly non-linear process, mostly non-stationary and which is affected, or could be explained, with different external factors such as different levels of seasonality, weather, special days and events, roadworks, among others; many have been the approaches taken to deal with such problem from a data-driven modelling perspective.

From a very general perspective, the different data modelling techniques could be classified according to the data model they use. For instance, some approaches, especially in the beginnings of the field, have used a scalar-based data model such as for example typical approaches relying on time series modelling, in order to describe the traffic flow at a specific point in space and time. On the other hand, the vector-based data model relies on the use of a dimension —either spatial or temporal— to describe the current traffic state. Lastly, a matrix-based model makes use of both dimensions —spatial by means of other points in the network and temporal through the use of past recent information— in their data model to describe the current traffic state.

Another way to make a distinction among data models is by how they are formulated based on parametric statistics or a non-parametric approach.

2.2.1 Parametrics and non-parametric statistics

Parametric and non-parametric statistics are branches of statistics which differ on the assumptions done about the data generation process and how it can be effectively modelled. More specifically, the former assumes that sample data comes from a population that can be adequately modelled by a probability distribution that has a fixed set of parameters θ. Ocassionally, the assumption about a specific probability distribution is not performed but the fixed set of model parameters remains. On the other hand, in non-parametric statistics the number of parameters θ in the model is not fixed and can increase —or decrease— as new data is collected.

In the case of parametric statistical models, the advantage is clear as the model complexity is bounded by definition. Furthermore, as long as the modelling assumptions are met during the inference stage then the model's sound basis is guaranteed. Some well-known parametric statistical models include time series modelling or Kalman filter. Moreover, traffic simulation models also fall into this category. The possible drawback of the parametric approach is that these models are not able to utilize additional information which can be present in a large or unbounded dataset.

On the other hand, a non-parametric model does not rely on a fixed parameter set θ and this implies that more data is usually required for the model fitting stage. However, this also implies

that this approach is more flexible because the amount of information that can be allocated in θ can grow as the amount of data grows, making this approach more suitable for those scenarios where the size of the dataset is unbounded. This characteristic allow such models to learn more complex relationships that those which they were initially designed for. A possible pitfall, however, is that computational cost during fitting or inference stages can be greater compared to those with a predefined and fixed parameter set θ. Well-known examples of non-parametric approaches include k-nearest neighbors, decision trees or neural networks. The latter is not a true non-parametric approach in the sense that the parameter set is not unbounded, but, in the practice it is extremely large —especially in deep neural networks— that allow them to be classified as it.

For the development of this thesis, a non-parametric approach has been adopted given that the results of this thesis is intended to run in real-time scenario where size of incoming data is unbounded and, moreover, complexity of the modelled problem can grow up to such an extent where the assumptions performed during the modelling stage would have been violated in such non-stationary and dynamical systems.

2.2.2 Machine learning

Machine learning (ML) is a subfield of computer science which also contributes to the broader area of study of Artificial Intelligence. Originally, the ML field evolved from the study of pattern recognition and computational learning theory and it lies on the boundary of several different academic disciplines: namely computer science, statistics, mathematics and engineering. ML explores the study and construction of algorithms that can learn from and make predictions on data. Such algorithms operate by building a model from example inputs in order to make data-driven predictions or decisions, rather than following strictly static program instructions. A more formal definition by Tom M. Mitchell [182]:

> A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E.

Machine learning aims to create solutions to complex problems automatically, faster and more accurately than a manually programmed solution and in a larger scale. The latter characteristic is key as ML is often seen as an extension to classical statistical learning to deal with large-scale problems which are usually described as *big data*. In fact, there always has existed some controversy between practicioners of both communities [43, 175]. Some comprehensive references about machine learning and statistical modelling are [38, 102, 109, 182, 187]

The recent popularity of the field and its growing study and application is motivated by the abundance of data to learn from, the increase of computation power and at lower cost, and because the field has matured both in terms of identity and in terms of methods and tools. The abundance of data is key especially in the traffic scenario as there is an increasing amount of available traffic flow data ranging from individuals to macroscopic data collected by stationary detectors and complemented with additional data such as floating car data coming from GPS and mobile phones, making measurable more things and more widely as there are more observable parts of the network.

The common basic type of problem which ML intend to solve in an effective and automated manner are regression, classification, clustering or rule extraction from data. A common taxonomy of ML algorithms is by the learning procedure; namely supervised learning —when data is labelled, and the learning method makes corrections according to the datum labels, e.g. in regression or classification—, unsupervised learning —data has no labels and common problems include clustering, dimensionality reduction or association rule learning— or a hybrid semi-supervised learning approach.

During the development of this thesis, both approaches —supervised and unsupervised learning— have been used for different parts of the thesis that cooperate each other forming a whole. Next, some of the most important concepts and methods from the machine learning, probability and statistical learning theory which are used in this thesis are described.

2.2.2.1 Empirical risk minimization (ERM)

One of the cornerstone in this thesis is how we find the proper spatiotemporal correlations between different areas in the network in order to predict the evolution of the traffic flows dynamically.

In its most elemental form given a regression problem, we have two spaces $X \in \mathbb{R}^p$ and $Y \in \mathbb{R}$ and we seek to learn a function $h : X \to Y$ where h is also known as hypothesis. Given a supervised learning setting, we have a *training set* of n labelled examples $(x_1, y_1), \ldots, (x_n, y_n)$ where $x \in X$ is the input and $y \in Y$ is the desired output that we want to learn in the form of the hypothesis $h(x_i)$.

We also assume that we are given a non-negative real-valued loss function $L(y, \hat{y})$ which measures how different the prediction $\hat{y}$ of a hypothesis is from the true outcome y. The prediction $\hat{y}$ is defined as the inner product of X and a vector of weights —also called coefficients— with a value per feature: $\langle \beta, X \rangle$.

The risk associated with hypothesis $h(x)$ is then defined as the expectation of the loss function: $R(h) = \mathbf{E}[L(y, h(x))]$. For the problem of finding the set of predictive spatiotemporal correlations

in the network, we are interested in the squared-error loss function:

$$L(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

The aim of a learning algorithm is to find a given hypothesis h^* among a fixed class of functions $\mathcal{H}$ for which the risk $R(h)$ associated with this loss function is minimal:

$$h^* = \arg\min_{h \in \mathcal{H}} R(h)$$

Another way to say so is by assuming a joint probability distribution $P(x, y)$ over X and Y and having a training set which consists of n instances $(x_1, y_1), \dots, (x_n, y_n)$ drawn independent and identically distributed (i.i.d.) from $P(x, y)$. For this case, the risk associated with hypothesis $h(x)$ is also defined as:

$$R(h) = \mathbf{E}[L(y, h(x))] = \int L(y, h(x))\, dP(x, y)$$

The problem is that in general $R(h)$ cannot be directly computed as the true distribution $P(x, y)$ is unknown to the learning algorithm —which is referred to as *agnostic learning*—. Therefore, an approximation called empirical risk R_{emp} can be computed by averaging the loss function on the training set:

$$R_{\text{emp}}(h) = \frac{1}{n} \sum_{i=1}^{n} L(y_i, h(x_i))$$

The empirical risk minimization (ERM) principle [251] states that the learning algorithm should choose a hypothesis $\hat{h}$ which minimizes this empirical risk:

$$\hat{h} = \arg\min_{h \in \mathcal{H}} R_{\text{emp}}(h)$$

In our case, ERM is carried out with a linear kernel over X. X is a matrix as it contains the spatiotemporal data describing the current traffic state. More specifically, X is of size $n \times p$ where each of the n observations has p coordinates composed of $p_s \times p_t$ elements, where p_s refers to each of the spatial points —*id*— within the road network with measurement data and p_t refers to the vector of lags used for each of these measurements:

$$X = \begin{bmatrix} x_{1,id_1} & x_{1,id_1-1} & \cdots & x_{1,id_1-p_t} & \cdots & x_{1,p_s} & x_{1,p_s-1} & \cdots & x_{1,p_s-p_t} \\ & \ddots & & & & & & & \\ & & \ddots & & & & & & \\ & & & \ddots & & & & & \\ & & & & \ddots & & & & \\ & & & & & \ddots & & & \\ & & & & & & \ddots & & \\ & & & & & & & \ddots & \\ x_{n,id_1} & x_{n,id_1-1} & \cdots & x_{n,id_1-p_t} & \cdots & x_{n,p_s} & x_{n,p_s-1} & \cdots & x_{n,p_s-p_t} \end{bmatrix}$$

Depending on the size of the road network at hand, X is usually of a high-dimensionality p —often of the order of tens of thousands—, whereas the sample size n is dependent of the available data. Thus, it is often usual to have situations where $p >> n$ which leads to an ill-posed problem whose solution is not unique. Furthermore, we are not interested in solutions where the traffic forecast for a given area is dependent on the full traffic state of the whole network because it has hardly physical sense. Even though we could manually cut the amount of features based on some distance criteria, this could lead to suboptimal solutions in terms of forecasting performance, but also often the cut-off would not be enough as p could continue being high-dimensional. Therefore, we are interested in an optimal and automated method for feature selection to select only those strongest spatiotemporal correlations leading to a simpler and parsimonious model selection. This parsimonious model selection is known to have additional benefits such as better interpretability, fewer chances of overfitting issues —because in high dimensions, ERM overfits the data and gives poor estimators even for simple linear models [91]— and the result is more computational efficient.

For the sake of comprehensiveness we also describe y as the vector of the desired response for each of the n observations:

$$y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

This response matrix contains the desired output for the desired forecasting horizon t_h which must be learned with the use of the known information until the current time t_0 encoded in X. This means we have adopted a direct multi-step forecasting approach based on learning this process for each of the multiple forecasting horizons t_h. This procedure is more common in machine learning

compared to more classical statistical modelling such as times series [58, 237]. The main reason for this choice lies on the higher robustness of direct multi-step estimation when the model is misspecified especially for non-stationary data generating processes [58].

2.2.2.2 Sparsity regularization learning

Based on the motivation given in the previous subsection about the need of reducing the size of explanatory variables for each of the forecasting processes, we seek to find the subset of features from X that best describe the output Y. In this sense, the general class of sparsity regularization methods seek to exploit this assumption during the learning stage of the problem to find that feature subset. As commented before, this has additional benefits such as a better model interpretability, less chances of overfitting issues —because in high dimensions, ERM overfits the data and gives poor estimators even for simple linear models [91]— and provides the optimal trade-off between complexity and accuracy.

Given the expectation of the squared loss function we want to minimize:

$$\min_{\beta \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} (y_i - \langle \beta, x_i \rangle)^2$$

The idea is to add to the previous optimization problem a constraint based on a vector norm over the features' weights β. This constraint can be directly plugged into the goal function with the use of a Lagrange multiplier, λ, by the introduction of a penalty term:

$$\min_{\beta \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} (y_i - \langle \beta, x_i \rangle)^2 + \lambda \|\beta\|_p$$

This p-norm may have different flavours as shown in Figure 2.2. The intuition behind is that through the optimization process a solution is to be found where the feasible region of the solution matches that of the loss contour with the ℓ_p region. This has an effect on the shrinkage of the coefficients β, and such effect depends on the type of ℓ_p norm used. Small ℓ_p norms such that $0 \leq p \leq 1$ have the effect of obtaining sparse estimates as there are more chances that the contour of the residual-sum-of-squares function touches them along the coordinates —because of their more angular shape— and thus estimating them exactly to zero. On the other hand, ℓ_p norms such that $p \geq 2$ have still the effect of coefficients shrinkage but not setting them exactly to zero and thus no sparse estimation. This is shown in Figure 2.3 where ℓ_1 and ℓ_2 norms are visually compared. This effect is amplified when the number of coefficients increases in high-dimensional problems.

More specifically, the ℓ_0 norm is equivalent to the best subset selection method, but it is computationally unfeasible especially in high-dimensions as it is equivalent to an exhaustive search evaluating all possible subsets of variables. In the same way, computation of $0 < L_p < 1$ norms suffer from numerical stability. Therefore, the most natural approximation for obtaining sparse estimations is through the ℓ_1 norm which, althought not differentiable, is convex. This convex relaxation allows a more efficient solving of the problem by the use of specific efficient algorithms. These optimization algorithms take into account both the differentiable convex part of the loss function along with the non-differentiable convex part of the penalty. For example, projected or proximal gradient descent methods with subgradients [31], coordinate descent methods coupled with the soft-thresholding operator [92, 90] or homotopy methods such as least angle regression (LARS) [80] whose drawback is that it does not scale up to large problems as well as some of the other methods.

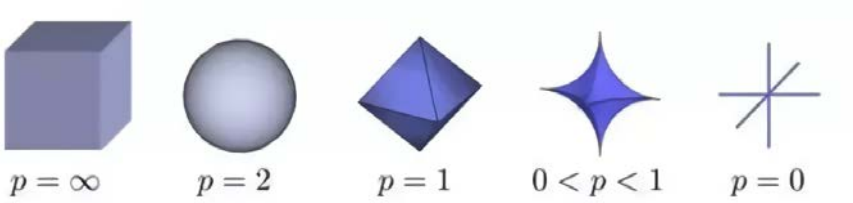


Figure 2.2: ℓ_p ball in three dimensions. As the value of p decreases, the size of the corresponding ℓ_p space also decreases. Source: [117].

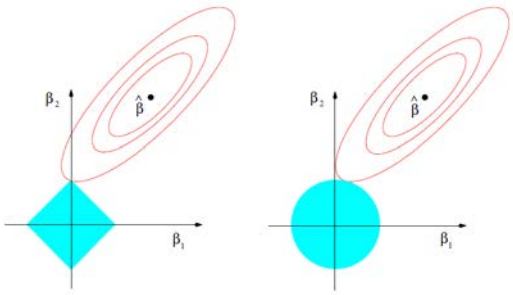


Figure 2.3: Estimation picture of the feasible solution space in a two-dimensional space when using the norms ℓ_1 on the left and ℓ_2 on the right. The solid blue areas are the constraint regions of these norms and the red ellipses are the contours of the residual-sum-of-squares function. The point $\hat{\beta}$ depicts the usual (unconstrained) least-squares estimate. Source: [117].

In the machine learning field, the ERM procedure for estimating least-squares models applying ℓ_1 regularization to obtain sparse estimates is known as lasso —least absolute shrinkage and selection operator— and was popularized by R. Tibshirani [242]. It is defined as follows:

$$\min_{\beta \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} (y_i - \langle \beta, x_i \rangle)^2 + \lambda \|\beta\|_1$$

From a probabilistic point of view, ℓ_1 norm would correspond to setting a Laplace prior centered at zero on the coefficients and selecting the maximum a posteriori (MAP) hypothesis after observing the data [201]. On the other hand, ℓ_2 norm would correspond to setting a Gaussian prior over the coefficients.

Concluding, with the use of such regularization methods in the estimator we incur in increasing the model bias, but, at the same time, also reducing its variance which is beneficial because variance dominates in high dimension within the bias-variance trade-off [73].

A complementary method that aims to reduce the computational burden, but that also has an indirect effect on obtaining sparse estimates is the use of *screening rules*. These screening rules aim to a priori eliminate predictors from the problem before solving it based on the connection between predictors inner products with the residuals, as those with a small inner product have less chances to become a nonzero coefficient.

In this sense, there are some *safe* screening rules that are more conservative as they perform the discard while still delivering the exact numerical solution. A example of this is the dual polytope projection (DPP) rule [259]. The global DPP discard the x_j predictor if:

$$|x_j^T y| < \lambda_{max} - \|x_j\|_2 \|y\|_2 \frac{\lambda_{max} - \lambda}{\lambda}$$

While the sequential version of the DPP discard the x_j variable at stage λ' if:

$$|x_j^T (y - X\hat{\beta}(\lambda'))| < \lambda' - \|x_j\|_2 \|y\|_2 \frac{\lambda' - \lambda}{\lambda}$$

where $\lambda < \lambda' \leq \lambda_{max}$ and $\lambda_{max} = max_j |\langle x_j, y \rangle|$ is the λ which corresponds to having all coefficients set to zero, as typically the lasso problem is solved by multiple iterations decreasing the λ value starting from λ_{max} —proccess typically called *regularization path* of the lasso—.

On the other hand, there exist screening rules which are less conservative thus achieving better performance by allowing occasional failures. For instance, the global strong rule [243] —which is a variant of the global DPP rule— discards x_j if:

$$|x_j^T y| < \lambda - (\lambda_{max} - \lambda)$$

And the sequential strong rule discards x_j at λ if:

$$|x_j^T (y - X\hat{\beta}(\lambda'))| < \lambda - (\lambda' - \lambda)$$

As previously said, sequential version of screening rules are applied when lasso problem is solved over a grid of decreasing λ values —regularization path—.

Occasional failures in sequential strong rule can be remediated for each value of λ by checking the Karush-Kuhn-Tucker (KKT) conditions on the subset of predictors and, if any of these violates the conditions they are added back to the subset and the problem is solved again. Otherwise, problem solving continues to the next λ value in the grid. Thus, sequential strong rule has been used in this thesis as it alleviates computational burden without sacrificing the exact solution.

Additionally, other kind of heuristic screening rules are applied which will be described in later chapters.

2.2.2.3 Multi-task learning

Multi-task learning (MTL) is a broad paradigm in the realm of machine learning itself, whose main goal is to improve the generalization performance —i.e. out-of-sample accuracy— of a model by leveraging the domain-specific information contained in a set of correlated tasks. MTL entails the following benefits:

1. The model selection is biased to prefer hypotheses that other tasks also prefer, and thus improving the generalization for new tasks in the same domain,
2. Data augmentation is implicitly done by averaging the noise patterns among tasks,
3. It allows to differentiate between relevant and irrelevant features especially when the data is noisy or high-dimensional as other tasks will provide additional evidence for the relevance or irrelevance of those features,
4. It has a regularization effect by avoiding the risk of overfitting the random noise of a single task.

Therefore, the main motivation of appliying MTL is to bias the model selection towards those hypothesis that best jointly explain the set of related tasks, and thus obtaining more realistic and consistent models.

Most of the MTL research has been focused on neural networks [52], and their contemporary version of deep learning [212]. The most usual approach is based on sharing a common representation. This sharing can be made explicitly by having a neuron or a whole output layer for each task while all the hidden layers interconnections are shared, which is known as connectionist approach or hard parameter sharing. Instead of sharing the parameters connections, another approach is to have a

separate network structure for each task but adding some constraint to force the distance between parameters to be reduced, which is also known as regularization or soft parameter sharing.

However, MTL has been also developed with non-neural models such as linear models, kernel methods, decision trees or Bayesian algorithms. One of the more common approaches to apply MTL is to enforce sparsity across tasks by applying norm-regularization also known as block-sparse regularization. This usually assumes that only a few features are used across all tasks [19], by means of generalizing the ℓ_1 norm to the MTL setting.

As described in the previous subsection about sparsity regularization, the regularization term $\lambda\|\beta\|_1$ imposed by the ℓ_1-norm penalizes each β_j component independently, which means that input features may be suppressed —keeping their coefficients as zero— independently from each other. Thus, more structure can be given to the norms defining the regularization term. For example, by assuming a prior partition of the feature space in G groups and thus having a subset β_g $\forall g \in G$, the regularization term turns out to be:

$$\lambda \sum_{g=1}^{G} \|\beta_g\|_q$$

where $\|\beta_g\|_q$ is the group ℓ_q-norm. A common choice for ℓ_q is the ℓ_2-norm:

$$\|\beta_g\|_2 = \sqrt{\sum_{j=1}^{|G|} (\beta_g^j)^2}$$

where β_g^j is the j-th feature of group $g \in G$. This norm is referred to as group lasso [272], and it forces entire coefficient groups β_g towards zero, rather than individual coefficients.

This group norm can be further generalized to the case of multiple linear regression where there are multiple responses to be jointly optimized. As aforementioned, it is called block-sparse regularization —or mixed-norm constraints ℓ_1/ℓ_q— and consists of establishing some specific ℓ_q norm on each individual coefficient vector shared across the set of tasks —as a synonym of responses— K, following a ℓ_1 norm over the previous vector norms which in the end results in having entire coefficient vectors set to zero across tasks.

Thus, for the case of multiple response using ERM, the single response vector y is replaced by a matrix Y of size $n \times K$ where n is the sample size and K the set of K jointly learned tasks, and the coefficient vector β of length p features is replaced by a matrix $\mathcal{B}$ of size $p \times K$. Then, the absolute individual penalty on each single coefficient β_j is replaced by a group-lasso penalty on

each coefficient K-vector β_j for a single predictor x_j, where each group $\mathcal{B}_j$ corresponds to the jth row of the $p \times K$ coefficient matrix $\mathcal{B}$:

$$\min_{\mathcal{B} \in \mathbb{R}^{p \times K}} \frac{1}{n} \sum_{i=1}^{n} \|y_i - \langle \mathcal{B}, x_i \rangle\|_F^2 + \lambda \sum_{j=1}^{p} \|\mathcal{B}_j\|_2$$

where $||\cdot||_F$ denotes the Frobenius norm[1]. In such case, for the ℓ_1/ℓ_2 penalized multiple Gaussian-response linear models that are used in this thesis, the sharing involves which features are selected since when a feature is selected then a coefficient is fit for each response, which turns out to be useful when there are a number of correlated responses or tasks to learn. There are other approaches, however, that also aim to achieve within-group sparsity [222].

Nevertheless, deciding which tasks are to be grouped to be jointly learned is a crucial decision in the MTL paradigm. As Caruana [52] demonstrated with different experimental scenarios, the overall performance only improves when similar and related tasks are jointly learned, whereas the opposite, learning unrelated tasks, may lead to suboptimal performance —named as negative transfer—. For instance, the block-sparseness approach is very dependent on the extent to which the features are shared across tasks [190], and others [135] have proposed approaches which deal with block-sparse and element-wise regularization separately. There is wide work referring to the task relatedness [141, 160, 215].

Finally, in addition to the block-sparse regularization, another form of multi-task learning approach has been applied in the development of this thesis related to the process of automated rule search which will be described in later chapters.

2.2.2.4 Data streams and online learning

Historically, machine learning research and practice have focused on batch learning usually using small and finite datasets. In the batch learning scenario, all the available collected data is used as training data —along with the corresponding common data splitting techniques in ML— for the learning algorithm that outputs a decision model after processing all the data. The main assumption that underlies on it is that data is generated from a stationary probability distribution.

However, this is contradictory with the fact that, when data models are put into production in real-time environments, they must respond in dynamic environments where data is collected over time and it is theoretically unbounded in size. Thus, the ability to incorporate new data and to dinamically adapt to changing environments becomes essential for learning algorithms. It can be

[1]The Frobenius norm of a matrix is simply the ℓ_2-norm applied to its entries

differentiated, however, between a smooth and gradual change —named as *concept drift*— that should be incrementally incorporated in the learning process and an abrupt and sudden change —named as *concept shift*—. Moreover, methods to forget outdated data are also necessary. All this leads to a whole research field for the development and application of ML algorithms for data streams [36].

What characterizes a data stream [188] is:

1. The data elements in the stream arrive on-line and continuously.
2. There is no control over the order in which data elements arrive, either within a data stream or across data streams.
3. Data streams are theoretically unbounded in size.
4. Random access to past data must be reduced in the sake of efficiency, or even avoided in some applications.

In complex systems and for large time periods, changes in the distribution of the examples and in data relations are expected. Therefore, the solution does not rely on just using an algorithm specifically adapted for incremental learning, but it is all parts of a whole. Since the most fundamental methods such as counting problems or basic statistics aggregates are need to be adapted to deal efficiently in such online environments. As an example, the sample mean can be recursively estimated with every new observation x_i:

$$\overline{x}_i = \frac{(i-1)\,\overline{x}_{i-1} + x_i}{i}$$

Similarly, the standard deviation can be recursively estimed:

$$\sigma_i = \sqrt{\frac{\sum x_i^2 - \frac{(\sum x_i)^2}{i}}{i-1}}$$

The use of a sliding window to take into account the most recent past is also common procedure in the realm of data streams. These can also be used as a method for data synopsis, along with data reduction methods such as sampling [253], histograms [111] or wavelets [150].

A smooth and gradual change —i.e. when the target concept is gradually changed over time— is usually handled with the use of an incremental learning approach. In this regard, some ML models are more naturally prone to such incremental learning than others. This is the case of, for example, *k-nearest neighbors* or probabilistic methods; while others require suitable changes. In this thesis, we have relied on the frequently used decision trees which is a common ML model known for its

interpretability when compared to other kind of black-box ML models. In the classical or original sense [44], it is a graphical model in the form of a tree starting from a root node without any incoming edge and where every other node has exactly one incoming edge and zero or two —or more than two if it is not a binary decision tree— outgoing edges. Those nodes without outgoing edges are called leaves, while the rest are called internal nodes as shown in Figure 2.4. Each edge originating from an internal node is labeled with a splitting predicate. The set of splitting predicates on the outgoing edges of an internal node must be non-overlapping —i.e. the conjunction of any two predicates evaluates to false— and exhaustive —the disjunction of all predicates evaluates to true—.

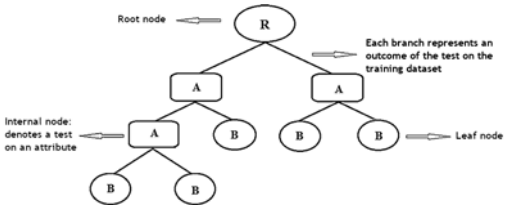


Figure 2.4: Example of a decision tree.

The decision tree model was not originally aimed to online or non-stationary problems, but several remarkable works have appeared in the literature that deal with this problem [74, 96]. Indeed, some parts of the work in this thesis are built on work by Gama [94] and colleagues [12, 131] to deal with data streaming problems. More specifically, their research has been centered on the development of methods for learning in streaming scenarios with the need to be quickly adaptable to non-stationary and dynamic processes, and especially through the building of decision trees. We use this work to address common problems found in real-time applications with real data —such as noisy or faulty data sources— and tailor the development towards the thesis goals and Aimsun Live requirements in the problem of spatiotemporal flow prediction.

In regard to the sparse regularized learning commented in the previous subsections, it is usually solved in batch mode. However, the problem can be solved iteratively without using the full dataset at once. If stated in a probabilistic form, the iterative resolution is obvious and natural by updating posterior probabilities given new evidence and prior probabilities. When not formulated probabilistically —which is very frequently, given that the problem is usually stated as a high-dimensional problem and Bayesian methods are not that scalable—, then the combination of loss function and penalty term must be solved. For this, a closed-form solution is difficult because of the non-differentiable part of the penalty term and the possible collinearities on data. Thus, it is commonly solved with first-order optimization methods which can be set to work as an iterated solver using mini-batches of data. For instance, in our case coordinate descent (coordinate-wise gradient descent) is used to obtain the parameter estimates because it applies well to the case

where $n \ll p$, and it has been successfully applied to problems with high-dimensionality [92, 90, 116], demonstrating to be efficient in large problems [191]. In online learning, the coordinate descent is also applicable using mini-batches of incoming data, a small learning rate and the soft-thresholding technique [219]. Additionally, instead of updating all the coordinates —features— in every update, it is possible to rely on the variable selector to select probabilistically a subset of coordinates. Another approach for online learning with sparse solutions is called Truncated Gradient based on the work by [162], which has other hyperparameters with no direct relationship with the penalty value λ from the batch setting. Other references dealing with the same problem are [248, 271, 57, 99, 16, 84]. Nevertheless, stability in online sparse solvers is of great importance [173].

Another concept related to incremental learning is *gradual forgetting*. In this case, the fundamental idea in time-forgetting mechanisms is to use a function for aging the examples and, thus, giving less importance or weight to older examples reaching a point where examples older than a certain age are forgotten. The age decay can be as simple as using a sliding window scheme or a decay factor [157] or be more sophisticated by detecting when changes occur. Regarding this fact, it is related the realm of change detection when the underlying data generation process or data relations being modelled is distinct enough from the initial conditions, which is common in non-stationary environments. These changes may also take place in an abrupt or sudden manner, thus degrading the perfomance very quickly. Therefore, incremental learning is a necessary property but not sufficient to deal with such abrupt changes. Around change detection —and, thus, concept drift or shift— there is a wide work in the ML literature [97]. In general, methods to deal with concept drift can be analyzed from different perspectives: data management, detection methods, adaptation methods, and decision model management.

The data management determines how memory is handled, i.e. it can be a full memory model where data examples are gradually forgotten using a fading factor that, for instance, can be linear [157] or exponential [153]. Another way to handle the memory is with the use of sliding window techniques which could be of a fixed size or an adaptive window size according to when data changes [35].

The detection model determines the techniques and mechanisms for drift detection, so it can identify the change-point or a small time-window where the change occurs along with a quantification of the change. There are two main approches: one is based on monitoring the evolution of specific performance indicators —such as performance metrics or data properties—, while the other is based on monitoring the difference between two time-windows —one corresponding to a reference past summary and the other over the most recent examples—. Examples of the first approach include the *FLORA* family of algorithms [262], the cumulative sum algorithm (CUSUM) and its variant

Page-Hinkley (PH) test [199]. Examples of the second class include [95, 147]. More specifically, the PH test is a sequential test for monitoring an abrupt deviation in the average of a Gaussian signal, which considers two cumulative variables, m_L^T and m_U^T, defined as the cumulated differences between the observed values and their mean till the current moment:

$$m_U^T = \sum_{t=1}^{T}(x_t - \overline{x}_T - \gamma), \quad m_L^T = \sum_{t=1}^{T}(x_t - \overline{x}_T + \gamma),$$

where $\overline{x}_T$ is the online mean of the observed variable till time T, and γ corresponds to the magnitude of changes that are allowed. The values $M_U^T = \min(m_U^t, t = 1, \ldots, T)$ and $M_L^T = \max(m_L^t, t = 1, \ldots, T)$ are also computed at every time step t. Finally, the PH test evaluates the differences: $m_U^T - M_U^T$ and $M_L^T - m_L^T$. When any of these differences is greater than a given threshold λ, an alarm is raised because of a detected change in the distribution which could be a positive or negative change in the average of the signal. The threshold λ is set according to the admissible false alarm rate. Increasing this threshold will entail fewer false alarms, but might miss some changes.

The adaptation model can be distinguished between blind and informed methods. As the name suggests, blind methods are defined by adapting the learner at regular intervals without taking into account if a change has really occured. On the other hand, informed methods rely on a detection model in order to modify the decision model.

Lastly, the decision model management determines how the decision models are handled in memory, and when they should be created, removed or maintained simultaneously. For instance, the dynamic weighted majority algorithm (DWM) is an ensemble method for tracking concept drift which maintains an ensemble of base learners and performs a weighted-majority vote of their response. In addition DWM dynamically creates and deletes experts in response to changes in performance. Another important aspect is the granularity of decision models. When a change occurs, it does not have impact in the whole instance space, but in particular regions. Thus, having more granular models —e.g. decision trees— helps to perform more efficient adaptations.

2.2.2.5 Probabilistic modelling

Normally in data science, models can be formulated from two perspectives: using a model-based probabilistic approach, or using an empirical or risk minimization approach. The former results in a *generative* model [187], while the latter leads to a *discriminative* model [252, 251]. Generally speaking, a discriminative model learns the decision boundary between the classes of a problem,

while a generative model explicitly models the actual distribution of each class. More specifically, a discriminative model —which was described in the previous subsection of empirical risk minimization (ERM)— does not learn the joint probability distribution $\mathrm{P}(X, Y)$ between the input space X and the output space Y, instead it learns the conditional probability distribution $\mathrm{P}(Y|X)$. On the other hand, a generative model does learn the joint probability distribution $\mathrm{P}(X, Y)$, having $\mathrm{P}(X|Y)$ and $\mathrm{P}(Y)$, and deriving the conditional $\mathrm{P}(Y|X)$ using Bayes theorem. In the end, both modelling perspectives are predicting using the conditional probability distribution $\mathrm{P}(Y|X)$, but the way how it is learned differs. An important note, however, is to remark that probabilistic modelling is not only tied to Bayesian statistics, as probabilistic models can also be fit from a Frequentist perspective using Maximum Likelihood Estimation (MLE).

No approach is uniformly better than other for doing data modelling, and it depends on the purpose and aimed operating mode of the model, the available prior knowledge about the structure of the problem, the amount of available data for model fitting, the complexity and dimensionality of the problem as well as the available computational resources.

For example, non-probabilistic algorithms are not able to generate new samples after learning the mapping between the input space X and the output space Y, because they simply give a separating hyperplane between classes. For this same reason, as discriminative models do not learn the joint probability distribution $\mathrm{P}(X, Y)$ and they do not have either the conditional $\mathrm{P}(X|Y)$, it is more difficult —or less natural— for them to function in outlier detection whereas generative models generally do.

In general, generative models often outperform discriminative models on smaller datasets because their generative assumptions place certain structure which performs as regularization preventing overfitting, whereas the discriminative counterpart might pick up on spurious patterns in such small dataset that do not really exist. However, in the long run as the dataset grows, discriminative models can outperform in case the generative assumptions are not satisfied, since discriminative algorithm make fewer assumptions. An interesting discussion on comparing both approaches is [192].

Generative models are built on top of the principles of probability theory. Probability theory is the natural way to quantify the uncertainty in measurements, parameters or around the model. A probability expresses the degree of confidence that an outcome or an event —a number of outcomes— will occur. The set of all these possible outcomes of a particular experiment is called the sample space Ω. The function which performs the mapping from a sample space Ω into the real numbers is a random variable X which can be either discrete or continuous $X : \Omega \to \mathbb{R}$.

From the axioms of probability theory [155]:

$$\mathrm{P}(X = x) \geq 0$$

$$\sum_{x \in \Omega} \mathrm{P}(X = x) = 1, \quad \int_{x \in \Omega} \mathrm{P}(X = x)\, dx = 1$$

$$\mathrm{P}(X \cup Y) = \mathrm{P}(X) + \mathrm{P}(Y), \quad \text{with } X \cap Y = \emptyset$$

It also derives the addition law of probability or sum rule:

$$\mathrm{P}(X \cup Y) = \mathrm{P}(X) + \mathrm{P}(Y) - \mathrm{P}(X \cap Y)$$

The conditional probability of an event $X = x$ is the probability that the event will occur given the knowledge that an event $Y = y$ has already occurred. It is denoted by $\mathrm{P}(X|Y)$, and it expresses the updated beliefs given the new evidence. If the two events $X = x$ and $Y = y$ were independent $X \perp\!\!\!\perp Y$ then $\mathrm{P}(X|Y) = \mathrm{P}(X)$, and if they were mutually exclusive then $\mathrm{P}(X|Y) = 0$. The product rule states that:

$$\mathrm{P}(X \cap Y) = \mathrm{P}(X|Y)\mathrm{P}(X)$$

where $\mathrm{P}(X \cap Y)$ is the joint distribution $\mathrm{P}(X, Y)$ of random variables X and Y. It is easy to observe that $\mathrm{P}(X, Y) = \mathrm{P}(X)\mathrm{P}(Y)$ when the random variables $X \perp\!\!\!\perp Y$, and when they are mutually exclusive then $\mathrm{P}(X, Y) = 0$. There is also conditional independence: $X \perp\!\!\!\perp Y|Z \Leftrightarrow \mathrm{P}(X, Y|Z) = \mathrm{P}(X|Z)\mathrm{P}(Y|Z)$.

Using the product rule $\mathrm{P}(X \cap Y) = \mathrm{P}(X|Y)\mathrm{P}(X)$ and $\mathrm{P}(Y \cap X) = \mathrm{P}(Y|X)\mathrm{P}(Y)$, by equating them we will get $\mathrm{P}(X|Y)\mathrm{P}(X) = \mathrm{P}(Y|X)\mathrm{P}(Y)$, and then the Bayes theorem is obtained:

$$\mathrm{P}(X|Y) = \frac{\mathrm{P}(X)\mathrm{P}(Y|X)}{\mathrm{P}(Y)}$$

which is the basis for inference and learning in generative models when a Bayesian approach is used. Then, the common nomenclature with a model or parameter set θ and observed data $\mathcal{D}$ is:

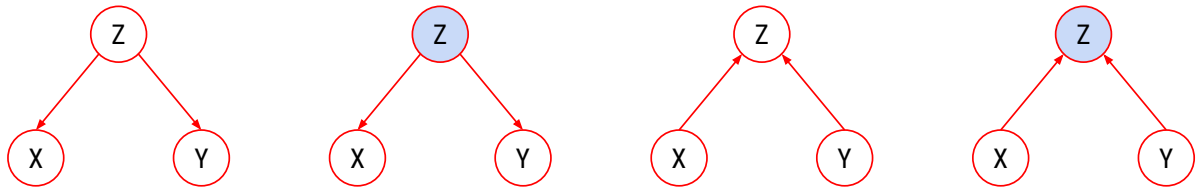
$$\mathrm{P}(\theta|\mathcal{D}) = \frac{\mathrm{P}(\theta)\mathrm{P}(\mathcal{D}|\theta)}{\mathrm{P}(\mathcal{D})}$$

That is, the posterior probability equals the prior probability $\mathrm{P}(\theta)$ times the likelihood ratio $\frac{\mathrm{P}(\mathcal{D}|\theta)}{\mathrm{P}(\mathcal{D})}$. The product rule can be generalized to more than two random variables, $X_1, \ldots, X_n$, as the chain rule:

$$\mathrm{P}\left(\bigcap_{k=1}^{n} X_k\right) = \prod_{k=1}^{n} \mathrm{P}\left(X_k \middle| \bigcap_{j=1}^{k-1} X_j\right)$$

In probabilistic inference, typical queries include marginal queries where the task is to compute $\mathrm{P}(Y|X = x)$ to obtain a marginal distribution, and the Maximum a Posteriori (MAP) queries —most probable explanation— where the task is to find $y* = \arg\max_{y \in \Omega} \mathrm{P}(Y|X = x)$.

All this probability theory can be merged with a graph-based representation which leads to the field of probabilistic graphical models (PGMs). PGMs use graphs to represent the complex probabilistic relationships between the random variables, allowing to compactly represent distributions of variables and in a intuitive manner —such as conditional independences—. In PGMs, nodes represent random variables and edges reflect dependencies between variables. Popular PGMs for inference include Bayesian networks based on a directed acyclic graph (DAG) or Markov random fields based on undirected graphs and factor graphs.



(a) Case 1 (b) Case 2 (c) Case 3 (d) Case 4

Figure 2.5: Bayesian networks cases. Blue nodes are those whose evidence has been observed.

Bayesian networks are a case of PGM whose structure is a DAG, thus not allowing cycles among the variables. The graph-based structure allows to easily impose dependences between variables and conditional independences which reduce complexity in the inference. Their interpretability is also a clear advantage. The factorisation rule for Bayesian networks (DAGs) is:

$$\mathrm{P}(x_1, \ldots, x_n) = \prod_{i=1}^{n} \mathrm{P}(x_i|\mathrm{Pa}(x_i))$$

where $\mathrm{Pa}(x_i)$ denotes parent of x_i.

For example, in Figure 2.5 it can be observed 4 cases of small Bayesian networks. In the case (a), X and Y are not independent each other so $\mathrm{P}(X, Y) \neq \mathrm{P}(X)\mathrm{P}(Y)$. But in the second case

(b), as random variable Z is observed, we can say X is independent of Y given the evidence Z —conditional independence $X \perp\!\!\!\perp Y \mid Z$—, thus $\mathrm{P}(X, Y|Z) = \mathrm{P}(X|Z)\mathrm{P}(Y|Z)$. Lastly, both cases (c) and (d) denote $X \perp\!\!\!\perp Y$ and $X \perp\!\!\!\perp Y \mid Z$, respectively, then $\mathrm{P}(X, Y) = \mathrm{P}(X)\mathrm{P}(Y)$.

Other benefits using PGMs is the exploitation of optimizations to reduce the computational complexity such as variable elimination or message passing. Furthermore, inference algorithms can be exact or approximate. More details are found in Chapter 8 of [38] and especially in [154].

3 Literature survey

3.1 Short-term traffic prediction

The field of short-term traffic forecasting has been active since the early 1980s [4] when it started to be integrated along most of Intelligent Transportation Systems (ITS). It has become a crucial component of them and, thus, it is being widely considered in many areas of transportation research. The traffic characteristics which are usually modelled include the volume, density, speed or travel times of the traffic network, and the forecasting horizon target is over the period of a few seconds to a few hours in the future. Given that road traffic is the visible result of the complex interplay between traffic demand —i.e. the amount of travelers making a trip at a particular place and time—, and traffic supply —i.e. the network infrastructure—, it is usual to find during the modelling stage that the input–output relationship is noisy and such relations in data are multivariate and highly nonlinear [170]. Additionally, the process is usually high-dimensional, non-stationary and tackled in real-time.

In the literature, there are two main approaches to deal with road traffic prediction: model-driven and data-driven [30]. Model-driven approaches try to reproduce the road network behaviour through simulation, and depending on the level of detail and the underlying traffic flow theory they are based on, they could be distinguished among microscopic, mesoscopic, macroscopic and hybrid variants [28, 247]. Aimsun [5] is an example of a commercial product that integrates different simulators for such tasks [53]. One requirement for such model-driven approaches to obtain accurate predictions is to have a detailed knowledge about the network geometry including junctions, lanes, roundabouts and signals, but also certain operational details such as speed limits or control plans. Then, during simulation all the entities, including vehicles, pedestrians and interactions among them, are moved through the network' roads. In this way, it is possible to obtain a prediction of the whole network state in the future.

Model-driven approaches are especially well-suited for long-term predictions that include transportation planning, as for example those planning changes to the existing infrastructure which require modelling future —months or years ahead— traffic conditions while disregarding any type

of real-time information. Model-driven approaches are within the parametric category of models as their number of parameters are fixed in advance as well as the model structure. However, their main drawback is that a detailed knowledge about each road network being modeled is required to obtain accurate predictions, and also it is especially important the inherent fact that both, road network infrastructures as well as traffic demand, are continuously changing and such changes must be reflected into the model structure and assumptions to maintain the accuracy. Further ore, estimating boundary conditions is a challenging task for physics-based approaches, e.g. when ramps have not measurement devices to make them observable, requiring specific methods to deal with it [186, 213]. The ability of this kind of models to model urban arterials and provide good forecasts is also more limited due to the existing traffic flow dispersion [209].

On the other hand, the data-driven approach aims to reproduce the input–output mapping but usually neglecting the underlying data generation process —i.e. well-founded mathematical models that are based on macroscopic and microscopic theories of traffic flow— and disregarding, in general, the network topology. Despite this, integrating the network spatio-temporal information within the short-term traffic prediction task is of ultimate importance [81]. This data-driven research branch has taken advantage from the fact that over time different measuring devices, such as induction loop detectors, controllers, video surveillance systems and even GPS devices installed on buses and taxis, have been deployed within road networks to measure and verify road traffic conditions, thus increasing the network observability. In addition, nowadays with the spread of smartphones, it is easier to collect location, incident information, travel times and routes typically used by drivers. Among the data-driven algorithms, some authors in the literature distinguish between a classical statistical perspective and a more novel research area based on the increasing amount of data mining methodologies and algorithms for analyzing vast quantities of data [142].

At the beginning of its development, most of the research was focused on a classical statistical data-driven approach to predict traffic at a single point [4, 227]. Such classical statistical approaches, commonly ARIMA and derivates or filtering techniques such as the Kalman filter stated as a time series approach, usually match with the above parametric definition in the sense that their number of parameters, model structure and assumptions are fixed in advance [263]. Over time, there have been attempts to enhance the approach, e.g. using a Bayesian perspective [105] or including the existing spatio-temporal correlations in the road network [183]. The main drawbacks of using time series models are that they can not deal efficiently with non-linear processes, and it is hard to integrate environmental data sources into them. Moreover, these parametric models can achieve a good performance when traffic shows regular variations, but the forecast error is obvious when the traffic shows irregular variations.

The growing data availability including the fusion of multiple data sources makes possible to relate traffic conditions to external information sources such as weather, incidents, road works and other special events. This data fusion of heterogeneous data sources coupled with the increased storage capacity and processing power has led to the advent of a novel research area based on the application of more sophisticated methodologies derived from the data science and machine learning fields [189, 214]. Another reason for the focus placed on ML-based methods is that most classical approaches have weaknesses under unstable traffic conditions, complex road settings, as well as when dealing with large datasets [256]. A part of this modelling branch is also called non-parametric in some literature, which implies that the number of parameters can grow indefinitely as more data is gathered as part of the learning process, so the point is that the number of parameters is flexible and not fixed in advance, thus getting a model structure and parameters' values that are totally determined from the data [189]. Nevertheless, a lot of research in this field has assumed an ad-hoc selection of the predictors for each of the network locations depending on the network topology, instead of performing automatic selection based on data. Finally, most of the literature deal with the problem of traffic forecasting from a classical perspective using a bounded dataset and not paying enough attention to the problem of concept drift and changes that may occur both in the supply or the demand, and thus the initial assumptions over the model are not further checked and the model does not evolve or learn with new data. Therefore, in the following lines, the focus is placed on those works whose model proposal is intended to deal in some way with this problem.

Regarding examples of applied methodologies, several research has applied a k-Nearest neighbours schema (k-NN), also known as non-parametric regression. k-NN is a well-suited technique for real-time given that online learning is naturally done with the addition of new cases to the database. In the literature, typically a simple form of k-NN is used. More sophisticated forms include [112] which used a k-NN schema based on weighted Euclidean distance using past time series from a specific spatial point and coupled with some enhancements to deal with specific problems: e.g. a locally weighted smoothing (loess) is adopted in order to reduce the inherent noisy traffic data, the weighted Euclidean distance strengths the recent measurements, and winsorization is used to limit extreme values. Given the continuous addition of new cases to the database, some issues must be handled in order to avoid an excessive penalty in the computational cost. In [129], for instance, a framework named *Spinning network* is proposed to deal with the continuously growing database size and the associated cost with searching and calculating the nearest observations. [149] integrated a k-NN based on weighted Euclidean distance approach with historical time-series traffic patterns which were built on different qualitative criteria such as weekdays or holidays in order to improve the accuracy with the longer forecasting horizons. In any case, a common drawback of the k-NN methodology is the extreme sensitivity to the selected distance function as well as the

selected number of k-neighbors which, with high number of neighbors, can suffer from the so-called curse of dimensionality.

Another common data-driven methodology is based on graphical models, which seem suitable to model road traffic networks. For example, [72] use continuous conditional random fields for speed forecasting, giving probabilistic intervals around predictions and handling missing data with robustness; however, their experiments deal with a small dataset where lot of information —weekends, holidays, and days with a large fraction of missing or corrupted data— has been pulled out for the tests, and as they point out the framework lacks the use of contextual information and thus it is insufficient during special events. Bayesian networks have been appropriately applied to the traffic prediction task. *JamBayes* [126] is commonly shown as a Bayesian inference application to identify the traffic jam formation, it has the ability to include different kind of contextual information and to deal with uncertainty through a probabilistic graphical representation; and even that apparently it is suited for real-time operating, almost no details are given about the structure and parameter learning stages which are especially important for real-time operations. In [234], a Gaussian mixture model (GMM) is used whose parameters are estimated via the competitive expectation maximization (CEM) algorithm. They also adopt the idea of conditional independence, that is, given the adjacent upstream traffic flows at different time delays, traffic flow at the current link is assumed to be independent of other upstream traffic flows. In [55] a method based on Bayesian networks is proposed in order to deal with traffic demand matrix estimation (OD flows) using the link flows and knowing the network topology, they use the Bayesian networks for the matrix estimation step going from individual link flows and turning proportions to aggregated demand matrix (OD flows) and then the traffic assignment, such as Stochastic User Equilibrium (SUE), is used to go from aggregated demand matrix to individuals link flows and turning proportions using certain assumptions. They make some assumptions, e.g. that OD flows follow a multivariate normal distribution as well as that conditional distribution on each link flow given the OD flows follows a normal distribution. Additionally, they use algebraic derivations to update parameters with new evidence one to one but they state that Markov Chain Monte Carlo (MCMC) should be used when the number of nodes in the transportation network is large. In [13, 14], a probabilistic parametric model is proposed, called multiregression dynamic model, built upon the definition of the directed acyclic graph (DAG) of the network where the goal is to estimate the flow propagation in the *forks* and *merges* within the network, and therefore it is essential for the method to have available data at such sites.

Other applications related to real-time and online operation is [56] where a support vector machine for regression is used for one-step ahead prediction and updating the parameters after each obser-

vation is seen. [69] proposed a fuzzy rule-based system optimized with genetic algorithms for the modelling and short-term forecasting of traffic flow in urban arterial networks. Such an approach has the advantage of suitably addressing data imprecision and uncertainty, and it enables the incorporation of expert's knowledge on local traffic conditions within the model structure, however it is also sensitive to the prior definitions of fuzzy memberships derived from expert knowledge.

Linear regression coupled with regularization techniques [166, 233] has been also used for traffic prediction. [124] gives some reference about approaches to deal with online *lasso*, and more specifically, with the sparse variations of the input signals over time with use of ℓ_1-norm combined with ℓ_2-norm for numerical stability. Their focus is on predicting link travel times using taxi probe data. In [139], a regime-switching analysis, e.g. from free-flow periods to congested periods, is performed in order to later build a linear model within these time periods [140, 183].

The Kalman filter stated in the form of time series modelling [196, 270, 171] has been also applied to short-term traffic forecasting.

Neural networks have been widely used in transportation research, including for example traffic control through reinforcement learning [2, 37, 230]. Regarding short-term traffic prediction, they have been considered suitable as the input–output data relationship is noisy and nonlinear and thus there is a vast amount of literature about different kind of implementations [133, 255], even in their most recent form with deep learning [130, 172]. [267] combines different neural network structures —such as long short-term and convolutional neural networks— for mining spatio-temporal patterns. In [106], they build an autoencoder neural network trained to minimize the reconstruction loss, and, because they used a complete data matrix to train the model, they use different matrix completion methods to benchmark, remarking the importance of the missing data imputation problem in networks of traffic sensors. [208] propose another deep learning approach for short-term traffic prediction performing a preliminar step using regularized linear regression for feature selection, and they remark the importance of performing trend filtering in order to smooth the inherent noise in traffic data. Another long short-term memory deep neural network is proposed in [273], where missing data is remedied by using adjacent data in temporal order. Recently, other deep architectures have been applied to the task of traffic forecasting, such as graph neural networks [168, 218]. However, besides the good behavior in accuracy terms of the neural networks approach, a review comparing statistical methods and neural networks [142], pointed out that researchers often implement this approach blindly, ignoring some of their shortcomings such as limited inherent explanatory power, so it is important to know when to use them as this model has limited explanatory power.

Besides, most of the data-driven traffic prediction works found in the literature have been focused

on predicting the traffic from an individual task perspective, neglecting to fully leverage the implicit knowledge shared in a road-network through space and time. This is inconsistent, however, given the recent growing availability of traffic data and, more specifically, the spatial coverage along the road networks which makes it sense to take benefit of the inherent existing spatiotemporal correlations in order to infer the future traffic state. Moreover, papers which have used a multi-task learning paradigm applied to traffic flow modelling have focused the attention on predicting multiple forecasting steps as the set of related tasks, and they have been relied on a neural network modelling architecture. Examples of this application include the works by [130, 136, 235], with a few exceptions [236].

A final remark between model-driven and data-driven approaches is that while data-driven approaches usually forecast the traffic state on specific points —those observable points with measurements—, model-driven approaches simulate the whole network allowing more comprehensive prediction outputs, including *what-if* scenarios. For this reason, some hybrid approaches have appeared, for example [223] proposed an approach based on parametric macroscopic traffic flow model but enhanced with some parts derived from data-driven methods: a Gaussian mixture model for missing data replacement and handling of uncertainty, and an incident detection method based on support vector machines with corrections using a Bayesian network. Other examples of hybrid approaches integrating simulation with data-driven methods are [1, 93, 200].

For additional references and an extended review of the various works on short-term traffic forecasting, the reader is referred to the works of: [250, 170] especially for travel time prediction. [254, 256], contain a comprehensive table with numerous references covering multiple research works. [142] focuses on the comparison between statistical methods and neural networks, listing many research work references in a tabular form. For a review on the importance of including spatio-temporal information into the short-term traffic forecasting, as well as a comprehensive list of works about it, see [81].

3.2 Traffic state identification

Efficient estimation of local traffic states from the fundamental diagram at each detection site is crucial for many real-time traffic management applications both in urban and freeway networks. Usually, these traffic states are inferred from the bivariate relationship between traffic flow and density using a deterministic approach. However, due to traffic congestion and position of detection sites especially in urban networks, this relation is highly scattered making these methods not suitable to handle the associated uncertainty in the process.

The fundamental diagram describes the flow-density and speed-density relationships as well as the speed of kinematic waves and shock waves (i.e. jam fronts) in the case of a freeway scenario where flow conservation and equilibrium state conditions are met. There has been different approaches to model the shape of the fundamental diagram, including pioneer work by Greenshields [110], and Drake's model [76], which are univariate models. Bivariate models distinguished formulation for the congested and uncongested regimes. These include the triangular fundamental diagram and Daganzo's truncated triangular fundamental diagram [62]. The distinction between both branches allowed to analytically estimate parameters such as the free speed equivalent to the slope of the uncongested regime, and the shock wave speed corresponding to the slope of the congested regime. Another traffic phenomenon is related to the traffic hysteresis and the associated capacity drop which is included in some of the fundamental diagram models such as findings by Koshi et al. [156] and the Wu's diagram which has an inverse λ shape [266].

Still, all these approaches are deterministic, leading to a single hyperplane in the flow-density plane, without leaving space for the stochasticity. However, this modelling assumption is problematic because of the wide scattering found in the bivariate relationship, especially in the congested regime. In fact, this wide scattering has its roots in several factors which are not modeled explicitly, namely drivers' behavior, vehicle and environmental conditions, among others. One such work that aimed to deal with the existing stochasticity can be found in [258].

In an urban road the relation is even more chaotic because of the existence of transient phases which are difficult to distinguish from equilibrium phases, and that it is caused by irregular interruptions such as traffic lights, pedestrian crossings or side-street parking. Moreover, the location of inductive loop detectors along the urban roads plays a crucial role in the shape of the resulting local fundamental diagram, especially if they are located just before or after a signaling or intersection, as studied in [257]. However, there have been attempts in the literature to find this flow-density relationship in urban areas. Work by Leclercq [163] is a notable example, where flows and occupancies coming from several inductive loop detectors in the city of Toulouse are used. The aim is to characterize those traffic states which are stable enough and filter the transient ones. He applies data preprocessing and uses a frequency histogram to filter those regions in the plane with higher data frequency in free-flow, while keeping the extremes which are related to the congested regime.

Another research branch that has generated a lot of literature in the recent years is the application of an urban-scale macroscopic fundamental diagram to the whole, usually urban, network. Daganzo and Geroliminis [65] provided experimental evidence, using data from the city of Yokohama, that a macroscopic fundamental diagram with low scattering exists linking space-mean flow and density.

However, Buisson and Ladier [48] showed with data from Toulouse that heterogeneity has a strong impact on the wide scattering and final shape of a macroscopic fundamental diagram. Mazloumian et al. [178] emphasized that the spatial aggregation of traffic variables cannot guarantee a well-defined relationship between the average density and flow, especially when the network is congested, because, in urban networks, congestion is by nature unevenly distributed in space by factors such as demand, road infrastructure and control.

All these works on traffic state identification models are based on a single best-fit curve approach. Another kind of methods in the literature also classify different regions of the fundamental diagram considering a certain amount of stochasticity in the data.

Xia and Chen [268] applied a nested clustering technique based on an agglomerative clustering algorithm to freeway data from California. However, their method is too time-consuming for an online implementation, as it retrains the model each time that a traffic data record arrived in real time. Later, in [269] they proposed a modification of the previous model, where instead of storing all the historical traffic data, they stored statistical features of data in an online agglomerative clustering algorithm, and using Bayesian Information Criterion (BIC) to determine automatically the optimal number of clusters. Kianfara and Edara [146] carried out a comparison with freeway data using hierarchical clustering, K-means clustering and Gaussian mixture model (GMM) for the task of classifying the fundamental diagram in two states: congested and uncongested. Azimi and Zhang [23] also compared K-means, fuzzy C-means, and CLARA algorithms to classify the fundamental diagram into six states using freeway data. Lastly, Antoniou et al. [18] applied, using highway data within an integrated framework for traffic state transition estimation and speed prediction, a model-based clustering (Gaussian mixture model) for the classification of regions in the fundamental diagram, where the number of traffic states was decided using BIC and some manual verification.

In all these referenced works, generally an information-theoretic criterion is used to determine the optimum number of clusters or identified states in the fundamental diagram, such as the one with lowest Bayesian Information Criteria (BIC). In essence, this criterion chooses the model with best trade-off between the maximized value of the likelihood function of the model and an introduced penalty by the number of parameters estimated by the model. While very reasonable, it can still lead to solutions where the number of identified clusters is higher than the desired one, in order to be interpretable or in accordance with the fundamental diagram; or, on the other hand, it can lead to solutions with a small but mixed set of traffic states. Moreover, it often happens that certain locations have not observed the full spectrum of traffic states, e.g. when a section has not been totally congested ever, and in such situations there is no guarantees to make accurate predictions

with such information-based criterion. Finally, the interpretability of the discovered clusters is sometimes problematic, because many models can explain the observed data and not all have a reasonable semantics.

3.3 Incident detection

Recurrent congestion [75, 225] exhibits a daily pattern and its location and duration is usually known by regular commuters and traffic operators. It is mainly caused by excess travel demand, inadequate traffic capacity or poor signal control [114]. On the other hand, non-recurrent congestion may suddenly occur at any time of day and location, as its occurence depends on the local conditions of the road network, as well as travel demand and traffic capacity. A non-recurrent congestion is mainly caused by unexpected events like traffic accidents or vehicle breakdowns. It can also occur due to planned engineering works, special events —e.g. football matches or concerts— or inclement weather [161]. Furthermore, non-recurrent congestion events are considered to be the major source of variability in traffic forecasts [195].

Nearly all non-recurrent congestion events are caused by the simultaneous action of three factors: high traffic load, a bottleneck, and disturbances of traffic flow caused by individual drivers [247]. A high traffic load is necessary as otherwise traffic disturbances cannot grow and propagate since traffic is unconditionally stable for sufficiently low densities. A bottleneck is that *weakest link* with a local reduction of the road capacity, and it can be permanent —e.g. on-ramps and off-ramps, lane closures, road narrowings or curves— or temporary —e.g., when caused by accidents—. The last needed factor is perturbations in the traffic flow itself such as inattentive drivers braking abruptly, speeding cars, lane changes, or trucks overtaking each other.

Despite the fact that the definition, roots and consequences of non-recurrent congestion events are clear; most automatic incident detection algorithms in the literature are developed for freeways, as the detection in urban network remains as a challenging task. Besides, most algorithms in the literature are based on the measurements from loop detectors, partially because loop detector systems have been the most widely used traffic measurement method and are of relatively low cost compared to other detection technologies. However, they present other kinds of pitfalls due to their Eulerian nature as they are placed fixed along the cross-sections. This makes the detection algorithm very dependant on the position of such detection sites, and standard performance measures such as detection rates (DR), false alarm rates (FAR) and mean time to detect (MTTD) are very dependant on their physical placement and distribution through the road network.

Some of the existing reviews in the literature [24, 68, 202, 232] already perform a classification to summarize the historical trend of applied methodologies in the task of traffic incident detection in freeways. Another distinction can be made in the approach these algorithms are learned. If data with labelled known incidents is used, then the problem is usually stated as discriminative classification problem —supervised learning approach— and the goal is to calibrate the algorithm parameters or to define the distributions of *normal* and *abnormal* traffic. On the other hand, the problem can stated using a distance-based approach or stated as a outlier detection problem —unsupervised learning approach—.

Historically, comparative algorithms were designed to compare the value of traffic measurements —volume, occupancy or speed— and their differences against certain pre-established threshold values, thus these algorithms are based on expert-designed rules. They are one of the oldest family of algorithms widely applied in the field of traffic research, and the most widely known are the different versions of the California algorithm [204], which has served as inspiration for multiple posterior modifications and improvements [216]. Other kinds of similar comparative algorithms are [60, 177]. Their major drawback is the difficulty of making them transferable to different networks as they rely heavily on expert knowledge in order to define the rules. For instance, California #8 algorithm involves 21 individual tests, but there exist many variants on the literature. Furthermore, there is the additional difficulty of calibrating these algorithms as this depends on observing incident observations for each monitoring site.

Distance-based approaches have been common in the literature. Standard statistical methods have been used to determine whether observed detector data differ significantly from estimated or predicted traffic characteristics. For instance, the standard normal deviate (SND) algorithm [77] developed by the Texas Transportation Institute is based on detecting significant deviations from the mean of the lanes' occupancy. In the University of California Berkeley (UCB) algorithm [169], the statistical fluctuations of time occupancy are recognized as random walks and those values out of range are indicative of traffic incidents. Another distance-based approach, similar to the previous one, relies on time series modelling methods which assume that traffic normally follows a predictable pattern over time, and therefore they detect incidents when the observed signal deviates significantly from the modelled time series. Although these methods can be stated as unsupervised problem, obviously the challenging part of such approach is to define what is the normal behavior —i.e. pattern— and which features can discriminate properly.

Another widely applied approach is based on aspects of the macroscopic modelling as for example the dynamic model [264]. The McMaster algorithm [207] is based on the catastrophe theory model and relies on a segmentation of the fundamental diagram using data for each detection site, and

the incident detection makes also use of each detection site independently.

In the practice, though, most of freeway management centers has historically relied on a modified version of the California algorithm or the McMaster algorithm for incident detection.

Additionally, much research concluded that raw detector data —e.g. coming from inductive loop detectors— is often inappropriate for incident detection if traffic noise cannot be filtered out. This is because when data is corrupted by noise, the incident patterns can be hidden in the traffic data making them difficult to be detected by certain algorithms, whereas certain fluctuations produced by noise sources can be often detected as incidents. In this sense, smoothing and filtering techniques have been applied to remove noise from traffic data that cause false alarms and hence permit true traffic patterns to be more visible in order to detect true incidents.

More recent approaches to deal with the problem of incident identification have made use of statistical and machine learning methods, and integrating alternative sources of data measurement. For example, [148] proposes a probabilistic approach based on a Bayesian network integrating external factors such as weather for urban networks and using labelled data of reported incidents. In [39, 185] an approach based on frequent subgraph mining is presented to deal with anomalous events. In [15], a spatio-temporal clustering is performed over the link journey times (LJTs) —i.e. the estimated journey time through a link at an established time interval— which are retrieved using automatic number plate recognition cameras, and thus those clusters of substantially high LJTs are identified as incidents. [20] make use of individual vehicle data from GPS trajectories to apply traffic flow theory in order to detect incidents in an urban expressway. In [151], they propose a probabilistic mixture model —also known as probabilistic *topic* model— of Poisson distributions for the speeds of a segment —i.e. a link— which is processed from probe-car data in an expressway, then they check for anomalous observation in the set of mixtures with the use of the Kullback–Leibler (KL) divergence. A similar approach is adopted in [121] with the difference of using data from inductive loop detectors from a real-data freeway and a simulated urban network, and therefore using a mixture of Gaussian distributions instead over the differences of flow and occupancy between adjacents of each detection site, and using the Mahalanobis distance instead of the KL divergence for checking the outlierness distance. In [224], they perform statistical tests to check for statistical differences in the normal and incident scenarios in data which comprehends speeds, travel times, acceleration and lane-change ratio from simulated probe-car data. [118] also uses simulated data, from inductive loop detectors, in order to evaluate a rule-based fuzzy logic approach.

3.4 Contributions

As can be seen, there is a vast literature about short-term traffic forecasting. Despite this fact, there is much less of it that approaches directly the problem of change detection and model self-adaptation. This is surprising, however, because it is well known the high volatility in the relationship between traffic demand and supply, especially in the context of urban networks. In addition, it is not frequent to find details about how to deal with missing or faulty data, as authors usually preprocess data removing rare observations, special days, etc. This constrasts with the situation found in real-time where noisy and faulty data sources are not uncommon. Moreover, there is very little work that integrates such forecasting methods with an automated process of graph pattern mining with ability to measure the outlierness, which has the potential benefit of anticipating anomalous conditions within the road network. Many data-driven works neglect the interpretability of the proposed model despite the fact that the end-user —i.e. a traffic engineer or traffic manager— needs not to be an expert data scientist. In line with this, there is almost no probabilistic approach that deals with the task of automatic traffic state identification, and very few giving a probabilistic approach to the task of incident detection in spite of its good properties to be interpreted. Additionally, many of these incident detection methods usually are trained in an supervised approach with provided incident data, but this data is hard to obtain and it is known of its low reliability. Besides, incident detection methods for urban networks are not frequent.

Finally, it is very rare to find an integrated approach, such as we propose in this thesis, that deals with the real-time traffic forecasting in a self-adaptative manner, graph pattern mining, outlierness detection, and incident detection. We aim to fill in this gap.

4 Overview of the proposed solution

In this thesis, the aim has been first to identify the existing limitations in Aimsun concerning the real-time analytical prediction system for Aimsun Live, and then to propose an integrated system in order to solve or alleviate such shortcomings. The identified limitations in Aimsun Live can be summarized as follows:

- Fixed architecture of parametric modelling without taking into account the network traffic dynamics.
- Forecasting models were built offline in batch mode with no subsequent online learning.
- No detection nor reaction to changes —either in the supply or the demand— over time.
- No detection of anomalous patterns in the network traffic dynamics.
- No incident detection.
- Very limited handling of missing data.
- No automated process to decide which amount of data to use.
- Lack of robustness in real-time operation.
- Limited interpretability by traffic engineers and managers.

4.1 Modelling assumptions and motivation

4.1.1 Spatiotemporal correlations

The main assumption behind the adopted predictive modelling is to use the information from the current traffic network state in order to infer its evolution in the subsequent short-term time steps —namely from 5 minutes until 60 minutes ahead—. This assumption is motivated by the fact that traffic flow behaves like a stream of a fluid where it is conserved and propagated through the road network, likewise the macroscopic traffic flow theory does with the use of hydrodynamic theory of fluids. Certainly, this implies we ignore about the behaviour of individual vehicles and concern only with the behaviour of sizable aggregate of vehicles. As already mentioned, the network traffic state is described based on the traffic flows, either density or occupancy and the average time speeds

which are measured by detection devices —usually loop detectors— which are spread along the traffic roads in order to provide periodic measurements also in real-time.

In our case, this assumption leads to the search and exploitation of existing spatio-temporal correlations in the road network. One important difference between our proposal and a solution purely based on simulation is that, as we do not aim to provide so in-detail output information as macroscopic simulation would do, we can also relax some of the assumptions compliance as well as reduce the level of detail of the required knowledge, such as for the network geometry for instance. In this sense, one of the main motivations which has led the development of the proposed approach is to provide a robust solution which makes the most of the observable data.

More specifically, in a transportation road network there is a lot of shared information given that all roads are connected. This and the existance of entrance-exit points motivates the seek and calibration of the proper spatiotemporal correlations for an accurate forecasting. Moreover, it is frequent that not the whole road network is observable —i.e. there are not installed detection devices to measure the traffic—, so the temporal aspect of these correlations can become extremely useful in practice. On the other hand, these spatiotemporal correlations are dynamic as they are not just time-dependent, but also conditioned on the different movement patterns underlying the transportation system which responds to the existing traffic demand.

4.1.2 Mobility patterns

These mobility patterns —that we refer to as *rules*— are to be sought by exploiting the graph structure of the road network, following an evidence-based decision making procedure. Without a doubt, historical patterns are far from being stationary and they must evolve in the same manner traffic demand do because of changes in the needs or behaviour from the users of the transportation system. In this sense, the process must be performed online and it must be adaptable in order to react to changes.

Despite the model is constantly evolving to match the current traffic demand, we are using past seen traffic patterns coupled with real-time traffic information to predict the future. This means that it can happen at some point that the rule that matches the current qualitative conditions is not actually the best fit in terms of traffic pattern but still there would be another rule which fits better such traffic conditions —i.e. we refer to this as recurrent conditions—. At first, this situation may be handled through graph matching for a better rule selection. But in the end, the system should reorganize itself to better acommodate such traffic patterns and it will depend on how fast demand changes occur and the system's ability to detect them and adapt itself. On the other hand, there

could also happen that current rule do not match properly the current traffic conditions because there are non-recurrent events in the network —e.g. an incident, traffic rerouting by authorities…— that modify the usual drivers' route choices. This latter case would be an example of non-recurrent traffic conditions which are hard to handle by a predictive data-driven approach —beyond the ability to properly identify them— because of its implicit unpredictability, and is best suited for a simulation-based solution with different case scenarios which works paired with the proposed approach.

4.1.3 Non-stationarity and change

These situations can be viewed as part of a more broad adversarial machine learning [128]. This means that the presence of such non-stationarity must be taken into account in the learning process so the system must adapt to the gradual changes as it learns, as well as react on sudden shifts. Of course, any approach which does not fulfill with online learning is condemned to become outdated, which invalidates the forecasting utility. In this sense, we also believe that non-stationarity is a fundamental problem bounded to forecasting and empirical modelling and thus, the key is not just to choose the best model but to find those relationships that survive long enough to be useful [120]. For this reason, the core idea of using more data for learning is always best is half true, because learning with a significant proportion of outdated data may produce suboptimal results.

4.1.4 Non-parametric modelling

This is what has led us to place as minimum assumptions as possible on the data modelling and the reason why we have adopted a non-parametric approach where data relationships are found and controlled using evidence-based criteria in a online manner for an accurate adaptation to changes in the traffic demand or supply. We thus depart from to other common approaches in time-series forecasting such as parametric time-series approaches which rely on a proper beforehand definition of trend and seasonality components during the modelling stage. In addition, such time-series forecasting models are especially vulnerable to concept drift because as previously said the underlying generating process of the time series observations may change, making forecasting models obsolete and, furthermore, making challenging to identify long seasonality components when new data is limited.

Above all and especially in statistical modelling, the assumptions placed by the analyst during the modelling stage determine which type of conclusions may be extracted and how significant are they. On the other hand, within the machine learning culture this point is not so crucial as

the peformance of the forecasting task. Often, this is seen as two different cultures within the statistical modelling field [43, 175]. In this sense, as aforementioned, we have decided not to place very strong assumptions in the data model and thus, approaching the problem from a machine learning perspective.

4.1.5 Interpretable reasoning

However, although treating the problem with a black-box model is common in many machine learning techniques, we have endeavoured to not sacrifice the human interpretability of the model, especially regarding end users such as traffic engineers or managers.

We assume that aforementioned rules are expressed using qualitative variables as it can ease the interpretation for end-users —e.g. traffic engineers or managers— as well as allowing for easier diagnostic, unlike other black-box modelling techniques in the statistical and machine learning fields. In addition, other parts of the thesis rely on probabilistic models as they are more human-interpretable and easier to diagnose.

Besides the aforementioned usual nonstationarity behaviour found within the process of traffic modelling, there exists other underlying statistical characteristics bounded to it. For instance, traffic is highly nonlinear in relation to the roads' capacity and how traffic congestion is formed, and the process has a high degree of multicollinearity among the road network's locations. Heteroscedasticity is also present because of different existing variabilities due to e.g. measurement errors. The proposals within this thesis aim to deal with them in a efficient manner using techniques adopted from the machine learning field.

In essence, for the real-time traffic prediction task we are seeking the best predictive spatiotemporal correlations within the proper identified context —*rule*—. Nevertheless, despite the fact that traffic prediction has been the driving motivation for the thesis development, the proposed ideas and solutions in this thesis are generic enough to be applied in any other problem where, ideally, their definition is that of the flow of information in a graph-like structure with special interest in environments susceptible to changes in the underlying data generation process. Moreover, the modular architecture of the proposed solution facilitates the adoption of small changes to the components that allow it to be adapted to a broader range of problems.

4.2 Modelling goals

The modelling goal is to build a set of rules in an online manner —*adaptative rules*— for the forecasting task in a graph-like structure. These rules are to be learned and managed autonomously following an evidence-based criteria with incoming streaming data in a non-stationary environment that evolves over time.

The main modelling goals are:

- The model complexity grows to adapt to that of the modelled process —non-parametric approach—, meaning that the number of rules is not fixed beforehand. This certainly requires control and regularization mechanisms in order to keep generalization and avoid overfitting.
- Adaptation to change in an online learning scenario. Both from a point of view of gradual changes as well as sudden changes through concept drift detection.
- Minimum number of assumptions in order to delegate in data the finding of relationships following an evidence-based criteria. This way, it would be possible to start modelling the problem from a point where the collected historical data is still scarce or the structural information about the graph is diffuse. This is motivated by the inherent volatility in the traffic demand an supply characteristics, thus an empirical approach is followed which reduces the uncertainty and the forecasting error.
- Autonomy. It is very important for a real-time system to have the ability to autonomously make decisions and being able to self-calibrate with new streaming data. This let the system to be more reactive and efficient about the usage of data as it free the end users from deciding which data size is more appropiate and how often a maintenance must be scheduled to build again the models with new data.
- Interpretability. The end-user of the system —e.g. a traffic engineer or traffic manager— does not have to be an expert data scientist to be able to interpret the output as well as have a high level interpretation of how the system works and what is the reasoning behind it. This is what has motivated our modelling decisions instead of selecting other popular techniques within the machine learning field, such as deep learning, as the interpretation of their internal workings is more black box.
- Robustness when facing outliers and missing data, which is rather common in real-time noisy traffic data.
- Prone to be scalable by the design of a modular architecture with emphasis on a parallelizable exploitation of large amounts of data.

Lastly, given that we are using a data-driven approach for real-time traffic forecasting instead of

a purely traffic simulation approach, we consider an accurate forecast is as important as a prompt warning of anomalous traffic conditions —e.g. non-recurrent events or incidents— which may lead to an unsatisfactory forecast accuracy but their identification constitute a great value both to traffic managers as well as an input for simulation-based traffic prediction systems.

4.3 Architecture of the proposed solution

The system architecture proposed in this thesis, which is shown in Figure 4.1, aims to deal with these problems. This architecture is thought to be modular but working as a whole for the purpose of real-time traffic forecasting.

As shown in Figure 4.1, the proposed system architecture is composed of several components for different responsibilities. First, previous to the real-time operation mode, the system must receive a list of inputs during an offline configuration stage. This list of inputs includes:

- The graph of the network which is being modelled —e.g. a city, an inter-urban network, etc.— which ideally derives from the work done by transport modellers and traffic engineers at Aimsun that build the Aimsun model of the network, otherwise it should be derived from external services such as the free project *OpenStreetMap* [197] along with a post-process of map-matching.
- The second input is the set of prior assumptions that the data analyst or ideally the traffic engineer has thought might be relevant for the specific network or city. These are expressed in the form of qualitative factors —e.g. the weekdays, time, weather or special days calendar, etc.— and they do not need to be exhaustive as the system is thought to find relevant patterns associated with such factors and their interactions. This means that such prior knowledge can be vague or very informative according to the expert's experience and knowledge.
- The last input is optional and consists of the available historical traffic data. It allows the system to start its real-time forecasting tasks from scratch without considering previously collected traffic data, thus starting to learn the data relations and relevant patterns with no accumulated experience.

During the real-time operation mode, the system continuously receives streams of data that are used for real-time forecasting tasks, and also for improving the experience of the system by learning from this data probably with certain periodicity —e.g. once every midnight or weekly—. This streaming data is used to feed different components of the system, namely the main one corresponds to *Adarules* whose main responsibility is that of performing the real-time forecasting and data mining. Adarules seeks to unveil recurrent patterns from data in the network graph, but also to check for

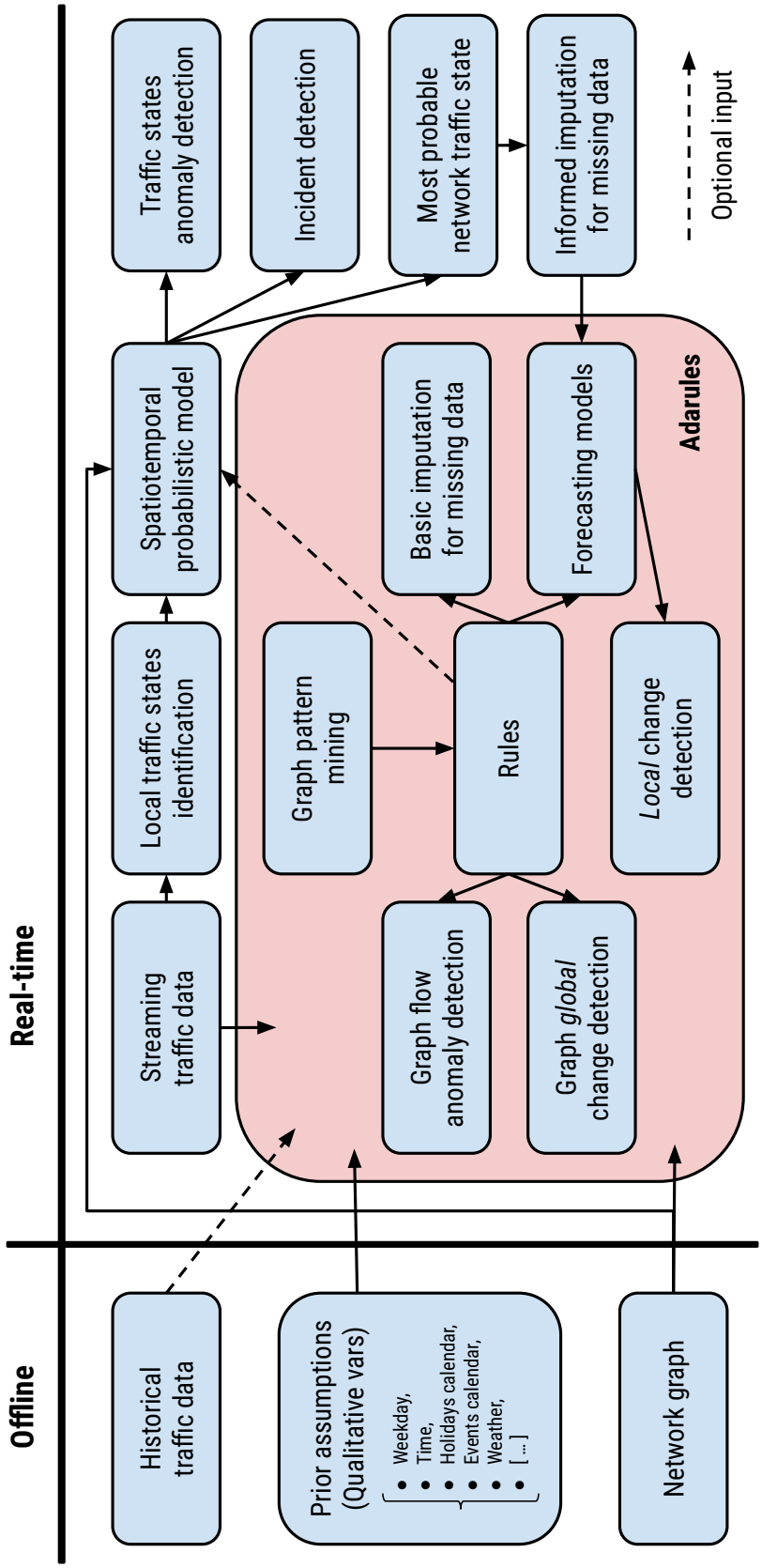


Figure 4.1: Functional architecture and workflow of the proposed system.

the on-going validity, updating, and removal of these patterns over time. On top of this process of mining graph patterns, it is built the processes of detecting anomalies of flow in these graph patterns, and the detection of changes in these patterns —i.e. global changes in the scope of the network— in order to react by giving proper warnings or changing the patterns' structure or even removing them. These patterns can also be used for basic imputation of missing data as they are associated with recurrent traffic conditions. Forecasting models of traffic conditions are built according to the process of pattern mining, so they can take advantage of such recurrent condition which is a standard procedure in machine learning and data mining. The detection of local changes —i.e. not affecting the scope of the complete network— are detected by this set of forecasting models. Streaming data also feeds the process of local traffic states identification for all the network which, in addition, serves as input for the spatiotemporal probabilistic model. This component is responsible for detecting anomalies in the traffic states, performing incident detection, and estimating the most probable network traffic state which can be also useful for an informed imputation of missing data.

The first point concerning the fixed structure of the modelling assumptions implied that the same assumptions and modelling structure was being applied for every new project and network, and thus disregarding the underlying traffic dynamics for each network. More specifically, the criteria was based on building a forecasting model for every location with detection data, every forecasting horizon and every specific time of the day. This leads to a large number of forecasting models entailing a considerable computational cost. Furthermore, besides the fact of ignoring the specific traffic network dynamics, there were additional issues related to the sample size of the dataset used to calibrate the models which could lead to suboptimal results. This is because as every model has only one observation per day, the learning process could find spurious correlations in such small dataset. For these reasons, in this thesis a non-parametric approach is proposed —*Adarules*— which uses certain prior knowledge in the form of qualitative factors put by the data analyst or the traffic engineer in order to automatically find patterns in the network graph. This means that for every network the system will unveil automatically these graph patterns from data according to the underlying traffic dynamics. This performs the modelling in a more realistic way. Its non-parametric nature allows to accommodate the complexity as it is needed, which means that networks with more complex dynamics will have, as expected, more patterns to be extracted in order to characterise the underlying demand properly. This also entail a more efficient usage of data as it is segmented per pattern and there is no need to further split the data, thus it allows models to be fit using more data granting them to find proper dependences and correlations in data and reducing the risk of spurious correlations. Computational efficiency is also improved as it is now tied to the number of found pattern in the network.

In addition, the previous large list of forecasting models were built in offline with all the available historical data without assesing for changes in the data which could make that some of that data were outdated already. The training of the models was performed in batch mode, and then they were not updated until the next maintenance, thus ignoring any online learning or detection of possible changes in the travel demand or the supply of the network. On the other hand, Adarules has automated methods to detect and react to changes both at a global scale in regard to the found graph patterns in the network, as well as at a local scale concerning the spatiotemporal correlations within the network. It also means that online learning is performed in the system both when new patterns are found, when change is detected in some of them, or periodically with mini-batches of data —incremental learning— to take into account smooth gradual changes. This way, either changes in the underlying traffic demand or in the network supply are properly detected making the system responsive.

An important issue is also the lack of anomalous traffic pattern detection. This is particularly important in order to be able to anticipate unreliable traffic forecasts given certain anomalous traffic conditions. Therefore, before there was no more option than waiting until observing how well was actually the forecasts performing. This was risky because those traffic forecasts were used to feed subsequent processes such as dynamic adjustments of the base travel demand for simulation, without any clue about their reliability. Additionally, this had also consequences on the trustworthiness of the forecasts by traffic managers. In the proposed system, it is possible to measure how anomalous are the current dynamics of the traffic network both in terms of the current matched graph pattern by Adarules which is based on the degree of anomaly of the flows through the graph, and in terms of the probabilities of the current traffic state measured by the spatiotemporal probabilistic model.

In the same way, there was not any implemented method for incident detection —which can be seen as a particular form of anomalous traffic pattern—. Besides the obvious shortcoming of not having such useful information for traffic managers, it could have consequences on the reliability of future forecasts and their utilization for other processes. The proposed spatiotemporal probabilistic model based on traffic states is able to calculate probabilities on the detection of traffic incidents, as well as calculating a normalized score which is based on their severity, temporal persistence, and spatial propagation.

Missing data, which is quite usual in traffic data and during real-time operations, was not handled at all. This fact was very harmful to the online performance of the system and only very basic and vague patterns —such as one pattern per weekday— were used to replace missing data in order to feed the forecasting models which were not able to deal with such missing data in a natural

way. For this purpose, there are two proposals which aim to deal with such problem. The first is a fairly basic method based on the replacement by the average value in the context of every graph pattern, as these already represent recurrent traffic conditions. The second one is based on the spatiotemporal probabilistic model in order to calculate the current most probable network traffic state which can serve as a method for a more informative imputation of missing data.

In regard to the available historical traffic data before the beginning of each Aimsun Live project, there was always a controversial decision about the amount of data to use in order to feed the process of training the forecasting models. Apparently, one might think that the more data the better but in some cases, this could lead to learning from outdated patterns. In addition, for some projects the amount of available collected traffic data was scarce and, thus, it was hard to make certain assumptions such as seasonality for instance. In any case, it was a time-consuming task for data analysts and a hard decision to take for traffic engineers. In the current proposal, this process of deciding what data to use is automatically tied to the process of patterns mining on the network graph which is already based on the observed data. In the same way, as aforementioned, there are methods in Adarules to also detect outdated patterns.

During the real-time operation, it is very common that a considerable part of the traffic measurement devices become temporally faulty or provide unreliable or noisy measurements. Although this is also related to how missing data is handled, it is also important to include methods that take into account the reliability of such data sources in order to avoid using them for the forecasting task as they are unreliable. This could lead, for example, that some forecasting models simply take a smaller amount of inputs but with a higher degree of reliability in order to become more robust. This fact was simply ignored in the past, and forecasting models could take any number of inputs —typically a large number— regardless of their reliability. Now, this fact is considered and integrated as part of the fit of the forecasting models.

Lastly, an important consideration that has guided part of the decisions taken in this thesis is the degree of interpretability of the whole system. Historically, it has been always a controversial issue of how interpretable machine learning methods are, especially nowadays with the reappearance of models based on neural-networks such as deep learning. In the past, there was limited interpretability in the analytical forecasting process in Aimsun Live, as there was a lot of forecasting models where each one, in addition, could have a large number of dependencies making difficult to diagnose the forecasting decisions. In this regard, the proposal has relied on methods and models whose output has a higher degree of interpretability, aimed mainly for traffic engineers and managers. For example, the pattern mining in the network graph is tied to qualitative factors which are placed as prior assumptions by them —traffic engineers and managers— and thus, they can

evaluate how much sense these identified rules have. Forecasting models have been thought to include a fewer number of dependencies but with stronger predictive correlations. Additionally, the traffic states and the spatiotemporal probabilistic model built on top of them, are models which allow the diagnose and interpretation of its decisions, as well as a more natural output based on probabilities.

4.4 Datasets used for this thesis

For the validation of the different methods proposed in this thesis, two datasets with different characteristics have been used. The first corresponds to the M4 and M7 motorways in Sydney, Australia. The dataset is provided by the New South Wales Government - Roads and Maritime Services. Real macroscopic traffic flow data whose time span is two years, corresponding to periods January, 2015 – December, 2016, has been used in order to include the impact of seasonality and all public holidays. The network consists of 455 double-loop detectors spread uniformly at every 500 metres, as presented in Figure 4.2, measuring traffic flow, occupancy and speed. This type of network —highway— is usually easier to forecast because traffic is usually more homogeneus at a network-level, as well as for the lack of traffic disruptions and the less occurence of non-recurring congestion and events. However, there is an additional challenge present in this specific network which is tied to the position of boundary detectors, i.e. those on-ramp and off-ramp detectors which serve as entrances and exits to the motorways respectively, because of the lack of observability beyond these sites.

The second dataset corresponds to the urban network of Santander City, Spain. The data is provided by the Santander City Council, and the data is publicly accessible through its open data portal [67]. In this case, a similar period of two years corresponding to January, 2016 – December, 2017 has been used. The urban network is composed of 4106 links which are measured with 489 single-loop detectors, as shown in Figure 4.3, observing traffic flow and occupancy without ability to observe the vehicles' speed. The urban topology of network is naturally more complicated to forecast because of its mesh connectivity, as well as the higher variability in data associated with traffic flow disruptions —such as traffic signals and lights, higher number of incidents, among others— and the more complex knowledge about drivers' routing decisions especially during special days or non-recurrent events and congestion. Furthermore, this dataset exhibits several challenging but interesting problems for the tests of non-stationarity and adaptation to change. Figure 4.5 shows the overall normalized flow aggregated over all the detectors for each traffic network, showing the differences in the temporal dynamic of traffic as well as the different variability associated with the

higher complexity in urban environments. In both cases, the frequency of data sampling is $\Delta t = 15$ minutes and these macroscopic measurements are already aggregated to the level of multi-lane stations, i.e. each of the data measurements are aggregated for the different road lanes at a specific detection site.

The choice of $\Delta t = 15$ minutes as data sampling time for the learning and validation stages of the different experimental scenarios in this thesis is made for several reasons:

1. Mitigating the inherent noise in road network measuring devices,
2. Reducing the running time for the experiments in the current research work without compromising the validity of the results,
3. Convenience for commercial purposes from Aimsun SLU and its product Aimsun Live,
4. Furthermore, when the system learning is done with a frequency of sampling time $\Delta t = 15$ minutes, it means that observed variables are also associated with a measurement window size $\Delta W = 15$ minutes. Then, whenever real-time requirements need to update the forecasts in a shorter time span —e.g. $\Delta t = 5$ minutes—, the same model —whose learning was performed using $\Delta t = 15$— is valid as long as the observed variables are processed to be consistent with the same data distribution as used during the learning. This implies to keep the same measurement sliding window size $\Delta W = 15$ minutes for that new shorter $\Delta t = 5$ frequency of data sampling.

According to the theoretical fundamental relations of traffic flow described in Chapter 2 and Chapter 3, these can be also observed in the data used for this thesis. As an example, Figure 4.4 shows the fundamental relations between traffic flow Q against occupancy O, traffic flow Q against speed V, and speed V against occupancy O, respectively and only using data from a specific detection site for each network. In the case of Santander, only the fundamental relation between traffic flow and occupancy is shown because of the inability of single-loop detectors to observe vehicles' speed.

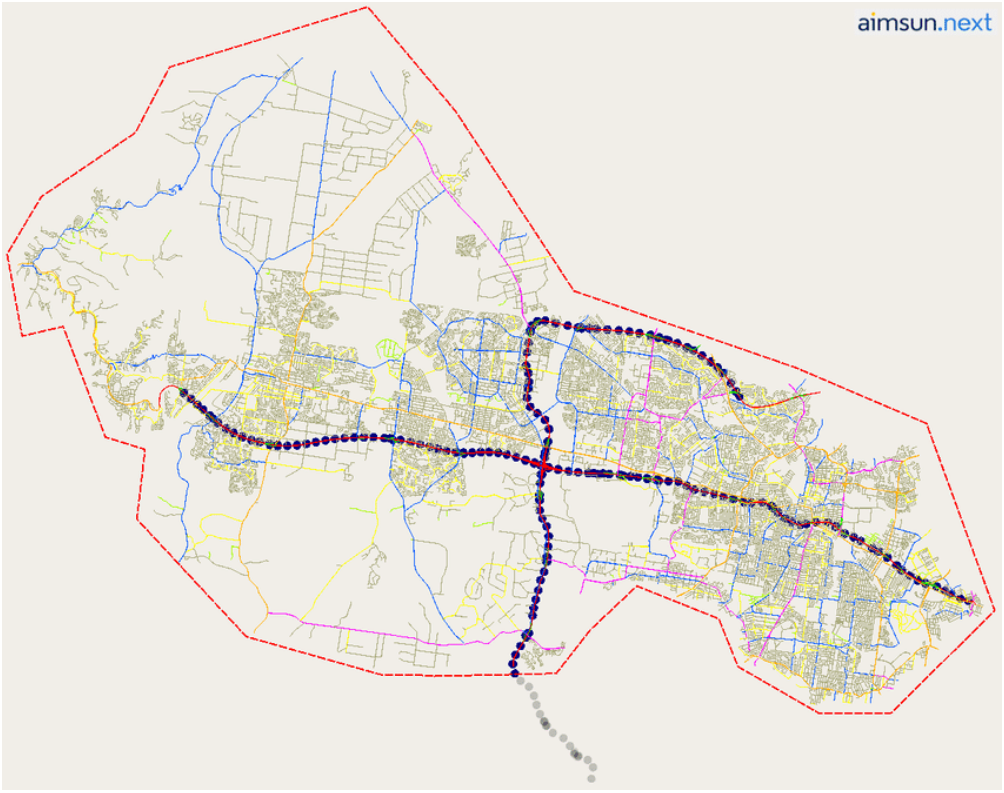


Figure 4.2: M4 (46-kilometre-long) and M7 (41-kilometre-long) motorways in Sydney, where loop-detectors are presented as dark blue dots.

Figure 4.3: Location of loop-detectors (red dots) and signalized intersections (yellow dots) in Santander (36 km^2, 4106 links).

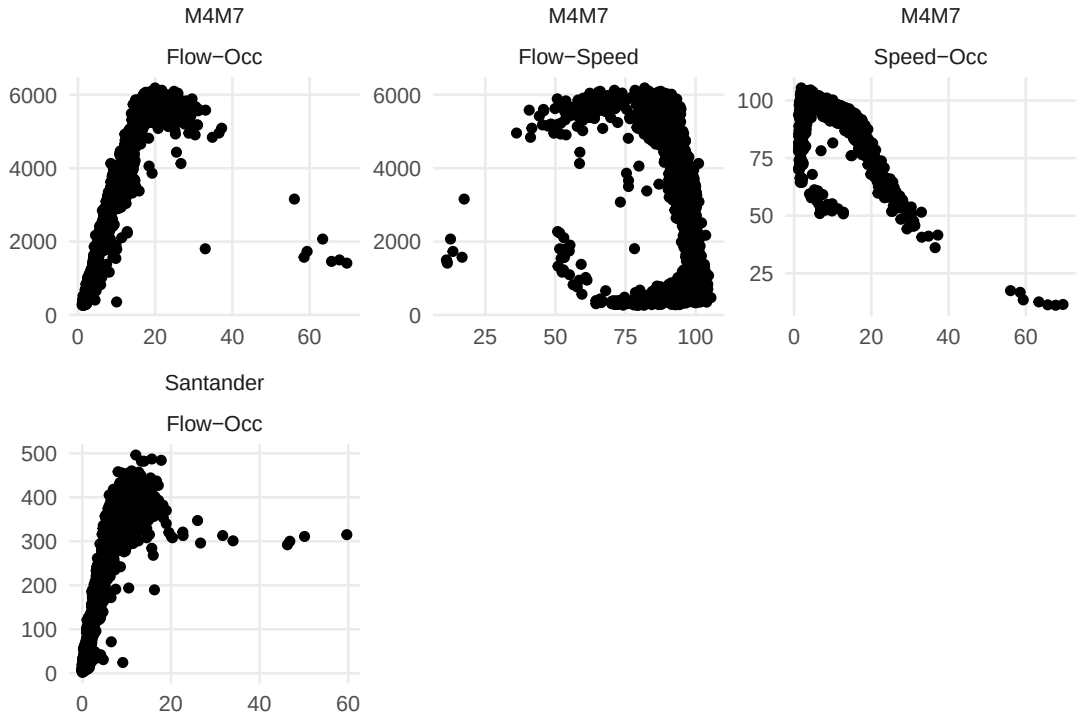


Figure 4.4: Some plots showing the pairwise relation in macroscopic flow data as stated in the fundamental diagram of traffic flow. Each left-side variable corresponds to the y-axis, while right-side variables correspond to the x-axis. Traffic flow is shown in vehicles per hour, occupancy is shown as a percentage and speed is shown in miles per hour.

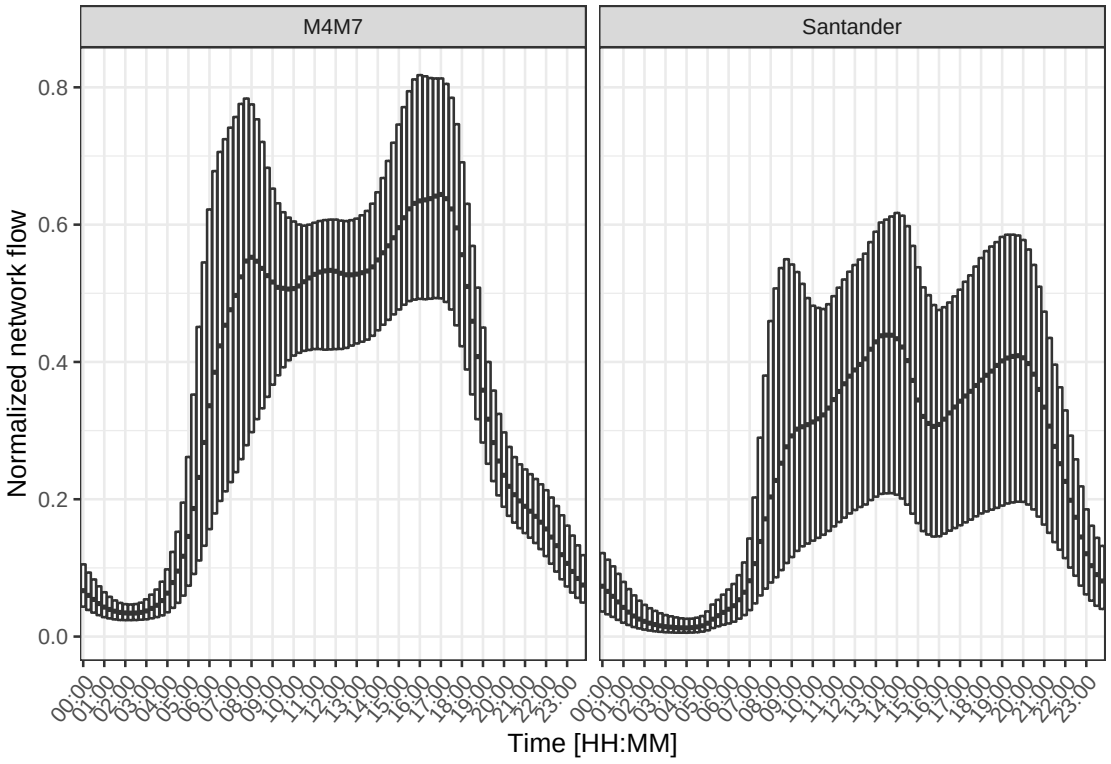


Figure 4.5: Network flow normalized for the Santander and M4M7 networks. In every time step of $\Delta t = 15$ minutes, the box shows the interquartile range —25% to 75%— along with the median —50%— as the horizontal line within the box.

5 The Adarules algorithm: towards a non-parametric approach

Adarules [181] —whose name derives from *Adaptive rules*— is a framework for predictive modelling based on a multi-level decision-tree structure which is built online using streaming data. It is composed of rules —in the form: *if antecedent(s) is satisfied, then […]*— that corresponds to recurrent conditions in the modelled problem —i.e. patterns— and they are automatically extracted and maintained in an autonomous manner from such streaming data within a non-stationary system which can change over time.

These rules —that in the task of traffic forecasting can be seen as mobility patterns in the network— are to be learned and managed autonomously by exploiting the associations and recurrent conditions within the problem —in the case of traffic, it is the graph structure of the road network—, following an evidence-based decision making procedure. However in the real world, historical patterns are far from being stationary and they must evolve and being updated over time. For instance in traffic, these patterns must evolve in the same manner traffic demand do because of changes in the needs or behaviour from the users of the transportation system. Therefore, the system is thought to perform in a real-time operation mode with the ability to be adaptable in order to react to changes.

In addition to the automated mining of patterns over time, one of the main features in Adarules for the forecasting ability is the proper identification of spatiotemporal correlations under the specific context or rule. In the case of traffic forecasting, the fact that in a transportation road network there is a lot of shared information given that all roads are connected and the existance of entrance-exit points motivates the seek and calibration of the proper spatiotemporal correlations for an accurate forecasting. Moreover, given that it is frequent that not the whole road network is observable —i.e. there are not installed detection devices to measure the traffic—, the temporal aspect of these correlations can become extremely useful in practice. On the other hand, these spatiotemporal correlations are dynamic as they are not just time-dependent, but they are also conditioned on the different movement patterns underlying the transportation system which responds to the existing traffic demand.

We have developed Adarules so that it complies the goals listed in Section 4.2. Namely:

- **Adopting a non-parametric approach**

  This implies adapting the modelling complexity as the complexity of the problem grows, thus the number of rules is not fixed beforehand and it can increase or decrease over time. This makes the approach well-behaved for the streaming scenario where the size of the data is theoretically unbounded.

  It also makes an implicit efficient usage of the sample size for fitting the forecasting models and automates the process of finding relevant patterns for every new problem being modelled.

- **Adaptation to change**

  Adaptation to changes in an online learning scenario. Both from a point of view of gradual changes as well as sudden changes through concept drift and shift detection. This adaptation can also be considered from the point of view of the scope of the change, either if it is a global change whose impact is significant on a wide part of the problem or it is local whose impact is solely on specific forecasting models. The handling of such changes is also different as different reaction methods are integrated such as gradual forgetting and rearrangement of the rules.

- **Reducing the number of assumptions**

  Relying on data in order to find relationships following an evidence-based criteria. This way, it would be possible to start modelling a problem even when the collected historical data is scarce.

- **Autonomy**

  It is very important for a real-time system to have the ability to autonomously make decisions and being able to self-calibrate with new streaming data. This let the system to be more reactive and efficient about the usage of data as it frees the end users from deciding which data size is more appropiate and how often a maintenance must be scheduled to build again the models with new data.

- **Interpretability**

  The end-user of the system —e.g. a traffic engineers or traffic managers— does not have to be an expert data analyst to be able to interpret the output as well as have a high level interpretation of how the system works and what is the reasoning behind it. This is what

has motivated our modelling decisions instead of selecting other popular techniques within the machine learning field as the interpretation of their internal workings is more black box.

Even though Adarules is able to find rules associated with any kind of variable —either discrete or continuous—, during its application in this thesis it has been decided to rely only on qualitative variables as it can ease the interpretation for end-users as well as it would allow an easier diagnostic.

- **Scalable**

  The partition of the data into the different rules makes the solving of the problem more scalable by exploiting its parallelization capability. A great effort has been made to achieve an efficient implementation that exploits the data in parallel at the level of rule and prediction model. Moreover, the implementation relies heavily on matrix calculus to speed up the computations.

  In addition, as working with large amounts of data is challenging because most of the time it doesn't into the computer's main memory, a great effort has also been put on processing chunks of data instead of the entire dataset at once. Thus, the system implementation is ready to deal with theoretical unbounded data or *big data*.

- **Robustness dealing with outliers and missing data**

  Given that the data-driven modelling proposal is intended for real-time forecasting instead of a pure simulation approach, we consider an accurate forecast is as important as a prompt warning of anomalous pattern conditions —i.e. non-recurrent conditions— which may lead to unsatisfactory forecast accuracy. Therefore, their identification constitutes a great value both as an output itself, but also when it is used as an input for other processes.

- **Modular software architecture**

  Therefore, it makes easier to replace certain components or methods of the framework according to the needs.

5.1 Methodology

The foundations of Adarules lay on an automatic knowledge discovery through rule identification. The fundamental components of Adarules are:

- There is one ruleset $\mathcal{R}$ for each of the modelling tasks. These tasks could be for instance a single measurement station in the network, a group of links or the whole network. The structure of such ruleset corresponds to a decision tree.
- Every ruleset $\mathcal{R}$ contains multiple rules. A rule R is analogous to a specific pattern found in data —e.g. certain similar traffic conditions—. A rule corresponds to a leaf node within the aforementioned decision tree.
- Every rule R has the form *Antecedent* $A \Rightarrow$ *Consequent* C.
- A literal L is a single condition over a specific attribute x_i with a specific split-point v: with the form $(x_i > v)$, $(x_i \leq v)$ if x_i is numerical, or $(x_i = v)$ if x_i is categorical data. $L(x_i)$ returns *True* if x_i satisfies L, and *False* otherwise.
- The antecedent A is composed from multiple literals L; the antecedent as a whole is evaluated as *True* or *False*. A rule R is said to cover an example or observation x if all of its literals are satisfied. Every literal from the series within an antecedent of a rule is extracted from every node in the decision tree on its path from the root node to the leaf node of that rule.
- The consequent C of a rule R may have multiple forms (constant value, summary statistic or a more complex function). It is built from the examples gathered in the scope of R.

The Adarules system has already been published in the scientific literature [181]. However, the final implementation used in this thesis results differ in certain key points with respect to the published in [181]. The main differences are:

- In [181] both contextual qualitative variables, as well as traffic continuous variables —flow, occupancy, and speed— from the road network, were considered as candidates to be evaluated during the splitting procedure. In this thesis, only contextual qualitative variables are considered during the splitting procedure for the reasons given in Section 5.1.1.
- The Adarules underlying data structure in [181] was a boosting of ternary decision trees —missing values were also considered as a third split in every rule—. This implies multiple rules could cover a given observation, then weighting their outputs according to their respective forecasting error. Conversely, in this thesis a single decision tree is built for a given modelling task —either a single detector or the entire road network as will be later described—, and thus, only a single rule is triggered for a given observation.
- The split function used during the pattern mining process in [181] was based on entropy minimization of the outcome variable. In this thesis, a scoring function based on the underlying graph structure of the problem is used.

Adarules design follows a modular architecture, as shown in Figure 5.1, making easier to replace the proposed algorithms within the different components according to different needs in other problems.

In the following, the main components within the Adarules architecture are described in detail.

5.1.1 Pattern mining

Rules are analogous to high-level features or underlying patterns in the problem at hand, e.g. a road network. Therefore, they correspond to a certain behaviour that is repeated with enough frequency under specific conditions, which is known as recurrent conditions in traffic. The initial hypothesis h_0 is that no pattern exists, which is equivalent to let Adarules start from scratch with only the root node R_0. From then on, every rule has a chance periodically to run an expansion evaluation process. If the evaluation process is favorable, that rule disappears and it is specialized into two new rules with the respective observations and statistics. Every expansion is dependent on a specific attribute and split-point —value or set of values— and hence the resulting number of two leaf nodes after the expansion. This leads to a binary decision tree as the underlying data structure which represents the patterns where every leaf corresponds to a rule. Nevertheless, if every split-point considers the special case of *missing value* as a third split, then the structure happens to be a ternary decision tree. However, although possible and developed within Adarules, this scenario has not been considered in this thesis as only qualitative features with always observed values are the only ones that have been considered as candidates for the expansion process.

The frequency of this evaluation, which takes place for each rule separately, is crucial as a low frequency can lead to slow learning and finding of the patterns while a high frequency can make the process too sensitive to transient noise. The parameter $N\text{min}$ dictates the minimum amount of observations which must be seen, separately on each rule scope, to proceed with an evaluation for rule expansion. This threshold $N\text{min}$ is preset to an initial value $N\text{min}_0 = 100$, that is later dynamically adjusted. The motivation behind the dynamical adjustment of $N\text{min}$ is to initially perform frequent evaluations for rule expansion, and then decrease the frequency when a rule is stable in order to avoid an unnecessary computational burden. Therefore, every time a rule undergoes a change of state —i.e. it has been created—, its $N\text{min}$ value is set back to $N\text{min}_0$ for a frequent pattern mining. On the other hand, every time a rule fails in the process of being expanded —i.e. no more specific patterns could be found with observed data so far—, its current $N\text{min}$ value is multiplied by a certain decay factor $Nmin_\gamma = 2$ which results in slowing the expansion attempts in order to wait until collecting further data.

The rule expansion evaluation process seeks to determine which is the attribute and split-point combination that best perform by evaluating with a specific scoring function on the examples seen so far. Once chosen, it will become a new literal that will tell apart the two new rules. A left-side rule with the form $(x_i \leq v)$ and a right-side rule with the form $(x_i > v)$ if x_i is a numerical or

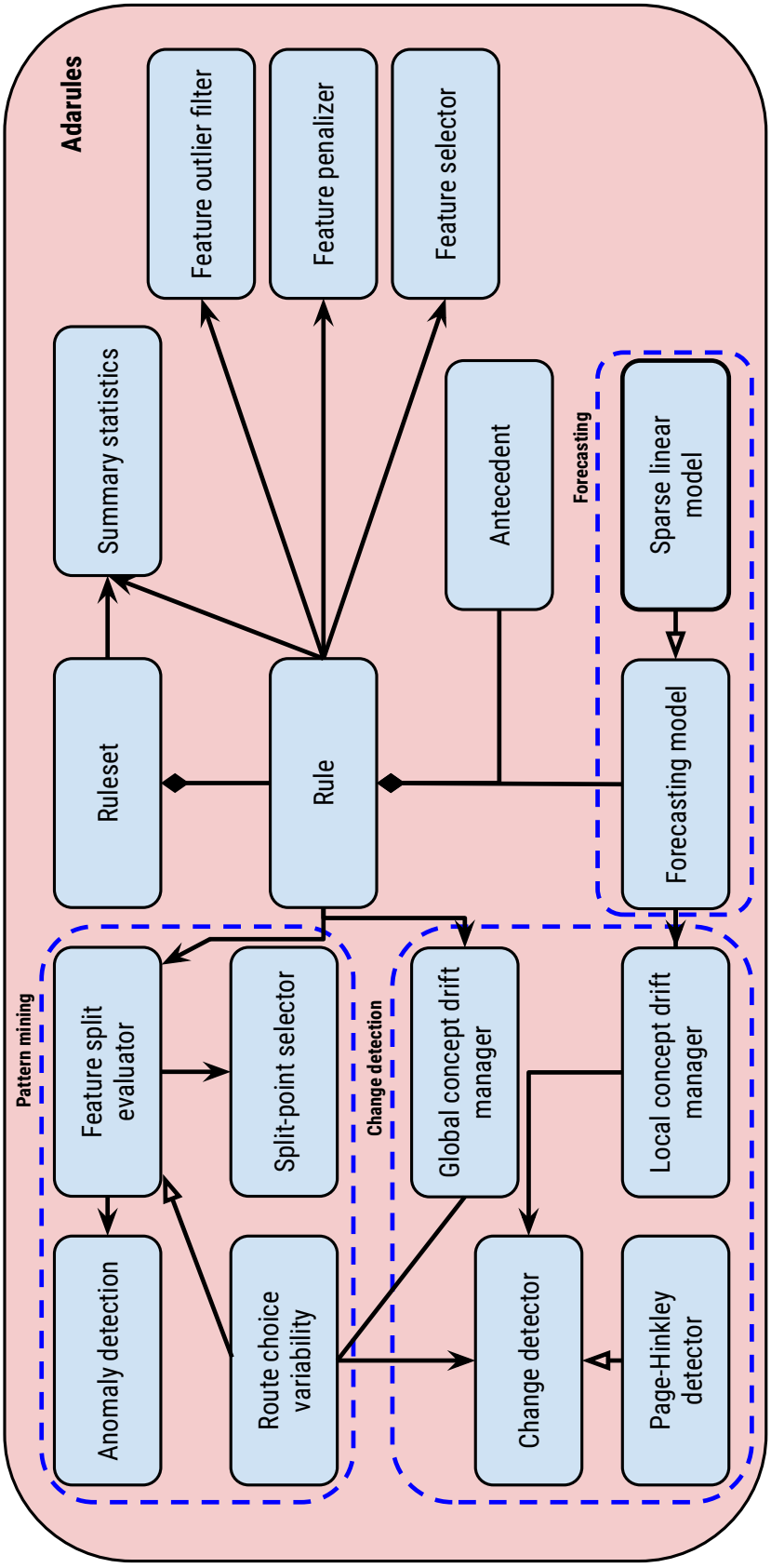


Figure 5.1: Modular architecture of Adarules. Main classes and their relationships are shown.

time-based attribute, whereas ($x_i \in v$) and ($x_i \notin v$), respectively, when x_i is a qualitative attribute. This binary splitting could be also turned out into a ternary splitting by considering the missing value state as the third child node.

An important clarification about the type of variables used in the process of pattern mining. Even though Adarules is able to find rules associated with any kind of variable —either discrete or continuous—, during its application in this thesis it has been decided to rely only on qualitative variables which are contextual (weekday, hour of the day, holidays calendar, season, timestamp). The reasons for such decision are:

1. To potentially ease the interpretation for end-users as well as to allow an easier diagnostic for the identified rules.
2. A modelling choice to isolate changes in the traffic demand-supply relationship from the rules' antecedents. This is intended to separate responsibilities between the modules for evaluating feature splits and change detection. An example of this could be: there is an antecedent relying on the flow for a given detector A > a, but then the road where such detector is placed changes. Then, immediately the definition of this rule would become invalid.
3. Reducing the number of splitting combinations to be considered also obviously reduces the computational cost.

5.1.1.1 Split-point selector

The only duty for this component is to generate a set of candidate split-points values for each of the input attributes acting as candidates to split on. These split-points can be a unique value or a set of values, and they will form the attribute and split-points combinations to be evaluated during the process of node expansion.

More specifically, the split-point selector used during the validation experiments for this thesis has followed a simple approach. For each of the selected continuous attributes including the timeline attribute such as *timestamp*, the split-points are those single values statistically selected by the cumulative probabilities —i.e. quantile functions— of the attribute with the aim of representing its full distribution. While in the case of discrete attributes the selection is based on the generation of continuous sub-intervals using their factor levels.

5.1.1.2 Scoring function

The scoring function is responsible for giving a score to every combination of attribute and split-point. Thus, after evaluating all candidate combinations of attributes and split-points, the one

with the highest score is selected to perform the node expansion in the decision tree which defines the rules.

The scoring function used in this thesis is motivated by the directed graph-structure of the problem at hand and the passing of information between nodes within the topological space. The assumption is that there underlie patterns in the flow of information across the graph. In the case of a road network, every node in the graph corresponds to a specific spatial point in the network —usually, a point with traffic measurements provided by a sensor—, and the edges between nodes correspond to the underlying network geometry connecting roads each other. This definition and discretization of the space naturally match well with the common procedure of measuring traffic in roads, i.e. measurements from inductive loop detectors placed over the road network like the ones used within this thesis.

Then, given nodes[1] v_1, v_2 from a graph G and an edge e connecting both nodes $v_1 \rightarrow v_2$ that represents the directed flow of information from node v_1 to node v_2 as shown in Figure 5.2, it is assumed that there exists a probability distribution that describe such information flow. Therefore, in this way it is possible to measure not only the expected value but also the uncertainty around the information flow among nodes. However, it is expected that such uncertainty —or variability— is going to be reduced under certain recurrent conditions which respond for instance to mobility patterns in the network. The goal, thus, is to perform a proper identification of the underlying flow patterns between both nodes according to their recurrence under certain conditions. The characterization of such probability distribution is given by the differentiation of v_2 minus its upstream node v_1, i.e. in the case of the road network it corresponds to subtracting the vehicle flows $v_2 - v_1$ at a given time interval. The choice of the subtraction operation to define this probability distribution over the information flow instead of other reasonable choices such as the division operation corresponds to (1) numerical stability and (2) proper differentiation not only by relative differences but also by the magnitude of the flows of information.

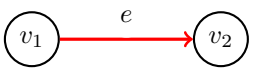


Figure 5.2: Basic graph example where v_1 and v_2 nodes correspond to specific points within the discretized space of the road network, and it is assumed there is a probability distribution over the directed flow of information represented by e.

For every evaluation of the combinations of attribute and split-point, the goal is to determine if there is a relevant pattern in the resulting probability distribution over information flow between two nodes after performing the split. To this end, a statistical test performs a comparison between

[1]For a matter of clarity, let us remind that *node* can refer to those R within the decision tree defining the rule set in Adarules, or a node v from the graph within the topological space of the network.

the existing probability distribution and the resulting one after the split in order to check for statistical significant differences by means of the two-sample Kolmogorov-Smirnov (K-S) test [176, 226]. The K-S test is a nonparametric method for comparing two samples, being sensitive to differences in both location and shape of the empirical cumulative distribution functions of the two samples. Its simplicity is also an advantage as it can be also used for measuring the distance D^* between two probability distributions in an efficient manner, especially if compared to other distances for probability distributions such as e.g. the Wasserstein distance. More formally, the two-sample K-S test evaluates the difference between the empirical distribution functions (CDFs) of the distributions of the two sample data vectors over the complete range in each data set, performing the maximum absolute difference between the CDFs of these distributions which is defined as the Kolmogorov–Smirnov statistic D^*:

$$D^* = \sup_x |\hat{F}_1(x) - \hat{F}_2(x)|$$

where $\hat{F}_1(x)$ and $\hat{F}_2(x)$ are the CDFs of the first and the second sample respectively, and sup is the supremum function corresponding to the maximum value within a vector. Thus the D^* statistic is bounded in $[0, 1]$. Another advantage of the K-S test is that, being a non-parametric test, it relies on the CDFs and makes no assumptions about the underlying true distributions of $\hat{F}_1(x)$ and $\hat{F}_2(x)$ which might be unknown.

This K-S statistic D^* can be employed to assess if the null hypothesis —that the two samples are drawn from the same distribution— can be rejected. More specifically, the null hypothesis H_0 is rejected at a certain level of significance α if:

$$D^* > c(\alpha)\sqrt{\left(\frac{n_1 + n_2}{n_1 n_2}\right)}$$

where n_1 and n_2 are the sizes of the first and second sample respectively. The value of $c(\alpha)$ is given by:

$$c(\alpha) = \sqrt{-\frac{1}{2}\ln\alpha}$$

Typically a level of significance $\alpha = 0.05$ is used. This way, it can be assessed in a rigorous manner if both distributions —before and after the split— are significantly different. Nevertheless, when the sample size is large enough, even a seemingly small difference in the CDFs could be considered significant. Thus another check is performed in order to assess if the new *significantlly*

different distribution after the split is due to an increment of the uncertainty measured as the dispersion before and after the split. In such case, the split —even resulting in a different probability distribution— is not desirable.

In order to account for how the shape of the probability distribution changes after the split, an additional check is added. The dispersion of the probability distribution before the split σ_1 and after the split σ_2 are estimated —by using the standard deviation for instance, as it is easy to be calculated online— along with the ratio between them:

$$\sigma^* = \frac{\sigma_1}{\sigma_2}$$

Then, we a define a certain threshold σ_τ, and a vote for variance reduction (VR) is cast in the following way:

$$\text{VR} = \begin{cases} -1, & \text{for } \sigma^* < (1 - \sigma_\tau) \\ 0, & \text{for } (1 - \sigma_\tau) \leq \sigma^* \leq (1 + \sigma_\tau) \\ 1, & \text{for } \sigma^* > (1 + \sigma_\tau) \end{cases}$$

A common value for the threshold is $\sigma_\tau = 0.10$. In order to mark a certain split as favorable, the following conditions must be satisfied:

1. The K-S test must have rejected the null hypothesis H_0 that the two samples are drawn from the same probability distribution.
2. In order to account only for *big enough* differences, a threshold D_τ is imposed so that the condition $D^* > D_\tau$ must be satisfied. A value for this threshold can be $D_\tau = 0.025$.
3. The new probability distribution after the split must be less uncertain than before the split, so $VR \geq 0$.

If the split complies with these conditions, a score is assigned based on the sample size of the split n_2 with regards to the sample size before the split n_1, and the K-S statistic:

$$g^*(x) = \frac{n_2}{n_1} D^*$$

Otherwise, if the split does not comply with the above conditions, it is not considered in the scoring process. Finally, the split with highest score is selected for the node expansion.

5.1.1.2.1 Extending the scoring function to a multi-task mining approach The described scoring method is defined for a single task, i.e. a unique directional flow of information between a pair of nodes. However, this definition can be easily extended to consider a set of related tasks. In the case of the problem of traffic forecasting whose underlying structure is the directed graph from the road network, the extension involves considering both the temporal and spatial dimension of the problem.

More specifically, the temporal dimension implies that we are not only interested in finding a flow pattern for the current time, but also to consider its temporal consistency in multiple time steps ahead related to those of the considered forecasting horizons —e.g. from $t = 0$ until $t = 60$ in Δt steps—. This means considering multiple probability distributions for each consecutive time interval, but dealing with such joint probability distributions is unfeasible because of computational costs and more importantly because true distributions are not known. Nevertheless, performing independent individual statistical tests on each temporal distribution may carry another series of risks given that statistical hypothesis testing is based on rejecting the null hypothesis if the likelihood of the observed data under the null hypotheses is low, and thus if multiple hypotheses are tested, the chance of a rare event increases and so does the likelihood of incorrectly rejecting a null hypothesis, i.e. incurring in a type I error (or false positive). For this purpose, methods for multiple hypothesis testing [78] such as the Bonferroni correction [40] aims to compensate by testing each individual hypothesis at a significance level of α/m where α is the desired overall level of significance and m is the number of hypotheses to test. However, this type of procedures to control the family-wise error rate (FWER) which is the probability of making a type I error comes with the cost of being conservative when there are a large number of tests or when the tests statistics are positively correlated, thus reducing the statistical power and increasing the probability of type II error (or false negatives). Therefore, given that K-S statistics D^* of multiple tests consecutive in time are positively correlated, a more practical approach has been adopted that assumes p-values, D^*, and VR values are random variables and then taking the expected value of them.

On the other hand, the spatial dimension defines how different nodes in the graph with their own relations are taken into account in such multi-task setting. With this aim, a voting schema has been adopted where every node casts a *vote* during the scoring stage. More specifically, after performing the temporal evaluation for each node, these cast a vote that can be negative, positive or abstention. Such vote decision per node is defined by:

$$\upsilon = \left\{ \begin{array}{ll} -1, & \text{for VR} < 0 \\ 1, & \text{for } p\text{-value} < \alpha \ \wedge \ D^* > D_\tau \ \wedge \ \text{VR} >= 0 \\ 0, & \text{for otherwise} \end{array} \right\}$$

Let us define υ^-_{max} as the maximum admissible fraction of negative votes among all nodes in order to further considering a given split decision criteria. Analogously, υ^+_{min} is the minimum fraction of positive votes among all nodes to further considering a given split decision criteria. General values for these thresholds can be $\upsilon^+_{min} = 0.75$ and $\upsilon^-_{max} = 0.15$. From those splitting evaluations which meet these criteria, $\upsilon^- \leq \upsilon^-_{max}$ and $\upsilon^+ \geq \upsilon^+_{min}$, the one with the highest final score is chosen for the node expansion, where the score for each split is given by:

$$G^*(x) = \upsilon^+ \frac{n_2}{n_1} E(D^*)$$

,

where υ^+ is the fraction of positive votes among all nodes in a specific split, n_1 and n_2 are the samples sizes before and after the split respectively, and $E(D^*)$ is the expected value of the random variable defined with the K-S statistic D^* values from those nodes whose VR ≥ 0.

Therefore, the scoring function $G^*(x)$ gives a score for all the graph at multiples consecutive time intervals, using the underlying assumed spatiotemporal representation constituted by multiple probability distributions coming from every connection in the graph between a pair of nodes and at multiple time intervals representing the flow of information between nodes.

5.1.1.3 Anomaly detection

After having characterized different patterns or rules using the aforementioned pattern mining procedure which is based on the flow of information through a directed graph, it is possible to use such probability distributions among nodes in order to quantify the outlierness of an observation. As aforementioned in the scoring function subsection, the underlying true distribution on such flow connections among nodes is not known but a reasonable assumption is to assume normality.

Thus a simple parametric approach as calculating the Z-score —also called standard score— which relies on an underlying normal distribution, can be used to quantify the outlierness of an observation. The Z-score is a metric that indicates how many standard deviations σ an observation x is from the population mean μ. In practice, however, as the population mean and standard deviation are

not known, the sample mean $\overline{x}$ and sample standard deviation S are used instead. It is equivalent to perform a standardizing or normalization of the variable x, and it is formally defined by:

$$z = \frac{x - \mu}{\sigma}$$

The motivation behind using the Z-score as a metric to quantify the outlierness relies on the shape itself of a Gaussian distribution around certain location. Therefore, any observation that has a Z-score higher than 3 can be considered an outlier, and likely to be an anomaly. As the Z-score increases above 3, observations become more obviously anomalous.

The method can be extended to the whole graph by considering a random variable Z^* composed of every Z-score from the graph connection. Then, the expected value of the distribution can be inferred.

The procedure for anomaly detection can be extremely useful to detect flow anomalies in the graph that can be centered in certain regions of the graph or spread through it. Thus providing useful information about weird conditions that can result in wrong forecasts. Moreover, this simple method for anomaly detection has the advantage of being very efficient to compute during real-time operation.

5.1.1.4 Basic filling in of missing data

Statistics for the complete feature space are updated in real-time with new incoming streams of data. Usually, basic descriptive statistics such as the mean, standard deviation, minimum, maximum and sample size are gathered. These are collected overall and specifically for each one of the rules. Thus, by means of the use of gathered statistics, it is possible to replace missing values from incoming streams with an unbiased estimator which also depends on the current identified pattern.

This property, albeit simple, is cheap and can be very useful in practice, as missing data is not uncommon at all in real-time applications.

5.1.2 Change detection

Change detection and adaptation in Adarules is tackled from two perspectives: a global and a local one.

The global perspective aims to detect those changes whose impact concerns a significant part of the graph and it involves restructuring the patterns by modifying the Adarules underlying tree structure.

This is motivated by the definition itself of pattern mining given in previous subsections. More specifically, given the aforementioned spatiotemporal representation over the complete network graph based on probability distributions for the flow of information between nodes, the goal is to detect changes in a significant part of such distributions. To this end, these distributions are assumed to be Gaussian and thus are represented by a mean μ and a standard deviation σ which are updated online with new streams of data. Then, these probability distributions can be continuously monitored in streaming with new observations in order to detect change —concept drift or shift— in such distribution. For each spatial node v_i in the graph, every time-specific t_j distribution is monitored for change $\vartheta^{\mathrm{G}}(v_i, t_j)$. A change is detected in a node v_i by the logical disjunction of the detected changes in its series of time intervals:

$$\vartheta^{\mathrm{G}}(v_i) = \vartheta^{\mathrm{G}}(v_i, t_1) \vee \ldots \vee \vartheta^{\mathrm{G}}(v_i, t_{max})$$

In order to raise a global change alarm in the context of a rule R_i, it is checked if change has been detected in a significant region ϑ_τ of the graph among nodes v: $\vartheta^G(G) > \vartheta^G_\tau$. In general, a *significant part* can be defined as half of the network graph, i.e. $\vartheta^{\mathrm{G}}_\tau = 0.5$. When such global change is detected in a node R_i of the ruleset, that rule becomes outdated and a restructuring must be performed in the decision tree of Adarules. However, in order to perform such restructuring only when certainly global change has been confirmed, a second check is performed to assess if that global has also affected to *enough* (e.g. $\vartheta^{\text{G-Siblings}}_\tau = 2/3$) related nodes of R_i in the decision tree —its sibling/s and their descendants— or, equivalently, such global change has been detected *enough* consecutive times in R_i (e.g. $\vartheta^{\text{G-ConsecutiveSolo}}_\tau = 3$) with the aim of reducing the risk of false positives. Once the global change is confirmed, the restructuring around rule R_i in the decision tree is performed as shown in the example in Figure 5.3, the marked node R_i —in red— along with its sibling nodes and their descendants —with dashed borders— are all merged again in a parent node R —in green— thus less specialized as a pattern but that will have future opportunities to be expanded as its Nmin parameter is reset to its default value.

There is a special case of global change detection which is not monitored nor performed the same way as described above, that occurs when a certain node R is expanded during the rule expansion process using the *timestamp* as splitting attribute. In such a scenario —shown as blue nodes in Figure 5.3—, a broadcast message is sent from that leaf node R to the whole decision tree in Adarules in order to revert the whole structure back to the root node and creating two child nodes R_1 and R_2 —in green— that will split up the data based on the selected timestamp t_{split}. From then on, only data after t_{split} will be considered for future rule expansions. Actually, this would be equivalent to set that new node R_2 as the new root node, thus ignoring older data —corresponding

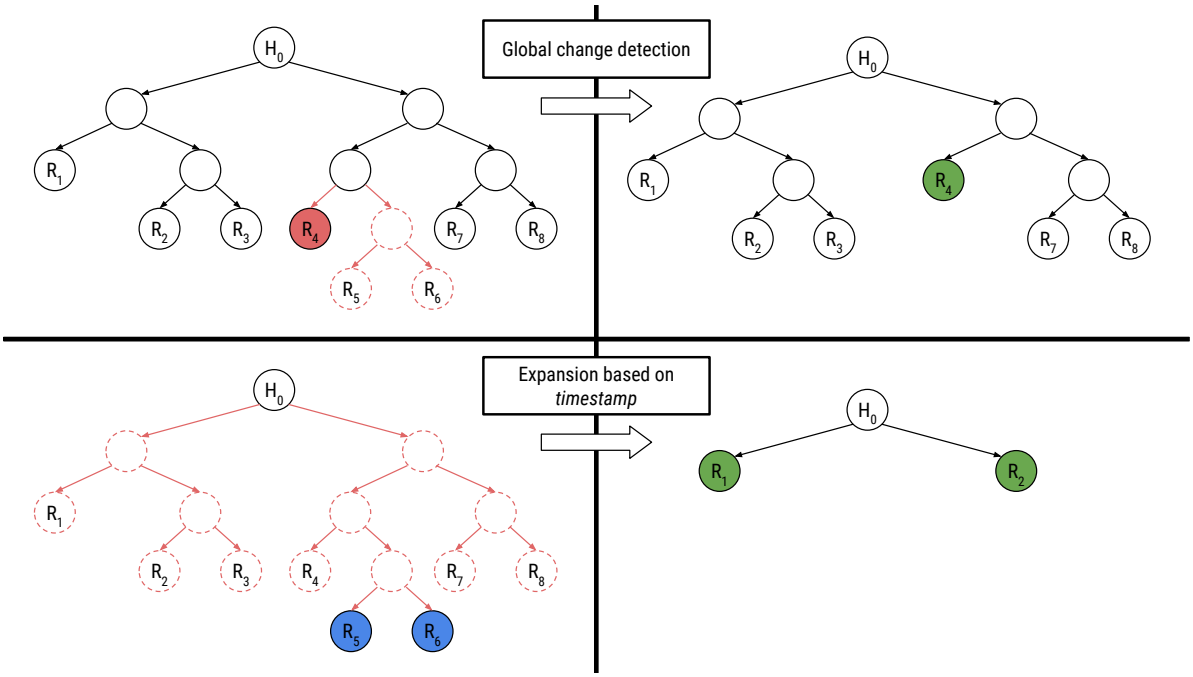


Figure 5.3: Restructuring performed within Adarules decision tree after a *global change* is detected in a node R, or whenever an expansion based on *timestamp* has been carried out in a node R.

to the node R_1—. Even though this would be the actual procedure in a real-time production system, we have decided to let *alive* such node R_1 in order to have the possibility to perform post-mortem analysis in the current implementation.

The second kind of perspective considered is concerned about a change within a local scope in the graph in the context of a specific rule R, $\vartheta_R^{\mathrm{L}}(v)$. More specifically, the forecasting error is monitored the same manner as described above for each v_i from the graph in the context of a specific rule R_i, and since the expectation of the residuals from the forecasting models is zero by definition, it can also be detected when a deviation occurs alerting of concept drift or shift. When this happens, actions are taken by reducing the weight on those observations before the local change took place in the context of that rule R_i and for that node v_i, thus performing an effect of gradual forgetting. This is carried out by using a decay factor $\lambda_{\vartheta^{\mathrm{L}}}$ to be multiplied on the observations' weights —which initially could be 1—. For instance, applying a decay factor $\lambda_{\vartheta^{\mathrm{L}}} = 2/3$.

The underlying algorithm that has been applied for monitoring change detection in each of the aforementioned cases is the Page-Hinkley (PH) test. More specifically, the PH test is a sequential test for monitoring an abrupt deviation in the average of a Gaussian signal, which considers two cumulative variables, m_L^T and m_U^T, defined as the cumulated differences between the observed values and their mean till the current moment:

$$m_U^T = \sum_{t=1}^{T}(x_t - \overline{x}_T - \gamma), \quad m_L^T = \sum_{t=1}^{T}(x_t - \overline{x}_T + \gamma),$$

where $\overline{x}_T$ is the online mean of the observed variable till time T, and γ corresponds to the magnitude of changes that are allowed. The values $M_U^T = \min(m_U^t, t = 1, \dots, T)$ and $M_L^T = \max(m_L^t, t = 1, \dots, T)$ are also computed at every time step t. Finally, the PH test evaluates the differences: $m_U^T - M_U^T$ and $M_L^T - m_L^T$. When any of these differences is greater than a given threshold λ, an alarm is raised because of a detected change in the distribution which could be a positive or negative change in the average of the signal. The threshold λ is set according to the admissible false alarm rate. Increasing this threshold will entail fewer false alarms, but it might miss some changes. Thus, an empirical evaluation has been performed in order to assess an optimal tuning of such parameters, γ and λ, for the PH test.

The goal of such empirical evaluation is to determine a setting of parameters which perform well in terms of detection ratio —true postives—, mean time to detect and a low false alarm ratio —false positives—. For this aim, the experimental design has contempled simulated data which is generated from two processes: a normal distribution and a log-normal distribution, this latter is intended in order to account for right-skewed data. Different changes of magnitude have been considered in order to evaluate the detection algorithm performance as shown in Figure 5.4, where every change of magnitude is made with respect to the immediately preceding state each one composed of a sample size $N = 10000$. The testing values considered for each parameter are:

$$\gamma = \begin{bmatrix} 0.005 \\ 0.05 \\ 0.1 \\ 0.5 \end{bmatrix}, \quad \lambda = \begin{bmatrix} 100 \\ 200 \\ 500 \\ 1000 \end{bmatrix},$$

that constitute the evaluating grid shown in Table 5.1. The summary of results is shown in Table 5.2 describing the percentage of true positives $\%\text{TP}^+$ —only for positive changes as all the changes of magnitude are positive—, the mean time to detect the change (MTTD) and the total number of false positives $\#\text{FP}^{+,-}$ alarming a change —either positive or negative— when there was not any. The detection algorithm is able to perform in a streaming mode digesting every element sequentially, but a mini-batch digesting approach has been adopted based on a size of 100 elements per chunk in order to perform more efficient algebraic operations. Then, for instance a MTTD of 2 would be equivalent to detecting the change in the data chunk between the element 100th to the 200th. As can be seen, results among different parameter settings are not extremely different,

Table 5.1: Hyperparameter grid to explore in the empirical evaluation for the Page-Hinkley algorithm.

γ	λ
0.005	100
0.050	100
0.100	100
0.500	100
0.005	200
0.050	200
0.100	200
0.500	200
0.005	500
0.050	500
0.100	500
0.500	500
0.005	1000
0.050	1000
0.100	1000
0.500	1000

and it is observed that lower levels of γ parameter implies a higher risk of false positives especially when combined with a low λ. An important fact is that the first change of magnitude equivalent to 1.5 is extremely subtle and thus it is not really convenient to flag it as an abrupt change in the signal. This lead to consider only those combinations with a 80% of $\%\text{TP}^{+}$. Among these, those with shortest MTTD are desired, and being conservative to reduce the risk of false positives with new signals, the final parameter combination selected for the PH change detection algorithm are $\gamma = 0.5$ and $\lambda = 200$.

5.1.3 Feature penalizer

The feature penalizer is aimed to generate penalties —understood as the opposite of the probability of being relevant for a given task— into the feature space and under a given task, on the basis of prior structural assumptions.

In the case of features coming directly from the road network graph, the penalties are calculated on the basis of four penalty components: (1) spatiotemporal $\rho_{\mathcal{ST}}$, (2) fraction of missing values ρ_{η}, (3) near zero variance ρ_{σ_0} and (4) differences between the flow levels $\rho_{\Delta Q}$.

The spatiotemporal component of the penalty $\rho_{\mathcal{ST}}$ relies on the distance between the pair of nodes v_1 and v_2 both in the spatial and temporal dimensions, usually v_1 corresponds to the current task and v_2 performs as *feature* for such task. In general, a distance function is sought with the form

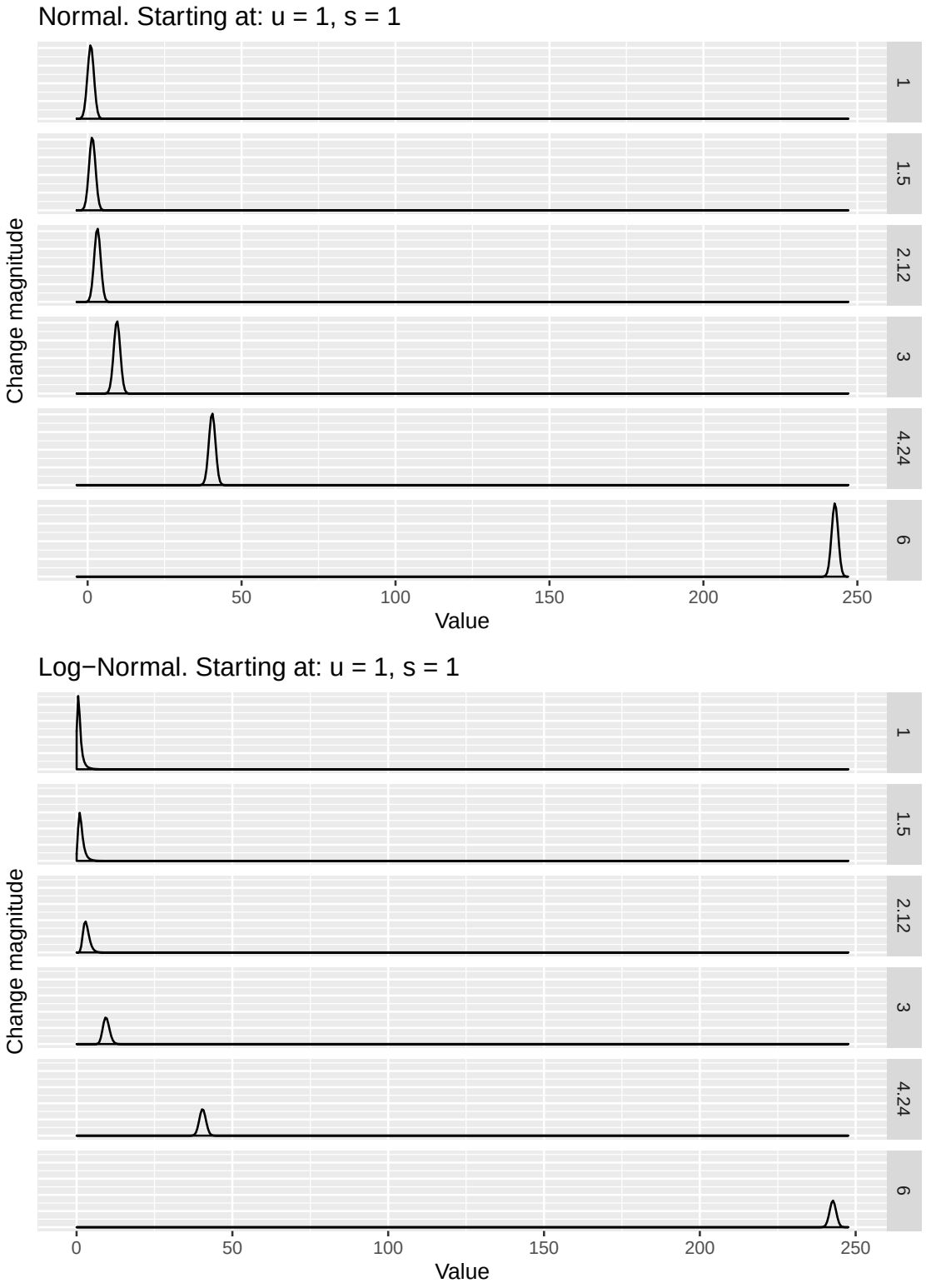


Figure 5.4: Normal and log-Normal distributions with different consecutive changes of magnitude using synthetic data. Every change of magnitude is made with respect to the immediately preceding state.

Table 5.2: Summary results from the assessment of PH test with different parameters.

γ	λ	Data	$\%TP^{+}$	MTTD	$\#FP^{+,-}$
0.005	100	Log-Normal	100	3.20	54
0.005	100	Normal	100	3.00	55
0.005	200	Log-Normal	100	3.60	10
0.005	200	Normal	100	3.20	8
0.005	500	Log-Normal	100	6.80	0
0.005	500	Normal	100	7.80	0
0.005	1000	Log-Normal	80	3.25	0
0.005	1000	Normal	80	4.25	0
0.050	100	Log-Normal	100	3.40	0
0.050	100	Normal	100	3.40	0
0.050	200	Log-Normal	100	3.80	0
0.050	200	Normal	100	4.00	0
0.050	500	Log-Normal	80	3.75	0
0.050	500	Normal	80	3.75	0
0.050	1000	Log-Normal	80	4.50	0
0.050	1000	Normal	80	4.75	0
0.100	100	Log-Normal	100	3.40	0
0.100	100	Normal	100	3.40	0
0.100	200	Log-Normal	100	3.80	0
0.100	200	Normal	100	4.20	0
0.100	500	Log-Normal	80	3.75	0
0.100	500	Normal	80	3.75	0
0.100	1000	Log-Normal	80	4.75	0
0.100	1000	Normal	80	4.75	0
0.500	100	Log-Normal	100	4.00	0
0.500	100	Normal	80	3.00	0
0.500	200	Log-Normal	80	3.25	0
0.500	200	Normal	80	3.25	0
0.500	500	Log-Normal	80	3.75	0
0.500	500	Normal	80	4.00	0
0.500	1000	Log-Normal	80	5.00	0
0.500	1000	Normal	80	5.25	0

$f : V \to \mathbb{R}$. As there is already a natural distance function that takes into account both dimensions at the same time, i.e. the travel time, such function $tt : V \to \mathbb{R}$ can be directly used. As the underlying problem structure is a directed graph and the function $tt(v_1, v_2)$ is asymmetric, we define:

$$tt^*(v_1, v_2) = \min\{tt(v_1, v_2), tt(v_2, v_1)\}$$

Thus, the spatial part $\mathcal{S}$ of the spatiotemporal component is given by the travel time already in time units $\mathcal{S}(v_1, v_2) = tt^*(v_1, v_2)$. The temporal part $\mathcal{T}$ of the component is defined by its relation with the lag operator L. Given a time series $X = \{x_1, x_2, ..., x_t, ...\}$, the lag operator is defined as $L^k X_t = X_{t-k}$. Then, the temporal part takes into account this lag operator and the associated sampling frequency Δt of data: $\mathcal{T}(v_2) = k\Delta t$, and the spatiotemporal component is defined as the addition of both parts:

$$\rho_{\mathcal{ST}}(v_1, v_2) = \mathcal{S}(v_1, v_2) + \mathcal{T}(v_2)$$

Finally, there exists a threshold $\rho^\tau_{\mathcal{ST}}$ that can be fixed in advance at a certain constant value c or dynamically adjusted according to a given forecasting horizon $h(t)$:

$$\rho^\tau_{\mathcal{ST}} = \begin{cases} h(t) \\ c \\ \infty \end{cases}$$

The objetive of such threshold is to impose an infinity penalty to those features whose spatiotemporal penalty exceeds the threshold: $\rho_{\mathcal{ST}}(v_1, v_2) > \rho^\tau_{\mathcal{ST}} \implies \rho_{\mathcal{ST}}(v_1, v_2) = \infty$. Obviously, when $\rho^\tau_{\mathcal{ST}} = \infty$ there is no such imposition.

For the rest of the penalty components, a sliding window ω of a given size, e.g. the last 30 days, is used in order to account for the most recent data statistics.

Penalty for the fraction of missing values ρ_η aims to impose a higher penalty to those features with a higher ratio of missing data recently, as these can be considered unreliable features. Having a function $\eta(v_2, \omega)$ that returns the fraction $[0, 1]$ of missing values for the feature v_2 in the window ω, we define $\rho_\eta(v_2) = \eta(v_2, \omega)$. Additionaly, a threshold is also defined $\rho^\tau_\eta = 0.90$, generally, so that:

$$\rho_{\eta}(v_2) = \begin{cases} \infty, & \text{for } \rho_{\eta}(v_2) > \rho_{\eta}^{\tau} \\ 0 & \text{for } \rho_{\eta}(v_2) \leq \rho_{\eta}^{\tau} \end{cases}$$

The motivation behind the near-zero-variance penalty component ρ_{σ_0} is to impose a penalization on those features with an extremely low observed variance. This is due, first, to the numerical instabilities —e.g. divisions by zero or numerical precision issues— they can cause in practice because of resampling methods that are used through the thesis. Moreover, this is also supported by the prior domain knowledge as predictors whose variance are extremely low can be due to the fact that either they are irrelevant as they represent a small part of the flows, or they correspond to regions of the graph that have not seen enough flow until that moment with the consequent risk of experiencing a structural change —for instance, a new road where the traffic has been restricted until construction is finished thus letting to pass the road traffic—. Therefore, a threshold $\rho_{\sigma_0}^{\tau} = 0.01$ is defined, so that $\rho_{\sigma_0}(v_2) \leq \rho_{\sigma_0}^{\tau} \implies \rho_{\sigma_0}(v_2) = \infty$, where $\rho_{\sigma_0}(v_2) = \sigma(v_2)$ returns the variance of v_2 using the window ω:

$$\rho_{\sigma_0}(v_2) = \begin{cases} \infty, & \text{for } \rho_{\sigma_0}(v_2) \leq \rho_{\sigma_0}^{\tau} \\ 0 & \text{for } \rho_{\sigma_0}(v_2) > \rho_{\sigma_0}^{\tau} \end{cases}$$

Lastly, the penalty component regarding the differences between the flow levels $\rho_{\Delta Q}$ aims to impose more penalty to those features v_2 with a significantly lower flow level Q compared to that of the current task v_1. To this end, $\rho_{\Delta Q}$ is defined as:

$$\rho_{\Delta Q}(v_1, v_2) = \begin{cases} 0, & \text{for } \bar{Q}(v_2) \geq \bar{Q}(v_1) \\ \frac{\bar{Q}(v_1) - \bar{Q}(v_2)}{\bar{Q}(v_1)} & \text{for } \bar{Q}(v_2) < \bar{Q}(v_1) \end{cases}$$

where $\bar{Q}$ corresponds to the average of the flow using the window ω.

After having all the separate penalty components, their composition into a single penalty ρ takes the spatiotemporal penalty $\rho_{\mathcal{ST}}$ as the *natural* baseline and the rest as influential factors that can increase it as they are in the range $[0, 1]$ along with the special value of ∞:

$$\rho(v_1, v_2) = \rho_{\mathcal{ST}}(v_1, v_2) \cdot (1 + \rho_{\eta}(v_2)) \cdot (1 + \rho_{\sigma_0}(v_2)) \cdot (1 + \rho_{\Delta Q}(v_1, v_2))$$

5.1.4 Feature selector

The aim of the features selector component is to provide these upon request. This selection can be deterministic —e.g. performing a full selection of features within a category or the whole feature space, or performing a cyclical selection over features, etc.— or could be stochastic —e.g. by assigning probabilities to features as a way to a priori calculate their suitability for the specific task.

In this thesis, it operates for two goals. One is to provide features as possible choices for the rule expansions process, which is carried out by simply selecting all the qualitative features among different categories —date and time, weather, calendars of special days, etc.—. The other purpose is to also serve relevant features during the process of fitting the forecasting models, which is performed on demand. This also includes trimming irrelevant features, which is performed in combination with the feature penalizer for those features v_j whose penalty $\rho(v_i, v_j) = \infty$ for a given task v_i, as part of the screening rules schema used for fitting the forecasting models.

5.1.5 Feature outlier filter

In a real-time application, it is crucial to detect univariate spurious outliers, i.e. those that are not truly part of the process but are caused by faulty measurement devices instead, in order to perform online filtering.

To this end, the interquartile range IQR [122] is a robust measure of scale, i.e. less sensitive to outliers, showing how the data is spread about the median. It is defined by:

$$\mathrm{IQR} = Q_3 - Q_1$$

From that, a certain threshold is used, typically $\mathrm{IQR}_\tau = 1.5$ which is used to multiply the IQR value. Then, detection of suspected outliers below the range of data can be performed by those which are lesser than $Q_1 - \mathrm{IQR} \cdot \mathrm{IQR}_\tau$, while detection of suspected outliers above the range of data can be performed by those which are greater than $Q_3 + \mathrm{IQR} \cdot \mathrm{IQR}_\tau$.

Estimation of quartiles Q_1 and Q_3 benefit from the streaming data using a sliding window ω of a given size, e.g. the last 30 days. Such estimation is individual per feature.

In addition, for those features whose support is well-defined, their minimum and maximum valid measurement range still holds. For instance, traffic measurement —either flow, occupancy or speed— cannot be negative.

5.1.6 Forecasting model: sparse model for spatiotemporal correlations

The proposed forecasting modelling approach has the form of linear dependence between every region in the network with respect to all the road network state and at different times, i.e. spatiotemporal correlations. Given that traffic is a highly non-linear process where changes in traffic dynamics occur suddenly, a linear model may not seem appropriate for short-term traffic prediction. However, the linear model is built using the examples covered by each rule, letting that these unveil the nonlinearities in the traffic dynamics leading to the creation of specialized linear models, which also benefit from simpler and more efficient procedures to learn.

The functional form of such modelling approach can be represented by:

$$Y(t+h) = f(X(t))$$

where:

- matrix Y corresponds to the network traffic state —flow, occupancy or speeds— at a given future time $t+h$ which is h time steps ahead of current time t,
- matrix X is the matrix containing all the necessary input information at a given set of time: $(t, t-1, \ldots, t-L_{max})$,
- f is the analytical model representing the relationship between X and Y.

The matrix X is built containing explicative variables of the model, i.e. those variables considered as necessary to explain the behaviour and evolution of Y, which may include the network spatial information involving the current network traffic state –including upstreams and downstreams of the selected measurement source to predict—, along with past values $(t, t-1, \ldots, t-L_{max})$ of these to infer the future evolution of the network traffic state over time.

The motivation behind using the temporal facet, i.e. lags of variables, is to include an important source of information to infer the evolution of the network traffic state over time, especially when there is an important lack of network observability, i.e. that not all the network has traffic measurement devices in order to provide observations.

However, it is not convenient to use data from every road segment to develop a forecast for a given location for two reasons: (1) it would be computationally prohibitive, and (2) locality effect in the causal relations between congestion patterns on different road segments could be hidden by such high-dimensionality. Moreover, there could be other correlated effects in data but which are not evident, for example when an arterial road is an alternative to a highway, those roads

will be strongly related even that they are not topological neighbours. To this end, methods of regularization provide with an automated approach to perform feature selection and model selection.

In addition, there exists a large effect of multicollinearity among features in a road network graph, and in such situation, some modelling approaches simply would fail in their attempt to fit the data —such as ordinary least squares regression (OLS)—, while other approaches that perform regularization on the feature space can surpass this effect. For example, a ridge penalty would shrink the coefficients of correlated predictors towards each other while the lasso penalty tends to pick one of them and discard the others. The latter is more interesting to the problem of short-term traffic prediction as we are interested in the *strongest* correlations to make more robust the solution: a sparse linear model for forecasting.

In this sense, the adaptive penalties generated from the *feature penalizer* component can perform as input of the fitting process for these sparse models, performing as prior information and helping to guide the fitting process towards more meaningful and robust solutions, and alleviating the aforementioned problem of multicollinearity. It is also convenient as the standard *lasso* approach does not have oracle properties as the *adaptive lasso* does have [274].

Thus, we are interested in a sparse linear model for the forecasting whose empirical risk minimization —for computational reasons in such high-dimensional setting— has the following form:

$$\min_{(\beta_0,\beta)\in\mathbb{R}^{p+1}} \frac{1}{2}\frac{1}{N}\sum_{i=1}^{N} w_i(y_i - \beta_0 - x_i^T\beta)^2 + \lambda\sum_{j=1}^{p}\frac{c_j}{\bar{c}}|\beta_j|$$

The first part corresponds to a squared loss, i.e. the minimization of the residual sum of squares, which weight every i-th observation out of N by a relative weight $w_i \in [0,1]$ describing its importance within the entire dataset. The second part corresponds to the regularization part over the vector of coefficients β whose relative importance within the optimization problem is determined by the regularization hyperparameter λ which is tuned used resampling techniques on data such as cross-validation. Every p-th feature is adaptively penalized according to a certain given value c_j which is normalized using $\bar{c} = \frac{1}{p}\sum_{j=1}^{p} c_j$. Such formulation is known as *adaptive lasso* in ERM literature.

Concerning the learning procedure, coordinate descent (coordinate-wise gradient descent) has been used to obtain the parameter estimates β because it applies well to the case where $N \ll p$, being successfully applied to this problem in high-dimensionality settings [90] and demonstrated to be efficient in large problems [191]. This family of first-order methods optimizes a target function with

respect to a single parameter at a time, iteratively cycling through all parameters until convergence is reached. Thus, application of coordinate descent to solve the problem involves performing per-feature updates until convergence.

More specifically, having X features already centered and standardized —and optionally also the response Y—, so that:

$$\sum_{i=1}^{N} x_{ij} = 0, \sum_{i=1}^{N} x_{ij}^2 = N, \quad \forall j = 1, 2, ..., p$$

and given current values for the coefficients $\beta_k = \tilde{\beta}_k$, $k \neq j$, computing the gradient with respect to β_j derives L_j:

$$L_j(\beta_j, \lambda) = \frac{1}{2N} \sum_{i=1}^{N} w_i \left(y_i - \sum_{k \neq j} x_{ik} \tilde{\beta}_k - x_{ij} \beta_j \right)^2 + \lambda c_j |\beta_j|$$

It can be denoted $\tilde{r}_{ij}$ as the partial residual with respect to the j-th feature which removes from the outcome the current fit from all but the j-th predictor:

$$\tilde{r}_{ij} = y_i - \tilde{y}_{ij}$$

where $\tilde{y}_{ij}$ is defined as:

$$\tilde{y}_{ij} = \sum_{k \neq j} x_{ik} \tilde{\beta}_k$$

and denoting $\tilde{z}_j$ as:

$$\tilde{z}_j = \frac{1}{N} \sum_{i=1}^{N} w_i x_{ij} \tilde{r}_{ij}$$

Finally the update to the coefficient β_j is defined as:

$$\tilde{\beta}_j = \frac{\mathcal{S}\left(\tilde{z}_j, \lambda\right)}{1 + \lambda}$$

where $\mathcal{S}(z, \lambda)$ is the soft-thresholding operator with value $\mathcal{S}(z, \lambda) = \text{sign}(z)(|z| - \lambda)_+$ which translates[2] its argument z toward zero by the amount λ, and sets it to zero if $|z| \leq \lambda$ as shown in Figure 5.5.

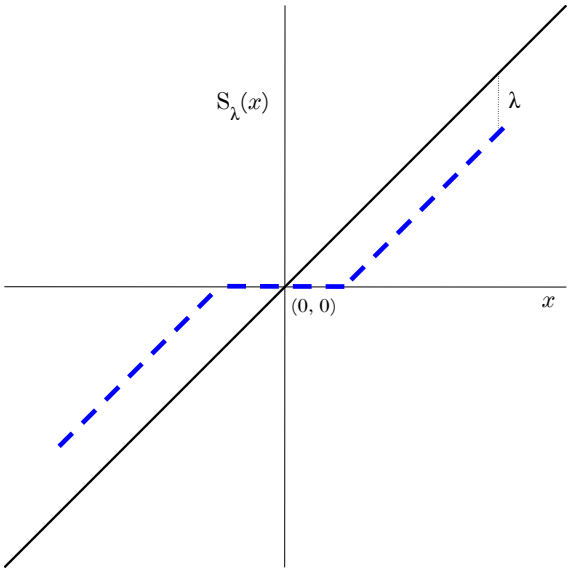


Figure 5.5: Soft thresholding function $\mathcal{S}(x, \lambda) = \text{sign}(x)(|x| - \lambda)_+$ is shown in blue (dashed lines), along with the 45° line in black.

Such parameter estimation could also be applicable in an online manner using mini-batches of streaming data, a small learning rate η and the soft-thresholding technique [219]. Moreover, instead of performing a *naïve* cyclical update of all the coordinates in every iteration, it would be possible to rely on the feature selector to probabilistically select a subset of more relevant coordinates to be updated. Even though its potential online applicability for an active incremental learning strategy, eventually in this thesis a passive incremental learning strategy has been chosen for the forecasting models. This strategy relies on the local change detection previously described in order to estimate again the coefficients using the updated weights vector w over the dataset observations. This is part of a gradual forgetting strategy which eventually could simply discard old data after a certain number of local changes have been detected. On the other hand, if no local change is detected in a significant amount of time, then parameters are estimated again including all the new observations, and relying on resampling techniques such as cross-validation (CV) in order to avoid a loss of generalization performance.

Solving for the lasso problem requires determining the optimal amount of sparsity which is imposed by the λ hyperparameter as stated in the ERM-like formulation. An strategy to address this is through what is known as *regularization path* of the lasso, which relies on solving the problem over a grid of λ values on a log scale ranging from the sparsest λ_{max} which corresponds to having all coefficients set to zero to the least sparse equivalent to an ordinary least-squares solution $\lambda_{min} =$

[2] r_+ denotes the positive part of $r \in \mathbb{R}$, equal to r if $r > 0$ and 0 otherwise

λ_{OLS}. More specifically, let $\lambda_0 > \lambda_1 > ... > \lambda_k$ be a grid of decreasing λ-values, where $\lambda_0 = \lambda_{max} = max_j|(x_j^T x_j)^{-1} x_j^T y|$, and $\lambda_k = \lambda_{min} = 0$ if the design matrix is full rank, or, otherwise $\lambda_{min} = \varepsilon\lambda_{max}$ with a given small $\varepsilon = 10^{-4}$. The path starts to be solved from λ_0, proceeding along the grid using value of $\hat{\beta}$ at the previous λ_{k-1} solution as the initial solution for the current λ_k, in a procedure known as *warm start*. Then, for every specific λ_k, the coordinate descent algorithm performs as stated in Algorithm 5.6.

Algorithm 1: Coordinate Descent (CD) algorithm for a given λ_k

Input : $\beta^{(0)}$: initial vector of coefficients from λ_{k-1}
s^*: accumulated passes over the data in the λ-path so far
s_{max}: max number of iterations (passes over the data for all the λ path)
Output: β: vector of coefficients for current λ_k

$s \leftarrow 0$
while *(¬ converge)* $\wedge$ *(*$s^* \leq s_{max}$*)* **do**
  foreach $j = 1, ..., p$ **do** cyclical CD over the active set $\mathcal{A}$
    Calculate $\tilde{z}_j = \frac{1}{N}\sum_{i=1}^{N} x_{ij} r_i + \tilde{\beta}_j^{(s)}$
      where $r_i = y_i - \tilde{y}_i = y_i - \sum_{j=1}^{p} x_{ij}\tilde{\beta}_j^{(s)}$ is the current residual
    Update $\tilde{\beta}_j \leftarrow \frac{\mathcal{S}(\tilde{z}_j, \lambda)}{1+\lambda}$
    Update $r_i \leftarrow r_i - (\tilde{\beta}_j^s - \tilde{\beta}_j^{s-1}) x_{ij} \quad \forall i, ..., N$
  end
  $s \leftarrow s + 1$
  $s^* \leftarrow s^* + 1$
end

Figure 5.6: Coordinate Descent algorithm.

The convergence criterion focuses on the impact of the change on the fitted values using a weighted norm of the coefficient change vector. Let v_j be the weighted sum-of-squares for the feature x_j using the weights vector w for every observation i:

$$v_j = \sum_{i=1}^{N} w_i x_{ij}^2$$

If there is an intercept $\hat{\beta}_0$ in the model, these x_j will be centered by the weighted mean, and hence this would be a weighted variance. After the previous state of the coefficient $\hat{\beta}_j^{(n-1)}$ has been updated to $\hat{\beta}_j^{(n)}$ in the n-th iteration, then Δ_j measures the weighted sum of squares of changes in fitted values for feature x_j and can be computed as:

$$\Delta_j = v_j \left(\hat{\beta}_j^{(n-1)} - \hat{\beta}_j^{(n)}\right)^2 = \frac{1}{N}\sum_{i=1}^{N} w_i \left(x_{ij}\hat{\beta}_j^{(n-1)} - x_{ij}\hat{\beta}_j^{(n)}\right)^2$$

After every complete cycle of coordinate descent, the maximum difference over all j is checked

$\Delta_{max} = max_j \Delta_j$ and when such largest change is negligible $\Delta_{max} < \varepsilon$, then convergence has been achieved.

It is important to note the parameter s_{max} that dictates the maximum number of iterations — number of passes over the N observations within the dataset— to perform if the solution has not converged before. For this thesis, an intended $s_{max} = 1000$ has been set for two reasons: (1) seeking for more sparse and thus more real-time robust and interpretable models, and (2) computational saving.

Such computational efficiency in solving the lasso problem is also achieved with the use of an active set $\mathcal{A}$ and the strong screening rules. More specifically, after a single CD iteration $s = 0$ through the set of p variables at a new λ_k starting from the warm start $\hat{\beta}(\lambda_{k-1})$, the active set $\mathcal{A}$ can be defined as the set of non-zero features at that time. From then on, the cyclical CD iteration is only performed on such features contained in the active set $\mathcal{A}$ during the subsequent s iterations. Upon convergence, the test $\frac{1}{N}|\langle x_j, r\rangle| < \lambda_k$ —where r is the current residual— is performed over all the omitted variables, and if all of them pass the exclusion test then the solution has been reached for the entire set of p variables. Otherwise, those variables that fail the test are included back in $\mathcal{A}$ and the process is repeated. In practice, an ever-active set $\mathcal{A}$ is maintained, i.e. any feature that had a non-zero coefficient somewhere along the regularization path until current λ_k is kept in $\mathcal{A}$. Similarly, such active set $\mathcal{A}$ can be enhanced with the use of strong screening rules assists in the identification of a subset of features likely to be candidates for $\mathcal{A}$, by defining the strong set $\mathbb{S}$ for a given λ_k as:

$$\mathbb{S} = \left\{ j \mid \ |\frac{1}{N}\langle x_j, r\rangle| > \lambda_k - (\lambda_{k-1} - \lambda_k) \right\}$$

where r is the residual at $\hat{\beta}(\lambda_{k-1})$. Therefore, the solution is computed restricting the attention to only the elements of $\mathbb{S}$. Apart from rare exceptions, the strong set $\mathbb{S}$ will cover the optimal active set $\mathcal{A}$. The use of such strong rules is extremely useful to perform a much more efficient problem solving especially when p is very high-dimensional. Moreover, the set of heuristic rules described in the *feature selector* section are also applied before starting to solve the lasso at λ_0.

Finally, after fitting the complete regularization path it becomes necessary to select a specific value of the λ path, say, λ^*, in order to use $\hat{\beta}(\lambda^*)$ as the final estimator. This model selection step is usually performed using information criterion methods such as AIC or BIC that integrate the performance using an error measurement (e.g. the MSE) coupled with a penalty over the number of parameters, or by resampling methods such as CV which has been the choice for this thesis. Hence the necessity to have a fast algorithm to compute the complete regularization path over a

grid of λ values, which is achieved with the use of the strong screening rules for efficient restriction of the active set along with the highly efficient parameter updates and the use of *warm starts* through the regularization path. Moreover, coordinate descent is especially fast for solving the lasso because the coordinate-wise minimizers are explicitly available and thus an iterative search along each coordinate is not needed. Secondly, it exploits the sparsity of the problem as for large enough values of λ most coefficients will be zero and will not be moved from zero. Using the CD approach to solve the lasso, it is very easy to allow for upper and lower bounds on each coefficient in the model —e.g. allowing a non-negative lasso—, by simply setting back them to the specified bound when such coefficients would attempt to exceed an upper or lower bound during the CD cycle.

5.1.6.1 Extension to a multi-task learning approach

In the case of having multiple responses, the aforementioned ERM formulation can be extended to tackle a multi-task learning approach, where the single response vector y is replaced by a matrix Y of size $N \times K$ where N is the sample size and K the set of K jointly learned tasks, and the coefficient vector β of length p features is replaced by a matrix $\mathcal{B}$ of size $p \times K$. Then, the absolute individual penalty on each single coefficient β_j is replaced by a group-lasso penalty on each coefficient K-vector $\mathcal{B}_j$ for a single predictor x_j, where each group $\mathcal{B}_j$ corresponds to the jth row of the $p \times K$ coefficient matrix $\mathcal{B}$:

$$\min_{\mathcal{B} \in \mathbb{R}^{p \times K}} \frac{1}{N} \sum_{i=1}^{N} w_i \| y_i - \langle \mathcal{B}, x_i \rangle \|_F^2 + \lambda \sum_{j=1}^{p} \frac{c_j}{\bar{c}} \| \mathcal{B}_j \|_2$$

For this case, within the ℓ_1/ℓ_2 penalized multiple Gaussian-response linear models, the sharing involves which features are selected across tasks K since when a feature x_j is selected, then a coefficients vector $\mathcal{B}_j$ becomes non-zero with an individual coefficient β_{jk} fit for each response K, which turns out to be useful when there are a number of correlated responses or tasks to learn. For example, these tasks could comprehend the forecasting of multiple steps ahead, multiple traffic responses —flow, occupancy, and speed together—, or multiple road sections within the network.

Similarly as before, coordinate descent techniques are one reasonable choice, whereas in this case a block coordinate descent is performed on each vector β_j while holding all the others fixed.

5.2 Pseudocode

5.2.1 Main corpus - streaming scenario

Figure 5.7.

Algorithm 2: Main corpus of the streaming scenario for Adarules

Input : $\mathcal{R}$: A given ruleset
$\Delta N_{\mathcal{R}}$: Size of the buffer of elements to be digested
$\mathcal{P}_{\text{PredictionInterval}}$: Probability for the prediction interval
$\mathcal{O}_{\text{Graph}}$: Whether to measure or not outlierness using the graph structure
$\mathcal{O}_{\text{TrafficStates}}$: Whether to measure or not outlierness using traffic states
$\mathcal{ID}$: Whether to perform or not incident detection

```
1  begin
2      foreach observation x_i in streaming do
3          Call R.predict(x_i, P_PredictionInterval, O_TrafficStates, O_Graph, ID)
4          if size(buffer) = ΔN_R then
5              Call R.digest-observations(X, F_Thresholding, C_TrafficStates)
6          else append elements into buffer
7              X ← (X, x_i)
8          end
9      end
10 end
```

Figure 5.7: Main corpus of the streaming scenario for Adarules.

5.2.2 Ruleset: digest observations

Most of the operations (filtering, filling in of missing data, generating features, ...) are performed in streaming or, preferably, in chunks to avoid having in memory all the data.

Figure 5.8 and Figure 5.9.

5.2.3 Ruleset: predict observations

Figure 5.10.

5.2.4 Rule: digest observations

Figure 5.11.

Algorithm 3: Ruleset $\mathcal{R}$: digest observations

Input : X: Input data matrix $[N \times p]$
$\mathcal{F}_{\text{Thresholding}}$: Whether to filter or not data using thresholding
$\mathcal{C}_{\text{TrafficStates}}$: Whether to fill in or not missing data using the informed approach based on traffic states

Output: $\mathcal{R}$ state is modified

begin
  Store new raw data X
  Update statistics Θ_{ω} within the sliding window ω
  if $\mathcal{F}_{Thresholding}$ **then** filter spurious outliers using *thresholding*
    `// F_τ⁻ and F_τ⁺ are the vectors of features' lower and upper bounds,`
    `// respectively, for those which are known`
    `// IQR_τ is the threshold for the interquartile range method`
    $X \leftarrow$ filter-thresholding(X, $\mathcal{F}_{\tau^-}$, $\mathcal{F}_{\tau^+}$, Θ_{ω}, IQR_{τ})
  end
  Update global statistics Θ for features using new data in X
  $X^* \leftarrow$ split-evaluator.generate-graph-features(X)
  if $\mathcal{R}$ *is using Traffic states* **then**
    $\mathcal{T}_{\mathcal{S}} \leftarrow$ trafficstates-model.classify-trafficstates(X)
  end
  $R^* \leftarrow$ evaluation of each rule $R \in \mathcal{R}$ on dataset X
  `// Perform filling in of missing data using the informed approach`
  if $\mathcal{C}_{TrafficStates}$ **then**
    $X_{\text{Clean}} \leftarrow$ fill-missing-data-with-trafficstates(X, $\mathcal{T}_{\mathcal{S}}$, R_{ID})
  end
  `// Perform filling in of (remaining) missing data using the basic approach`
  $X_{\text{Clean}} \leftarrow$ fill-missing-data-by-rule(X, Θ, Θ_R)
  Store X_{Clean}
  foreach *rule* R *in* R^* **do**
    Select the sets $X_R \subseteq X$ and $X_R^* \subseteq X^*$ belonging to R
    R.digest-observations(X_R, X_R^*)
    if *Global drift detected in* R **then**
      $k \leftarrow$ current time index within X_R
      Append k to the vector in R $R.\vec{\vartheta}_{\text{G}} \leftarrow (R.\vec{\vartheta}_{\text{G}}, k)$
    else
      Forget previous detected global changes in R by emptying its vector R.ConceptDriftGlobalDetected $\leftarrow \emptyset$
    end
  end
  Select the set $R^*_{\vartheta_{\text{G}}} \subseteq R^*$ of activated rules R^* with global change $R^*_{\vartheta_{\text{G}}} \leftarrow \{R \in R^* : \vartheta_{\text{G}}(R) = 1\}$
  Select the set $R^*_{\vartheta_{\text{G}}{}^*} \subseteq R^*_{\vartheta_{\text{G}}}$ that meet the criteria $\vartheta_{\tau}^{\text{G-Siblings}} \vee \vartheta_{\tau}^{\text{G-ConsecutiveSolo}}$
  foreach *rule* R *in* $R^*_{\vartheta_{G}{}^*}$ **do**
    Perform restructuring of node R within the decision tree $\mathcal{R}$
  end
  Select the set $R^*_{\text{E}} \subseteq R^*$ of rules to be expanded $R^*_{\text{E}} \leftarrow \{R \in R^* : E(R) = 1\}$

Figure 5.8: Ruleset $\mathcal{R}$: digest observations. Part I.

(34)
(35) **foreach** *rule* R *in* R^*_E **do**
(36) Perform expansion of node R by creating two child nodes $R_{\text{Left}}, R_{\text{Right}}$
(37) **end**
(38) Update spatiotemporal bayesian networks $\mathcal{ST}_{\mathcal{BN}}$ using new data X
(39) **foreach** *local node* v *in* $\mathcal{R}.V$ **do**
(40) Call the feature penalizer to generate penalties $\rho_\omega(v)$ using statistics in Θ_ω
(41) **foreach** *rule* R *in* R^* **do**
(42) Call the feature penalizer to generate penalties $\rho_R(v)$ using statistics in R
(43) $\rho(v) \leftarrow \max(\rho_\omega(v), \rho_R(v))$
(44) $R(v)$.digest-observations(X_R, $\rho(v)$)
(45) **end**
(46) **end**
(47) Select the set $\left(R^*_{\vartheta_{\text{L}}}(v) \subseteq R^* \;\; \forall\, v \in \mathcal{R}.V\right)$ of rules with local change
$R^*_{\vartheta_{\text{L}}}(v) \leftarrow \{R(v) \in R^* : \vartheta_{\text{L}}(R(v)) = 1\}$
(48) **foreach** *rule* R *in* $R^*_{\vartheta_{\text{L}}}$ **do**
(49) Select the set $V^* \subseteq \mathcal{R}.V$ that meet the spatial or temporal criteria
(50) **foreach** *local node* v *in* V^* **do**
(51) Generate vector of weights $w_R(v)$ for $R(v)$ according to local changes in $R(v)$
(52) Update learners in $R(v)$ using $X_R, w_R(v), \rho(v)$
(53) **end**
(54) **end**
(55) **end**

Figure 5.9: Ruleset $\mathcal{R}$: digest observations. Part II.

5.2.5 Rule(v): digest observations

Figure 5.12.

5.2.6 Rule(v): predict observations

Figure 5.13.

5.2.7 Filter spurious outliers using thresholding

Figure 5.14.

5.2.8 Change detection

Figure 5.15.

Algorithm 4: Ruleset $\mathcal{R}$: predict observations

Input : X: Input data matrix $[N \times p]$
$\mathcal{F}_{\text{Thresholding}}$: Whether to filter or not data using thresholding
$\mathcal{C}_{\text{TrafficStates}}$: Whether to fill in or not missing data using the informed approach
$\mathcal{P}_{\text{PredictionInterval}}$: Probability for the prediction interval
$\mathcal{O}_{\text{Graph}}$: Whether to measure or not outlierness using the graph structure
$\mathcal{O}_{\text{TrafficStates}}$: Whether to measure or not outlierness using traffic states
$\mathcal{ID}$: Whether to perform or not incident detection

Output: $\hat{Y}, \hat{Y}_P$: Forecasts for every $v \in \mathcal{R}.V$, and prediction intervals
$\hat{\mathcal{O}}_{\text{Graph}}$: Measure of outlierness using graph patterns
$\hat{\mathcal{O}}_{\text{TrafficStates}}$: Probabilities of outlierness using traffic states
$\hat{ID}$: Probabilities of incident detection using traffic states

begin
  if $\mathcal{F}_{Thresholding}$ **then** filter spurious outliers using *thresholding*
    `// $\mathcal{F}_{\tau^-}$ and $\mathcal{F}_{\tau^+}$ are the vectors of features' lower and upper bounds,`
    `// respectively, for those which are known`
    `// $\text{IQR}_\tau$ is the threshold for the interquartile range method`
    `// $\Theta_\omega$ are the statistics within the sliding window`
    $X \leftarrow$ filter-thresholding(X, $\mathcal{F}_{\tau^-}$, $\mathcal{F}_{\tau^+}$, Θ_ω, IQR_τ)
  end
  if $\mathcal{R}$ *is using Traffic states* **then**
    $\mathcal{T}_{\mathcal{S}} \leftarrow$ trafficstates-model.classify-trafficstates(X)
  end
  $R^* \leftarrow$ evaluation of each rule $R \in \mathcal{R}$ on dataset X
  `// Perform filling in of missing data using the informed approach`
  if $\mathcal{C}_{TrafficStates}$ **then**
    $X \leftarrow$ fill-missing-data-with-trafficstates($X, \mathcal{T}_{\mathcal{S}}$)
  end
  `// Perform filling in of (remaining) missing data using the basic approach`
  $X \leftarrow$ fill-missing-data-by-rule(X)
  foreach *local node* v *in* $\mathcal{R}.V$ **do**
    foreach *rule* R *in* R^* **do**
      Select the set $X_R \subseteq X$ belonging to R
      $(\hat{Y}, \hat{Y}_P) \leftarrow R(v)$.predict($X_R$, $\mathcal{P}_{\text{PredictionInterval}}$)
    end
  end
  if $\mathcal{O}_{Graph}$ **then**
    $X^* \leftarrow$ split-evaluator.generate-features(X)
    $\hat{\mathcal{O}}_{\text{Graph}} \leftarrow R$.measure-graph-outlierness(X^*, Θ^*_R)
  end
  if $\mathcal{O}_{TrafficStates}$ **then**
    $\hat{\mathcal{O}}_{\text{TrafficStates}} \leftarrow \mathcal{ST}_{\mathcal{BN}}$.measure-trafficstates-outlierness($\mathcal{T}_{\mathcal{S}}$)
  end
  if $\mathcal{ID}$ **then**
    $\hat{ID} \leftarrow \mathcal{ST}_{\mathcal{BN}}$.incident-detection($\mathcal{T}_{\mathcal{S}}$)
  end
end

Figure 5.10: Ruleset $\mathcal{R}$: predict observations.

Algorithm 5: Rule R: digest observations

Input : X_R: Input data matrix $[N \times p]$ whose observations belongs to this rule R
X_R^*: Graph-features data matrix $[N \times q]$ q is the number of graph connections
Output: E_R: result of the possible attempt to expand R
ϑ_R^G: Whether if a global change has been detected in rule R or not

begin
  Update rule statistics Θ_R for features using new data in X_R
  Update rule graph statistics Θ_R^* using new data in X_R^*
  `// Perform change detection in every data distribution over the graph (spatial and temporal)`
  $\vartheta^G(v,t) \leftarrow$ change-detection(X_R^*, Θ_R^*)
  Evaluate if there exists global change in rule R checking the whole graph G
  $\vartheta_R^G \leftarrow \vartheta_R^G(G) > \vartheta_\tau^G$
  `//` $N\texttt{min}_R$ `is the number of observations seen since the last expansion in` R
  if $N_R^* > Nmin_R$ **then**
    $E_R \leftarrow$ evaluate-rule-expansion(X_R, X_R^*)
    if E_R **then**
      return ϑ_R^G, R_{Left}, R_{Right}
    else
      `// If attempt to expand` R `has failed, delay its timing to make a new attempt`
      $N\text{min} \leftarrow N\text{min} \cdot N\text{min}_\gamma$
    end
  end
  return ϑ_R^G, $E_R = False$
end

Figure 5.11: Rule R: digest observations.

Algorithm 6: Rule $R(v)$: digest observations

Input : X_R: Input data matrix $[N \times p]$ whose observations belongs to this rule R
$w_R(v)$: Vector of weights for the observations belonging to R in the context of local node v
$\rho(v)$: Vector of adaptive penalties for all the features in the context of local node v
Output: $\vartheta_R^{\text{L}}(v)$: Whether if a local change has been detected for node v in the context rule of R or not

begin
  Generate prediction Y_R
  Update error statistics $\mathcal{E}_R(v), \mathcal{E}_R(v,l)$ for rule $R(v)$ and for every learner $l \in \mathcal{L}$ using Y_R
  $\vartheta_R^{\text{L}}(v) \leftarrow$ change-detection$(Y_R, \mathcal{E}_R(v))$
  Update learners $l \in \mathcal{L}$ using new data X_R according to $w_R(v), \rho(v)$
  return $\vartheta_R^L(v)$
end

Figure 5.12: Rule $R(v)$: digest observations.

Algorithm 7: Rule $R(v)$: predict observations

Input : X_R: Input data matrix $[N \times p]$ whose observations belongs to this rule R
$\mathcal{P}_{\text{PredictionInterval}}$: Probability for the prediction interval
Output: $\hat{Y}, \hat{Y}_P$: Forecasts for every $v \in \mathcal{R}.V$, and prediction intervals

```
1 begin
2     foreach learner l in ℒ do
3         Y(l), Ŷ_P(l) ← l.predict(X_R, 𝒫_PredictionInterval)
4     end
5     Combine predictions Y(l), Ŷ_P(l) into Ŷ, Ŷ_P
6     return Ŷ, Ŷ_P
7 end
```

Figure 5.13: Rule $R(v)$: predict observations.

Algorithm 8: Filter spurious outliers using thresholding

Input : X: Input data matrix $[N \times p]$
$\mathcal{F}_{\tau^-}$: Vector of features' lower bounds (for those which are known)
$\mathcal{F}_{\tau^+}$: Vector of features' upper bounds (for those which are known)
Θ_ω: Feature statistics in the context of a recent sliding window ω
IQR_τ: Threshold used for the interquartile range method
Output: X^F: Filtered X

```
1 begin
2     X^F ← Filter features values in X using lower limits in ℱ_τ⁻ for those which are known
3     X^F ← Filter features values in X using upper limits in ℱ_τ⁺ for those which are known
4     X^F ← Filter (for the rest of) features values using lower and upper limits given by
        IQR(X^F, Θ_ω, IQR_τ)
5     return X^F
6 end
```

Figure 5.14: Filter spurious outliers using thresholding.

Algorithm 9: Change detection

Input : x: Vector of standardized data
θ: vector of statistics for x and Page-Hinkley hyperparameters
Output: ϑ: Whether if a change has been detected in x or not

begin
  $T \leftarrow \text{length}(x)$
  foreach *time* t *in* T **do**
    $\bar{x}_t \leftarrow \text{mean}(\bar{x}_t, x_t)$
    $m_U^t \leftarrow m_U^t + (x_t - \bar{x}_t - \gamma)$
    $m_L^t \leftarrow m_L^t + (x_t - \bar{x}_t + \gamma)$
    $M_U^t \leftarrow \min(M_U^t, m_U^t)$
    $M_L^t \leftarrow \max(M_L^t, m_L^t)$
    $\vartheta^+ \leftarrow (m_U^t - M_U^t) > \lambda$
    $\vartheta^- \leftarrow (M_L^t - m_L^t) > \lambda$
    if $\vartheta^+ \vee \vartheta^-$ **then**
      break
    end
  end
  return ϑ
end

Figure 5.15: Change detection.

5.2.9 Classify Traffic states

Figure 5.16.

Algorithm 10: Classify Traffic states

Input : X: Input data matrix $[N \times p]$
Output: $\mathcal{T}_{\mathcal{S}}$: Classification of traffic states for each observation in the context of every node v

begin
  foreach *local node* v *in* V **do**
    $\mathcal{T}_{\mathcal{S}}(v) \leftarrow$ the most probable local traffic state using the calibrated components in the trafficstates-model(v) with data $X(v)$
  end
  return $\mathcal{T}_{\mathcal{S}}$
end

Figure 5.16: Classify Traffic states.

5.2.10 Filling in of missing data using the basic approach based on graph patterns

Figure 5.17.

Algorithm 11: Filling in of missing data using the basic approach based on graph patterns

Input : X: Input data matrix $[N \times p]$
Θ: Global statistics for every feature in X
Θ_R: Statistics in the context of every rule R for every feature in X
Output: X_{Clean}: X data with filled missing data

```
1 begin
2 |  Fill in every missing data feature in X with their corresponding expected values from Θ_R
3 |  return X_Clean
4 end
```

Figure 5.17: Filling in of missing data using the basic approach based on graph patterns.

5.2.11 Generating graph features

Figure 5.18. *Graph feature* is an abstract variable that is built using the network flows and the definition of the network graph, and using the scoring function described in Section 5.1.1.2. They are used in the context of the pattern mining processes to characterize such patterns.

Algorithm 12: Generating graph features

Input : X: Input data matrix $[N \times p]$
Output: X^*: Graph-features data matrix $[N \times q]$ q is the number of graph connections

```
1 begin
2 |  foreach local node v in V do
3 |  |  Get upstream nodes v_u of v
4 |  |  foreach time step Δt in T do e.g. from t = 0 until t = 60 in Δt steps
5 |  |  |  X*(v, v_u, t) ← v(t) − v_u(t)
6 |  |  end
7 |  end
8 |  return X*
9 end
```

Figure 5.18: Generating graph features.

5.2.12 Measure outlierness based on graph features

Figure 5.19.

5.2.13 Incident detection using the probabilistic traffic states approach

Figure 8.5 and Figure 8.6.

Algorithm 13: Measure outlierness based on graph features

Input : X^*: Graph-features data matrix $[N \times q]$ q is the number of graph connections
Θ^*_R: Graph feature statistics in rule R
Output: $\mathcal{O}(v)$: Outlierness degree in every node v of the graph

begin
  foreach *local node* v *in* V **do**
    Get upstream nodes v_u of v
    $\mathcal{O}_v(v) \leftarrow$ Calculate Z-score(v) using data in $X^*(v, v_u)$ according to statistics in Θ^*_R within graph pattern R
  end
  return $\mathcal{O}(v)$
end

Figure 5.19: Measure outlierness based on graph features.

Algorithm 14: Incident detection using the probabilistic traffic states approach

Input : $\mathcal{T}_{\mathcal{S}}$: Traffic states for every node v
R_{ID}: Name or identifier for the corresponding graph patterns / rules
Output: $ID(v)$: Severity of incidents in every spot of the network graph

begin
  `// The first stage consists of assigning a raw score based on the identification of anomalous congestion spots (congestions which are non-recurrent in terms of probability) through the network graph`
  foreach *local node* v *in* V **do**
    $v_u \leftarrow$ upstreams(v)
    $v_d \leftarrow$ downstreams(v)
    $v^L \leftarrow$ lags(v)
    `// Find suspicious observations` $n \in N$ `where` v `spot is congested and any of its neighbour spots` v_x `are not`
    $\mathcal{S}_{idx} \leftarrow \mathcal{T}_{\mathcal{S}}(v)$ is congested
    $\wedge\, \exists\, v_x : \left(\mathcal{T}_{\mathcal{S}}(v_{u_1}) \vee ... \vee \mathcal{T}_{\mathcal{S}}(v_{u_n}) \vee \mathcal{T}_{\mathcal{S}}(v_{d_1}) \vee ... \vee \mathcal{T}_{\mathcal{S}}(v_{d_n}) \text{ is not congested}\right)$
    foreach *observation* n *in* $\mathcal{S}_{idx}$ **do**
      Get the set of non-congested neighbour spots $v_{NC} \subseteq \{v_u, v_d\}$ of v
      $v^L_{NC} \leftarrow$ lags(v_{NC})
      `// Which is the joint probability for the current traffic state and its lags in` v, v^L `given the evidence` E`?`
      `// Evidence` E `is composed of those traffic states from non-congested neighbours` v_{NC}, v^L_{NC} `and the current graph pattern (rule)` R
      $E \leftarrow (\,\mathcal{T}_{\mathcal{S}}(v_{NC}) \cup \mathcal{T}_{\mathcal{S}}(v^L_{NC})\,) \;\cup\; R_{ID}$
      `// Joint probability distribution` J_v `of` $v \cap v^L$ `given the evidence in` E
      $\mathcal{J}_v \leftarrow \mathrm{P}(\mathcal{T}_{\mathcal{S}}(v) \cap \mathcal{T}_{\mathcal{S}}(v^L) \,|\, E)$
      `// Calculate outlierness raw score` $Oraw_{v,n}$ `using the joint probability distribution` $\mathcal{J}_v$
      $\hat{O}raw_{v,n} \leftarrow \frac{(\max \mathcal{J}_v) - (\mathcal{J}_v(\mathcal{T}_{\mathcal{S}}(v) = \tilde{v} \cap \mathcal{T}_{\mathcal{S}}(v^L) = \tilde{v}^L))}{\max \mathcal{J}_v}$
    end
  end

Figure 5.20: Incident detection using the probabilistic traffic states approach. Part I.

(15)
    // The second stage consists of assigning a final score for those *enough* raw anomalous spots
(16) $\hat{O}_\tau raw \leftarrow 0.50$
(17) **foreach** *observation* n *in* $\mathcal{S}_{idx}(v)$ **do**
(18) For those *enough* anomalous spots $\hat{O}raw > \hat{O}_\tau raw$
    // The final score is composed of three different contributions: ($\hat{O}_1 \in [0,1]$) a relative severity of the existing congestion (50%), ($\hat{O}_2 \in [0,1]$) the temporal recurrence of the existing non-recurrent congestion (25%), and ($\hat{O}_3 \in [0,1]$) the spatial propagation of the incident through the network graph (25%)
(19) $\hat{O}_1 \leftarrow \mathcal{T}_{\mathcal{S}}(v) - H(\mathcal{T}_{\mathcal{S}}(v_{NC}))$
    // Here $H(x)$ is the harmonic mean of vector x
(20) $t_{max} \leftarrow 30$ minutes
(21) $\hat{O}_2 \leftarrow \min(\hat{O}_2 + \Delta t / t_{max}, 1)$
(22) Find recursively congested neighbours of v in order to assess the spatial propagation of the incident
(23) $\hat{O}_3 \leftarrow$ number of nodes linked to v in a congested state through the graph up to a given maximum
(24) $ID(v) \leftarrow 0.50 \cdot \hat{O}_1 + 0.25 \cdot \hat{O}_2 + 0.25 \cdot \hat{O}_3$
(25) **end**
(26) **return** $ID(v)$
(27) **end**

Figure 5.21: Incident detection using the probabilistic traffic states approach. Part II.

6 Validation of Adarules under different change scenarios

The aim of this chapter is to evaluate the performance of the Adarules algorithm described in the previous Chapter 5 in terms of forecasting accuracy, model complexity and interpretability.

To this end, the validation has been performed using the two datasets described in previous sections. More specifically, the evaluation is performed on both networks: the M4/M7 motorways network from Sydney (Australia) and the urban network from the city of Santander (Spain). The M4/M7 network consists of 455 double-loop detectors spread uniformly at every 500 metres measuring traffic flow, occupancy and speed; while the Santander network is measured by a total of 489 single-loop detectors that observe the traffic flows and occupancies. All of these detectors have been used as input information for Adarules in order to capture the traffic dynamics and to build the forecasts. However, the evaluation of the forecasting error has only been performed on a selection of both sets of detectors in order to better manage —in terms of computational efficiency and time availability— the large set of experiments and their evaluation. Therefore, a selection of 20 detectors has been chosen for each network. In Figure 6.1 for M4/M7 network, and Figure 6.2 for the Santander network, the whole set of detectors located through the network —in blue— as well as those selected to validate the results —in red— are shown. This selection is given in more detail in the Table 6.1 which contains, for each network, all these detectors' identifiers and their description. The selection of detectors has been done manually by checking in detail both networks, and their geometry with the aim of giving a fair and unbiased representation of each network. The reasons for such selection —instead of evaluating the results on the whole set of detectors— are:

1. Choosing detectors spread through the network and with different surrounding geometry settings. This leads to obtaining a representative picture of the network by focusing on those key places through the road network, instead of having certain network areas over-represented simply because there are a larger number of detectors placed in such areas.
2. Hastening the whole evaluation process without sacrificing the validity of the results.

For both networks, a time period of two years has been used for the experiments, splitting such data in training data to build the models while leaving the rest only to validate the forecasting generalization. This data splitting procedure depends on the experiment and the baseline model, and thus it will be described later. The time period for the M4/M7 dataset spans from January, 2015 – December, 2016, and the time period for the Santander dataset spans from January, 2016 – December, 2017. Different flow profiles for the selected validation detectors can be observed in Figure 6.3 for the M4/M7 network and in Figure 6.4 for the Santander network using the two-year datasets, which gives certain insight about the daily traffic dynamics by showing different flow profiles with the associated variability and a strong skewness depending on the hour of the day.

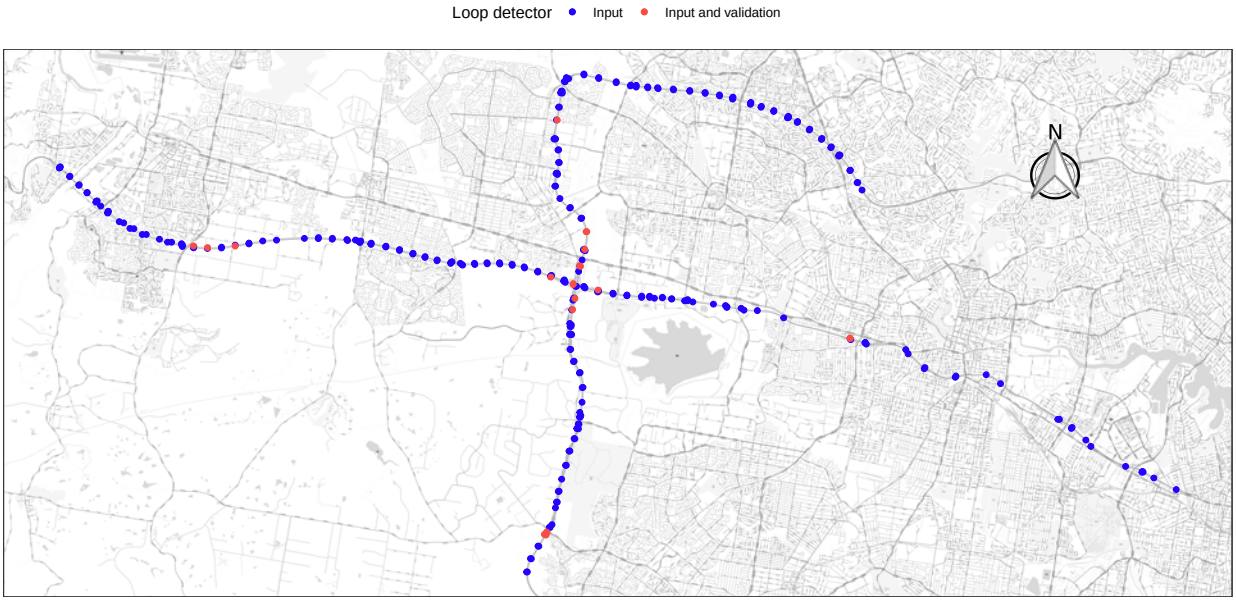


Figure 6.1: Network layout for the M4/M7 motorways in Sydney with the position of all the detectors used as input information for Adarules —as blue points—, as well as those which are used to evaluate the forecasting accuracy —as red points—.

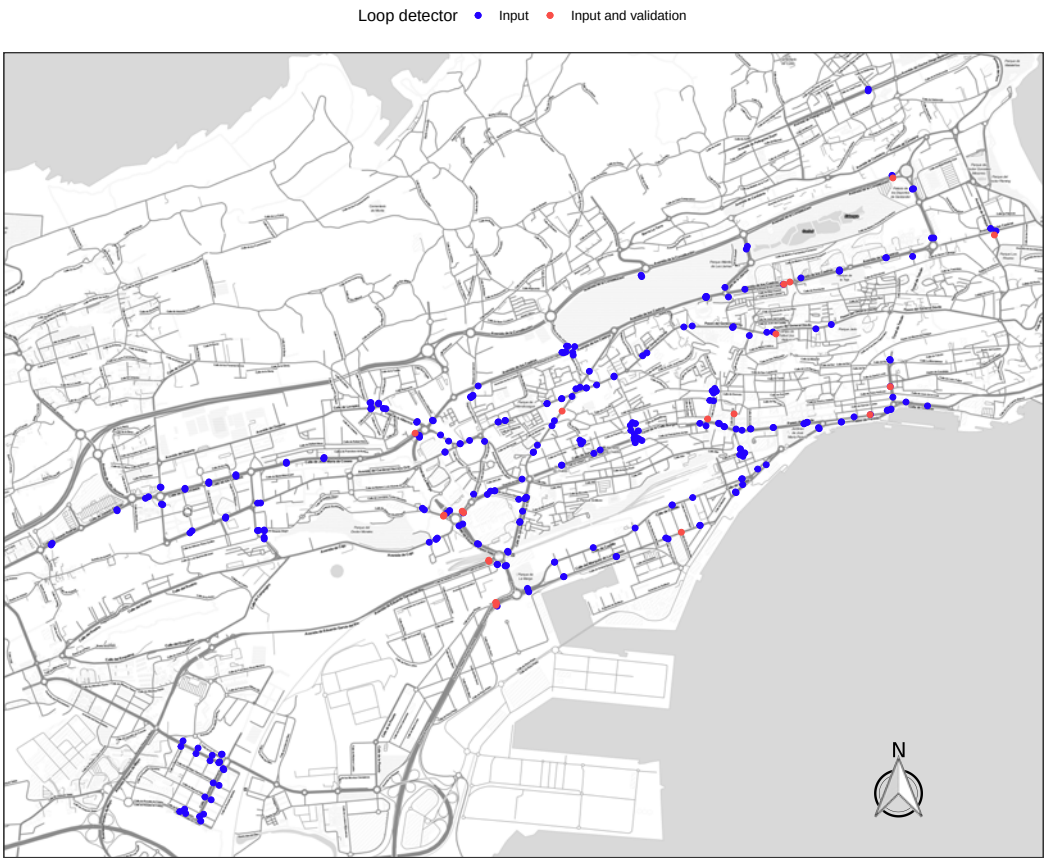


Figure 6.2: Network layout for the Santander urban network with the position of all the detectors used as input information for Adarules —as blue points—, as well as those which are used to evaluate the forecasting accuracy —as red points—.

Table 6.1: Selection of detectors for the evaluation in the different experiments for Adarules validation.

Identifier	Latitude	Longitude	Road type	Details
			Santander urban network	
1001	43.45361	-3.829633	2-lane arterial [70 km/h]	Entering the city before a roundabout
1021	43.45116	-3.829085	1-lane on-ramp [80 km/h]	Exiting the city after a roundabout
1022	43.45111	-3.829028	2-lane arterial [50 km/h]	Exiting the city
1023	43.45099	-3.829055	2-lane arterial [50 km/h]	Entering the city
1027	43.45367	-3.829654	2-lane arterial [60 km/h]	Exiting from the city after roundabout
1031	43.45534	-3.813856	3-lane road [50 km/h]	Between an intersection and a traffic light
1035	43.45659	-3.831832	3-lane road [40 km/h]	Exiting the city between traffic light and roundabout, including bus lane
1036	43.45629	-3.833374	2-lane road [50 km/h]	Entering the city between traffic light and roundabout
1037	43.45636	-3.833322	2-lane road [50 km/h]	Exiting from the city between roundabout and traffic light
1908	43.45650	-3.831721	3-lane road [50 km/h]	Entering the city between two roundabouts with traffic light
2014	43.46208	-3.811698	2-lane road [50 km/h]	Between two pedestrian crossings, in the city centre
2034	43.46400	-3.796720	2-lane road [50 km/h]	Between two traffic lights, in the city centre
2057	43.46238	-3.809530	2-lane road [40 km/h]	Between a traffic light and an intersection
2910	43.46234	-3.798376	3-lane road [50 km/h]	Between two traffic lights, including bus lane
3027	43.47023	-3.804956	2-lane road [50 km/h]	After a roundabout
3035	43.47303	-3.788181	2-lane road [50 km/h]	Before a traffic light
3078	43.46123	-3.835693	2-lane road [50 km/h]	Exiting from the city between a roundabout and a pedestrian crossing
3085	43.47643	-3.796507	2-lane road [50 km/h]	Exiting before pedestrian crossing and roundabout

Table 6.1: Selection of detectors for the evaluation in the different experiments for Adarules validation. *(continued)*

Identifier	Latitude	Longitude	Road type	Details
3088	43.47010	-3.805456	2-lane road [50 km/h]	Before a join and a roundabout
3924	43.46255	-3.823636	2-lane road [50 km/h]	Between two traffic lights
M4/M7 motorways network				
MS004029B	-33.81531	150.960449	3-lane motorway	Heading East at the end of M4
MS004029X	-33.81512	150.960290	1-lane off-ramp	Heading East, exiting the M4 before MS004029B
MS004048E	-33.79967	150.862664	1-lane on-ramp	Heading East, joining M4 after exit from M7 in Light Horse Interchange
MS004050A	-33.79788	150.853041	3-lane motorway	Heading West in M4 in the middle of Light Horse Interchange
MS004050B	-33.79761	150.853036	3-lane motorway	Heading East in M4 in the middle of Light Horse Interchange
MS004052E	-33.79552	150.844381	1-lane on-ramp	Heading West, joining M4 after exit from M7 in Light Horse Interchange
MS004076A	-33.78550	150.722001	3-lane motorway	Heading West in M4
MS004078B	-33.78600	150.711301	3-lane motorway	Heading East at the start of M4
MS004079E	-33.78547	150.705801	1-lane on-ramp	Heading East, joining M4 and before MS004078B
MS009023A	-33.87778	150.842866	2-lane motorway	Heading South at the end of M7
MS009023B	-33.87760	150.842691	2-lane motorway	Heading South at the start of M7
MS009023E	-33.87857	150.842546	1-lane on-ramp	Joining M7 after MS009023A
MS009023X	-33.87832	150.841876	1-lane off-ramp	Exiting M7 before MS009023B
MS009040A	-33.80595	150.852810	2-lane motorway	Heading South in M7 after Light Horse Interchange entrance
MS009041E	-33.80240	150.853716	1-lane on-ramp	Heading South in M7 coming from Light Horse Interchange
MS009043A	-33.79203	150.855874	2-lane motorway	Heading South in M7 through Light Horse Interchange

Table 6.1: Selection of detectors for the evaluation in the different experiments for Adarules validation. *(continued)*

Identifier	Latitude	Longitude	Road type	Details
MS009044E	-33.79183	150.855468	1-lane on-ramp	Heading North in M7 coming from Light Horse Interchange
MS009045A	-33.78659	150.857515	2-lane motorway	Heading South in M7 heading to Light Horse Interchange
MS009046A	-33.78086	150.858310	2-lane motorway	Heading South in M7 heading to Light Horse Interchange
MS009046B	-33.78088	150.858043	2-lane motorway	Heading North in M7 coming from Light Horse Interchange
MS009055A	-33.74490	150.846849	2-lane motorway	Heading South in M7 after the entrance MS009057E

The first set of experiments aim to determine which learning configuration —either single-task or multi-task learning— performs better in Adarules concerning both the pattern mining process to create rules and the learning process to build the forecasting models. To this end, the comparison is assessed in terms of forecasting accuracy (R^2, GEH, RMSE and normalized RMSE) and model complexity measured as the number of rules identified. The evaluation is performed in both network datasets, for every available traffic measurement (traffic flow, occupancy, and speed) and for two forecasting horizons —15 and 60 minutes ahead—. Two learning strategies are evaluated and compared:

- **Single-task versus multi-task rule mining**: This comparison aims to measure the exploitation of the existing spatial information in the problem. The idea is to measure the effect, in accuracy and complexity, of letting Adarules to unveil the patterns in the network graph using only limited information —i.e. only the flow distributions from the adjacent nodes from each of the twenty nodes being evaluated in a single-task approach—, or using the complete network graph information —i.e. all the nodes in the network graph with their directed connections in a multi-task approach—. The difference lies in the network scope used to identify new graph patterns —rules— as well as determine when these become outdated. The underlying network traffic state information is equally available to both approaches, i.e. forecasting models can use the information from all the detectors in the network in order to perform the forecasts.
- **Single-task versus multi-task forecasting models**: This comparison aims to measure the exploitation of the existing temporal information in the problem. In this case, the idea is to measure the effect over the accuracy of the forecasting models —the sparse model for spatiotemporal correlations— of using a single-task or multi-task learning approach. The multi-task here refers to jointly learning these spatiotemporal correlations for multiple forecasting steps ahead —e.g. 15, 30, 45 and 60 minutes ahead— instead of learning them separately for each one of them. Same as before, the underlying network traffic state information is equally available to both approaches, i.e. forecasting models can use the information from all the detectors in the network in order to perform the forecasts.

Second, the strategy combination of single-task and multi-task learning —in both pattern mining and forecasting models— which achieves a better ratio of accuracy and low complexity is used to afterwards perform a battery of tests with different change scenarios. Namely, this battery of tests comprehends:

1. A scenario with no artificial changes induced, i.e. real data from the two subsequent years is used. Thus, only the *real* change —either in the demand or the supply— is present for this

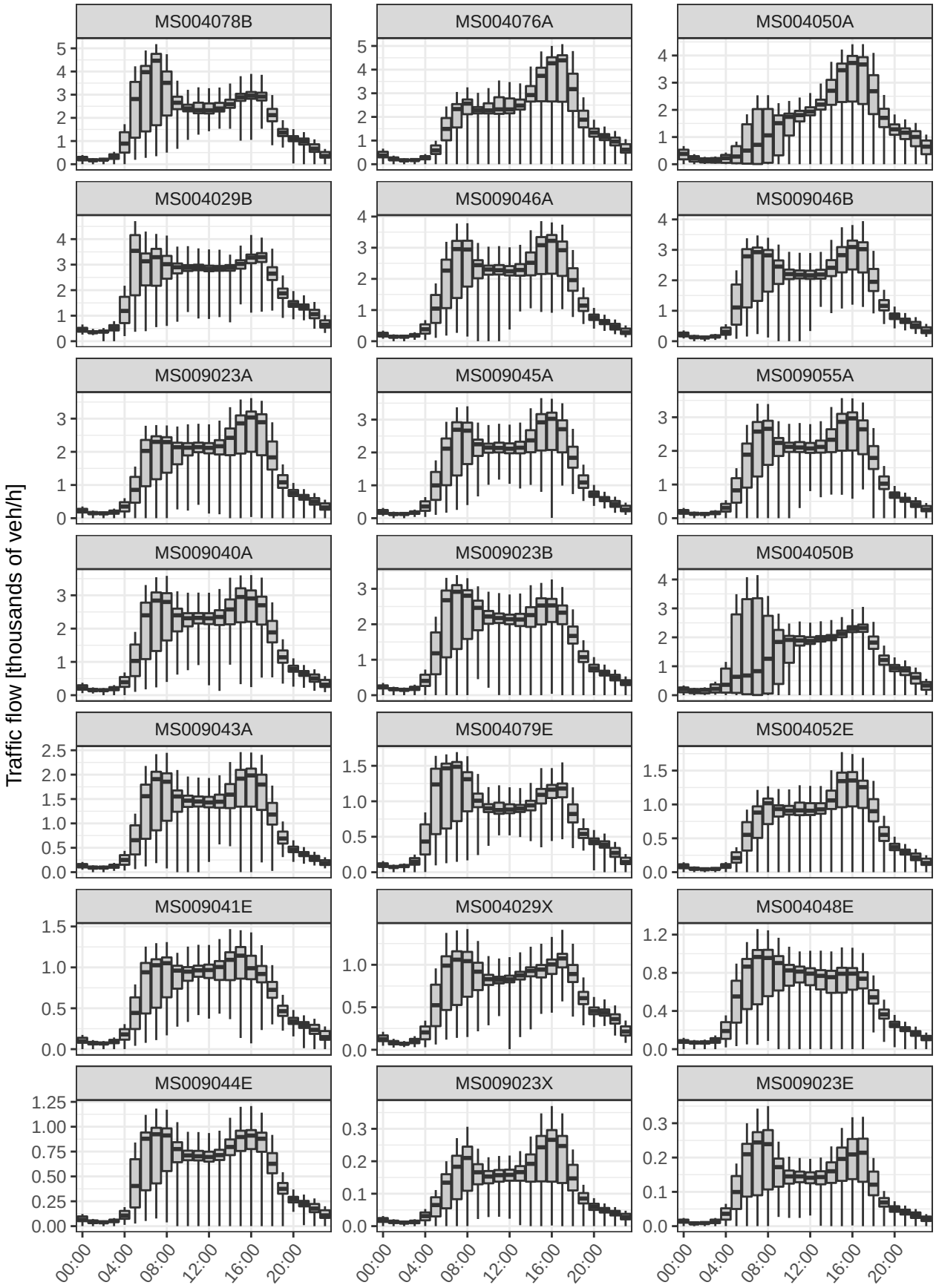


Figure 6.3: Traffic flow for the detectors used in validation for the M4/M7 network, showing the temporal dynamics summarized over the two-years period. Boxplot reflects the interquartile range (25^{th} and 75^{th} percentiles) with the median (50^{th} percentile) as the horizontal line. Outlying lines show the range.

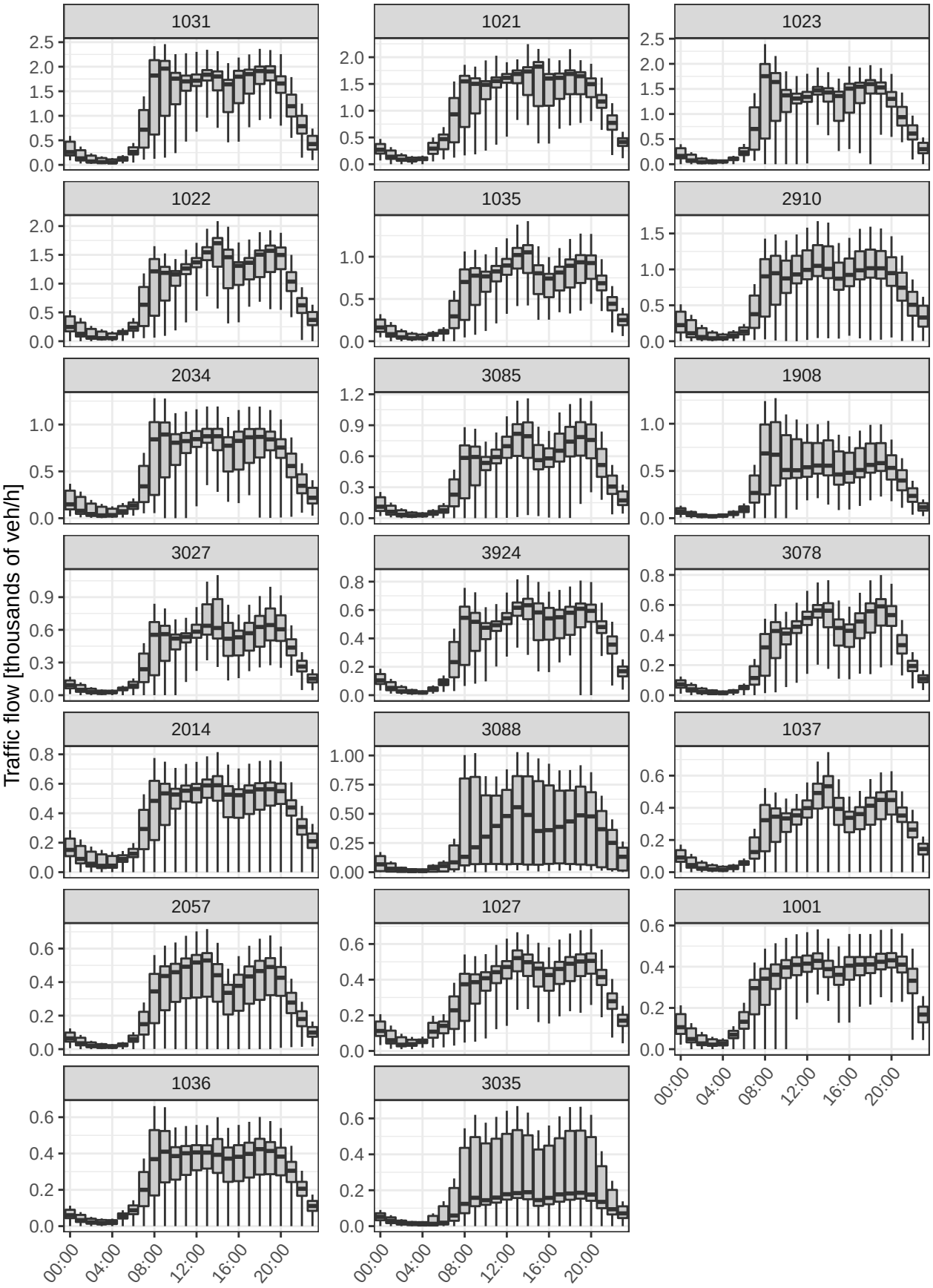


Figure 6.4: Traffic flow for the detectors used in validation for the Santander network, showing the temporal dynamics summarized over the two-years period. Boxplot reflects the interquartile range (25^{th} and 75^{th} percentiles) with the median (50^{th} percentile) as the horizontal line. Outlying lines show the range.

experiment in the datasets.

2. A scenario with explicit no-drift during the second year. This means that the first year on each dataset is the real one, but the subsequent year data is fake by using a specific month of the second year —e.g. May— and replicating it for twelve times. The goal of this experiment is to test the performance in an absolutely no-change scenario, i.e. how the forecasting accuracy and the model complexity evolves over time.
3. A first year with the real data, and then starting from the second year and every two months (January, March, May, July, September, November) a fake change is introduced over all the network by selecting 200 detectors at random where the traffic variables —flow, occupancy, and speed— from 100 of these detectors are incremented by a 4% while the other 100 detectors experience a 4% decrease in the traffic variable. This is maintained until, two months later, another round of smooth changes takes place while accumulating the one from the previous swapping. The goal is to determine the Adarules ability to react and adapt to these gradual changes.
4. A first year with the real data, and then starting from the second year and every two months a fake change is introduced over all the network by swapping the AM and PM periods. This means that every two months (January, March, May, July, September, November) the traffic is swapped and, thus, traffic during the night takes place during the day and vice versa. This is maintained for two months until the next swap takes place. The goal is to determine the Adarules ability to react and adapt to these abrupt changes.
5. A first year with the real data, and then starting from the second year and every two months a fake change is introduced over all the network by swapping 100 detectors identifiers selected at random. This means that every two months (January, March, May, July, September, November) the traffic from these detectors is swapped with others in the network. This is maintained until, two months later, another swapping takes place while accumulating the one from the previous swapping. The goal is to determine the Adarules ability to react and adapt to these extreme abrupt changes.

6.1 Evaluation metrics

All these experiments are evaluated in both network datasets, for every available traffic measurement (traffic flow, occupancy, and speed) and for two forecasting horizons —15 and 60 minutes ahead—.

Experimental results, including those from the first and the second stage of the experiments setup,

are evaluated using different criteria.

- *Forecasting accuracy*

Forecasting accuracy or its inverse the forecasting error is the main driving motivation for this thesis focused on real-time short-term traffic prediction. The metric used for this criteria is based on the root mean squared error (RMSE) and its normalized version (nRMSE) using the range of the forecasting variable.

The RMSE is a common performance indicator in the machine learning domain and it is defined by:

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

The normalized RMSE takes the RMSE and divides it by the range of the variable:

$$\text{nRMSE} = \frac{\text{RMSE}}{y_{max} - y_{min}}$$

The nRMSE is an easy-to-understand performance indicator that relates the forecasting error with a normalized value in $[0, 1]$ or $[0, 100]\%$, where the lower value is the better. It makes also possible and easier to compare the performance between different datasets and models. We prefer using RMSE as it penalizes large differences more than absolute metrics such as the Mean Absolute Error (MAE). This is important to give more importance to large forecasting errors in situations of high-traffic instead of, for instance, forecasting error in a situation of low traffic —e.g. during the night—. This effect is also taken into account when doing the normalization using the range instead of other alternatives such as the mean or the variance. Moreover, RMSE is also consistent with the typical squared loss function used to train the forecasting models [107].

Other common forecasting accuracy metrics, especially in the transportation domain, are the GEH-statistic [86] and the well-known coefficient of determination R^2 [29]. Nevertheless, they are strongly correlated with the RMSE used in this thesis since they all make use of the squared error. The difference lies in how they normalize the resulting forecasting error, and we have preferred the normalization by the range of the variable for the reasons given in the previous paragraph.

The forecasting accuracy is visually analyzed using a specific type of plot. This type of plot shows —for a given traffic variable, network dataset, and forecasting horizon— the temporal evolution —in monthly bins— of a given key performance indicator (KPI) such as the nRMSE. For each monthly bin, a distribution is shown —*violin plot*— containing the $N = 20$ KPIs for each of

the selected network points for validation which have been aggregated over the whole monthly period. In addition, a black horizontal line is depicted representing the median —50% of the sample— of every group —a given learning approach, baseline, ...—. These plots are placed in Appendix II: Detailed results for the validation of Adarules under different change scenarios for the interested reader, with the aim of reducing an excessive length of the current chapter. Instead, within the current chapter, the information regarding the median per group and monthly period is summarized in a table per experiment containing all the network datasets, traffic variables and forecasting horizons.

- *Model complexity*

As Adarules is a non-parametric approach that evolves and thus can increase or decrease its size over time, it is important to measure the current complexity of the system. To this end, the number of rules and its evolution over time is measured and used as a performance indicator so it can be compared between different Adarules runs.

- *Intepretability*

Despite the fact that efforts have been put during the development of this thesis to drive the choices towards methods and models with higher transparency to user end-user, however, it is still difficult to objectively measure the **human** interpretability of a machine-learning model.

For this reason, the best way to assess this criterion is simply by getting feedback from the expert user by showing the modelling results —i.e. the identified rules in the graph and the spatiotemporal correlations in the forecasting models— to the traffic engineers at Aimsun to ask them how interpretable and consistent they seem to them.

6.2 Baselines

Two baseline approaches have been considered in order to compare them to Adarules whose codenames are **HA** and **ANA**:

1. **HA**: An historical average or seasonal naïve forecast $\bar{y}(DoW, t)$ for each considered traffic measurement (traffic flow, occupancy and speed) as a function of the day of the week or holidays (Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday, Holiday) and the time of the day (hour and minute).

2. **ANA**: The current existing methodology in Aimsun Live to build the data-driven short-term forecasting models. It consists on estimating the most predictive spatial correlations for every time of the day (HH:MM), every detector node v, every forecasting variable, and every forecasting horizon $\hat{t}_h$, but independently of the day of the week.

These two baselines have been implemented using a block-evolution updating schema [98], i.e. models perform periodic blind updates with a sliding window:

- Yearly: using the first year of the dataset to learn and validating over the second year.
- Quarterly: using three months to learn and validating on the following three months, then using these last three to learn and repeat four times —during the second year—.
- Monthly: using one month to learn and validating on the following month, then using that last month to learn and repeat twelve times —during the second year—.

6.3 Missing data

Given how usual the traffic data measurement devices are faulty, it is important to notice the proportion of missing data for each network dataset and traffic variable as shown in Figure 6.5. Therefore, it can be assessed the correlation between this proportion of missing data and its impact on the forecasting accuracy.

6.4 Adarules: pattern mining using a single-task or a multi-task approach

The first set of experiments aim to compare the ability to completely exploit the network spatial information —*multi-task mining* (MTM)— or not —*single-task mining* (STM)—- during the rule mining procedure. More specifically, the idea is to measure the effect, in accuracy and complexity, of letting Adarules to unveil the patterns in the network graph using only limited information —i.e. using only the flow distributions from the adjacent nodes from each of the twenty nodes being evaluated in a single-task approach (STM)—, or using the complete network graph information —i.e. all the nodes in the network graph with their directed connections in a multi-task approach (MTM)—. The difference lies in the network scope used to identify new graph patterns —rules— as well as determine when these become outdated. On the other hand, the underlying network traffic state information is equally available to both approaches, i.e. forecasting models can use the information from all the detectors in the network in order to perform the forecasts.

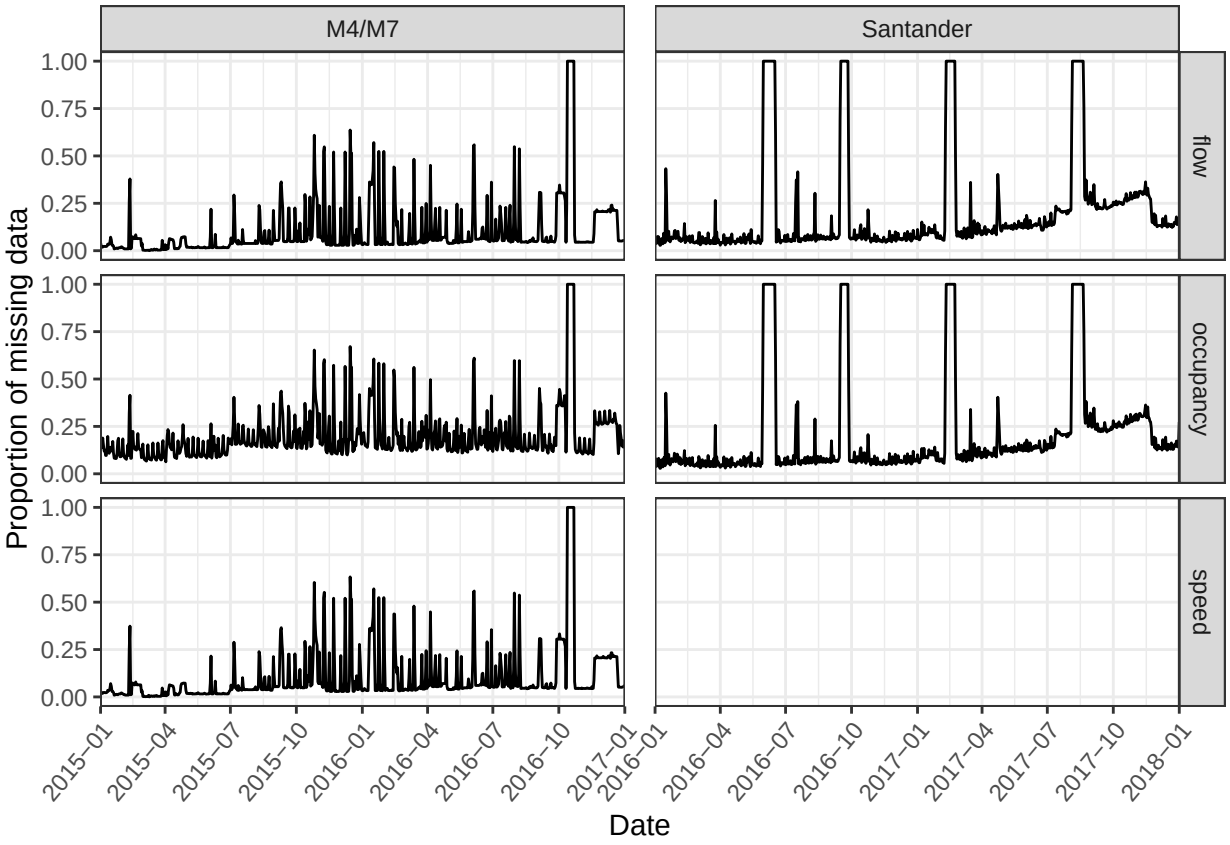


Figure 6.5: Proportion of the missing data for each network dataset and traffic variable as a function of the date.

In both cases —STM and MTM—, they have available the same set of contextual qualitative variables —day of the week, holidays, the hour of the day, etc.— to perform the splitting evaluations. As described in the previous paragraph, the difference lies in the network spatial used to evaluate such splitting evaluations. At a first glance, it may seem that limiting the spatial scope for the pattern mining process in STM could yield limited observability of the network thus perhaps not being able to observe some traffic dynamics effects through the network. On the other hand, such STM may be able to faster capture and in a more accurate way such patterns as it is observing only —and thus, giving more weight— the traffic surrounding the site where a given detector is placed.

The evaluation is performed in both network datasets, M4/M7 and Santander, for every available traffic measurement (traffic flow, occupancy, and speed) and for two forecasting horizons —15 and 60 minutes ahead—.

The first criterion used to assess both approaches —STM and MTM— relies on the complexity of the resulting model. This complexity is measured by the total number of rules found during the learning using the two years dataset for each network. The goal is both to evaluate how fast the number of rules increases depending on the mining approach, and which is the total number of rules. Both rule mining approaches start with one rule —*default rule*— during the first 15 days until enough data has been gathered before mining for rules. In Figure 6.6, it can be seen: the total number of rules for the MTM approach in red, and (a) the total number of rules over all the $N = 20$ rulesets for the STM approach in the dashed blue line —in a scale ($\cdot 10^{-1}$)—, as well as (b) the average number of rules per ruleset in the blue solid line together with the range of minimum and maximum number of rules among the $N = 20$ rulesets denoted by the blue ribbon.

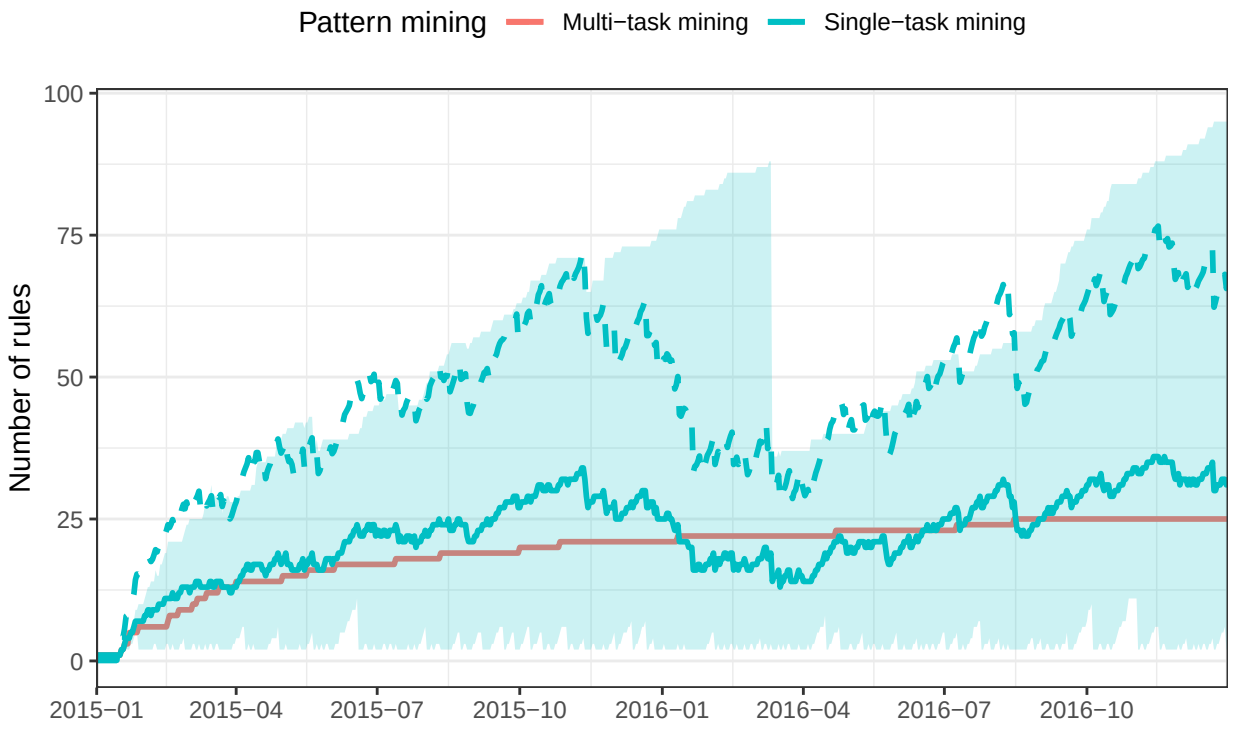
In the figure, it can be seen that both approaches unveil rules at a similar pace starting from the 15^{th} day in both networks. In the case of the M4/M7 network, the MTM approach decreases its rule mining pace after observing a couple of weeks, starting to raise the pace again until the 2015/04 where the rule mining slows down by increasing the rules on a logarithmic scale until number of rules is stabilized in 25 rules over the last four months —2016/07 to 2016/12—. In the case of Santander network, the MTM approach increases the number of rules linearly as a function of the time until a stabilization point is reached after observing the first three months. At this point, the number of rules increases slowly until reaching 36 rules at the end of the second year —2017/12—. This matches the intuitive rationale of having more number of rules describing a more complex network —Santander urban network— as the existing traffic dynamics are richer compared to a motorway network. Moreover, it is also interesting to observe the small slowdown in the M4/M7 MTM rule mining at the beginning given that it must be challenging to find graph pattern until

more data is observed given the lack of observability beyond entrance/exits and boundaries of the motorway. On the other hand, the STM approach has a similar trend at the beginning but then becomes much more unstable having abrupt changes in the number of rules —i.e. the number of rules either increases or decreases— probably due to spurious observed effects which are regularized in the case of MTM. Moreover, the total number of rules represented by the dashed line in a scale of $(\cdot\, 10^{-1})$ shows that counting all the existing rules becomes difficult to be interpreted —even though many of them will probably overlap or be redundant as they are simultaneously found at different spots of the network— with some time periods reaching a total of 600 over all the $N = 20$ rulesets in both networks.

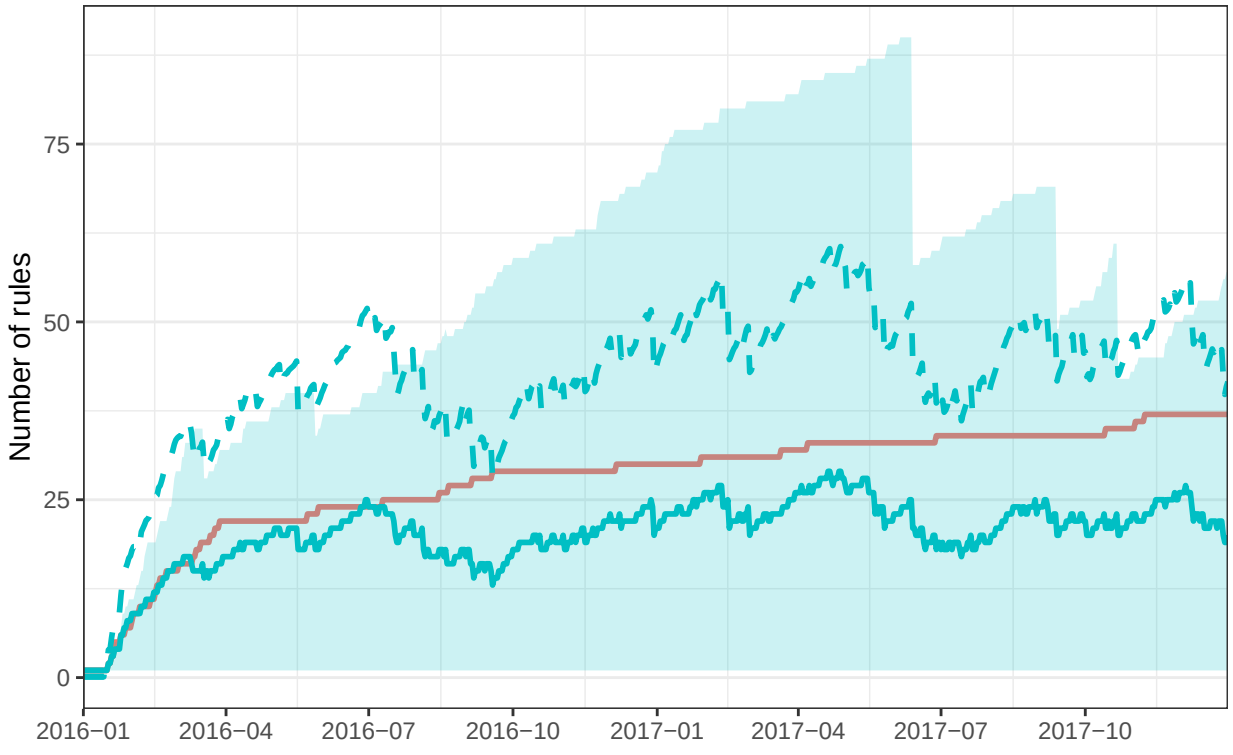
In this case, the assessment in terms of forecasting accuracy relies on the use of the normalized RMSE (nRMSE) as it is an easy-to-understand performance indicator that relates the forecasting error with a normalized value in $[0, 1]$ or $[0, 100]\%$, where a lower value is better.

One figure per forecasting traffic variable is included: Figure 9.1 for the flow, Figure 9.2 for the occupancy and Figure 9.3 for the speed. Each of these figures has subfigures for the two network datasets and for each forecasting horizon, resulting in four subfigures with the exception of speed which has only two because there is no measured speed data in the Santander dataset.

In these figures, it can be seen the evolution of the forecasting error —aggregated in monthly bins— during the two years for each network dataset. In Figure 9.1, the nRMSE is shown for the traffic flow forecasting. In the case of the M4/M7 network, it is noticeable a similar learning pace at the beginning of the two years. However, this pace changes after the fourth month (2015/05) and the MTM approach achieves a much lower nRMSE median value than its STM counterpart. It is also noticeable how not only the median value but also the shape of the distribution of the different nRMSE values across all the validation detectors is significantly different too, as the STM approach always get a much wider distribution showing that some of the detector forecasts are performing significantly worse compared to the MTM approach. In the case of the Santander network, the difference is not so evident but, still, it can be appreciated that the MTM approach achieves a lower nRMSE median value as well as having a shorter tail in the nRMSE distribution in almost all the months especially more visible during the second year. The situation is similar for both forecasting horizons —15 and 60 minutes— for both network datasets. For the cases of occupancy and speed forecasting, the differences are not that significant but still the tendency is always the same as the MTM approach achieves a lower forecasting error both in terms of the median value as well as the extreme values in the tail of the distributions. This information is also numerically depicted in Table 6.2 where the nRMSE values for the last twelve months are shown for all the network datasets, variables, and forecasting horizons.



(a) M4/M7 network.



(b) Santander network.

Figure 6.6: Comparison of the resulting model complexity for both pattern mining approches using Adarules: single-task and multi-task. Number of identified rules as a function of the time —every iteration corresponds to one day—. In the case of STM, the solid line reflects the average and its ribbon reflects the minimum and maximum number of rules among the $N = 20$ rulesets, whereas the dashed line reflects the total sum of rules $(\cdot 10^{-1})$ across the $N = 20$ rulesets.

The difference in the gain of forecasting accuracy between both networks may be probably due to the fact that learning more realistic patterns in a motorway network is harder when only a very reduced scope of it is observed. Additionally, the limited observability in the STM approach when finding rules may also lead to recognizing spurious effects as true patterns, and henceforth the unstable number of rules over time and the associated drops in the forecasting accuracy.

Lastly, it is also very interesting not only that the MTM is achieving a better forecasting performance in all networks, variables and forecasting horizons compared to the STM approach, but also that it is doing it much more efficiently because the number of identified rules is considerably lower over all the two-years period in both network datasets as previously commented. For this reason, the multi-task mining (MTM) approach has been chosen for the following experiments.

Finally, some important clarifications about the results related to the number of rules shown in Figure 6.6. Among every single execution of Adarules, there cannot be rules' duplicities because the underlying Adarules' decision tree is exhaustive —i.e. every single observation is covered just by one and only one rule—. This happens to be for the MTM execution and every of the STM executions for each detector. However, among the different STM executions —one per detector included in the evaluation— there could be duplicated rules since they are executed independently —e.g. a rule *Monday-Fridays at 7 a.m.* could be found for two different detector sites in independents STM executions—. Nevertheless, by looking at Figure 6.6 it can been observed that the average number rules for STM executions —blue solid line— is similar to the single number obtained for the MTM execution —red solid line—. This fact would validate the hypothesis that every single STM execution is finding the same set of rules that the obtained in the MTM execution. However, by looking at the blue ribbon in Figure 6.6, it can be seen how the range —the minimum and the maximum size of a given ruleset among the different STM executions— has a wide amplitude denoting that some STM executions found a number of rules equivalent to three times the average, while others STM executions were not able to find any valid rule probably because of a lack of data or enough variability on it. This last fact does not occur in the MTM execution since the rules are found at network-level and, hence, there is an implicit knowledge transfer to those network points where there is not enough data. In addition, it can also be observed certain abrupt drops in the upper bound of such blue ribbon, indicating that the maximum number of rules found by Adarules in any given STM execution has decreased due to a hard global change detected thus restructuring the ruleset decision tree entirely. This fact reflects how the simple observation of just the local traffic makes the pattern mining process more sensitive to variations in such local traffic since it lacks from a *regularization effect* provided by observing the traffic dynamics over the entire network.

CONCLUSION: In summary, the MTM approach brings the following benefits over the STM approach:

1. Knowledge transfer to network sites where data is not enough to perform pattern mining.
2. Smaller models, hence:
   a. Easier interpretation for a traffic engineer.
   b. More computational efficient procedures (see Figure 6.3.
3. Regularization effect to make the patterns less sensitive to noise or spurious patterns.
4. Better forecasting accuracy as measured by a lower —in most cases— or same forecasting error at worst.

Table 6.2: Comparison of the forecasting performance measured by nRMSE for both rule mining approaches. The KPIs are aggregated over the results during the last year in each of the datasets.

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
M4/M7 | flow | t = 15 | nRMSE												
STM	1.94	2.24	2.07	1.94	1.78	1.78	1.74	1.74	1.77	1.60	1.71	2.29
MTM	1.49	1.69	1.62	1.62	1.41	1.46	1.48	1.45	1.49	1.38	1.47	1.71
M4/M7 | flow | t = 60 | nRMSE												
STM	6.13	6.04	5.92	5.56	5.07	4.82	4.41	4.63	4.85	4.32	4.52	7.05
MTM	4.84	4.44	4.35	4.31	3.91	3.84	3.62	3.80	3.92	3.63	4.02	5.29
M4/M7 | occupancy | t = 15 | nRMSE												
STM	2.04	2.44	2.81	2.36	2.55	2.49	2.39	2.61	2.77	2.10	2.77	2.81
MTM	1.90	2.20	2.40	2.26	2.33	2.20	2.28	2.44	2.62	1.88	2.63	2.42
M4/M7 | occupancy | t = 60 | nRMSE												
STM	2.54	3.46	3.65	3.63	3.24	3.44	3.33	3.40	3.67	2.72	3.30	3.49
MTM	2.27	3.21	3.03	3.57	2.95	3.18	2.95	3.06	3.45	2.34	3.19	3.38
M4/M7 | speed | t = 15 | nRMSE												
STM	6.13	6.76	6.64	5.89	5.95	6.12	6.09	6.23	6.26	5.63	6.63	6.42
MTM	5.62	6.05	6.23	5.51	5.83	5.88	5.63	5.67	5.94	5.39	5.83	5.78
M4/M7 | speed | t = 60 | nRMSE												
STM	6.58	8.32	8.07	6.97	6.90	7.23	6.72	6.81	7.11	6.48	7.13	7.85
MTM	6.41	7.56	7.99	6.59	6.76	6.95	6.63	6.20	6.49	5.97	6.94	6.53

Table 6.2: Comparison of the forecasting performance measured by nRMSE for both rule mining approaches. The KPIs are aggregated over the results during the last year in each of the datasets. *(continued)*

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Santander | flow | t = 15 | nRMSE												
STM	1.98	1.71	1.60	1.95	1.68	1.82	2.19	1.97	1.84	2.00	1.89	1.99
MTM	1.71	1.37	1.56	1.72	1.69	1.73	1.86	1.94	1.52	1.85	1.76	1.75
Santander | flow | t = 60 | nRMSE												
STM	5.71	3.84	4.10	4.70	4.53	4.68	5.57	4.96	4.60	5.52	5.05	6.10
MTM	4.42	3.45	4.14	4.61	4.55	4.68	5.30	5.30	3.98	4.45	4.14	4.83
Santander | occupancy | t = 15 | nRMSE												
STM	3.12	2.70	2.90	2.91	2.78	2.85	3.67	3.32	3.01	3.74	3.98	4.53
MTM	2.80	2.56	2.73	2.89	2.64	2.85	3.47	2.95	3.06	3.79	3.71	4.50
Santander | occupancy | t = 60 | nRMSE												
STM	3.77	3.20	3.65	3.29	3.45	3.68	4.42	4.34	4.16	4.02	4.26	5.04
MTM	3.40	3.04	3.12	3.48	3.13	3.39	3.99	3.95	4.00	4.17	4.23	5.52

6.4.1 A note on computational efficiency

In this section, some facts about the computational aspects comparing the different methods will be given. The hardware used to run all the experiments has been the same desktop computer, whose specifications are:

- Processor: Intel(R) Core(TM) i7-7700K CPU @ 4.20GHz, 4 Core(s), 8 Logical Processor(s).
- Installed Physical Memory (RAM): 64.0 GB.
- SSD (where the software is installed and run): Toshiba KXG50ZNV512G XG5 NVMe PCIe M.2 512GB.
- HDD (where data and models are stored and read/write): Seagate Desktop HDD ST2000DM001 - hard drive - 2 TB - SATA 6Gb/s.

We have tried to measure the computational times as fair as possible, taking into account that the methods differ completely in the way they are built —i.e. static batch processes or incremental processes— as well as the kind of information they provide —there is no pattern mining, anomaly detection or fill in of missing values in the case of the ANA method—. For such reason, all methods have been executed using mini-batches of 1 month of data. Therefore, ANA models are iteratively

trained using one month, then predicting the next one, and repeating this 24 times since there are two years of data for each network dataset. On the other hand, Adarules executions have been run using mini-batches of 1 month of data too, which means Adarules system is updated —i.e. finding new patterns, updating patterns and statistics, detecting changes and anomalies, updating the forecasting models, etc.— with a new data stream 24 times too —once every month of data until having observed all the two years—. Every time Adarules receives a chunk of one month of data, it first generates the prediction for such month before updating the system with the new data.

In both cases, it has been used the entire network —315 detectors in M4/M7 network, and 254 detectors in Santander network— both as input and as an output of the network predictions. It has been evaluated only one forecasting variable —flow— and one forecasting horizon —15 minutes— .

Average computational performance for the different methods —Adarules single-task mining, Adarules multi-task mining, and ANA forecasting models used in Aimsun Live— using the two-years dataset from both networks —M4/M7 and Santander— are shown in Figure 6.3. First, it is interesting to check how the multi-task mining approach used in Adarules makes it to be much more efficient than its single-task mining counterpart —almost 100 times faster if considering running the STM approach in the entire network—. It can be also observed that the Adarules MTM proposal is much more computational efficient than current solution used in Aimsun Live (ANA) —almost 9 times faster depending on the network—. Another interesting fact is observing how the average time taken by Adarules MTM is higher in the Santander network compared to the M4/M7 network, in spite of M4/M7 network has some more detectors —i.e. more forecasting models to be built—. This may be probably because of the more complex traffic dynamics in the urban network of Santander —i.e. more pattern rules— as well as the potentially higher number of changes and drift that may lead to higher change detection and models' readjustment.

It is also worth to note some more facts about computational performance:

- Prediction time for Adarules, including rule pattern matching and forecasts calculation, is remarkably small making it very appropriate for real-time usage. The time taken to match the graph pattern —rule— in Adarules MTM and then calculating the forecasts for the entire network is less than a second.
- *Digest* time —i.e. to observe a new traffic data stream to perform all the system's updatings: pattern mining, anomaly detection, change detection, forecasting models' update, etc.— for a 1-month chunk of data using the entire network is approximately 10 minutes for both network datasets.

Table 6.3: Average computational performance for the different methods —Adarules single-task mining, Adarules multi-task mining, and ANA forecasting models used in Aimsun Live— using the two-years dataset from both networks —M4/M7 and Santander—.

Method	Time [h]	Peak RAM use [GB]	CPU threads
M4/M7 motorways network			
Adarules STM	1.75 h · 315 IDs	2.1	8
Adarules MTM	5.5	2.5	8
ANA	48	2.5	8
Santander urban network			
Adarules STM	2 h · 254 IDs	2.1	8
Adarules MTM	6	2.5	8
ANA	45.5	2.4	8

- Pattern mining using a 1-month data for the entire network —i.e. evaluating all the candidate splits and choosing the best in order to create a new rule— takes less than 3 minutes in both networks.
- A big emphasis has been put on the ability of Adarules to be able to handle large amounts of data, and thus the implementation is able to work with chunks of data —or data streams— instead of having all the data in memory. Therefore, internal parameters related to the chunk size is configurable according to the available computer resources.
- The big computational advantage of Adarules, when running in real-time, is its ability to perform incremental updates and only asking for the exact amount of historical data that is required and when it is required.

6.5 Adarules: forecasting model learning using a single-task or a multi-task approach

Therefore, from now on we forget about single-task mining (STM) and continue experiments with multi-task mining (MTM) approach as it achieved better forecasting accuracy and generalization error, while also resulting in lower model complexity as described in the previous set of experiments. The objective for this second set of experiments is to assess the ability of jointly learning multiple related tasks —the set of forecasting horizons 15, 30, 45, 60 minutes for a given traffic variable— at the forecasting models level by using a *multi-task learning* (MTL) approach or just independently learning every forecasting horizon in a *single-task learning* (STL) approach. This relates to how the spatiotemporal correlations are selected in the case of forecasting a specific traffic variable in a given network location, because in the MTL when a feature is selected it is shared across all the tasks —-the forecasting horizons— despite that their relevance may vary across them —i.e. far

away network locations will have a small relevance for the shortest forecasting horizon compared to longer forecasting horizons—. Conversely, in the STL approach, there is no such constraint and, therefore, spatiotemporal correlations for every forecasting horizon may vary freely and completely. The main driving motivation for applying this MTL approach is that the process of rule mining itself is considering not only spatial but also temporal dynamics when identifying the graph patterns, and thus it is reasonable to assume that spatiotemporal correlations will be shared across the timing span —e.g. 60 minutes— in the context of a given rule —if no anomalies or incidents are taking place— as these rules are learned to consider the traffic dynamics for the whole time span. The second motivation arises from the fact that we consider preferable to bias the feature selection towards those locations which are relevant in multiple subsequent time steps as this forces to select those strongest correlations relevant in all the multiple time steps. Once again, a clarification: the underlying network traffic state information is equally available to both approaches, i.e. forecasting models can use the information from all the detectors in the network in order to perform the forecasts.

The evaluation is performed in both network datasets, M4/M7 and Santander, for every available traffic measurement (traffic flow, occupancy, and speed) and for two forecasting horizons —15 and 60 minutes ahead—.

The assessment in terms of forecasting accuracy is performed using the normalized RMSE (nRMSE). There is one figure per forecasting traffic variable: Figure 9.4 for the flow, Figure 9.5 for the occupancy and Figure 9.6 for the speed. Each of these figures has subfigures for the two network datasets and for each forecasting horizon, resulting in four subfigures with the exception of speed which has only two because there is no measured speed data in the Santander dataset. These results can be also corroborated in the Table 6.4.

In this case, the differences between the MTL and STL approaches are more subtle among the different traffic variables. Still, when using the MTM approach it can be appreciated a small improvement in the forecasting horizon of 60 minutes which is more noticeable in the traffic flow variable especially during the second year in each network dataset, with respect to the STM approach. For this reason, the MTL approach for the forecasting models has been chosen in order to perform the rest of the experiments.

CONCLUSION: MTL performs slightly worse than STL at shorter forecasting horizons —15 minutes—, but performs better in longer forecasting horizons —60 minutes—. Furthermore, the advantage improves as time passes.

Table 6.4: Comparison of the forecasting performance measured by nRMSE for both forecasting learning approaches. The KPIs are aggregated over the results during the last year in each of the datasets.

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
M4/M7 | flow | t = 15 | nRMSE												
STL	1.49	1.69	1.62	1.62	1.41	1.46	1.48	1.45	1.49	1.38	1.47	1.71
MTL	1.60	1.81	1.67	1.71	1.49	1.52	1.54	1.44	1.58	1.42	1.53	1.77
M4/M7 | flow | t = 60 | nRMSE												
STL	4.84	4.44	4.35	4.31	3.91	3.84	3.62	3.80	3.92	3.63	4.02	5.29
MTL	4.62	4.20	4.22	4.09	3.65	3.70	3.55	3.53	3.69	3.30	3.67	4.92
M4/M7 | occupancy | t = 15 | nRMSE												
STL	1.90	2.20	2.40	2.26	2.33	2.20	2.28	2.44	2.62	1.88	2.63	2.42
MTL	1.87	2.20	2.37	2.36	2.38	2.22	2.34	2.50	2.65	1.84	2.69	2.51
M4/M7 | occupancy | t = 60 | nRMSE												
STL	2.27	3.21	3.03	3.57	2.95	3.18	2.95	3.06	3.45	2.34	3.19	3.38
MTL	2.25	3.18	3.03	3.51	2.96	3.46	2.88	2.98	3.42	2.44	3.22	3.30
M4/M7 | speed | t = 15 | nRMSE												
STL	5.62	6.05	6.23	5.51	5.83	5.88	5.63	5.67	5.94	5.39	5.83	5.78
MTL	5.59	6.19	6.25	5.53	5.79	5.80	5.74	5.55	5.85	5.32	5.72	5.76
M4/M7 | speed | t = 60 | nRMSE												
STL	6.41	7.56	7.99	6.59	6.76	6.95	6.63	6.20	6.49	5.97	6.94	6.53
MTL	6.28	7.42	7.68	6.56	6.63	6.84	6.53	6.14	6.39	5.92	6.92	6.52
Santander | flow | t = 15 | nRMSE												
STL	1.73	1.40	1.61	1.75	1.73	1.73	1.88	1.96	1.51	1.84	2.07	1.86
MTL	1.77	1.44	1.72	1.86	1.78	1.83	1.96	1.99	1.46	1.71	1.74	1.76
Santander | flow | t = 60 | nRMSE												
STL	4.38	3.33	3.97	4.49	4.36	4.51	5.09	5.33	3.97	4.42	5.83	5.33
MTL	4.24	3.31	3.90	4.22	4.69	4.53	4.89	5.33	3.50	4.25	4.00	4.59
Santander | occupancy | t = 15 | nRMSE												
STL	2.80	2.58	2.73	2.87	2.64	2.85	3.46	2.92	3.11	3.84	3.81	4.00
MTL	2.89	2.54	2.78	2.85	2.54	2.90	3.42	3.20	2.76	3.87	3.80	4.45
Santander | occupancy | t = 60 | nRMSE												

Table 6.4: Comparison of the forecasting performance measured by nRMSE for both forecasting learning approaches. The KPIs are aggregated over the results during the last year in each of the datasets. *(continued)*

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
STL	3.73	3.11	3.12	3.48	3.15	3.38	3.95	3.98	4.01	4.16	4.46	4.98
MTL	3.47	2.86	3.11	3.43	3.10	3.37	3.94	3.95	3.76	4.21	4.18	5.32

6.6 Adarules vs baselines: Real data scenario

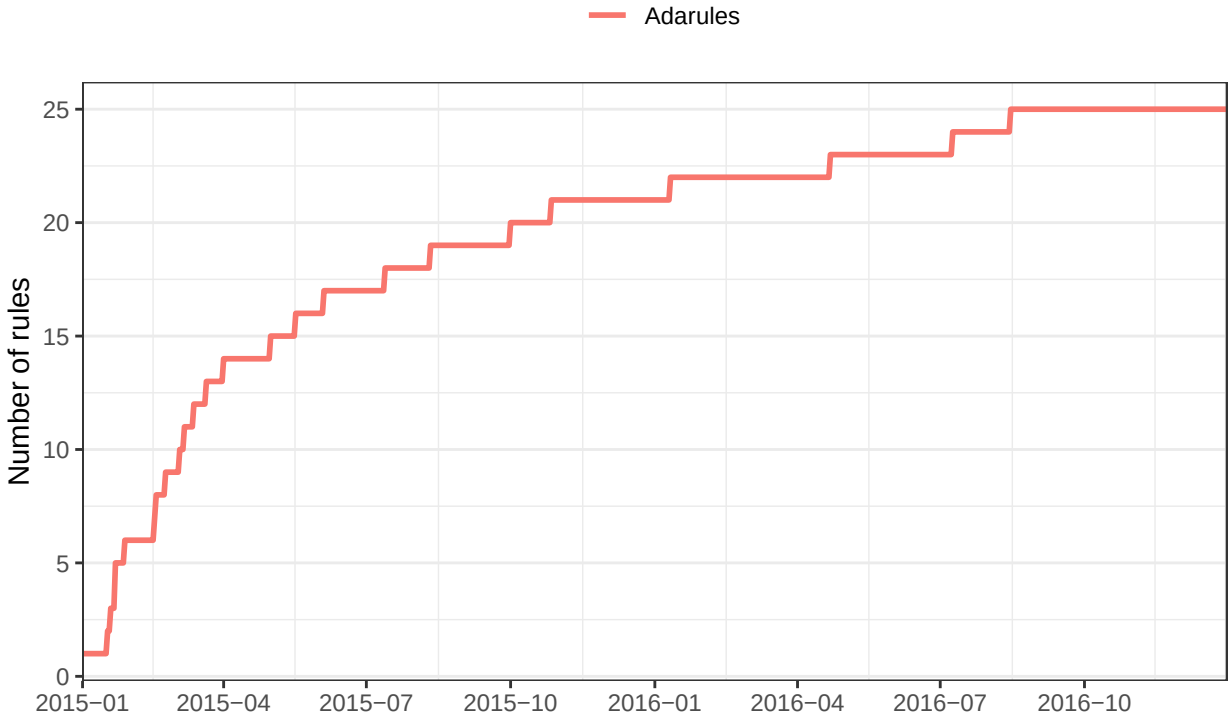
From this experiment onwards the objective is to evaluate the Adarules performance —configured using multi-task mining for rule mining and multi-task learning for forecasting models— against the set of baselines —the current methodology *ANA* and a seasonal naïve forecast *HA*— described in Section 6.2 in a set of different change scenarios.

This section describes the first scenario and it consists simply on using the real data from both network datasets in order to evaluate and compare the forecasting performance from Adarules, ANA and HA approaches in the three traffic variables and two forecasting horizons. The goal, then, is to compare the Adarules performance using the already existing changes and non-stationarities within the networks during the two years against the set of baselines.

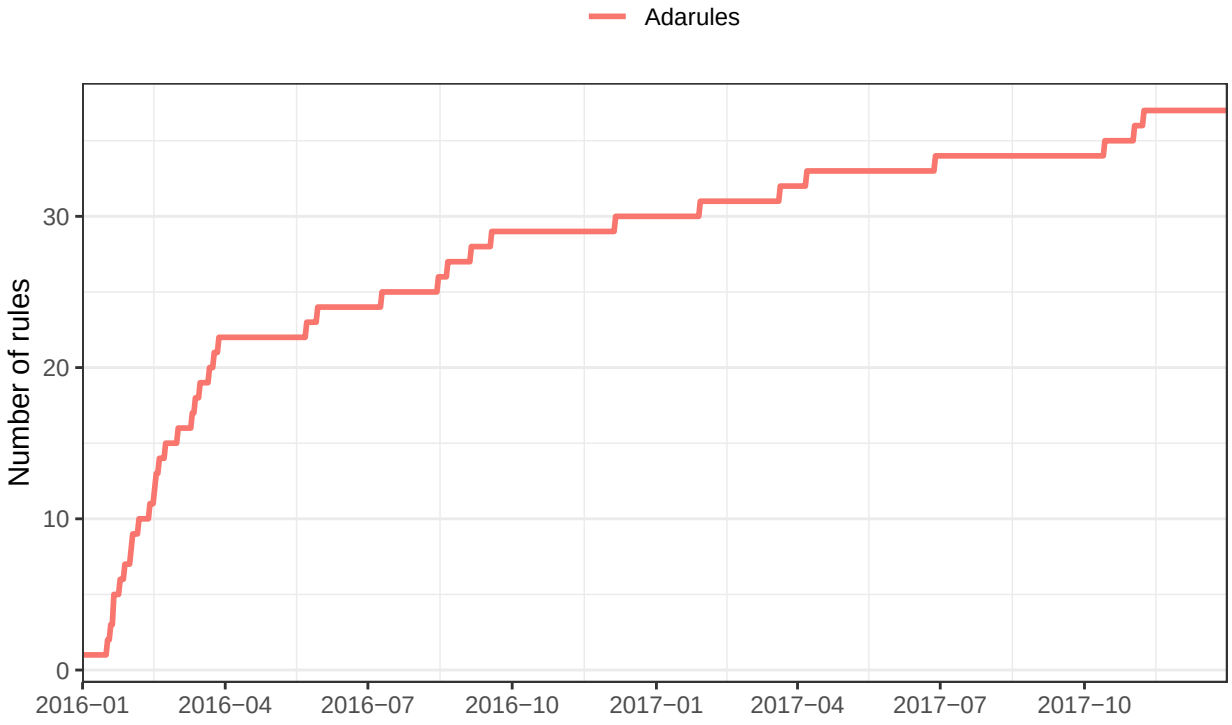
Given that all the subsequent experiments related to change scenario have been designed so that *synthetic* changes are introduced during the second year of each network dataset, the results of the experiments are shown, both graphically and in tables, only for the months during the second year. This does not imply that the first year of each dataset is not used. In fact, that first year is used to learn —as corresponds to every approach— the *real* situation before facing new *unexpected* situations.

Firstly, the resulting number of rules for Adarules in both networks (Figure 6.7) is identical to the one using multi-task-mining in the subsection dedicated to evaluating the MTM and STM approaches for rule mining —i.e. 25 rules in the case of the M4/M7 network, and 36 rules in the case of Santander network—. This makes sense as the underlying data used in both experiments is identical, Adarules has been configured in MTM rule mining and the process is deterministic.

Table 6.5 shows the antecedents for the resulting list of rules after the learning process is finished at the end of the two years in both datasets —25 rules in the case of M4/M7 and 37 rules in the case of Santander network—. There is an interesting situation regarding two specific rules found for the Santander network. These correspond to the pair of rules whose set of antecedents include



(a) M4/M7 network.



(b) Santander network.

Figure 6.7: Evolution of the Adarules complexity in the *real-data* scenario. Number of identified rules as a function of the time —every iteration corresponds to one day—.

the definition of a Datetime antecedent: Datetime $\leq$ 2016-05-18, and Datetime $>$ 2016-05-18. This may seem strange, but as was described in Section 5.1.2, this situation corresponds to a special case of global change detection where a broadcast message would be sent from that node to the whole decision tree in Adarules in order to revert the whole structure back to the root node and creating two child nodes that will split up the data based on the selected timestamp 2016-05-18, thus ignoring older data than such timestamp. However, we have decided to include the list of all the rules without proceeding with such reestructuring in order to show all the identified rules until that moment.

From the observation of the lists of rules and their antecedents in Table 6.5, one may think that a generic distance-based clustering such as the well-known k-means performed over the data would have been enough to achieve the same result. Nevertheless, it is important to remark that such clustering would have had to also consider the same set of qualitative variables as described in Section 5.1.1, and moreover, it would have to be flexible to be able to consider new prior information in the form of new candidates for the pattern mining process —e.g. weather, events calendar, or even quantitative measurements from traffic sensors— the same way Adarules is. Furthermore, even though the list of rules presented in Table 6.5 corresponds to a specific moment —at the end of the two years in both dataset—, the key point of the pattern mining process in Adarules is its online evolving nature. This implies that patterns (rules) are found dynamically as more data is observed, and additionally they are monitored to cope with and adapt to global changes. This contrasts with a classic clustering approach that is usually done offline with no incremental learning. Finally, the online nature of the pattern mining in Adarules makes it especially efficient for a real-time application.

Table 6.5: List of rules identified by Adarules in each network dataset just at the end of the two years. Size corresponds to the number of observations gathered under the scope of a specific rule.

Antecedent(s)	Size
M4/M7 motorways network	
Hour = [1 - 2]	5856
Hour = [0]	2928
Weekday = [Saturday, Sunday, Holiday] $\wedge$ Hour = [3]	908
Weekday = [Monday - Friday] $\wedge$ Hour = [3]	2020
Weekday = [Sunday, Holiday] $\wedge$ Hour = [4 - 5]	1000
Weekday = [Saturday] $\wedge$ Hour = [4 - 5]	816

Table 6.5: List of rules identified by Adarules in each network dataset just at the end of the two years. Size corresponds to the number of observations gathered under the scope of a specific rule. *(continued)*

Antecedent(s)	Size
Weekday = [Monday - Friday] ∧ Hour = [4]	2020
Weekday = [Monday - Friday] ∧ Hour = [5]	2020
Hour = [22, 23]	5855
Hour = [20 - 21]	5856
Hour = [19]	2928
Weekday = [Sunday] ∧ Hour = [18]	404
Weekday = [Monday - Saturday] ∧ Hour = [18]	2524
Weekday = [Sunday, Holiday] ∧ Hour = [6]	500
Weekday = [Sunday, Holiday] ∧ Hour = [7]	500
Weekday = [Sunday, Holiday] ∧ Hour = [8 - 9]	1000
Weekday = [Sunday, Holiday] ∧ Hour = [10 - 17]	4000
Weekday = [Saturday] ∧ Hour = [6 - 8]	1224
Weekday = [Saturday] ∧ Hour = [9 - 17]	3672
Weekday = [Monday - Friday] ∧ Hour = [9 - 13]	10100
Weekday = [Monday - Friday] ∧ Hour = [14 - 15]	4040
Weekday = [Monday - Friday] ∧ Hour = [17]	2020
Weekday = [Monday - Friday] ∧ Hour = [6]	2020
Weekday = [Monday - Friday] ∧ Hour = [8]	2020
Weekday = [Monday - Friday] ∧ Hour = [7, 16]	4040
Santander urban network	
Weekday = [Monday - Friday] ∧ Hour = [2 - 3]	3944
Weekday = [Monday - Friday] ∧ Hour = [1, 4]	3944
Weekday = [Saturday, Sunday, Holiday] ∧ Hour = [3 - 4]	1912
Weekday = [Saturday, Sunday, Holiday] ∧ Hour = [1 - 2]	1912
Hour = [5]	2928
Weekday = [Saturday, Sunday] ∧ Hour = [6]	840
Weekday = [Monday - Friday] ∧ Hour = [6]	2088
Weekday = [Monday - Friday] ∧ Hour = [0]	1972
Weekday = [Saturday, Sunday, Holiday] ∧ Hour = [0]	956
Weekday = [Saturday, Sunday, Holiday] ∧ Hour = [7]	956

Table 6.5: List of rules identified by Adarules in each network dataset just at the end of the two years. Size corresponds to the number of observations gathered under the scope of a specific rule. *(continued)*

Antecedent(s)	Size
Weekday = [Friday] $\wedge$ Hour = [7]	392
Weekday = [Monday - Thursday] $\wedge$ Hour = [7]	1580
Weekday = [Saturday, Sunday, Holiday] $\wedge$ Hour = [23]	955
Weekday = [Saturday, Sunday, Holiday] $\wedge$ Hour = [22]	956
Weekday = [Saturday, Sunday, Holiday] $\wedge$ Hour = [21]	956
Weekday = [Saturday, Sunday, Holiday] $\wedge$ Hour = [8]	956
Weekday = [Sunday, Holiday] $\wedge$ Hour = [9 - 10]	1072
Weekday = [Saturday] $\wedge$ Hour = [9 - 10]	840
Weekday = [Saturday, Sunday, Holiday] $\wedge$ Hour = [15 - 16]	1912
Weekday = [Saturday] $\wedge$ Hour = [14, 17]	840
Weekday = [Saturday] $\wedge$ Hour = [11 - 13, 18 - 20]	2520
Weekday = [Sunday, Holiday] $\wedge$ Hour = [14, 17]	1072
Weekday = [Sunday, Holiday] $\wedge$ Hour = [11 - 13, 18 - 20]	3216
Weekday = [Monday - Friday] $\wedge$ Hour = [23]	1972
Weekday = [Monday - Friday] $\wedge$ Hour = [22]	1972
Weekday = [Monday - Friday] $\wedge$ Hour = [21]	1972
Weekday = [Monday - Friday] $\wedge$ Hour = [9 - 11]	5916
Weekday = [Monday - Friday] $\wedge$ Hour = [15 - 16]	3944
Weekday = [Monday - Friday] $\wedge$ Hour = [17 - 19] $\wedge$ Season = [Spring]	1524
Weekday = [Monday - Friday] $\wedge$ Hour = [17 - 19] $\wedge$ Month = [August]	504
Weekday = [Monday - Friday] $\wedge$ Hour = [17 - 19] $\wedge$ Season $\neq$ [Spring] $\wedge$ Month $\neq$ [August] $\wedge$ Datetime $\leq$ 2016-05-18	600
Weekday = [Monday - Friday] $\wedge$ Hour = [17 - 19] $\wedge$ Season $\neq$ [Spring] $\wedge$ Month $\neq$ [August] $\wedge$ Datetime $>$ 2016-05-18	3288
Weekday = [Monday - Friday] $\wedge$ Hour = [8] $\wedge$ Month = [July - August]	320
Weekday = [Monday - Friday] $\wedge$ Hour = [8] $\wedge$ Month $\neq$ [July - August]	1652
Weekday = [Monday - Friday] $\wedge$ Hour = [20]	1972
Weekday = [Monday - Friday] $\wedge$ Hour = [14]	1972
Weekday = [Monday - Friday] $\wedge$ Hour = [12 - 13]	3944

The assessment in terms of forecasting accuracy is performed using the normalized RMSE (nRMSE).

There is one figure per forecasting traffic variable: Figure 9.7 for the flow, Figure 9.8 for the occupancy and Figure 9.9 for the speed. Each of these figures has subfigures for the two network datasets and for each forecasting horizon, resulting in four subfigures with the exception of speed which has only two because there is no measured speed data in the Santander dataset. These results can be also corroborated in the Table 6.6.

It can be observed the clear superiority of Adarules compared to the rest of baselines in any of their configurations —yearly, quarterly or monthly blind updates—. This is especially noticeable in the very short-term traffic forecasting —15 minutes— in both the median values of the nRMSE indicator through all the months as well as the tails of the distributions showing those worst nRMSE indicators in every month. The better performance of Adarules is reproducible in all the three traffic variables compared to the baselines. Between both networks, it can be seen how the performance in the M4/M7 network is slightly better to that of the Santander urban network which is logical due to the higher probability of non-recurrent and unexpected events within a city. Other interesting facts are observing the performance instabilities achieved by ANA baseline according to the period of its training, and also how the performance of the HA baseline is greatly influenced by the seasonality. In the comparison for the longer forecasting horizon —60 minutes— is less evident to observe great differences, but still, it is visible how Adarules performance shows more performance stability along the year than the baselines.

Among the different features of Adarules described in Chapter 5 that makes it obtain better forecasting results, one important difference with respect to the ANA baseline already being used in Aimsun Live is how data is used in order to build the forecasting models. In ANA, the data fragmentation was extremely harsh since every forecasting model was built using only data corresponding to the same time of the day (HH:MM) and hence, for a year-dataset only 365 observations were available to build the model. Conversely, Adarules rely on the online pattern mining process in order to accommodate the data —in rules— and therefore being able to build forecasting models with more data by taking into account the traffic dynamics more realistically.

CONCLUSION: Using the real data from both networks —M4/M7 and Santander—, Adarules achieves better forecasting accuracy than its baseline competitors —ANA and HA— in both networks, all the forecasting variables, and forecasting horizons. Rules interpretation have been checked to be in line with expected knowledge from an expert traffic engineer.

Table 6.6: Comparison of the forecasting performance measured by nRMSE between Adarules and baselines in the real data scenario. The KPIs are aggregated over the results during the last year in each of the datasets.

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
M4/M7 | flow | t = 15 | nRMSE												
Adarules	1.60	1.78	1.67	1.67	1.48	1.50	1.54	1.44	1.56	1.40	1.53	1.76
ANA-y	2.99	3.04	3.17	3.05	2.99	3.09	3.07	2.89	3.01	3.05	2.97	3.54
ANA-q	4.30	4.02	4.20	4.76	4.82	5.13	4.39	3.98	4.36	4.42	3.92	5.47
ANA-m	5.43	7.25	6.94	6.21	6.23	7.01	6.26	5.60	5.54	6.54	6.11	7.70
HA-y	6.59	4.04	5.03	4.18	3.92	4.13	4.00	4.39	5.30	5.06	5.85	9.43
HA-q	7.64	3.89	5.03	4.22	3.99	4.36	3.50	3.22	4.14	3.24	3.64	8.99
HA-m	6.59	3.98	4.81	3.92	3.34	3.86	3.45	3.34	4.10	3.62	3.99	8.87
M4/M7 | flow | t = 60 | nRMSE												
Adarules	4.58	4.16	4.18	3.97	3.65	3.69	3.54	3.53	3.68	3.31	3.67	4.92
ANA-y	6.26	5.71	6.18	6.09	5.76	6.15	6.19	5.89	6.30	6.36	6.10	7.37
ANA-q	7.77	6.50	7.20	7.12	7.04	7.34	7.00	6.54	7.08	7.62	6.66	8.56
ANA-m	8.14	8.33	8.51	7.67	7.19	8.23	7.46	6.91	7.62	8.33	7.20	9.98
HA-y	6.68	4.10	5.11	4.25	3.98	4.19	4.06	4.47	5.38	5.14	5.93	9.57
HA-q	7.76	3.94	5.12	4.28	4.04	4.42	3.55	3.27	4.21	3.29	3.68	9.12
HA-m	6.68	4.05	4.89	3.99	3.38	3.92	3.50	3.39	4.16	3.68	4.05	9.01
M4/M7 | occupancy | t = 15 | nRMSE												
Adarules	1.87	2.21	2.42	2.37	2.37	2.23	2.38	2.53	2.61	1.85	2.75	2.52
ANA-y	2.22	2.65	2.82	2.79	2.79	2.77	2.79	2.91	3.20	2.69	3.40	3.37
ANA-q	2.61	2.89	2.89	3.24	3.08	2.90	3.04	2.96	3.29	2.77	4.21	3.58
ANA-m	2.56	3.57	3.43	3.95	3.51	3.33	3.20	3.41	3.96	3.41	3.76	4.32
HA-y	2.58	2.93	2.91	3.29	2.77	3.05	2.97	3.12	3.68	2.46	3.48	3.86
HA-q	3.03	3.08	3.01	3.42	3.02	3.24	3.11	3.18	3.60	2.47	3.35	4.07
HA-m	2.58	2.93	2.89	3.23	2.74	3.04	2.96	3.09	3.54	2.38	3.30	3.87
M4/M7 | occupancy | t = 60 | nRMSE												
Adarules	2.25	3.21	3.07	3.25	3.01	3.46	2.94	3.01	3.45	2.48	3.24	3.33
ANA-y	2.72	3.30	3.43	3.48	3.29	3.45	3.28	3.38	3.91	3.19	3.86	3.78
ANA-q	2.94	3.36	3.86	3.79	3.55	3.70	3.60	3.49	3.96	3.16	6.08	4.20
ANA-m	2.68	3.69	3.92	3.92	3.69	3.90	3.53	3.46	4.34	3.49	5.67	4.94

Table 6.6: Comparison of the forecasting performance measured by nRMSE between Adarules and baselines in the real data scenario. The KPIs are aggregated over the results during the last year in each of the datasets. *(continued)*

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
HA-y	2.62	2.97	2.95	3.34	2.81	3.08	3.05	3.17	3.72	2.50	3.53	3.83
HA-q	3.07	3.12	3.06	3.47	3.06	3.28	3.17	3.23	3.64	2.51	3.40	4.08
HA-m	2.62	2.97	2.93	3.27	2.77	3.07	3.03	3.13	3.58	2.42	3.34	3.87
M4/M7 | speed | t = 15 | nRMSE												
Adarules	5.59	6.17	6.25	5.53	5.79	5.78	5.74	5.55	5.85	5.29	5.72	5.76
ANA-y	5.74	6.34	6.80	5.98	6.17	6.16	6.17	6.03	6.28	5.66	6.53	6.30
ANA-q	6.20	6.77	7.75	6.87	6.62	6.57	6.70	6.87	6.56	6.05	7.16	6.77
ANA-m	6.65	8.23	9.08	7.19	7.06	7.24	7.27	7.35	7.26	6.56	8.79	8.05
HA-y	6.79	7.29	8.28	6.68	6.81	7.09	7.05	7.38	6.64	6.58	8.21	7.51
HA-q	7.08	7.49	8.37	7.17	7.14	7.40	6.88	7.38	6.76	6.54	8.29	7.52
HA-m	6.79	7.26	8.25	6.65	6.60	6.97	6.80	7.34	6.48	6.20	8.01	7.40
M4/M7 | speed | t = 60 | nRMSE												
Adarules	6.29	7.42	7.70	6.57	6.64	6.85	6.54	6.15	6.39	5.90	6.92	6.52
ANA-y	6.40	7.49	7.99	6.80	6.81	7.04	7.13	6.67	7.01	6.60	7.79	6.88
ANA-q	6.52	7.76	8.19	7.25	7.28	7.45	7.16	7.36	7.09	7.03	8.44	7.38
ANA-m	6.87	8.80	9.29	7.72	7.66	7.64	7.52	7.66	7.53	7.50	8.96	8.56
HA-y	6.74	7.29	8.12	6.61	6.80	7.07	7.03	7.23	6.61	6.48	7.97	7.20
HA-q	6.97	7.53	8.28	7.12	7.07	7.39	6.84	7.22	6.67	6.44	8.02	7.37
HA-m	6.74	7.27	8.06	6.57	6.62	6.95	6.77	7.12	6.41	6.12	7.76	7.10
Santander | flow | t = 15 | nRMSE												
Adarules	1.62	1.44	1.71	1.84	1.78	1.81	1.95	1.95	1.46	1.69	1.71	1.75
ANA-y	1.83	1.70	1.87	2.11	1.81	2.23	2.18	2.21	2.42	3.09	2.50	2.45
ANA-q	2.43	2.22	2.55	2.56	2.63	2.64	2.72	2.52	2.99	2.68	3.99	5.40
ANA-m	4.17	4.09	4.91	3.82	3.22	3.63	4.29	3.22	7.07	3.59	5.38	4.38
HA-y	5.53	3.50	3.51	4.40	3.42	4.02	5.98	6.17	3.92	5.10	4.70	5.48
HA-q	5.25	3.48	3.16	4.76	3.68	5.14	5.90	6.47	3.87	5.43	6.53	6.42
HA-m	5.53	3.44	3.64	4.35	3.32	3.83	5.88	5.94	3.77	4.23	4.31	5.09
Santander | flow | t = 60 | nRMSE												
Adarules	3.81	3.31	3.91	4.24	4.72	4.53	4.86	5.31	3.50	3.99	4.02	4.57

Table 6.6: Comparison of the forecasting performance measured by nRMSE between Adarules and baselines in the real data scenario. The KPIs are aggregated over the results during the last year in each of the datasets. *(continued)*

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
ANA-y	4.61	4.12	4.60	5.36	4.59	5.21	5.58	5.33	5.29	5.54	5.39	6.30
ANA-q	5.16	4.75	4.95	5.56	5.37	5.56	6.15	5.00	6.18	5.55	7.04	7.23
ANA-m	6.49	5.41	5.67	6.70	5.39	6.09	6.64	4.95	10.31	6.32	6.49	6.40
HA-y	5.45	3.53	3.51	4.47	3.45	4.07	5.97	6.23	3.94	4.25	4.76	5.58
HA-q	5.24	3.52	3.20	4.82	3.73	5.14	5.88	6.40	3.91	4.54	6.41	6.46
HA-m	5.45	3.47	3.64	4.42	3.35	3.87	5.87	6.00	3.78	3.85	4.35	5.12
Santander | occupancy | t = 15 | nRMSE												
Adarules	2.89	2.54	2.78	2.85	2.55	2.91	3.42	3.21	2.74	3.88	3.80	4.46
ANA-y	3.28	2.82	3.04	3.25	2.92	3.26	3.72	3.92	4.00	4.46	4.27	5.40
ANA-q	3.64	3.14	3.28	3.14	3.08	3.41	4.27	4.03	3.88	3.93	4.07	5.68
ANA-m	4.12	3.46	3.63	3.75	3.22	3.63	4.56	3.86	7.86	4.13	4.25	5.11
HA-y	3.68	3.05	3.06	3.44	3.13	3.49	4.13	4.52	3.76	4.02	5.02	6.66
HA-q	3.82	3.01	3.20	3.46	3.32	3.70	4.30	4.45	3.67	4.16	4.48	6.22
HA-m	3.68	3.06	3.07	3.45	3.09	3.34	4.03	4.23	3.69	3.52	3.78	6.18
Santander | occupancy | t = 60 | nRMSE												
Adarules	3.47	2.87	3.12	3.44	3.10	3.36	3.93	3.96	3.77	4.27	4.18	5.33
ANA-y	3.70	3.06	3.20	3.53	3.33	3.51	4.17	4.31	4.20	4.95	5.00	6.30
ANA-q	4.04	3.27	3.44	3.54	3.41	3.76	4.60	4.33	4.23	4.11	4.88	6.44
ANA-m	4.61	3.92	3.80	3.96	3.36	3.82	5.08	4.49	7.43	3.92	4.42	5.92
HA-y	3.72	3.10	3.11	3.49	3.15	3.54	4.18	4.57	3.80	3.48	5.10	6.77
HA-q	3.88	3.05	3.23	3.51	3.37	3.76	4.35	4.50	3.71	3.51	4.55	6.31
HA-m	3.72	3.10	3.12	3.49	3.07	3.39	4.08	4.31	3.73	3.28	3.84	6.27

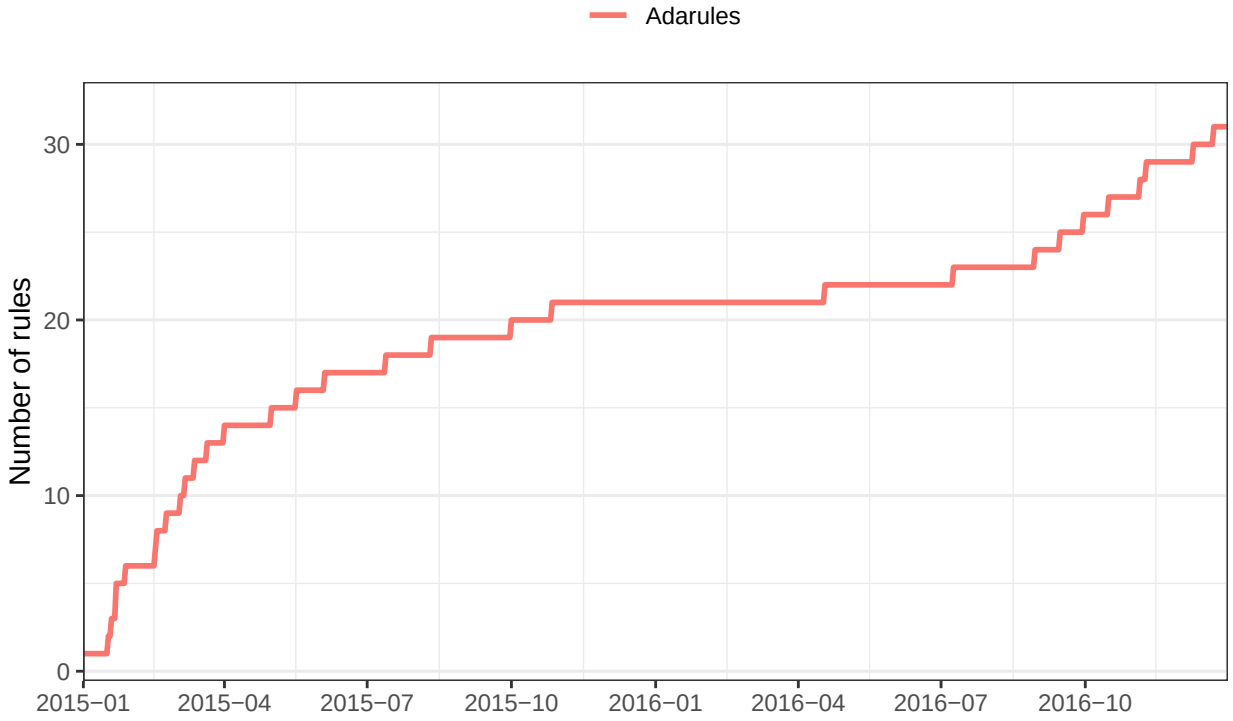
6.7 Adarules vs baselines: *Zero* drift scenario

This section describes the second scenario, which consists of replacing the second year in each network dataset using the month of May from the second year repeated twelve times. The objective is to evaluate and compare the forecasting performance from Adarules, ANA and HA approaches in the three traffic variables and two forecasting horizons when there is absolute certainty that there is no kind of change along the second year of the evaluation. This implies that the first year is used

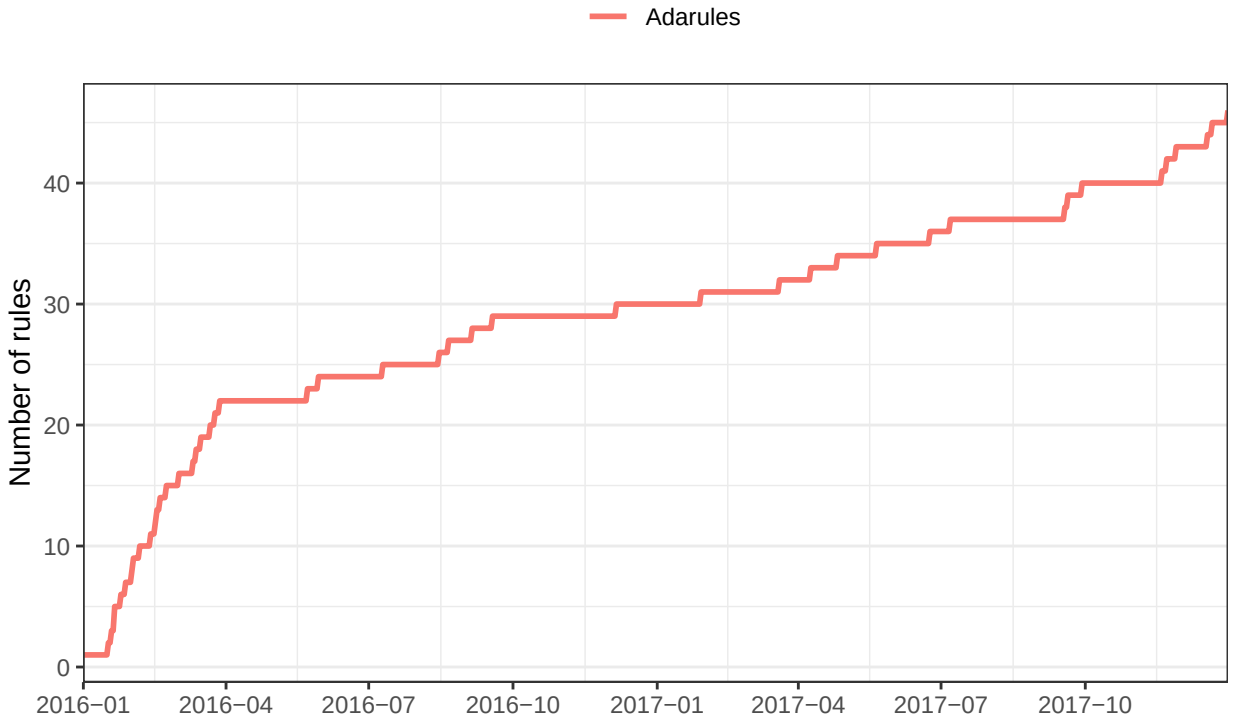
to be learned *as is* —with the original real data— by Adarules, and the rest of baselines according to their updating schedule. Therefore, the results of the experiments are shown, both graphically and in tables, only for the months during the second year.

Firstly, the evolution of the number of rules for Adarules in both networks (Figure 6.8) is identical during the first year to the previously observed within the experiment which used real-data (Figure 6.7). This was expected, as the data used during the first year of this second change scenario is also real data, and it is only during the second year when artificial changes are introduced in the datasets. Starting from the second year in each dataset, it is observed that the number of rules is growing much slower compared to the experiment with the real data. Again, this is the expected behaviour as all the data corresponding to the second year is just a copy of just one month. However, after several months —i.e. several copies of the same month of May— Adarules starts to identify new rules in both datasets. This may be due to the fact that, with enough sample size of data, even small differences can be identified as significant. Besides, data from both years —the first one with real data, and the second one with artificial copies— is suspected to differ to some extent. Even though this may seem a bit confusing at a first glance, an example may clarify it: supposing there exists a specific rule for the traffic between [2 - 4] p.m., and by just looking the traffic during the first year the pattern mining process is not able to differentiate the traffic at 2 p.m. from the traffic at 4 p.m. because of the existing variability. However, if the same data —the month of May— is received once and again, it may come to a point where such differences are accentuated and, thus, that rule may become more specialized for the traffic at 2, 3, and 4 p.m., respectively, because of more data sample size makes statistical tests more confidently reject hypothesis based on the statistical significance. This scenario seems to correspond to what is observed in Figure 6.8, as the evolution of the number of rules in both networks is virtually stable during the first months in the second year, and it is not until the middle of the year —after six copies of the same data has been observed— when the number of rules starts to raise again lineally. This hardly can be classified as *overfitting* because, as said, the rules' increase does not occur until six copies of May have been observed; instead, it is a sample of the good adaptation and confident specialization of Adarules in the face of a new scenario.

The assessment in terms of forecasting accuracy is performed using the normalized RMSE (nRMSE). There is one figure per forecasting traffic variable: Figure 9.10 for the flow, Figure 9.11 for the occupancy and Figure 9.12 for the speed. Each of these figures has subfigures for the two network datasets and for each forecasting horizon, resulting in four subfigures with the exception of speed which has only two because there is no measured speed data in the Santander dataset. These results can be also corroborated in the Table 6.7.



(a) M4/M7 network.



(b) Santander network.

Figure 6.8: Evolution of the Adarules complexity in the *zero drift* scenario. Number of identified rules as a function of the time —every iteration corresponds to one day—.

The first observable fact is that, overall, the forecasting performance is better compared to the previous experiment with real data. It is clear that this was the expected behaviour especially after observing more and more data with the same distribution. Overall, it can be seen that Adarules shows a good balance between what has been learned —*memory*— and the new situation where new data is always the same but probably not that different from what was seen in the past. Still, Adarules outperforms the rest of baselines especially in the very short-term —15 minutes—. Those ANA and HA baselines that use exclusively data from the second year to learn —some quarterly and monthly periods— can compete in performance with Adarules during the first months in the 60-minutes traffic forecast. The only exception is in the case of speed forecasting where the ANA baseline using a sliding window of 3 months to learn —quarterly— obtain a lower error when it starts to use exclusively 3 months from the new *fake* data and, conversely, the ANA version using only one month is not able to reach such performance even that the data distribution is always the same. Nevertheless, it can be appreciated how Adarules gradually improves its performance because of the gradual specialization as more data is observed.

CONCLUSION: Using data modified to incur in a *zero* drift scenario during the second year —all twelve months being a copy of May—, it is observed how Adarules performs an intelligent adaption by gradually specializing the rules when the data sample sizes are large enough —i.e. around mid of such second year— to confidently reject the statistical hypothesis, avoiding to overfit the data with low sample sizes during the first months of the second year. The forecasting accuracy with Adarules is far better than the baselines, being also interesting to observe the monthly gradual improvement of the accuracy for Adarules.

Table 6.7: Comparison of the forecasting performance measured by nRMSE between Adarules and baselines in the zero drift scenario. The KPIs are aggregated over the results during the last year in each of the datasets.

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
M4/M7 | flow | t = 15 | nRMSE												
Adarules	1.61	1.56	1.56	1.51	1.48	1.43	1.39	1.39	1.37	1.33	1.29	1.29
ANA-y	3.26	3.06	3.23	3.12	3.12	3.11	3.26	3.18	3.11	3.21	3.10	3.24
ANA-q	4.12	4.03	4.10	4.91	4.91	4.91	4.95	4.95	4.99	4.65	4.68	4.64
ANA-m	6.78	6.33	7.04	6.77	6.72	7.02	6.82	7.06	6.49	6.70	6.86	6.97
HA-y	3.95	3.86	3.95	3.98	3.95	3.98	3.95	3.95	3.98	3.95	3.98	3.95
HA-q	3.95	3.96	3.95	2.40	2.41	2.40	2.30	2.30	2.29	2.30	2.29	2.30
HA-m	3.95	3.41	3.21	3.05	2.87	2.77	2.56	2.50	2.45	2.41	2.38	2.36

Table 6.7: Comparison of the forecasting performance measured by nRMSE between Adarules and baselines in the zero drift scenario. The KPIs are aggregated over the results during the last year in each of the datasets. *(continued)*

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
M4/M7 | flow | t = 60 | nRMSE												
Adarules	3.86	3.78	3.72	3.57	3.44	3.38	3.25	3.21	3.18	3.08	3.03	2.99
ANA-y	5.96	5.89	5.93	5.96	5.89	5.96	5.96	5.91	5.96	5.92	5.95	5.97
ANA-q	7.01	7.10	7.01	6.54	6.48	6.54	6.33	6.33	6.40	6.34	6.40	6.33
ANA-m	8.42	7.95	7.71	7.67	7.57	7.83	7.54	7.76	7.92	7.50	7.53	7.73
HA-y	4.00	3.91	4.00	4.03	4.00	4.03	4.00	4.00	4.03	4.00	4.03	4.00
HA-q	3.99	4.00	3.99	2.43	2.44	2.43	2.33	2.33	2.31	2.33	2.31	2.33
HA-m	4.00	3.45	3.24	3.08	2.90	2.81	2.59	2.52	2.48	2.44	2.40	2.38
M4/M7 | occupancy | t = 15 | nRMSE												
Adarules	2.51	2.42	2.42	2.38	2.34	2.29	2.13	2.08	2.08	2.00	1.90	1.92
ANA-y	2.85	2.85	2.85	2.88	2.83	2.86	2.85	2.84	2.88	2.84	2.86	2.86
ANA-q	3.25	3.03	3.25	1.77	1.79	1.80	1.79	1.79	1.82	1.74	1.70	1.74
ANA-m	3.63	2.93	3.40	3.22	3.26	3.31	3.27	3.26	3.12	3.29	3.25	3.24
HA-y	3.08	2.77	3.08	3.06	3.08	3.06	3.08	3.08	3.06	3.08	3.06	3.08
HA-q	3.45	3.08	3.45	2.59	2.62	2.59	2.61	2.61	2.59	2.61	2.59	2.61
HA-m	3.08	2.66	2.91	2.84	2.79	2.74	2.71	2.68	2.64	2.65	2.61	2.62
M4/M7 | occupancy | t = 60 | nRMSE												
Adarules	3.29	2.88	3.20	3.09	2.96	2.92	2.77	2.75	2.60	2.51	2.46	2.52
ANA-y	3.68	3.37	3.67	3.72	3.64	3.72	3.68	3.67	3.72	3.66	3.72	3.68
ANA-q	3.94	3.55	3.94	2.09	2.09	2.09	1.97	2.01	2.01	1.77	1.78	1.77
ANA-m	4.16	3.21	3.54	3.54	3.57	3.62	3.45	3.47	3.55	3.57	3.70	3.64
HA-y	3.13	2.80	3.13	3.11	3.13	3.11	3.13	3.13	3.11	3.13	3.11	3.13
HA-q	3.50	3.12	3.50	2.63	2.65	2.63	2.65	2.65	2.62	2.65	2.62	2.65
HA-m	3.13	2.69	2.94	2.87	2.83	2.78	2.75	2.72	2.68	2.68	2.64	2.66
M4/M7 | speed | t = 15 | nRMSE												
Adarules	6.13	5.92	6.00	5.57	5.46	5.10	4.83	4.72	4.40	4.37	4.04	3.92
ANA-y	6.31	6.23	6.31	6.32	6.30	6.32	6.30	6.30	6.32	6.30	6.32	6.30
ANA-q	6.81	6.39	6.82	2.98	3.08	2.99	3.22	3.23	3.03	3.21	3.02	3.21
ANA-m	7.59	6.29	6.62	6.42	6.63	6.42	6.63	6.58	6.34	6.66	6.38	6.67

Table 6.7: Comparison of the forecasting performance measured by nRMSE between Adarules and baselines in the zero drift scenario. The KPIs are aggregated over the results during the last year in each of the datasets. *(continued)*

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
HA-y	6.92	6.75	6.92	6.86	6.92	6.86	6.92	6.92	6.86	6.92	6.86	6.92
HA-q	7.12	7.09	7.12	5.45	5.52	5.45	5.51	5.51	5.43	5.51	5.43	5.51
HA-m	6.92	6.49	6.48	6.23	6.14	5.98	5.94	5.85	5.72	5.63	5.50	5.52
M4/M7 | speed | t = 60 | nRMSE												
Adarules	6.98	6.70	6.78	6.24	6.00	5.60	5.39	5.06	4.88	4.75	4.58	4.42
ANA-y	7.20	7.12	7.19	7.12	7.19	7.12	7.20	7.19	7.12	7.19	7.12	7.18
ANA-q	7.51	7.44	7.51	3.10	3.10	3.12	3.25	3.26	3.12	3.47	3.11	3.48
ANA-m	8.08	6.95	7.01	6.93	6.84	6.77	7.02	6.96	6.76	7.05	6.85	7.00
HA-y	6.86	6.69	6.86	6.80	6.86	6.80	6.86	6.86	6.80	6.86	6.80	6.86
HA-q	7.06	7.00	7.06	5.40	5.47	5.40	5.46	5.46	5.38	5.46	5.38	5.46
HA-m	6.86	6.43	6.43	6.17	6.09	5.91	5.88	5.79	5.65	5.60	5.47	5.47
Santander | flow | t = 15 | nRMSE												
Adarules	1.88	2.01	1.95	1.90	1.79	1.85	1.84	1.58	1.47	1.35	1.33	1.30
ANA-y	2.31	2.38	2.35	2.36	2.29	2.36	2.35	2.32	2.37	2.31	2.36	2.37
ANA-q	2.97	3.07	3.01	2.02	2.01	2.02	2.05	2.05	2.06	2.09	2.11	2.09
ANA-m	4.37	3.14	3.30	3.04	3.08	3.10	3.16	3.02	3.20	3.09	3.17	3.23
HA-y	3.69	3.71	3.69	3.70	3.69	3.70	3.69	3.69	3.70	3.69	3.70	3.69
HA-q	3.92	3.97	3.92	2.07	2.08	2.07	2.06	2.06	2.06	2.06	2.06	2.06
HA-m	3.69	3.48	3.28	3.11	2.98	2.90	2.72	2.43	2.26	2.21	2.17	2.12
Santander | flow | t = 60 | nRMSE												
Adarules	4.56	5.03	4.89	4.60	4.17	4.31	4.19	3.44	3.02	2.52	2.46	2.40
ANA-y	4.79	5.02	4.90	4.95	4.72	4.94	4.92	4.82	4.97	4.79	4.94	5.02
ANA-q	5.63	5.88	5.72	4.35	4.31	4.35	4.38	4.38	4.42	4.34	4.38	4.37
ANA-m	7.48	5.71	5.57	5.66	5.57	5.70	5.55	5.66	5.70	5.68	5.71	5.57
HA-y	3.75	3.76	3.75	3.76	3.75	3.76	3.75	3.75	3.76	3.75	3.76	3.75
HA-q	3.97	4.02	3.97	2.09	2.10	2.09	2.08	2.08	2.08	2.08	2.08	2.08
HA-m	3.75	3.53	3.32	3.15	2.99	2.91	2.73	2.45	2.28	2.23	2.19	2.14
Santander | occupancy | t = 15 | nRMSE												
Adarules	3.08	2.90	2.90	2.80	2.67	2.66	2.43	2.55	2.46	2.06	1.97	1.87

Table 6.7: Comparison of the forecasting performance measured by nRMSE between Adarules and baselines in the zero drift scenario. The KPIs are aggregated over the results during the last year in each of the datasets. *(continued)*

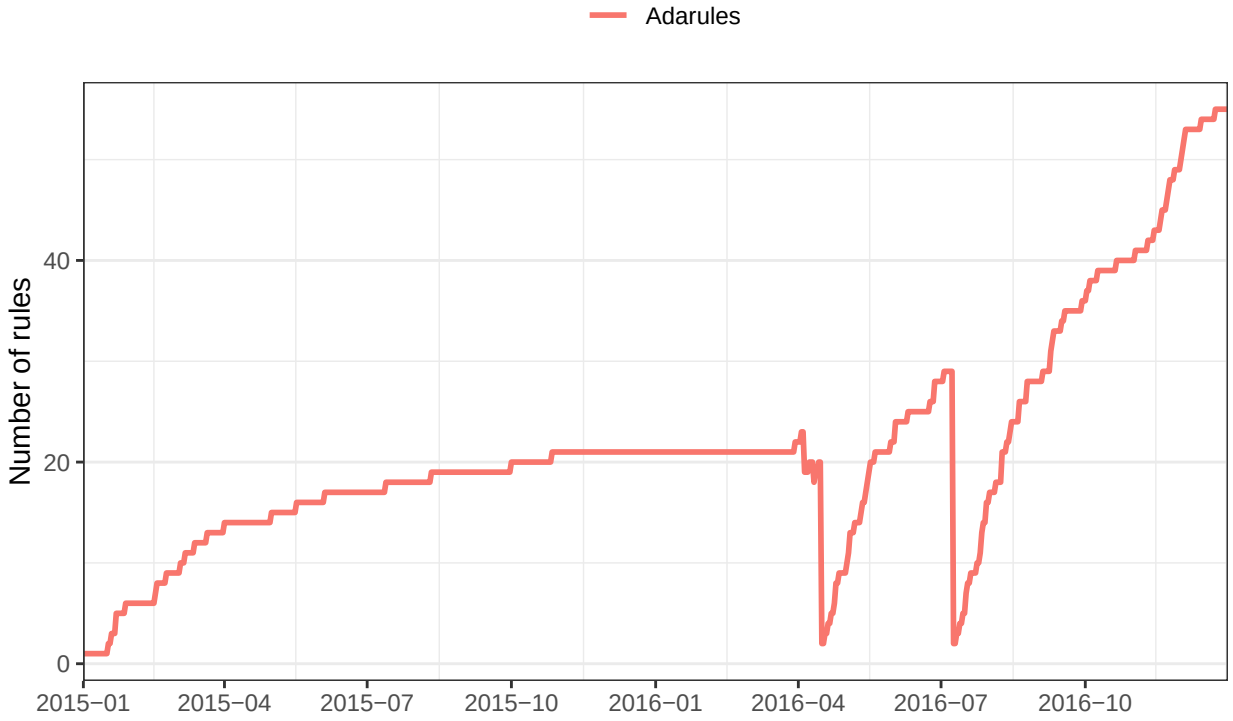
	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
ANA-y	3.24	3.24	3.27	3.27	3.22	3.28	3.26	3.24	3.28	3.24	3.28	3.27
ANA-q	3.95	3.87	3.97	1.31	1.36	1.31	1.41	1.41	1.40	1.51	1.52	1.51
ANA-m	4.84	2.92	3.02	2.97	2.99	3.05	2.95	2.87	2.94	2.96	2.90	2.93
HA-y	3.47	3.50	3.47	3.51	3.47	3.51	3.47	3.47	3.51	3.47	3.51	3.47
HA-q	4.20	4.25	4.20	2.46	2.45	2.46	2.44	2.44	2.46	2.44	2.46	2.44
HA-m	3.47	3.31	3.16	3.07	2.95	2.89	2.82	2.76	2.72	2.54	2.51	2.46
Santander | occupancy | t = 60 | nRMSE												
Adarules	3.66	3.53	3.35	3.22	3.01	3.04	2.78	2.85	2.66	2.19	2.10	2.02
ANA-y	3.62	3.67	3.64	3.66	3.61	3.67	3.63	3.63	3.67	3.62	3.67	3.64
ANA-q	4.48	4.45	4.50	1.42	1.45	1.43	1.38	1.38	1.33	1.40	1.39	1.41
ANA-m	5.21	3.22	3.31	3.23	3.23	3.24	3.20	3.22	3.15	3.27	3.18	3.26
HA-y	3.44	3.47	3.44	3.48	3.44	3.48	3.44	3.44	3.48	3.44	3.48	3.44
HA-q	4.18	4.25	4.18	2.41	2.40	2.41	2.39	2.39	2.41	2.39	2.41	2.39
HA-m	3.44	3.28	3.13	3.04	2.92	2.86	2.78	2.72	2.69	2.50	2.47	2.42

6.8 Adarules vs baselines: Gradual change scenario

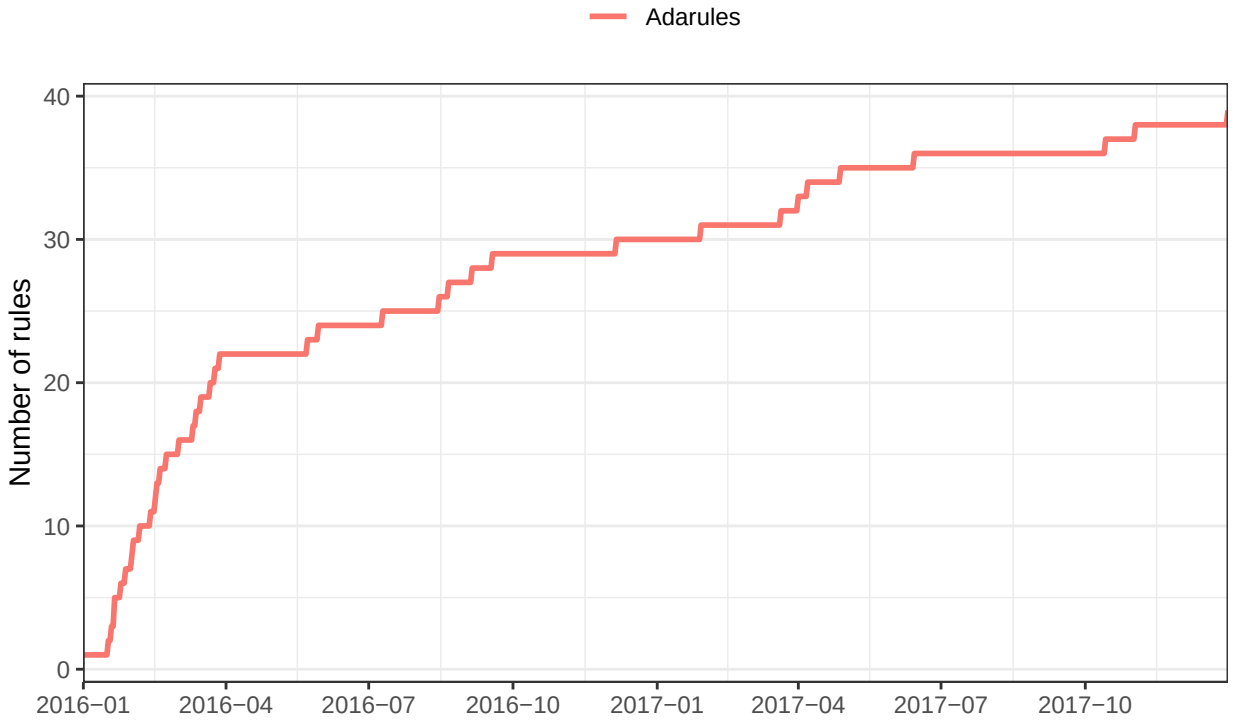
This section describes the third scenario and it consists of introducing a large number of small changes during the second year in each of the network datasets —M4/M7 and Santander—. More specifically, first year corresponds to real data and then during the second and every two months (January, March, May, July, September, November) a fake change is introduced over all the network by selecting 200 detectors at random where the traffic variables —flow, occupancy, and speed— from 100 of these detectors are incremented by a 4% while the other 100 detectors experience a 4% decrease in the traffic variable. This is maintained until, two months later, another round of smooth changes takes place while accumulating the one from the previous swapping. This implies that the first year is used to be learned *as is* —with the original real data— by Adarules, and as corresponds to every other baseline according to their updating schema. Therefore, the results of the experiments are shown, both graphically and in tables, only for the months during the second year. For this experimental scenario, the goal is to determine the Adarules ability to react and adapt to these gradual changes.

Firstly, the evolution of the number of rules for Adarules in both networks (Figure 6.9) is identical during the first year to the previously observed within the experiment which used real-data (Figure 6.7). This was expected, as the data used during the first year of this second change scenario is also real data, and it is only during the second year when artificial changes are introduced in the datasets. Starting from the second year, the evolution in the number of rules differs for each dataset. In the case of the M4/M7 network, there is certain stationarity in the number of rules during the first four months of the second year and then, after two consecutive executions of small artificial gradual changes —around the end of 2016/04—, Adarules detects several global concept drifts as observed in the small subsequent reduction in the number of rules. Following, Adarules detects and performs a timeline split as shown by the abrupt drop in the number of rules from around 20 to 2. This means that data before the specific datetime chosen in the splitting procedure is discarded, thus only considering observed data from that point in time. The reason probably because too many changes have been accumulated —and probably the tube-shaped network geometry also plays a role— which has lead Adarules to take the decision of performing such timeline splitting. Two months later, a similar situation occurs probably to make more accurate the timeline point where network flow patterns are different. After that point (2016/07), the number of rules in M4/M7 increases linearly as a function of time. On the other hand, in the case of Santander network, the evolution of rules and decisions performed by Adarules is different from that described for M4/M7. For the Santander network dataset, the number of rules grows quickly during the first three months and, afterwards, the pace for obtaining new rules is much slower during the second year. The key difference in how Adarules treats this kind of change in both networks may be explained because of as there is more uncertainty in the propagation of the flows within an urban network, these small changes are not detected as such hard breakpoints in time. Conversely, in a motorway network, the uncertainty around the propagation of the flows is much smaller and, thus, even small changes like these are probably going to be detected as global changes. This probably explains the two abrupt drops in the number of rules observed in the M4/M7 network, which are related to detected global changes in a given rule tied to the *timestamp* splitting attribute. As described in the previous paragraph, the selection of the 200 detectors every two months is random and, furthermore, these small changes are accumulated every two months. Thus, it could lead to a situation where a lot of small changes are focused on certain detectors that, on such motorway network with bounded variability, are eventually detected as *hard* global changes.

The assessment in terms of forecasting accuracy is performed using the normalized RMSE (nRMSE). There is one figure per forecasting traffic variable: Figure 9.13 for the flow, Figure 9.14 for the occupancy and Figure 9.15 for the speed. Each of these figures has subfigures for the two network datasets and for each forecasting horizon, resulting in four subfigures with the exception of speed



(a) M4/M7 network.



(b) Santander network.

Figure 6.9: Evolution of the Adarules complexity in the *gradual change* scenario. Number of identified rules as a function of the time —every iteration corresponds to one day—.

which has only two because there is no measured speed data in the Santander dataset. These results can be also corroborated in the Table 6.8.

In this case, it is clear how Adarules outperforms the rest of baselines in all the network datasets, traffic variables, and forecasting horizons. Even in the situation when Adarules structure is completely restored due to hard global concept drifts —in the M4/M7 network around the middle of the second year—, the performance is downgraded for a while but still more competitive than the baselines. Therefore, the response from Adarules is proper for such gradual changes in a network.

CONCLUSION: Adarules response is kept at suitable quality levels during the entire second year, showing the ability of Adarules to adapt itself when coping with — a lot of— small gradual changes. Furthermore, its accuracy outperforms that of the baselines competitors —ANA and HA—.

Table 6.8: Comparison of the forecasting performance measured by nRMSE between Adarules and baselines in the gradual change scenario. The KPIs are aggregated over the results during the last year in each of the datasets.

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
M4/M7 | flow | t = 15 | nRMSE												
Adarules	1.62	1.76	1.78	1.79	1.71	1.78	1.83	1.76	1.92	1.66	1.70	2.33
ANA-y	2.91	2.91	3.17	2.97	3.00	3.23	3.21	2.95	3.10	3.07	2.81	3.52
ANA-q	4.32	4.00	4.31	3.81	3.71	4.18	4.10	3.32	3.76	3.89	3.28	4.66
ANA-m	5.64	6.74	7.11	6.39	6.76	6.98	6.07	5.30	5.58	7.01	6.04	7.45
HA-y	6.67	4.14	5.03	4.69	3.97	4.46	3.93	3.87	5.20	4.86	5.68	9.19
HA-q	7.49	4.00	5.09	4.52	4.32	4.51	3.99	3.93	4.50	3.47	3.67	9.05
HA-m	6.67	3.98	5.01	4.49	3.91	4.26	3.67	3.60	4.20	3.65	3.79	8.87
M4/M7 | flow | t = 60 | nRMSE												
Adarules	4.72	4.16	4.35	5.33	4.64	4.83	5.45	4.36	4.48	4.01	3.98	6.86
ANA-y	6.20	5.52	6.12	6.31	5.58	6.17	5.99	5.85	6.17	6.30	5.90	7.29
ANA-q	7.78	6.57	7.48	6.81	6.05	6.37	6.89	6.42	6.64	7.19	6.28	8.17
ANA-m	8.21	8.44	8.61	7.76	7.45	8.14	7.81	6.81	7.64	8.32	7.04	9.54
HA-y	6.78	4.19	5.11	4.76	4.03	4.52	3.98	3.93	5.28	4.94	5.76	9.33
HA-q	7.61	4.05	5.17	4.59	4.39	4.58	4.06	3.98	4.56	3.52	3.72	9.19
HA-m	6.78	4.04	5.09	4.56	3.97	4.32	3.72	3.66	4.26	3.70	3.85	9.00
M4/M7 | occupancy | t = 15 | nRMSE												
Adarules	1.91	2.28	2.43	2.44	2.44	2.47	2.40	2.49	2.67	2.19	3.06	2.83
ANA-y	2.22	2.69	2.91	2.93	2.73	2.76	2.85	2.97	3.00	2.64	3.31	3.25

Table 6.8: Comparison of the forecasting performance measured by nRMSE between Adarules and baselines in the gradual change scenario. The KPIs are aggregated over the results during the last year in each of the datasets. *(continued)*

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
ANA-q	2.59	2.85	3.03	3.42	3.12	2.99	3.15	3.13	3.28	2.72	4.25	3.58
ANA-m	2.59	3.57	3.43	3.75	3.57	3.45	3.20	3.23	3.81	3.21	3.73	4.54
HA-y	2.58	2.85	2.97	3.36	2.80	3.09	2.84	3.15	3.59	2.54	3.52	3.77
HA-q	3.03	3.05	3.08	3.56	2.98	3.23	3.15	3.22	3.58	2.49	3.37	4.13
HA-m	2.58	2.86	2.96	3.30	2.75	3.10	2.85	3.12	3.42	2.41	3.19	3.85
M4/M7 | occupancy | t = 60 | nRMSE												
Adarules	2.31	3.04	3.20	3.54	3.23	3.44	3.32	3.24	3.64	2.51	3.62	3.53
ANA-y	2.77	3.26	3.50	3.69	3.49	3.62	3.27	3.40	3.88	3.19	4.04	3.76
ANA-q	2.94	3.33	3.93	3.97	3.57	3.70	3.50	3.40	3.90	3.18	5.21	4.09
ANA-m	2.80	3.59	4.05	4.06	3.75	3.92	3.37	3.44	4.17	3.46	5.17	4.94
HA-y	2.61	2.89	3.01	3.40	2.84	3.13	2.91	3.19	3.62	2.57	3.57	3.77
HA-q	3.07	3.07	3.12	3.61	3.02	3.27	3.18	3.26	3.62	2.53	3.41	4.14
HA-m	2.61	2.90	3.00	3.34	2.78	3.14	2.92	3.16	3.45	2.45	3.23	3.85
M4/M7 | speed | t = 15 | nRMSE												
Adarules	5.63	5.95	6.73	5.60	6.07	6.11	5.94	6.27	6.20	6.03	6.24	6.45
ANA-y	5.83	6.34	7.49	6.33	6.48	6.67	6.36	6.49	6.35	5.75	6.56	6.34
ANA-q	6.24	6.94	8.11	6.81	6.92	7.32	7.19	7.54	7.30	6.15	7.51	7.32
ANA-m	6.74	7.90	9.69	7.46	8.01	7.27	7.65	7.27	7.71	6.41	9.40	7.86
HA-y	6.81	7.39	8.67	7.15	7.22	7.55	7.19	7.65	7.22	6.86	8.58	7.62
HA-q	7.23	7.60	8.89	7.36	7.50	7.96	8.08	7.94	8.09	6.60	8.31	8.12
HA-m	6.81	7.32	8.67	7.00	7.16	7.45	7.34	7.63	7.41	6.59	8.04	7.66
M4/M7 | speed | t = 60 | nRMSE												
Adarules	6.22	7.59	8.07	6.81	7.15	7.19	7.23	6.88	7.22	6.73	7.97	7.01
ANA-y	6.41	7.59	8.45	7.35	7.57	7.76	7.41	7.27	7.85	6.68	8.07	7.10
ANA-q	6.69	7.78	8.86	7.55	7.86	8.04	8.17	7.71	8.37	6.39	8.41	7.79
ANA-m	6.96	8.18	9.63	7.65	8.35	7.46	8.42	7.45	7.84	7.83	9.56	8.20
HA-y	6.71	7.34	8.53	6.97	7.19	7.60	7.25	7.45	7.19	6.76	8.31	7.31
HA-q	7.17	7.65	8.71	7.26	7.40	7.85	8.05	7.76	7.96	6.51	8.20	7.98
HA-m	6.71	7.29	8.48	6.82	7.13	7.41	7.38	7.37	7.31	6.51	7.88	7.38

Table 6.8: Comparison of the forecasting performance measured by nRMSE between Adarules and baselines in the gradual change scenario. The KPIs are aggregated over the results during the last year in each of the datasets. *(continued)*

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Santander | flow | t = 15 | nRMSE												
Adarules	1.54	1.42	1.76	1.93	1.91	1.92	1.94	2.13	1.57	1.83	1.76	1.74
ANA-y	1.85	1.61	1.87	2.10	1.80	2.19	2.25	2.17	2.44	2.93	2.31	2.39
ANA-q	2.52	2.41	2.60	2.39	2.57	2.56	2.86	2.48	3.10	2.67	4.52	4.33
ANA-m	4.09	4.04	4.85	3.54	3.38	3.58	4.35	3.24	7.42	3.31	5.43	4.23
HA-y	5.57	3.55	3.74	4.26	3.70	4.63	6.12	6.54	4.83	5.78	5.04	5.94
HA-q	5.70	3.68	3.48	4.58	4.18	5.05	5.78	6.60	4.41	5.53	6.05	6.57
HA-m	5.57	3.48	3.85	4.29	3.62	3.78	6.00	6.17	4.37	4.56	4.51	5.50
Santander | flow | t = 60 | nRMSE												
Adarules	3.75	3.32	3.96	4.75	4.38	4.45	5.10	5.07	3.94	4.33	4.31	4.58
ANA-y	4.73	4.15	4.68	5.18	4.53	5.47	5.90	5.67	5.38	5.63	5.66	6.55
ANA-q	5.13	4.73	5.13	5.40	5.51	5.61	6.16	5.25	6.19	5.18	6.82	7.81
ANA-m	6.33	5.69	5.93	6.66	5.68	6.03	6.77	4.91	11.27	6.14	7.84	6.62
HA-y	5.35	3.58	3.77	4.29	3.74	4.69	6.04	6.59	4.86	4.25	5.10	6.00
HA-q	5.42	3.71	3.51	4.63	4.24	5.11	5.72	6.68	4.47	4.54	6.13	6.65
HA-m	5.35	3.51	3.90	4.33	3.66	3.80	5.93	6.10	4.42	4.32	4.58	5.57
Santander | occupancy | t = 15 | nRMSE												
Adarules	2.75	2.62	2.76	2.99	2.50	2.86	3.58	3.38	2.86	3.89	3.91	4.62
ANA-y	3.19	2.85	3.02	3.09	3.10	3.32	3.85	3.96	3.77	4.50	4.52	5.51
ANA-q	3.57	3.14	3.13	3.12	3.13	3.38	4.17	4.29	3.96	3.79	4.14	5.78
ANA-m	4.13	3.58	3.56	3.77	3.21	3.66	4.87	3.75	7.58	4.09	4.57	5.82
HA-y	3.63	2.99	3.03	3.24	3.21	3.52	4.33	4.40	3.61	4.03	5.47	6.79
HA-q	3.82	3.10	3.18	3.29	3.57	3.79	4.51	4.49	3.62	4.12	4.35	6.27
HA-m	3.63	3.00	3.00	3.25	3.18	3.39	4.21	4.17	3.88	3.53	3.82	6.27
Santander | occupancy | t = 60 | nRMSE												
Adarules	3.40	3.08	3.19	3.56	3.12	3.49	3.87	3.92	3.51	4.64	4.51	5.67
ANA-y	3.71	3.24	3.53	3.42	3.53	3.64	4.28	4.34	4.20	5.27	5.14	6.39
ANA-q	4.04	3.39	3.44	3.41	3.48	3.78	4.63	4.59	4.22	4.02	4.58	6.38
ANA-m	4.83	3.79	3.90	3.95	3.40	3.95	5.04	4.51	8.08	3.68	4.68	6.22

Table 6.8: Comparison of the forecasting performance measured by nRMSE between Adarules and baselines in the gradual change scenario. The KPIs are aggregated over the results during the last year in each of the datasets. *(continued)*

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
HA-y	3.67	3.03	3.08	3.29	3.26	3.57	4.33	4.40	3.65	3.55	5.56	6.90
HA-q	3.88	3.14	3.22	3.33	3.62	3.84	4.49	4.54	3.64	3.51	4.42	6.36
HA-m	3.67	3.04	3.05	3.29	3.23	3.44	4.21	4.22	3.94	3.36	3.88	6.36

6.9 Adarules vs baselines: Abrupt change (AM-PM) scenario

This section describes the fourth scenario and it consists of introducing abrupt changes by swapping the AM and PM periods during the second year in each network dataset. More specifically, first year corresponds to real data and then during the second and every two months (January, March, May, July, September, November) a fake change is introduced over all the network by swapping the AM and PM periods: i.e. the traffic is swapped and, thus, traffic during the night takes place during the day and vice versa. This is maintained for two months until the next swap takes place. This implies that the first year is used to be learned *as is* —with the original real data— by Adarules, and as corresponds to every other baseline according to their updating schema. Therefore, the results of the experiments are shown, both graphically and in tables, only for the months during the second year. For this experimental scenario, the goal is to determine the Adarules ability to react and adapt to these abrupt changes.

Firstly, the evolution of the number of rules for Adarules in both networks (Figure 6.10) is identical during the first year to the previously observed within the experiment which used real-data (Figure 6.7). This was expected, as the data used during the first year of this second change scenario is also real data, and it is only during the second year when artificial changes are introduced in the datasets. Starting from the second year, the rulesets run on each network dataset undergo similar changes in their structure and number of rules. Firstly, after the first change introduced in the month of January, several subsequent global concept drifts are detected as can be seen on the drops in the number of rules. The number of rules drops to half of the number of rules before the introduced changes in each dataset —from 22 to 14 rules in M4/M7, and from 30 to 16 rules in Santander—. Then, Adarules decides to simply restore the entire ruleset structure by forgetting certain past data as can be seen in abrupt drops of the number of rules. This is especially visible in the Santander network with several changes of this kind. Interestingly, Adarules seems to build the ruleset in a different way with the newly observed data in both datasets starting from around

July of the second year, as can be seen by the increasing number of rules over time with only minor corrections.

The assessment in terms of forecasting accuracy is performed using the normalized RMSE (nRMSE). There is one figure per forecasting traffic variable: Figure 9.16 for the flow, Figure 9.17 for the occupancy and Figure 9.18 for the speed. Each of these figures has subfigures for the two network datasets and for each forecasting horizon, resulting in four subfigures with the exception of speed which has only two because there is no measured speed data in the Santander dataset. These results can be also corroborated in the Table 6.9.

In this case, it is clear how Adarules outperforms the rest of baselines in all the network datasets, traffic variables, and forecasting horizons. Even in those situations when Adarules structure is completely restored due to hard global concept drifts, the performance is a bit downgraded for a while but Adarules is able to quickly learn and recover a good forecasting performance using the new observed data. Therefore, the response from Adarules is proper for such abrupt changes in a network.

CONCLUSION: Adarules is able to autonomously detect the abrupt changes and perform the required restructuring in order to quickly recover the expected level of forecasting quality. Moreover, the timings taken by Adarules to forget and learn new concepts seem appropriate given the comparison with the baseline competitors in terms of forecasting error.

Table 6.9: Comparison of the forecasting performance measured by nRMSE between Adarules and baselines in the abrupt change (AM-PM) scenario. The KPIs are aggregated over the results during the last year in each of the datasets.

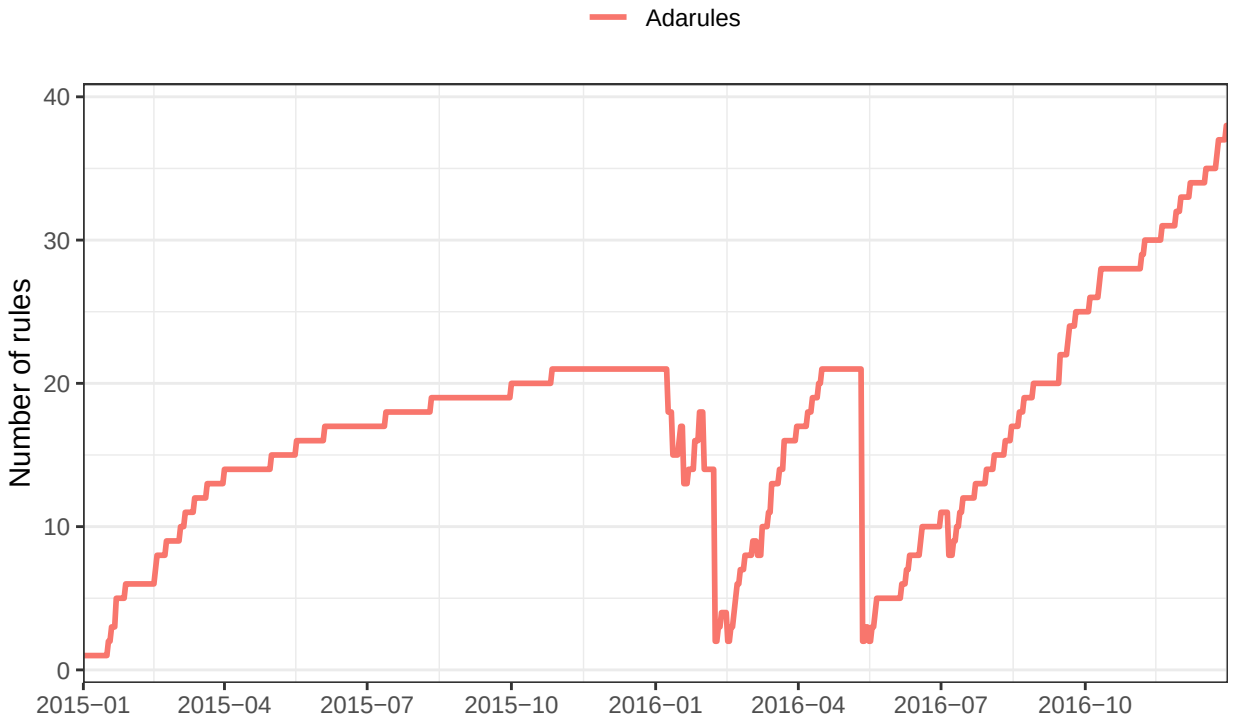
	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
M4/M7 | flow | t = 15 | nRMSE												
Adarules	4.77	3.16	6.35	4.92	4.74	2.63	6.21	4.01	3.82	4.13	6.56	4.01
ANA-y	9.60	10.21	3.06	3.10	10.61	10.60	3.03	2.94	10.73	10.38	2.89	3.51
ANA-q	13.30	14.66	4.40	12.19	7.95	8.54	13.24	13.47	8.15	13.36	7.20	7.66
ANA-m	19.45	6.88	26.28	6.21	21.43	7.09	22.52	5.70	24.00	7.79	25.37	7.48
HA-y	45.63	48.79	5.03	4.18	49.09	48.76	4.00	4.39	49.80	48.34	5.85	9.43
HA-q	46.07	48.91	5.03	32.10	18.72	19.64	33.03	34.47	18.24	33.21	18.30	18.87
HA-m	45.63	45.68	8.97	8.44	41.83	37.52	16.87	17.47	34.73	28.89	25.19	24.01
M4/M7 | flow | t = 60 | nRMSE												
Adarules	18.09	8.66	14.45	13.10	11.59	5.95	15.56	11.00	9.12	9.76	14.48	11.32
ANA-y	26.17	28.24	6.27	6.38	29.25	29.53	6.15	6.19	29.67	28.63	6.16	7.27
ANA-q	30.96	32.65	7.30	15.77	11.21	11.33	18.58	19.10	10.50	17.57	10.38	10.44

Table 6.9: Comparison of the forecasting performance measured by nRMSE between Adarules and baselines in the abrupt change (AM-PM) scenario. The KPIs are aggregated over the results during the last year in each of the datasets. *(continued)*

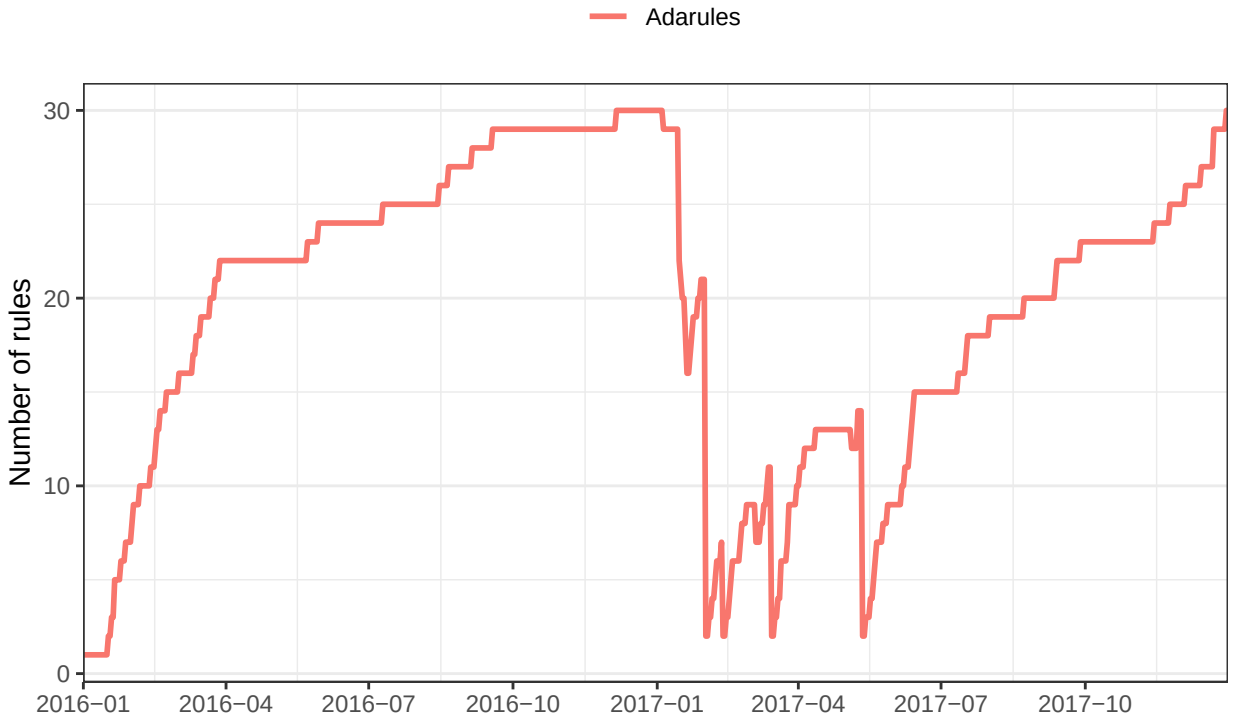
	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
ANA-m	39.25	8.71	41.53	7.69	37.32	8.51	38.69	7.24	40.33	8.80	41.90	9.91
HA-y	45.61	48.82	5.11	4.25	49.11	48.77	4.06	4.47	49.73	48.29	5.93	9.57
HA-q	46.07	48.96	5.12	32.06	18.78	19.62	33.05	34.42	18.16	33.09	18.24	18.91
HA-m	45.61	45.70	9.04	8.47	41.88	37.66	16.89	17.46	34.57	28.80	25.16	24.00
M4/M7 | occupancy | t = 15 | nRMSE												
Adarules	3.95	2.67	4.08	3.72	4.12	2.67	3.99	3.02	3.26	2.90	4.63	3.65
ANA-y	6.80	7.96	2.84	2.80	7.68	7.79	2.77	2.95	8.24	7.61	3.46	3.25
ANA-q	6.85	7.76	2.99	4.30	3.43	3.44	4.80	4.26	3.71	4.08	3.93	3.97
ANA-m	6.98	3.55	9.46	3.97	8.69	3.34	8.56	3.34	9.83	3.15	10.23	4.14
HA-y	9.37	10.69	2.91	3.29	10.01	10.50	2.97	3.12	10.98	10.01	3.48	3.86
HA-q	9.49	10.73	3.01	7.39	5.01	5.00	7.67	8.09	5.41	7.34	5.65	5.15
HA-m	9.37	10.17	3.38	3.76	8.70	8.28	4.55	4.71	8.21	6.27	6.70	5.83
M4/M7 | occupancy | t = 60 | nRMSE												
Adarules	5.48	4.04	5.16	4.79	5.23	3.73	5.38	4.12	4.51	3.80	5.93	4.79
ANA-y	7.90	8.89	3.42	3.45	9.30	8.85	3.31	3.43	9.54	9.06	3.90	3.79
ANA-q	7.95	9.04	3.81	4.92	4.10	4.27	5.01	5.04	4.21	4.60	4.20	4.23
ANA-m	8.63	3.68	10.61	3.84	9.68	3.95	10.00	3.53	10.06	3.28	11.94	4.85
HA-y	9.36	10.67	2.95	3.34	10.05	10.48	3.05	3.17	11.00	9.97	3.53	3.83
HA-q	9.50	10.73	3.06	7.42	4.95	5.00	7.70	8.12	5.45	7.35	5.71	5.19
HA-m	9.36	10.17	3.43	3.81	8.63	8.29	4.57	4.75	8.24	6.29	6.76	5.74
M4/M7 | speed | t = 15 | nRMSE												
Adarules	7.60	6.66	8.62	7.65	7.78	5.98	7.27	7.06	6.87	5.89	8.71	7.45
ANA-y	10.73	10.93	7.01	5.98	11.80	11.98	6.15	6.04	11.90	10.77	6.45	6.37
ANA-q	10.80	12.27	7.76	7.58	7.12	7.31	7.76	7.60	6.82	6.96	7.46	7.10
ANA-m	11.09	8.00	14.46	7.21	12.90	7.44	12.66	7.29	13.65	6.70	13.44	7.93
HA-y	13.49	14.91	8.28	6.68	14.67	14.85	7.05	7.38	15.17	14.43	8.21	7.51
HA-q	13.30	14.89	8.37	10.71	8.43	8.96	11.23	11.39	8.21	10.44	8.99	8.02
HA-m	13.49	14.26	8.52	6.92	13.06	12.31	7.81	8.18	11.06	9.55	9.94	9.00
M4/M7 | speed | t = 60 | nRMSE												
Adarules	9.95	8.63	10.34	8.73	8.94	7.67	8.66	7.91	8.68	7.43	10.41	9.76
ANA-y	12.22	13.48	7.96	6.81	13.30	13.70	7.10	6.78	13.86	13.15	7.75	6.94
ANA-q	12.29	14.08	8.30	8.66	8.12	8.37	8.53	8.45	7.72	8.05	8.56	7.57
ANA-m	12.34	8.90	15.56	7.58	14.65	7.90	14.11	7.65	14.15	7.71	20.78	8.56
HA-y	13.57	14.96	8.12	6.61	14.70	14.84	7.03	7.23	15.25	14.52	7.97	7.20

Table 6.9: Comparison of the forecasting performance measured by nRMSE between Adarules and baselines in the abrupt change (AM-PM) scenario. The KPIs are aggregated over the results during the last year in each of the datasets. *(continued)*

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
HA-q	13.44	15.01	8.28	10.57	8.48	8.98	11.18	11.30	8.28	10.54	8.90	7.99
HA-m	13.57	14.31	8.36	6.91	13.10	12.33	7.79	8.09	11.12	9.67	9.93	8.97
Santander | flow | t = 15 | nRMSE												
Adarules	5.00	2.67	4.28	2.45	3.54	2.88	3.51	3.03	3.07	2.88	4.51	3.13
ANA-y	6.63	6.77	2.07	2.20	7.16	7.05	2.21	2.69	8.70	8.32	2.13	2.41
ANA-q	9.34	9.44	2.97	6.55	5.71	5.72	7.03	9.03	5.34	5.40	9.07	9.83
ANA-m	14.04	4.44	19.15	3.79	12.68	3.87	13.69	3.56	20.51	3.54	21.19	4.13
HA-y	34.33	34.52	3.51	4.40	34.29	34.66	5.98	6.17	38.75	34.53	4.70	5.48
HA-q	34.78	35.03	3.16	19.98	15.94	15.92	24.20	26.65	12.81	22.95	18.15	17.58
HA-m	34.33	31.29	6.83	6.57	28.89	26.45	11.38	12.91	25.43	22.93	19.65	20.25
Santander | flow | t = 60 | nRMSE												
Adarules	14.71	6.67	13.64	6.52	8.92	7.10	8.70	8.13	9.11	6.68	11.84	8.31
ANA-y	20.82	20.71	4.55	5.41	25.44	23.83	5.68	6.19	28.79	30.82	5.39	6.39
ANA-q	22.57	22.49	4.94	10.09	9.03	10.29	11.64	14.74	9.16	12.13	12.87	13.82
ANA-m	24.73	6.40	28.09	6.73	24.21	6.26	27.92	5.81	36.42	6.39	33.52	6.44
HA-y	34.36	34.56	3.51	4.47	34.38	34.28	5.97	6.23	38.73	38.96	4.76	5.58
HA-q	34.78	35.05	3.20	20.05	15.92	15.61	24.16	26.53	12.84	22.93	18.25	17.81
HA-m	34.36	31.35	6.75	6.59	28.86	26.33	11.37	12.89	25.41	22.71	19.74	20.13
Santander | occupancy | t = 15 | nRMSE												
Adarules	5.23	2.98	4.30	2.83	3.21	3.40	4.85	3.71	4.14	3.90	6.88	4.85
ANA-y	7.39	6.44	3.02	3.26	6.12	6.02	3.73	3.99	10.51	6.85	4.26	5.38
ANA-q	9.13	7.99	3.24	4.03	3.86	4.20	6.00	5.71	4.94	4.91	6.24	6.54
ANA-m	9.87	3.47	8.49	3.76	7.92	3.64	9.55	3.90	17.40	4.09	13.37	5.18
HA-y	10.79	10.46	3.06	3.44	10.11	10.65	4.13	4.52	16.53	10.03	5.02	6.66
HA-q	11.22	11.38	3.20	6.74	5.99	6.15	9.38	12.14	5.81	6.87	9.32	9.22
HA-m	10.79	9.62	3.86	4.17	8.86	8.49	7.14	6.15	11.08	6.64	9.83	9.78
Santander | occupancy | t = 60 | nRMSE												
Adarules	6.36	3.63	6.02	3.99	4.57	4.60	6.69	5.33	6.23	4.38	8.13	6.73
ANA-y	10.06	10.11	3.43	3.53	9.46	9.67	4.17	4.60	13.72	10.33	4.93	6.30
ANA-q	10.60	10.39	3.43	4.27	4.46	4.87	7.35	7.51	6.29	5.58	6.93	8.11
ANA-m	10.98	3.74	13.94	4.10	10.84	3.74	11.17	4.77	19.24	3.86	16.48	5.74
HA-y	10.73	10.15	3.11	3.49	10.15	10.54	4.18	4.57	16.45	9.82	5.10	6.77
HA-q	11.10	11.27	3.23	6.71	5.96	6.08	9.47	12.15	5.75	6.77	9.26	9.29
HA-m	10.73	9.32	3.88	4.21	8.91	8.50	7.19	6.18	11.01	6.45	9.92	9.90



(a) M4/M7 network.



(b) Santander network.

Figure 6.10: Evolution of the Adarules complexity in the *abrupt change (AM-PM)* scenario. Number of identified rules as a function of the time —every iteration corresponds to one day—.

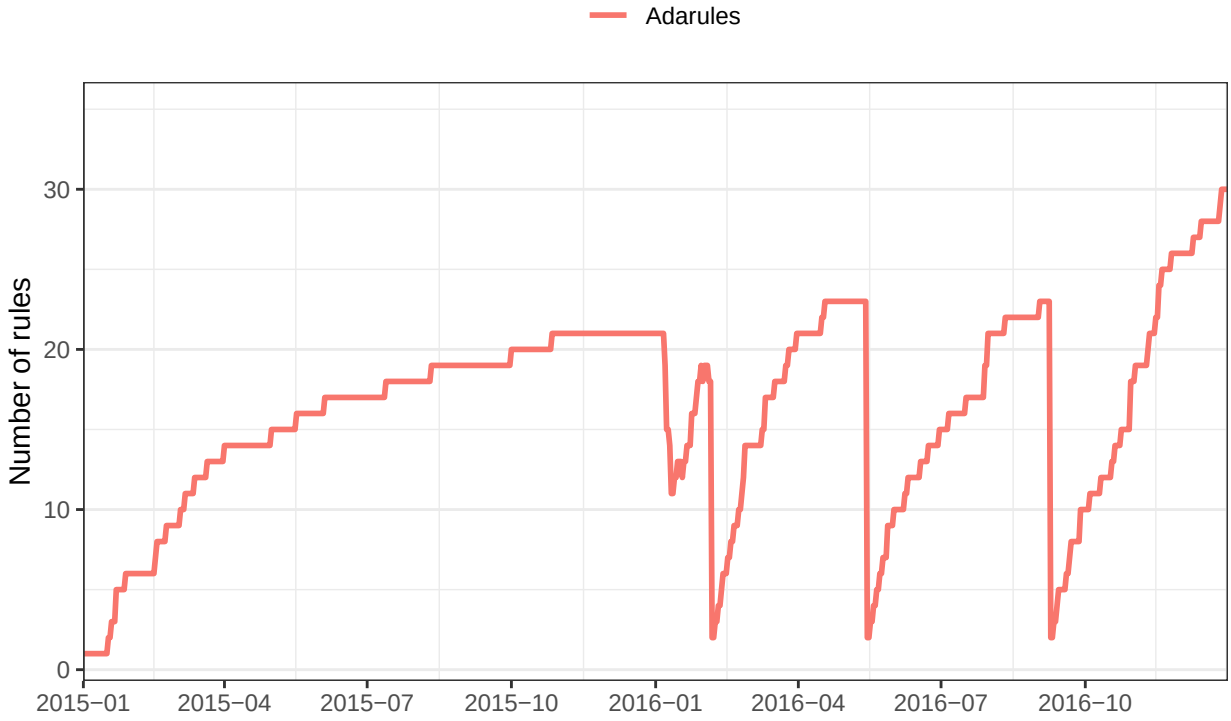
6.10 Adarules vs baselines: Abrupt change (IDs) scenario

This section describes the fifth scenario and it consists of introducing extreme abrupt changes by swapping 100 detectors identifiers selected at random during the second year in each network dataset. More specifically, first year corresponds to real data and then during the second and every two months (January, March, May, July, September, November) a fake change is introduced over all the network by swapping 100 detectors identifiers selected at random: i.e. the traffic from these detectors is swapped with others in the network. This is maintained until, two months later, another swapping takes place while accumulating the one from the previous swapping. This implies that the first year is used to be learned *as is* —with the original real data— by Adarules, and as corresponds to every other baseline according to their updating schema. Therefore, the results of the experiments are shown, both graphically and in tables, only for the months during the second year. For this experimental scenario, the goal is to determine the Adarules ability to react and adapt to these extreme abrupt changes.

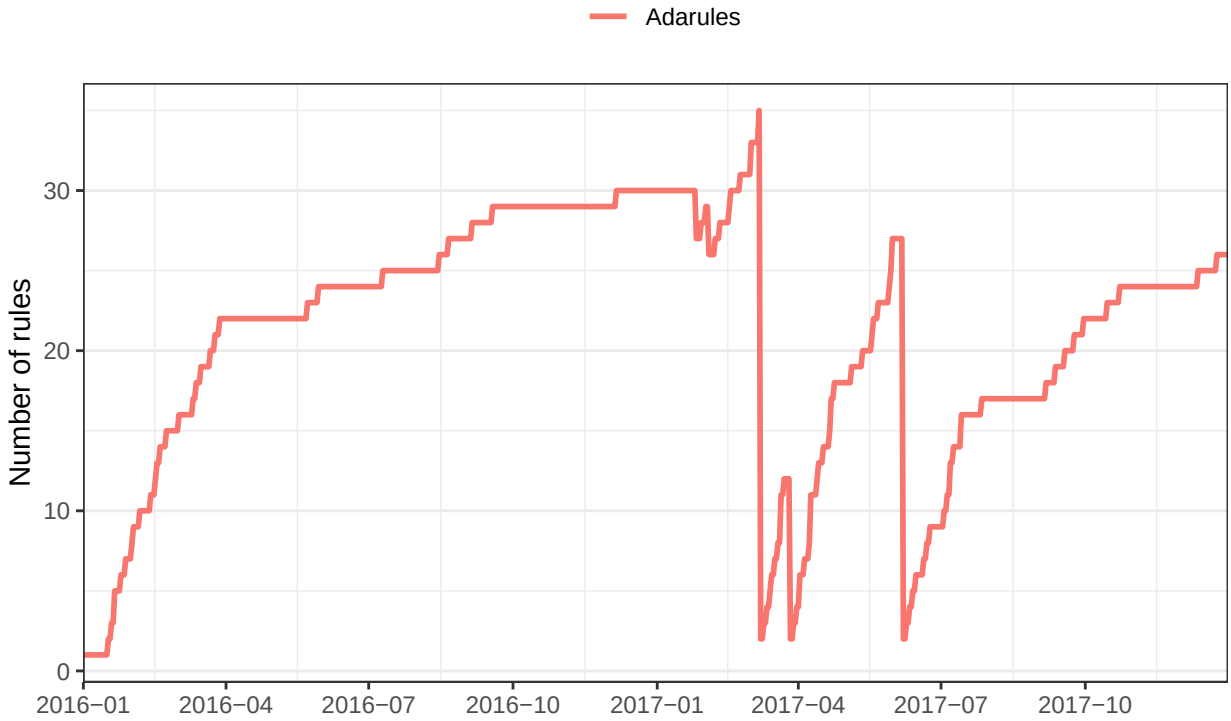
Firstly, the evolution of the number of rules for Adarules in both networks (Figure 6.11) is identical during the first year to the previously observed within the experiment which used real-data (Figure 6.7). This was expected, as the data used during the first year of this second change scenario is also real data, and it is only during the second year when artificial changes are introduced in the datasets. Starting from the second year, the rulesets run on each network dataset undergo similar changes in their structure in spite of the differences in the magnitude of the number of rules. In both networks, the number of rules experiments a drop during the first month of the second year as it is the first moment that an ID swap has been performed, and Adarules identifies it as several global changes. Then, the number of rules in M4/M7 decreases from 22 to 12, while on Santander the number drops from 30 to 26. Afterward, Adarules consider as best decision to restructure completely the ruleset after several of the bimonthly changes which are introduced in the datasets. This is performed similarly for M4/M7 and Santander networks.

The assessment in terms of forecasting accuracy is performed using the normalized RMSE (nRMSE). There is one figure per forecasting traffic variable: Figure 9.19 for the flow, Figure 9.20 for the occupancy and Figure 9.21 for the speed. Each of these figures has subfigures for the two network datasets and for each forecasting horizon, resulting in four subfigures with the exception of speed which has only two because there is no measured speed data in the Santander dataset. These results can be also corroborated in the Table 6.10.

In this scenario, it is again clear how Adarules outperforms the rest of baselines in all the network datasets, traffic variables, and forecasting horizons. Even in those situations when Adarules struc-



(a) M4/M7 network.



(b) Santander network.

Figure 6.11: Evolution of the Adarules complexity in the *abrupt change (IDs)* scenario. Number of identified rules as a function of the time —every iteration corresponds to one day—.

ture is completely restored due to hard global concept drifts, the performance is a bit downgraded for a while but still more competitive than the baselines. Therefore, the response from Adarules is proper for such extreme abrupt changes in a network.

CONCLUSION: Adarules is able to autonomously detect the extreme abrupt changes and perform the required restructuring in order to quickly recover the expected level of forecasting quality. Moreover, the timings taken by Adarules to forget and learn new concepts seem appropriate given the comparison with the baseline competitors in terms of forecasting error.

Table 6.10: Comparison of the forecasting performance measured by nRMSE between Adarules and baselines in the abrupt change (IDs) scenario. The KPIs are aggregated over the results during the last year in each of the datasets.

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
M4/M7 | flow | t = 15 | nRMSE												
Adarules	2.38	2.71	3.71	2.26	3.60	1.66	2.87	2.12	3.52	1.75	4.06	2.85
ANA-y	5.16	5.54	7.86	8.15	9.57	9.25	9.30	10.29	9.70	9.28	9.69	9.37
ANA-q	7.20	7.61	9.47	2.78	4.93	4.78	6.30	6.06	8.58	2.88	7.23	5.85
ANA-m	8.36	5.50	10.95	4.55	11.62	4.99	9.04	4.41	13.90	4.58	13.78	4.96
HA-y	6.35	6.09	14.93	15.60	18.86	13.91	15.72	16.08	22.08	18.03	18.93	18.33
HA-q	6.86	5.30	15.99	7.37	14.81	13.44	7.90	7.81	13.92	4.75	18.95	17.28
HA-m	6.35	6.07	13.83	13.33	14.84	11.20	11.02	11.61	12.10	12.68	13.31	12.01
M4/M7 | flow | t = 60 | nRMSE												
Adarules	6.15	5.96	8.23	6.22	8.99	4.60	6.29	4.92	8.27	4.37	8.73	6.12
ANA-y	13.34	15.17	26.66	26.93	27.19	26.14	24.83	25.76	27.03	24.85	33.10	29.44
ANA-q	14.18	15.67	33.72	5.37	12.66	12.46	12.99	14.04	28.76	6.07	15.22	14.98
ANA-m	26.95	6.50	25.28	5.68	22.30	6.05	25.67	5.03	19.60	6.06	31.72	7.26
HA-y	6.45	6.18	15.17	15.86	19.16	14.14	15.96	16.33	22.21	18.32	19.22	18.52
HA-q	6.96	5.38	16.25	7.49	15.07	13.66	8.11	7.94	14.22	4.82	19.26	17.45
HA-m	6.45	6.16	14.06	13.55	15.07	11.36	11.21	11.80	12.24	12.94	13.52	12.14
M4/M7 | occupancy | t = 15 | nRMSE												
Adarules	2.24	2.00	2.43	2.25	2.45	2.20	2.64	2.02	2.77	2.08	3.12	2.48
ANA-y	2.89	3.00	4.04	3.69	4.08	4.11	4.51	4.09	5.78	5.64	6.17	5.64
ANA-q	3.15	3.34	4.68	2.55	3.44	3.30	3.71	3.11	4.56	2.91	5.05	3.69
ANA-m	3.37	2.72	4.84	2.48	4.39	3.05	4.11	2.48	5.31	3.26	6.04	3.11
HA-y	3.27	3.49	5.26	4.72	4.04	4.29	5.79	6.26	6.59	6.13	7.77	6.89
HA-q	3.24	3.29	5.35	3.61	4.68	4.57	3.95	4.38	5.05	3.74	4.87	5.06
HA-m	3.27	3.32	5.04	4.48	4.13	4.15	4.57	4.15	5.22	4.43	6.15	4.36
M4/M7 | occupancy | t = 60 | nRMSE												

Table 6.10: Comparison of the forecasting performance measured by nRMSE between Adarules and baselines in the abrupt change (IDs) scenario. The KPIs are aggregated over the results during the last year in each of the datasets. *(continued)*

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Adarules	2.64	2.82	3.49	3.01	3.36	2.96	3.62	2.60	3.84	2.47	4.19	3.09
ANA-y	3.44	3.71	5.07	4.63	4.55	4.40	5.15	5.03	6.74	6.28	7.01	6.49
ANA-q	3.39	3.61	5.28	2.83	3.85	3.91	4.08	4.06	5.08	3.70	7.16	5.47
ANA-m	3.37	2.90	4.48	2.50	4.67	2.89	4.32	2.77	6.22	4.09	7.30	3.53
HA-y	3.30	3.53	5.34	4.79	4.08	4.34	5.86	6.32	6.69	6.24	7.89	7.02
HA-q	3.22	3.34	5.42	3.65	4.74	4.63	3.95	4.41	5.09	3.69	4.93	5.13
HA-m	3.30	3.36	5.11	4.54	4.20	4.19	4.52	4.19	5.25	4.22	6.23	4.34
M4/M7 | speed | t = 15 | nRMSE												
Adarules	7.07	6.90	9.12	8.09	8.31	5.81	7.05	6.91	7.68	5.78	8.38	7.36
ANA-y	7.68	8.46	10.60	9.01	13.25	12.87	12.84	12.58	16.55	13.55	14.50	18.89
ANA-q	8.00	8.74	10.98	7.61	10.01	9.89	8.10	8.46	14.06	6.61	11.94	12.02
ANA-m	8.48	7.56	11.87	7.32	11.09	6.51	9.37	6.02	9.61	6.70	17.53	7.82
HA-y	8.39	9.25	11.95	10.31	13.17	12.60	13.29	13.70	15.84	13.81	16.77	18.67
HA-q	8.67	9.13	11.40	7.92	12.05	11.83	8.08	9.06	13.90	7.67	16.79	20.76
HA-m	8.39	9.21	11.80	9.72	12.15	11.48	10.74	10.90	14.08	12.32	14.52	13.28
M4/M7 | speed | t = 60 | nRMSE												
Adarules	8.85	8.34	10.11	8.64	9.57	6.63	7.97	7.25	9.21	7.27	12.27	9.83
ANA-y	8.56	9.47	10.74	9.95	14.48	13.40	15.67	15.31	17.94	14.57	18.05	18.50
ANA-q	8.90	10.13	11.87	7.84	11.40	12.04	8.87	8.97	14.32	7.96	15.98	17.30
ANA-m	9.22	8.31	11.65	7.20	12.02	7.39	8.69	6.20	9.96	8.27	22.34	8.52
HA-y	8.33	9.30	11.76	10.25	13.11	12.60	13.20	13.81	15.56	13.36	16.68	18.55
HA-q	8.62	9.15	11.42	7.78	11.91	11.39	8.10	8.98	14.05	7.46	16.99	21.01
HA-m	8.33	9.22	11.71	9.64	12.00	11.32	10.83	10.95	13.98	12.31	14.64	12.72
Santander | flow | t = 15 | nRMSE												
Adarules	2.22	1.27	2.04	1.23	2.13	1.34	1.47	1.31	1.89	1.62	3.11	2.14
ANA-y	2.88	2.67	4.44	4.21	6.24	6.61	7.06	6.57	5.86	5.68	5.08	5.03
ANA-q	4.70	4.56	5.64	1.72	4.35	5.27	4.42	6.23	5.87	1.59	4.96	5.06
ANA-m	5.23	2.13	9.70	2.44	6.18	1.50	8.74	1.20	17.94	1.62	13.75	1.61
HA-y	5.21	3.47	6.31	4.90	13.69	9.33	15.51	16.04	16.38	16.78	9.08	10.01
HA-q	5.25	3.10	7.01	4.76	8.80	7.37	8.76	11.44	10.03	5.55	9.55	9.23
HA-m	5.21	3.38	5.99	4.82	8.75	6.87	12.50	11.41	13.64	13.26	9.80	8.12
Santander | flow | t = 60 | nRMSE												
Adarules	5.32	2.91	5.14	2.96	4.89	3.62	3.73	2.62	4.99	3.09	7.10	5.12
ANA-y	5.06	4.90	9.30	8.79	11.85	13.22	23.26	18.09	15.63	16.23	14.21	13.96

Table 6.10: Comparison of the forecasting performance measured by nRMSE between Adarules and baselines in the abrupt change (IDs) scenario. The KPIs are aggregated over the results during the last year in each of the datasets. *(continued)*

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
ANA-q	9.31	8.99	12.37	3.82	10.18	12.69	11.74	14.23	15.34	3.71	9.45	9.90
ANA-m	8.14	4.49	15.80	4.05	11.88	2.35	13.03	2.02	33.07	3.29	15.92	2.51
HA-y	5.12	3.52	6.68	5.10	13.91	9.44	15.72	16.22	16.63	16.39	9.23	10.06
HA-q	4.81	3.13	7.14	4.86	8.65	6.64	8.86	11.70	10.18	5.35	9.90	9.64
HA-m	5.12	3.42	6.31	4.96	8.87	6.95	12.66	11.55	13.89	13.24	10.00	8.22
Santander | occupancy | t = 15 | nRMSE												
Adarules	2.35	1.80	3.60	2.47	4.32	3.08	3.13	2.70	3.96	3.44	5.11	3.86
ANA-y	2.93	2.35	4.36	3.81	7.37	5.62	5.27	5.67	5.97	6.04	6.89	7.19
ANA-q	3.38	2.76	4.88	2.78	6.47	6.46	5.28	5.20	5.61	4.00	9.07	10.26
ANA-m	4.60	2.01	6.71	3.10	7.67	3.31	6.08	3.70	9.35	3.82	9.93	4.63
HA-y	2.74	2.03	5.50	4.01	7.44	6.56	6.41	6.01	6.17	8.23	8.85	8.62
HA-q	2.67	2.12	5.09	3.81	8.21	7.35	5.89	6.35	5.78	4.99	7.49	7.98
HA-m	2.74	2.00	6.06	4.63	7.35	5.79	5.25	5.40	5.14	5.55	7.41	6.64
Santander | occupancy | t = 60 | nRMSE												
Adarules	3.00	2.19	4.23	2.95	4.42	3.33	4.00	3.65	4.53	3.76	5.43	4.87
ANA-y	3.59	2.87	5.28	4.26	6.43	6.90	6.21	6.21	6.87	8.36	8.47	8.37
ANA-q	4.70	3.69	5.37	3.19	7.66	7.92	5.50	6.47	6.28	4.16	9.56	8.91
ANA-m	3.58	2.27	7.55	3.31	7.82	3.19	5.74	4.00	9.49	3.81	10.03	4.85
HA-y	2.77	2.05	5.57	4.55	7.60	6.62	6.45	5.94	6.20	8.34	9.11	9.05
HA-q	2.62	2.14	5.15	3.84	8.34	7.52	6.05	6.39	5.83	5.06	7.71	8.17
HA-m	2.77	2.02	6.07	4.70	7.21	5.87	5.24	5.31	5.20	5.62	7.64	6.74

7 Probabilistic model for robust traffic state identification

Efficient estimation of local traffic states at each detection site in urban and freeway networks is crucial for many real-time traffic management applications. Usually, these traffic states are inferred from the bivariate relationship between traffic flow and density —as stated in the fundamental diagram— using a deterministic approach. However, due to traffic congestion and position of detection sites especially in urban networks, this relation is highly scattered making these methods not suitable to handle the associated uncertainty in the process.

For this reason, we propose a probabilistic model that allows the inclusion of prior knowledge on traffic states and part of their relative parametrization according to the expert user's judgment. The model is formulated in a Bayesian framework where we also introduce several constraints as per the fundamental diagram shape to solve the common problem of identifiability in this kind of generative models used to estimate latent variables. The model has been published in the scientific literature [180].

Latent variables are those that are not directly observed but rather inferred from the set of variables which are directly measured. Distilling such latent variables during the data modelling process is widely recognized to be crucial, as they imply dimensionality reduction. Furthermore, these latent variables are generally more easily interpreted as they are usually tied to a more meaningful semantic interpretation, and lead to a concise representation of the observed data. Therefore, the focus is on finding such underlying latent variables present in the bivariate relationship between the traffic volume, usually measured as the number of vehicles per hour or another specific time resolution, and occupancy, i.e. the percentage of time that a vehicle occupied the detector. This research direction is mainly motivated by the theoretical and empirical findings found on the fundamental diagram of traffic flow [245] and Kerner's three-phase traffic theory [145], which already defines an existing correlation assuming that, on average, drivers exhibit same behavior under similar stationary conditions.

Still, estimating the traffic state associated with current traffic conditions in an urban context presents several difficulties because of non-homogeneous traffic conditions, flow perturbations —e.g. traffic lights— and the existence of transient states. Moreover, the existence of changes in the capacity —the foundation for the fundamental diagram— of a given link is much more frequent. These changes have their origin in various causes such as changes in the speed limit, the lanes distribution, etc. The model is built with the motivation to overcome these difficulties.

Although the model is not limited to it, we have so far used cross-sectional data captured by stationary sensors being induction loops the most ubiquitous ones [247]. For this reason, we have decided not to model the fundamental relationships where speed is used because of the inability of single-loop detectors to directly measure vehicle speed, but being estimated instead. We have used the time-aggregated traffic flow Q and the time-aggregated occupancy O using a specific time interval Δt.

The traffic flow Q is defined as the number of vehicles ΔN passing through specific location x within a time interval Δt:

$$Q(x,t) = \frac{\Delta N}{\Delta t}$$

Occupancy is the fraction of the time interval Δt during which the location x is occupied by a vehicle:

$$O(x,t) = \frac{1}{\Delta t} \sum_{\alpha=\alpha_0}^{\alpha_0+\Delta N-1} (t_\alpha^1 - t_\alpha^0)$$

where α represent each individual vehicle, t_α^0 the instant when the α^{th}-vehicle's front passes the detector and t_α^1 the instant when the α^{th}-vehicle's rear end passes the detector.

Finally, the performance of the model has been evaluated in the two networks: M4/M7 Motorways network in Sydney, and the urban city center of Santander. Results demonstrate the robustness of our approach to infer traffic states even with low data availability in some parts of the fundamental diagram.

7.1 Methodology

7.1.1 Capacity change detection

The capacity of a lane or a link in traffic is the maximum traffic flow (vehicles per hour) that can be accommodated in it during a given time period under prevailing roadway, traffic and control conditions. The fundamental diagram of traffic is built around the foundation of the *capacity* concept in order to explain the non-linear behaviour of traffic flow. And so does the probabilistic method proposed in this thesis.

However, the empirical or observed capacity of a given link or lane is subject to different influential factors such as the speed limit, the distribution of lanes —when considering multi-lane links—, the signalization or even the average driving mode. This possibility of structural change associated with the maximum capacity is much more frequent in an urban context due to the frequency of changes in signalization, roadworks, etc. Ideally, the method should be applied to individual lanes in order to reduce the degree of heterogeneity, so the method could be applied as a multi-level modelling approach from individual lane-detectors up to multi-lane stations. However, this is not always possible as sometimes the provided data has been already processed and aggregated. Besides, some kind of changes —e.g. signalization— should be informed by local authorities to traffic management centres for proper modelling corrections.

Still, we propose here an algorithm for the automated change detection of the observed capacity based solely on data, which is related to structural changes over time. In our case, we define the maximum capacity as the capacity from which a link becomes congested thus allowing a lower flow of vehicles while these spend more time —i.e. at a lower speed— crossing the link. This maximum capacity *pivot* is defined as a two-dimensional point of flow and occupancy —used as an equivalence for the unobserved traffic density— placed in the plane of traffic flow and occupancy. In Figure 7.1, it is shown the algorithmic details about how to detect multiple observed capacities on a given lane or link. In essence, the algorithm jointly utilizes the time series of flow and occupancy to identify the maximum observed capacities (flow, occupancy) over time. To this end, it is important to deseasonalize the maximum observed flow. Then, the key is to identify different clusters —modelled as bivariate Normal distributions— of pairs (flow, occupancy) that depict an observed capacity. Lastly, the mean of every of these identified components performs as a prior capacity.

An example on the application of this algorithm can be seen in Figure 7.2 where two different capacities have been identified by the algorithm which has classified every day according to the observed capacity. The original flow-occupancy diagram for this station (1024) is shown in Figure 7.3 where two different capacities can be visually distinguished very clearly.

Algorithm 15: Detecting changes in observed capacity over time

Input : $Q(t)$: Time series of traffic flow for a given lane/link
$O(t)$: Time series of occupancy for a given lane/link
$Q^{\tau} = 0.975$: Preset quantile for getting daily maximum flows
$\omega_D = 15$: Sliding window days considered for getting seasonal daily maximum flows
$\tau_Q = 0.05$: Bandwith around daily maximum flows
$\mathcal{M}_{max}$: Maximum number of mixture components
$\mathcal{B}^{\tau}$: Threshold used for the Bhattacharyya distance

Output: Ξ: List of distinct observed capacities within the time series

begin
  $Q^*(d) \leftarrow$ daily maximum flows applying Q^{τ} on every date in $Q(t)$
  `// Dissociate typical seasonality (e.g. weekends) from observed maximum flow`
  $Q^*(d) \leftarrow$ Apply rolling mean in Q^* as defined by ω_D
  foreach *date* d *in* D **do**
    `// Estimate the corresponding occupancy for every daily max flow` $Q^*(d)$
    $Q^*_{\tau}(d) \leftarrow Q^*(d) \cdot \tau_Q$
    $O^*(d) \leftarrow \text{median}(O(t))\ \forall t \in d$ and whose $Q(t)$ is within $Q^*(d) \pm Q^*_{\tau}(d)$
  end
  `// Using the` D `observations` $[Q^*(d), O^*(d)]$`, perform the estimation of the optimal (BIC) number of 2-D Gaussian mixture components up to` $\mathcal{M}_{max}$
  $\mathcal{M} \leftarrow$ Gaussian.mixture.components$(Q^*(d), O^*(d))$
  foreach *pair* $\mathcal{M}_i, \mathcal{M}_j$ *in* $\mathcal{M}$ **do**
    `// Calculate Bhattacharyya distance between` $\mathcal{M}_i, \mathcal{M}_j$
    $\mathcal{D}_{ij} \leftarrow$ Bhattacharyya.distance$(\mathcal{M}_i, \mathcal{M}_j)$
    if $\mathcal{D}_{ij} \leq \mathcal{B}^{\tau}$ **then**
      Merge components $\mathcal{M}_i, \mathcal{M}_j$ and proceed recursively with close enough neighbour components from either $\mathcal{M}_i$ or $\mathcal{M}_j$
    end
  end
  foreach *component* $\mathcal{M}_i$ *in* $\mathcal{M}$ **do**
    `// Every (remaining) component's mean is a detected prior capacity for the given link`
    return $\mathcal{M}_i$
  end
end

Figure 7.1: Change detection of the observed capacity based on the capacity defined by traffic flow and occupancy.

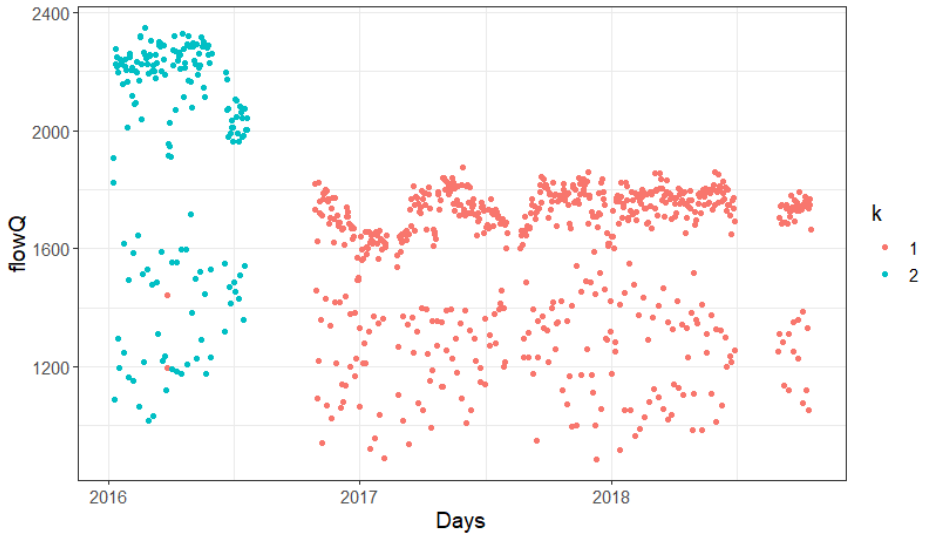


Figure 7.2: Classification of the daily maximum observed capacities for a given link-station (1024) in Santander. Every point correspond to the daily maximum observed capacity. The two color reflects the two detected capacities during these two years and the classification of the days.

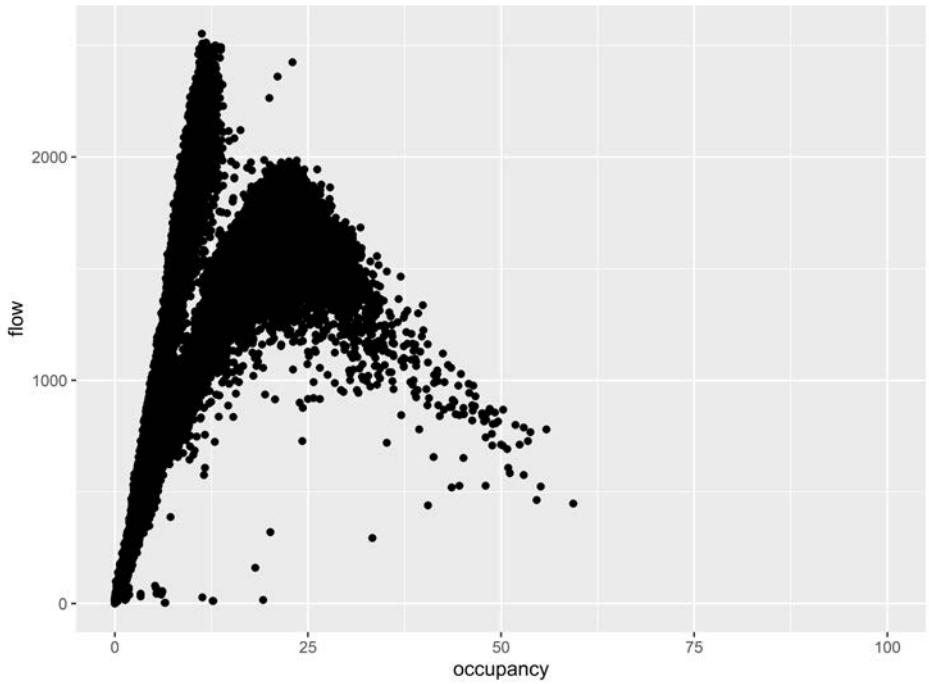


Figure 7.3: Flow-occupancy diagram for a given link-station (1024) in Santander. Two different capacities can be visually distinguished.

7.1.2 Traffic state identification

The proposed modelling method follows a probabilistic approach for finding the a-posteriori maximum probability model to explain the observed data while imposing certain constraints derived from prior domain knowledge. In this sense, we impose the number of underlying traffic states. This makes the model more interpretable by definition, and able to infer even in the regions with less observed data. Furthermore, the proposed model can classify the different traffic states associated with regions in the plane, and also fit the lines associated with the fundamental relation. In addition, the parametric nature of the problem formulation makes the approach more robust avoiding the overfitting.

7.1.2.1 Bivariate relationship of a single traffic state by means of a geometric interpretation

A single traffic state component is described by a bivariate Gaussian distribution. Intuitively it is located somewhere on the plane of the fundamental diagram as it will be shown later, and it is pulled by the degree of correlation between both variables, flow and occupancy, in that specific region of the fundamental diagram. The decision to use a bivariate Gaussian distribution is motivated because the *perfect* fit would be given by the line from the fundamental diagram itself, and thus, it is reasonable to assume that the existing area of uncertainty around such line (the scatter of points) could be modelled by the elliptic form of a bivariate Gaussian distribution. This distribution may be, indeed, normally distributed around such line, but the main reason of such decision is to facilitate computations that could be improved upon.

Such multivariate normal distribution is fully explained by its mean vector μ and its symmetric positive semi-definite covariance matrix Σ, being the following in the bivariate case:

$$\Sigma = \begin{bmatrix} \sigma(Q,Q) & \sigma(Q,O) \\ \sigma(O,Q) & \sigma(O,O) \end{bmatrix} = \operatorname{diag}(\Sigma)\, \Omega \operatorname{diag}(\Sigma),$$

where Ω is the correlation matrix for traffic flow Q and traffic occupancy O, and $\operatorname{diag}(\Sigma)$ represent the vector of their standard deviations. The covariance matrix Σ defines both the spread (variance) and the orientation (covariance) of the data. Still, in order to gain a more geometrical interpretation and for the ease of modelling, we can uniquely decompose such symmetric positive semi-definite matrix into a pair of vectors and magnitudes, i.e. the eigenvectors $\{v_1, v_2\}$ and the eigenvalues $\{\lambda_1, \lambda_2\}$. We recall that these define the shape of the data pointing into the direction of the largest spread of the data, and whose magnitude equals the spread in this direction. The eigendecomposition expresses matrix Σ in terms of its eigenvectors and eigenvalues $\Sigma = V \Lambda V^T$,

where V is the matrix containing the eigenvectors and Λ is the matrix containing the corresponding eigenvalues along the diagonal, and zeros elsewhere. The largest eigenvector $\vec{v_1}$ points into the direction of the largest spread of the data, whereas $\vec{v_2}$ is always orthogonal to $\vec{v_1}$ and points into the direction of the second largest variance of the data. Furthermore, these eigenvectors $\vec{v}$, which are unit vectors, can be easily reparameterized into a single value $\vec{v} = [\cos\theta \ \sin\theta]^T$, where θ is the plane angle. Once dealt with the orientation of the data shape, it is still necessary to reparameterize to deal with the magnitudes, related to the eigenvalues Λ, for each orientation in order to have the error ellipse representing the covariance matrix $(\frac{Q}{\sigma_Q})^2 + (\frac{O}{\sigma_O})^2 = s$, where s defines the scale of the ellipse. The left-hand side of this ellipse equation represents the sum of squares of independent normally distributed data samples, and such sum of squared Gaussian data points is known that follows a Chi-Square distribution with two degrees of freedom in this case. Thus, to use a 95% confidence level for instance, we use the 95^{th} quantile of a χ_2^2 distribution which is equivalent to $s \approx 5.991$. Now, using this scalar s we can make a direct conversion between the eigenvalues Λ and the ellipse's axis lengths by $2\sqrt{s\lambda}$.

We perform eigendecomposition to seek the orthonormal vectors of the covariance matrix XX^T of flow and occupancy for each traffic state component. A more general approach is to use Singular Value Decomposition (SVD) on the original data matrix X. SVD is more general in that it applies to non-square matrices, but we do not need this generality and thus we do not need to pay its higher computational cost.

7.1.2.2 Extending the nomenclature for different traffic states

The aforementioned methodology is valid for describing a specific traffic state, but it can be easily extended for K states through a probabilistic graphical model (PGM) notation, as shown in Figure 7.5. This PGM reflects the set of K traffic states where each one is described as a bivariate Gaussian distribution. These are shown in the right of the figure but taking into account that they are depicted without the reparameterization described in the last section only for a matter of ease of comprehension. Each of this K Gaussian components is described by its mean μ_k and its covariance Σ_k, both having their corresponding prior hyperparameters μ_k^0 and Σ_k^0 respectively. On the left side of the figure, there is the set of N points where each of this x_n data observations, composed of traffic flow and occupancy, is associated with one of the aforementioned K components. This association of the n^{th} datapoint and k^{th} traffic state is depicted through the z_n node which corresponds to a categorical multinomial distribution of size K. Finally, this multinomial distribution is associated with a symmetric Dirichlet distribution π. The core of the proposed model is to adjust its parameters represented as Ψ which best explain the observed data x, i.e. maximizing the model

probability:

$$P(\Psi|x) \mathrel{+}= P(\pi_k) + P(x_n|\mu_k, \Sigma_k) \quad \forall k \in K, \forall n \in N$$

Rather than the model probability, we actually maximize the log-probability (i.e. $\log P(\Psi|x)$) because of its nicer mathematical properties when dealing with derivatives. However, the semantics of the problem definition stays the same.

We have assumed $K = 5$ traffic states in our model which are related to the underlying correlation found on the fundamental diagram:

1. K_1: (Almost) no traffic,
2. K_2: Free-flow conditions,
3. K_3: Maximum capacity,
4. K_4: Traffic congestion conditions,
5. K_5: Total congestion.

The motivation of defining such traffic states conditions are motivated by the theoretical statements and empirical findings related to the shape of the fundamental diagram of traffic flow, and mainly because for their reasonable and very interpretable semantics. It is important to clarify that no automatic discovery of clusters is involved during the process that could be tagged afterward with the corresponding traffic state as in a typical clustering application. The Gaussian components are already explicitly labeled from the beginning taking advantage of the a priori knowledge derived from the fundamental diagram. In this way, the observed data is then used to calibrate geometrically these components that are not going to mix or overlap with each other because of the geometric constraints included in the formulation of the model. A diagram showing the layout of these $K = 5$ components in the flow-occupancy plane can be seen in the Figure 7.4.

Nomenclature of the PGM for each of the $K = 5$ traffic states is shown in Table 7.1, including all the parameters used in the model. The first column reflects the target component and its geometric interpretation: location, orientation or shape which refer to the mean and variance in a Normal distribution, respectively. The second and third columns show the parameter nomenclature and the corresponding reparameterization used in the model. The term *reparameterization* is defined to express a set of variables —such as the covariance of a multinormal distribution— in terms of others such as e.g. the eigenvalues and eigenvectors obtained from the eigendecomposition of the matrix. The fourth column tells whether the parameter is an input value given to the model, a constant value, a sampled parameter to be calibrated from observed data or a derived parameter

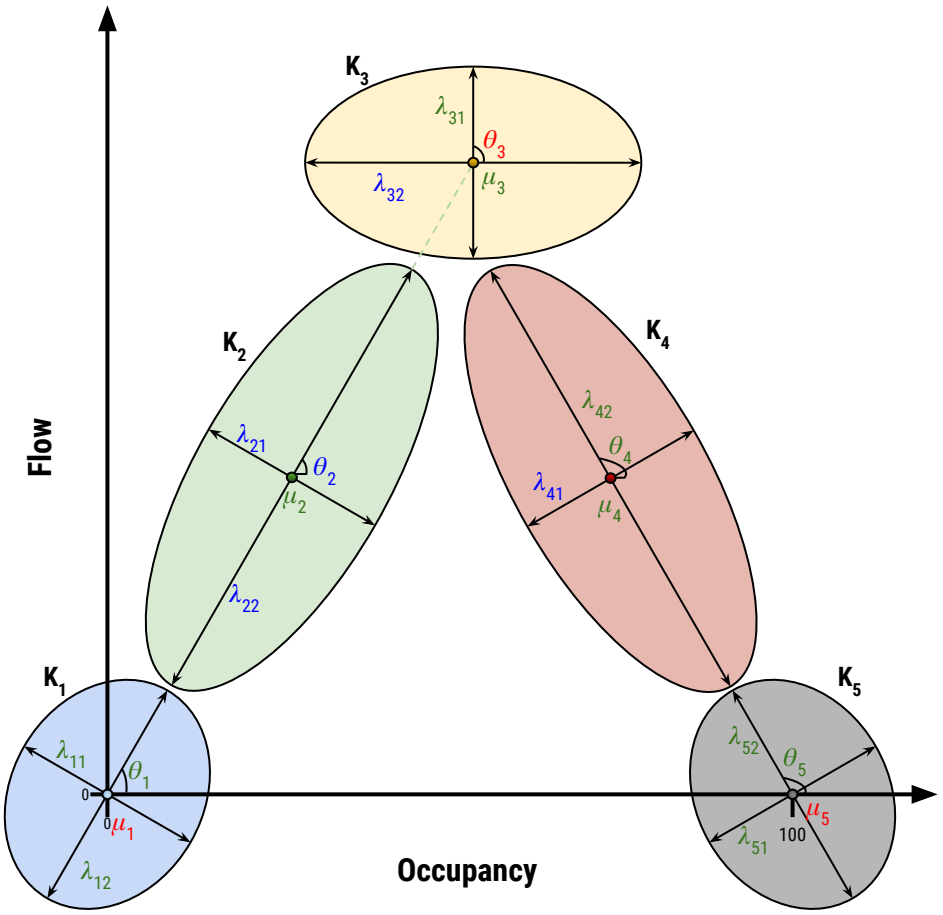


Figure 7.4: Block diagram showing the layout of the $K = 5$ traffic states components placed in the flow-occupancy diagram. Red parameters are fixed or constant, blue parameters are fit using the observed data, and green parameters are derived from others according to the imposed geometrical constraints. Parameters μ depict the mean of the components' —their center—; θ depict the components' orientation; and λ depict the components' shape through the eigenvalues.

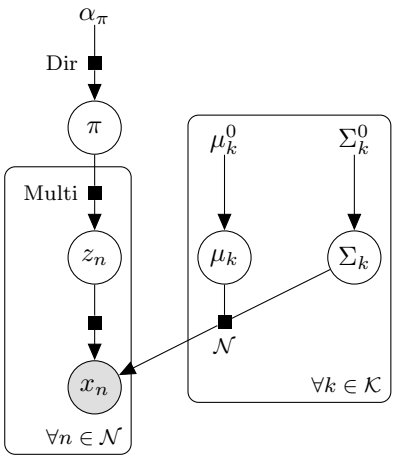


Figure 7.5: Probabilistic graphical model for traffic state identification.

Table 7.1: Nomenclature of the traffic states PGM.

Name	Reparameterization		Type	Value
K proportions	π_K		Sampled	$Dir(\alpha_\pi)$
K_1 occ ratio	τ_{11}		Input	0.05
K_1 flow ratio	τ_{12}		Input	0.15
K_1 location	μ_1		Constant	[0 0]
K_1 orientation	v_{11}, v_{12}	θ_1	Derived	θ_2
K_1 shape	Σ_1	λ_{11}	Derived	τ_{11}
K_1 shape	Σ_1	λ_{12}	Derived	τ_{12}
K_2 location	μ_2		Derived	$\mu_1, \lambda_{12}, \theta_2, \lambda_{22}$
K_2 orientation	v_{21}, v_{22}	θ_2	Sampled	$\mathcal{N}(\mu_{\theta_2}, \sigma_{\theta_2})$
K_2 shape	Σ_2	λ_{21}	Sampled	$\mathcal{N}(\mu_{\lambda_{21}}, \sigma_{\lambda_{21}})$
K_2 shape	Σ_2	λ_{22}	Sampled	$\mathcal{N}(\mu_{\lambda_{22}}, \sigma_{\lambda_{22}})$
K_3 flow ratio	τ_{31}		Input	0.20
K_3 location	μ_3		Derived	$\mu_2, \lambda_{22}, \lambda_{31}$
K_3 orientation	v_{31}, v_{32}	θ_3	Constant	0
K_3 shape	Σ_3	λ_{31}	Derived	τ_{31}
K_3 shape	Σ_3	λ_{32}	Sampled	$\mathcal{N}(\mu_{\lambda_{32}}, \sigma_{\lambda_{32}})$
K_4 location	μ_4		Derived	$\mu_5, \theta_4, \lambda_{52}, \lambda_{42}$
K_4 orientation	v_{41}, v_{42}	θ_4	Derived	μ_5, μ_3
K_4 shape	Σ_4	λ_{41}	Sampled	$\mathcal{N}(\mu_{\lambda_{41}}, \sigma_{\lambda_{41}})$
K_4 shape	Σ_4	λ_{42}	Derived	$\mu_5, \theta_4, \lambda_{52}$
K_5 flow ratio	τ_{51}		Input	0.15
K_5 location	μ_5		Constant	[1 0]
K_5 orientation	v_{51}, v_{52}	θ_5	Derived	θ_4
K_5 shape	Σ_5	λ_{51}	Derived	λ_{41}
K_5 shape	Σ_5	λ_{52}	Derived	τ_{51}

which is calculated from others. Therefore, the fifth column shows a default value, the distribution to be sampled or the parameters which are used to derive the corresponding parameter. Input refers to those parameters which are fixed as they represent relative ratios for some traffic states. This means they can be set according to the traffic engineer's criteria, or they can be left to the reasonable defaults. Sampled refers to those parameters which are calibrated using the observed traffic volume and occupancy data. More specifically, traffic states proportions π has a symmetric Dirichlet prior distribution, while the others five parameters are sampled from a prior normal distribution whose hyperparameters are derived from what we call the *maximum capacity pivot*. Finally, derived parameters are automatically calculated during the fitting by satisfying certain constraints and basic geometric manipulation.

Table 7.2 shows the set of constraints defined in the model formulation as described in the algorithm on Figure 7.6 to solve the probability maximization problem. These parameters are, thus, calibrated by the sampling or optimization method in order to maximize the resulting model probability given

Table 7.2: Parameter constraints during the sampling process.

Parameter	Constraint
θ_2	Support in $[\pi/6,\ \pi/2]$ rad
λ_{21}	Support in $[0, 0.15]$
λ_{22}	Support in $[0,\ dist([0,0],[1,1])]$
λ_{32}	Support in $[0,\ dist([0,0],[1,1])]$
λ_{41}	Support in $[0,\ dist(\mu_4,\ intersect(K_2^{x+}, y = \mu_{42}))]$

the observed data, while at the same time complying with the imposed constraints. The model has been implemented in the probabilistic programming language Stan [51] which provides full Bayesian inference for continuous-variable models through Markov chain Monte Carlo using very efficient samplers —such as the No-U-Turn sampler or Hamiltonian Monte Carlo—, but also gradient-based variational Bayesian methods for approximate Bayesian inference. Its computational efficiency along with the flexibility to specify the model have been the reasons for its choice. For example, the model of a given detector with one-year data can be fit using variational Bayes in the order of 5 to 10 minutes, using a single CPU core —therefore the fit of the entire network can be easily parallelized at detector level—.

The resulting model is a K-dimensional vector π with the data proportions for each traffic state and a set of (K) $D-$dimensional μ vectors with the means of the traffic states in the plane, together with the corresponding Dx$D-$dimensional covariance matrices Σ, where $D = 2$ because we are using two variables: traffic flow and occupancy. This means that we have K bivariate Gaussian distributions whose probability density functions can be used to classify new instances very efficiently in real time, given their good analytical and computational properties. Furthermore, if we are required to update the model online using incoming mini-batches of data, well-established and efficient frameworks such as Expectation-Maximization (EM) or streaming variational Bayes [46] could be applied in a straightforward way.

7.1.2.3 Geometrical constraints

Besides the constraining of the sampling space for those parameters shown in Table 7.2, there are additional geometrical constraints. These are more easily understood in conjuction with Figure 7.4. The imposed geometrical constraints within the model formulation include:

- Location —i.e. centers— of K_1 and K_5 are constrained to a specific location: $mu_1 = [0, 0]$ and $mu_5 = [100, 0]$.
- Certain parameter are directly derived from their immediate component as shown in the nomenclature in Table 7.1. For instance, orientation in components K_1 and K_5 is the same

Algorithm 16: Probabilistic model for traffic states

Input : X: Data with traffic flow and occupancy for a given capacity component $\mathcal{M}_i$
τ_{11}: K_1 occ ratio
τ_{12}: K_1 flow ratio
τ_{31}: K_3 flow ratio
τ_{51}: K_5 flow ratio

Output: μ_k: Vectors of means for the $K = 5$ Gaussian components
Σ_k: Matrices of covariances for the $K = 5$ Gaussian components
π_k: Vector of proportions among the $K = 5$ Gaussian components

begin
  Normalize $X = [O, Q]$ to be in range $[0, 1]$
  Set $\mu_1 = [0\ 0]$
  Set $\mu_5 = [1\ 0]$
  Set $\theta_3 = 0$
  Sample $\pi \sim Dir(\alpha_\pi)$
  Sample $\theta_2 \sim \mathcal{N}(\mu_{\theta_2}, \sigma_{\theta_2})$
  Sample $\theta_5 \sim \mathcal{N}(\mu_{\theta_5}, \sigma_{\theta_5})$
  Sample $\lambda_{21} \sim \mathcal{N}(\mu_{\lambda_{21}}, \sigma_{\lambda_{21}})$
  Sample $\lambda_{32} \sim \mathcal{N}(\mu_{\lambda_{32}}, \sigma_{\lambda_{32}})$
  Sample $\lambda_{51} \sim \mathcal{N}(\mu_{\lambda_{51}}, \sigma_{\lambda_{51}})$
  Derive Σ_k foreach $[\lambda_{k1}\lambda_{k2}]$
  Derive $[\mu_2, \mu_3, \mu_4]$ using constraints
  for $n := 1$ *to* N **do** *Data loop*
    for $k := 1$ *to* K **do** *Traffic states loop*
      $P(\Psi|x)$ += $P(\pi_k) + P(x_n|\mu_k, \Sigma_k)$
    end
  end
end

Figure 7.6: Pseudocode for the traffic states PGM.

as that of K_2 and K_4, respectively.

- The location μ_2 from K_2 is set according to the projection of vectors and lengths using: μ_1, θ_2, λ_{12} and λ_{22}.
- The orientation θ_4 from K_4 is set according to the angle formed between μ_5 and μ_3 through the arctan using both points.
- Let $\gamma_{k_3^{y-}}$ be the point in the plane defined by the point μ_3 projecting the distance from the minor eigenvector λ_{31} towards the orthogonal direction of θ_3 —i.e. $\theta_3 + 3\,\pi/2$—.
- Let $\gamma_{k_5^{y+}}$ be the point in the plane defined by the point μ_5 projecting the distance from the major eigenvector λ_{52} towards the direction of θ_5.
- The major eigenvector λ_{42} from K_4 is set according to the distance from $\gamma_{k_5^{y+}}$ to the point defined by the intersection of $\gamma_{k_3^{y-}}$ and the line which is defined by the projection from the major eigevector λ_{42} of K_4 towards the direction defined by θ_4.
- The location μ_4 from K_4 is set according to $\gamma_{k_5^{y+}}$, θ_4, λ_{52} and λ_{42}.
- Let $\gamma_{k_2^{y+}}$ be the point in the plane defined by the point μ_2 projecting the distance from the major eigenvector λ_{22} towards the direction of θ_2.
- The center μ_3 of K_3 is set by projecting the major eigenvector λ_{22} of K_2, from the point $\gamma_{k_2^{y+}}$ until the distance dictated by the vector $[\lambda_{31}\cos\theta_2, \quad \lambda_{31}\sin\theta_2]$.
- Let $\gamma_{k_2^{x+}}$ be the point in the plane defined by the tangent to K_2 which is parallel to its major eigenvector λ_{22}.
- Sampling space for parameter λ_{41} —i.e. the width of component K_4— is constrained to avoid overlapping with the free-flow component K_2. This is carried out by constraining λ_{41} upper bound to the distance between the center of K_4 and the point $\gamma_{k_2^{x+}}$: distance$(\mu_4, \gamma_{k_2^{x+}})$.

7.2 Experiments

The proposed model is evaluated in the selected highway —M4/M7— and urban —Santander— contexts with both datasets consisting of traffic flows and occupancy aggregated at different temporal granularity.

Firstly, a station —defined as an aggregation of loop detectors in consecutive lanes— from the M4/M7 Western Motorway has been modelled. In Figure 7.7, the two subfigures show segmentation of the flow-occupancy plane in the different traffic states by proposed model. The upper subfigure shows $K = 5$ traffic states identified by the model, with the corresponding latent variable proportions: $\pi = [0.184, 0.758, 0.02, 0.03, 0.003]$ ordered as described in the previous subsection. That is, data density for the first traffic state corresponding to “almost no traffic” is 18.4% of the

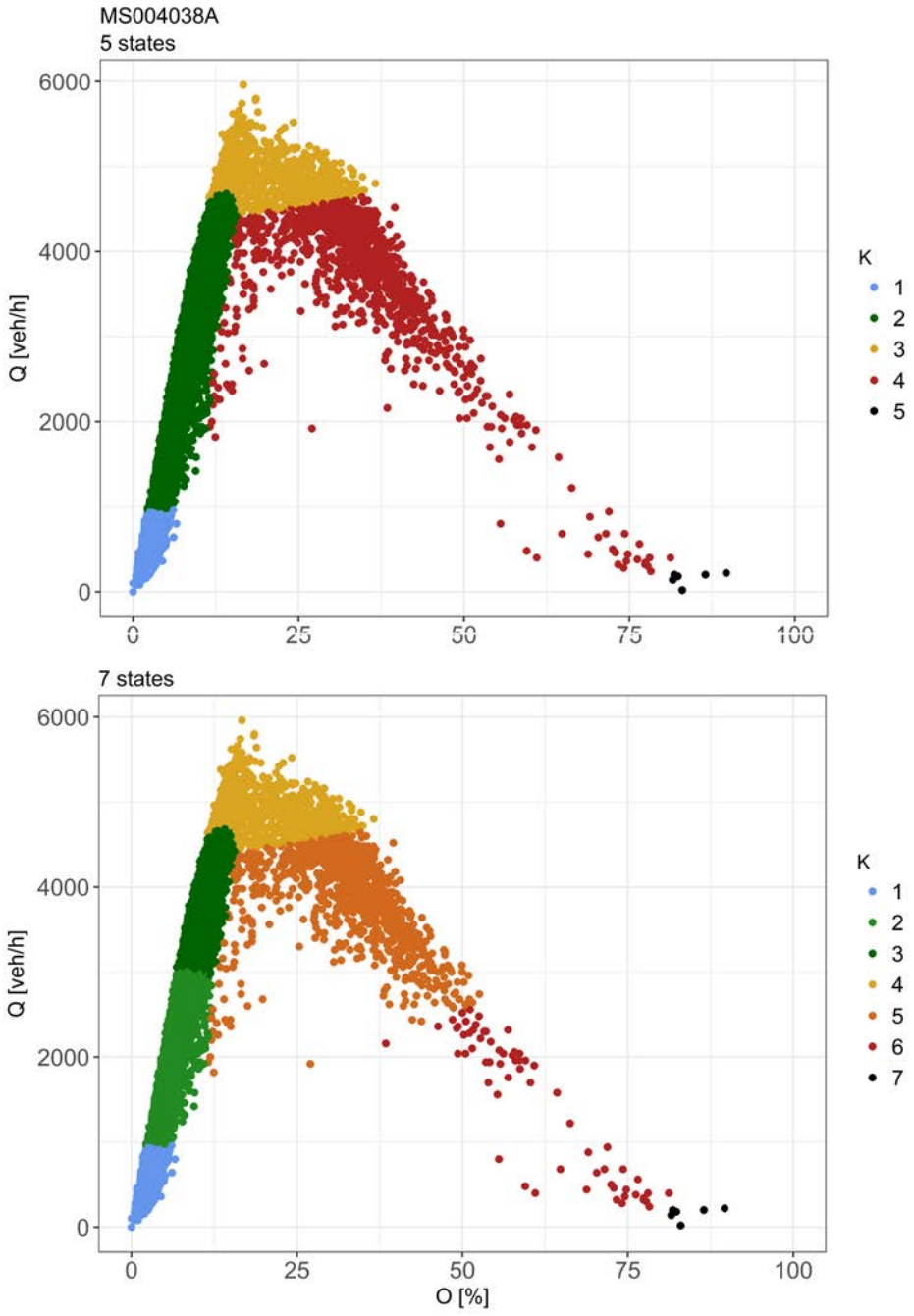


Figure 7.7: Traffic state identification results for a 3-lane station in the M4 Western Motorway (Sydney). Upper figure correspond to the original result from the model with $\pi = [0.184, 0.758, 0.02, 0.03, 0.003]$, whereas the figure in the bottom corresponds to a finer classification in 7 states using a post-classification algorithm.

overall data, 75.8% of the data reflects the second state for free-flow conditions, and less than 6% for the remaining traffic states. Furthermore, two additional substates were derived by splitting the free-flow state and the congested state into two using their nearest traffic states as points of attraction during the classification of each incoming observation. As a result, the lower subfigure, shows 7 traffic states identified by the model after applying a classification algorithm to identify traffic states even more precisely. This approach may be appropriate when a finer classification is required. In addition, independently of the number of traffic states, the proposed model is robust to identify observations associated with traffic hysteresis phenomena in the congested traffic state (by red and orange clusters respectively), which are just at the right hand side of the free-flow traffic state (presented in green cluster).

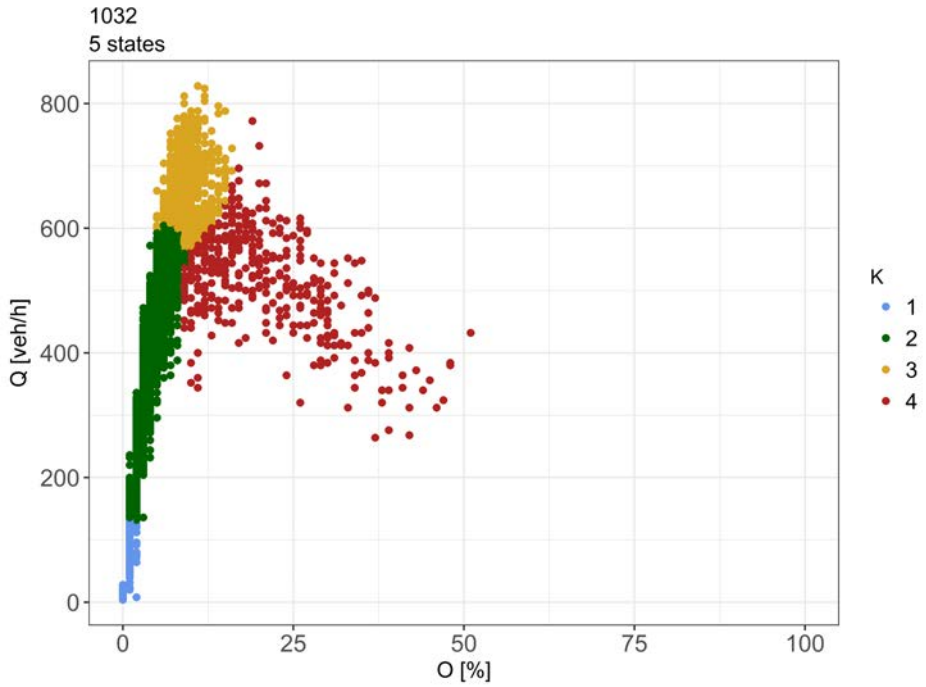


Figure 7.8: Traffic state identification results for a 2-lane station in an urban arterial in the city of Santander. Traffic states proportion is $\pi = [0.167, 0.565, 0.196, 0.072, 0]$.

Whereas in the case of the urban network dataset, Figure 7.8 shows the identified traffic states, where the state proportions are $\pi = [0.167, 0.565, 0.196, 0.072, 0]$, in an urban expressway in Santander city. In addition, Figure 7.9, demonstrates a model performance in a more complicated scenario corresponding to a station covering three lanes, located in an urban road of the city center. Resulting state proportions, $\pi = [0.284, 0.645, 0.071, 0, 0]$, demonstrate that this road section has never experienced congested traffic state. Figure 7.10 shows an interesting case where two different observed capacities were detected and thus leading to two different traffic states model estimations.

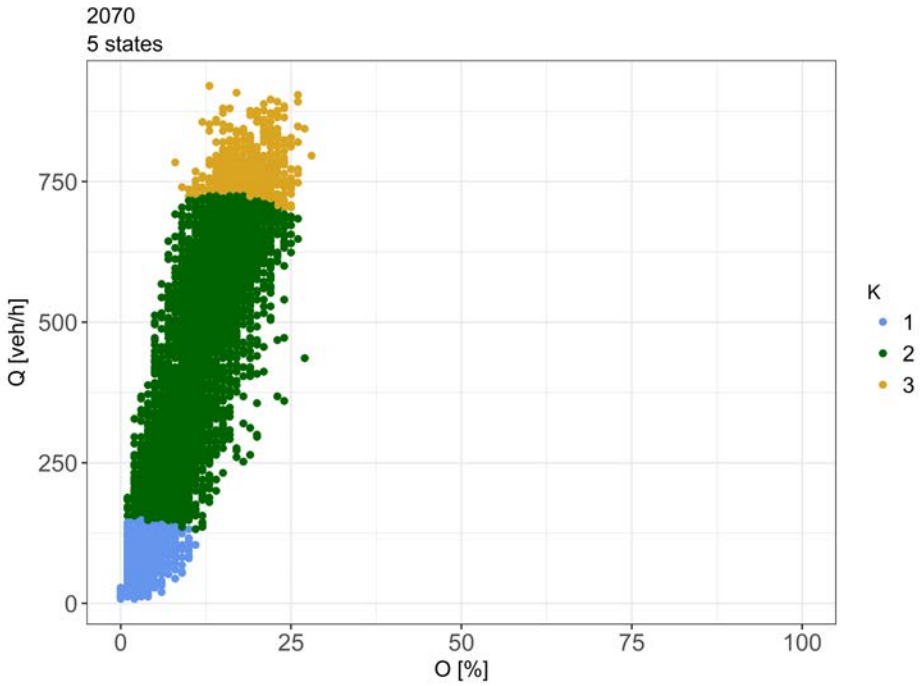


Figure 7.9: Traffic state identification results for a 3-lane station in an urban road in the city center of Santander. Traffic states proportion is $\pi = [0.284, 0.645, 0.071, 0, 0]$.

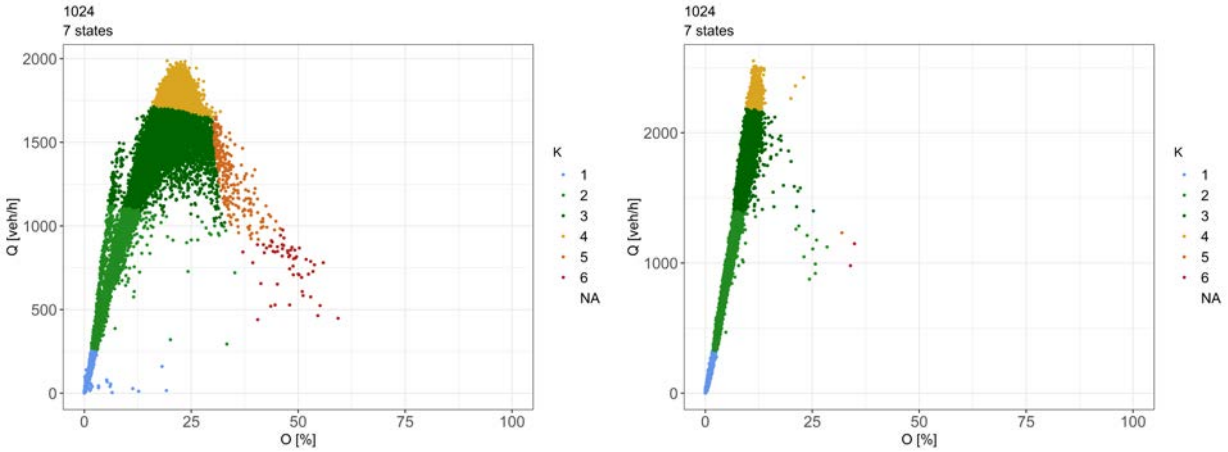


Figure 7.10: Traffic state identification results for a 2-lane station in an urban road in the city center of Santander, after two different observed capacities were detected in the dataset.

7.3 Discussion

The proposed probabilistic graphical model for the identification of different local traffic states is based on the bivariate relation from the fundamental diagram which correlates traffic flow and density. The model uses data collected from stationary loop detectors which are the most ubiquitous traffic data source, and thus we have used traffic flow and occupancy instead of density or speed to avoid making more assumptions than needed.

A majority of methods in the literature that rely on the fundamental relationship to determine traffic states are deterministic, neglecting the stochastic nature of such relation which is observed in the form of wide scattering, especially in the congested traffic state. This uncertainty is modeled here by using a more probabilistic approach. Furthermore, although the relation from the fundamental diagram applies for traffic in equilibrium conditions, we have been able to apply it to an urban scenario. The results are in line with the observations made in [163], where traffic is analyzed in spite of such a wide scattering that data seems like a cloud of points at a glance, yet there are still underlying patterns with data regions more dense than others. Nevertheless, it is recommended the application of a filtering function to smooth the high frequencies and get rid of many transient states, thus achieving a more robust estimation. This is especially important in urban data where data resolution is high and there are important flow disturbances such as traffic lights, pedestrian crossings, etc.

The proposed model [180] differs from others in the literature in that it uses a Bayesian perspective that allows to include the prior information from the domain expert. In cases where such expert knowledge is not available, we have proposed an algorithm to detect changes in the observed capacity related to structural changes from historical data. These structural changes may include permanent lane closures, new lanes, or permanent changes in the speed limit. This algorithm also serves to provide the prior capacity *pivot* in the plane. This prior information and imposing constraints solve the common problem of identifiability in this kind of generative models. It also solves the problem when observed data is not available in some regions of the fundamental diagram, as was one of the cases studied in this paper, where the model still correctly associates the identified traffic states. The resulting model is very efficient in storage terms and to be applied in real-time operations. Online updates can also be performed very efficiently.

8 Spatiotemporal probabilistic model for learning the traffic state dynamics

This chapter relies on the output generated by the traffic state model described in the previous Chapter 7. More specifically, a Bayesian network is placed upon the traffic states in order to exploit the spatiotemporal propagation of these in a probabilistic manner. The main motivation behind this model is to exploit the principle of locality in traffic, where a specific location is directly influenced only by its immediate surroundings. This is in contrast with the modelled effects in Chapter 5 where the problem was treated in a high-dimensional setting. Obviously, the motivation was different as the goal was to perform prediction at multiple time steps ahead —usually from 15 to 60 minutes—. In this case, the study is centered on modelling the local transition of traffic states considering both the spatial and temporal influence, with the aim of performing inference for the next time step. This is modelled using graph notation as shown in Figure 8.1 for a series of nodes $v_1, \dots, v_n$ —i.e. the complete network— or, more compactly, in Figure 8.2 showing the influential variables on a given node v_i. These variables are determined by the corresponding rule pattern at that moment, the upstream and downstream nodes of v_i at the previous time interval as well as the node v_i itself at the previous time interval. This notation serves as the basis to build a probabilistic graphical model such as a dynamic Bayesian network that relates the spatiotemporal interaction of the traffic states in the network by means of the joint probabilities and taking benefit of the existing conditional independence by the principle of locality.

Similarities can be found between the proposed modelling perspective and the work in [123]. They proposed a dynamic Bayesian network approach using traffic conditions of past time instances in the chosen link and its neighbours to estimate traffic conditions as a binary state variable (congested or not congested) along a small network of arterial roads using probe vehicle data. However, the approach is limited to classifying the traffic conditions into broad categories such as congested and not-congested. The proposed method here is generalized to different network types, and it is built on top of the traffic states information identified by the model from Chapter 7. Additionally, the method here also benefits from the rule pattern provided by Adarules in Chapter 5 in order to differentiate different network dynamics. Certain similarity can also be found with [261], although

their proposal is not aimed to incident detection nor learning the dynamics of the traffic states. Actually, they propose a framework which relies on third-party incident reports —i.e. labelled data— to classify as recurrent or non-recurrent those reported incidents. This classification is done by means of a Bayesian network to assess when certain congestion propagation pattern is anomalous enough using the joint probability. This joint probability is based on the congested or non-congested states at every site along the road. Our approach is built on top of a more rich and meaningful set of traffic states as described in the previous Chapter 7 and, furthermore, it is aimed for automatic incident detection —i.e. non-recurrent congestion— through the learning of the spatiotemporal traffic dynamics using the traffic states.

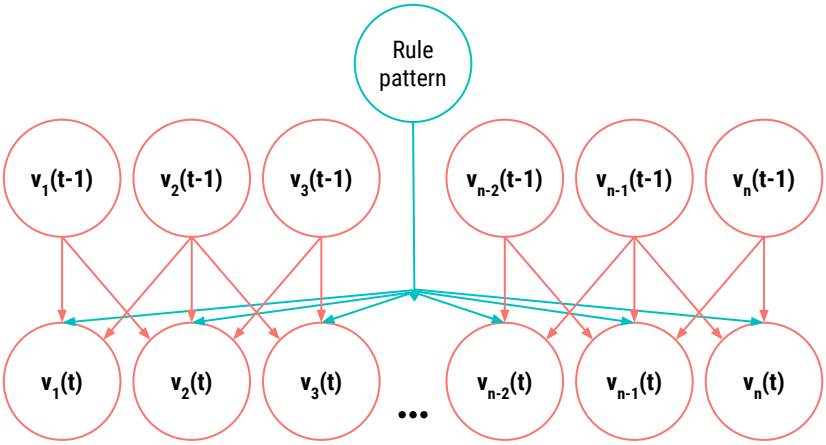


Figure 8.1: Spatiotemporal Bayesian network involving the traffic states to exploit the principle of locality in traffic: unfolded version for the complete network.

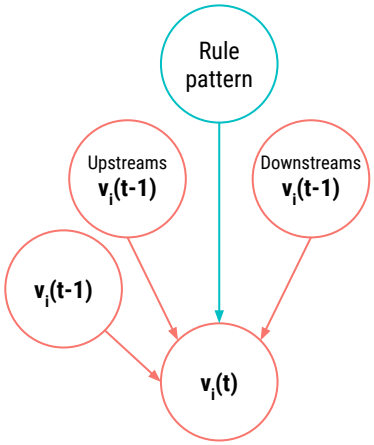


Figure 8.2: Spatiotemporal Bayesian network involving the traffic states to exploit the principle of locality in traffic: folded version for a given node v_i.

The reason of including both upstream and downstream traffic for every site is to consider the incoming traffic flow but also to capture and take into account the derived effects from abrupt or unexpected flow disruptions such as traffic incidents. These effects may be traffic shockwaves or queue spillovers [104].

There exist multiple potential applications for such general probabilistic spatiotemporal model. These applications are based simply on the different type of queries that can be answered by the spatiotemporal probabilistic model presented in the Figures 8.1 and 8.2. Namely:

General anomaly detection based on the current traffic states and optionally given the corresponding rule pattern. It implies performing a probabilistic query of how anomalous is the observation of a given traffic state in a certain local area. This is carried out using the conditional probability distribution of the selected local node according to the state of its surroundings and optionally conditioning on a given rule pattern selected by Adarules. The algorithmic procedure for this task is shown in Figure 8.3.

Filling in of missing data. This operation is equivalent to estimating the most probable network traffic state using the available observed data so far. This means calculating the most probable state of every node conditioning on those influential nodes with observed information as well as the current rule pattern selected by Adarules. The algorithmic procedure for this task is shown in Figure 8.4.

Incident detection. Because a traffic incident can be seen as a special case of anomaly, the spatiotemporal probabilistic model can be used to detect anomalous, or with a low probability, congestion spots also called *non-recurrent congestion* in order to raise an alarm and to evaluate the spatiotemporal propagation and its severity. The algorithmic procedure for this task is shown in Figures 8.5 and 8.6.

8.1 Quantifying the outlierness

The first step taken before the evaluation of the incident detection method is to perform a sanity check showing how the proposed spatiotemporal probabilistic model evaluates the outlierness. To this end, it has been taken a snapshot of Adarules at a specific time such as at the beginning of the second year for each dataset —M4/M7 and Santander networks—. This means that Adarules has observed and learned using data from the first year in each dataset, then being *frozen* so that it is not going to be updated with new data but only to evaluate the outlierness using the knowledge gathered until that moment. Then, starting from the second year in each network, data has been altered in order to include a *fake* drift where AM and PM time period have been swapped out, i.e. usual traffic during the day is moved to the night and vice versa. This artificial drift is maintained for two months —i.e. until the end of the data used for this case study—.

Two algorithms for anomaly detection have been developed in this thesis and applied to this case

Algorithm 17: Measure outlierness using the probabilistic traffic states approach

Input : $\mathcal{T}_{\mathcal{S}}$: Traffic states for every node v
R_{ID}: Name or identifier for the corresponding graph patterns / rules
Output: $\mathcal{O}(v)$: Outlierness degree in every node v of the graph

begin
  foreach *local node* v *in* V **do**
    $v_u \leftarrow$ upstreams(v)
    $v_d \leftarrow$ downstreams(v)
    $v_u^L \leftarrow$ lags(upstreams(v))
    $v_d^L \leftarrow$ lags(downstreams(v))
    $v^L \leftarrow$ lags(v)
    // Which is the probability of observing the current traffic state in v given the evidence E?
    // Evidence E is composed of the current graph pattern (rule) R and the past observation in its spatial surroundings v^L, v_u^L, v_d^L
    $E \leftarrow \mathcal{T}_{\mathcal{S}}(v_u^L) \cup \mathcal{T}_{\mathcal{S}}(v_d^L) \cup \mathcal{T}_{\mathcal{S}}(v^L) \cup R_{ID}$
    // Probability of observing the current state $\tilde{v}$ given the evidence in E
    $\mathcal{O}_v(v) \leftarrow \mathrm{P}(\mathcal{T}_{\mathcal{S}}(v) = \tilde{v} \,|\, E)$
  end
  return $\mathcal{O}(v)$
end

Figure 8.3: Measure outlierness using the probabilistic traffic states approach.

study. The first algorithm, described in Chapter 5, measures the outlierness for a given node v in the context of a rule R (Figure 5.19) relying on the identified pattern for the flow propagation in the network. For this reason, we call it 'anomaly detection using graph features' or 'graph outlier' as it makes use of the different distributions of flow propagation between nodes in the network graph. The second algorithm is described in this chapter, Figure 8.3, and it relies on detecting anomalous —or with low probability— spatiotemporal transitions of the underlying identified traffic state among nodes v in the network using a given rule R as support to further discriminate certain transitions. In essence, both methods are to detect anomalies but the type of anomaly, as well as the message they tell, is different.

As shown in Figure 8.7 for both datasets, the timeline depicts the aggregated network outlierness based on the graph pattern using the mean over all the individual anomaly scores from every node v in the graph. As described in Chapter 5, this anomaly score relies on the estimation of Z-scores and, thus, any value outside the range $[-3, 3]$ is suspicious to be anomalous. The first thing to note is how the outlierness score during the first year before the drift is wider in the Santander dataset. This is to be expected, as an urban network has more variability in the propagation of the flows through its links compared to a motorway network. The second fact to note is how this network-aggregated anomaly score starts to oscillate strongly in the second year reaching levels

Algorithm 18: Filling in of missing data using the probabilistic traffic states approach

Input : X: Input data matrix $[N \times p]$
$\mathcal{T}_{\mathcal{S}}$: Traffic states for every node v
R_{ID}: Name or identifier for the corresponding graph patterns / rules
Output: X_{Clean}: X data with filled missing data

begin
  foreach *local node* v *in* V **do**
    $v_u \leftarrow \text{upstreams}(v)$
    $v_u^L \leftarrow \text{lags}(\text{upstreams}(v))$
    $v_d \leftarrow \text{downstreams}(v)$
    $v_d^L \leftarrow \text{lags}(\text{downstreams}(v))$
    $v^L \leftarrow \text{lags}(v)$
    // Which is the most probable traffic state for v given the evidence E?
    // Evidence E is composed of the observed values coming from spatial neighbouring (upstream and downstreams) around v coupled with their temporal information (temporal lags), and optionally the current graph pattern (rule R)
    $E \leftarrow (\mathcal{T}_{\mathcal{S}}(v_u) \cup \mathcal{T}_{\mathcal{S}}(v_u^L)) \cup (\mathcal{T}_{\mathcal{S}}(v_d) \cup \mathcal{T}_{\mathcal{S}}(v_d^L)) \cup \mathcal{T}_{\mathcal{S}}(v^L) \cup R_{ID}$
    // Most probable joint state within the joint distribution
    $\mathcal{T}_{\mathcal{S}}(v) \leftarrow \max \mathrm{P}(\mathcal{T}_{\mathcal{S}}(v) \mid E)$
    Derive associated features (flow, occupancy, speed) associated with the traffic state $\mathcal{T}_{\mathcal{S}}(v)$ and fill in X
  end
  return X_{Clean}
end

Figure 8.4: Filling in of missing data using the probabilistic traffic states approach.

Algorithm 19: Incident detection using the probabilistic traffic states approach

Input : $\mathcal{T}_{\mathcal{S}}$: Traffic states for every node v
R_{ID}: Name or identifier for the corresponding graph patterns / rules
Output: $ID(v)$: Severity of incidents in every spot of the network graph

begin
  // The first stage consists of assigning a raw score based on the identification of anomalous congestion spots (congestions which are non-recurrent in terms of probability) through the network graph
  foreach *local node* v *in* V **do**
    $v_u \leftarrow$ upstreams(v)
    $v_d \leftarrow$ downstreams(v)
    $v^L \leftarrow$ lags(v)
    // Find suspicious observations $n \in N$ where v spot is congested and any of its neighbour spots v_x are not
    $\mathcal{S}_{idx} \leftarrow \mathcal{T}_{\mathcal{S}}(v)$ is congested
    $\wedge\, \exists\, v_x : \big(\mathcal{T}_{\mathcal{S}}(v_{u_1}) \vee ... \vee \mathcal{T}_{\mathcal{S}}(v_{u_n}) \vee \mathcal{T}_{\mathcal{S}}(v_{d_1}) \vee ... \vee \mathcal{T}_{\mathcal{S}}(v_{d_n})$ is not congested$\big)$
    foreach *observation* n *in* $\mathcal{S}_{idx}$ **do**
      Get the set of non-congested neighbour spots $v_{NC} \subseteq \{v_u, v_d\}$ of v
      $v^L_{NC} \leftarrow$ lags(v_{NC})
      // Which is the joint probability for the current traffic state and its lags in v, v^L given the evidence E?
      // Evidence E is composed of those traffic states from non-congested neighbours v_{NC}, v^L_{NC} and the current graph pattern (rule) R
      $E \leftarrow (\,\mathcal{T}_{\mathcal{S}}(v_{NC}) \cup \mathcal{T}_{\mathcal{S}}(v^L_{NC})\,) \;\cup\; R_{ID}$
      // Joint probability distribution J_v of $v \cap v^L$ given the evidence in E
      $\mathcal{J}_v \leftarrow \mathrm{P}(\mathcal{T}_{\mathcal{S}}(v) \cap \mathcal{T}_{\mathcal{S}}(v^L) \,|\, E)$
      // Calculate outlierness raw score $Oraw_{v,n}$ using the joint probability distribution $\mathcal{J}_v$
      $\hat{O}raw_{v,n} \leftarrow \frac{(\max \mathcal{J}_v) - (\mathcal{J}_v(\mathcal{T}_{\mathcal{S}}(v) = \tilde{v} \cap \mathcal{T}_{\mathcal{S}}(v^L) = \tilde{v}^L))}{\max \mathcal{J}_v}$
    end
  end

Figure 8.5: Incident detection using the probabilistic traffic states approach. Part I.

(15)
    // The second stage consists of assigning a final score for those *enough* raw anomalous spots
(16) $\hat{O}_\tau raw \leftarrow 0.50$
(17) **foreach** *observation* n *in* $\mathcal{S}_{idx}(v)$ **do**
(18) For those *enough* anomalous spots $\hat{O}raw > \hat{O}_\tau raw$
    // The final score is composed of three different contributions: ($\hat{O}_1 \in [0,1]$) a relative severity of the existing congestion (50%), ($\hat{O}_2 \in [0,1]$) the temporal recurrence of the existing non-recurrent congestion (25%), and ($\hat{O}_3 \in [0,1]$) the spatial propagation of the incident through the network graph (25%)
(19) $\hat{O}_1 \leftarrow \mathcal{T}_{\mathcal{S}}(v) - H(\mathcal{T}_{\mathcal{S}}(v_{NC}))$
    // Here $H(x)$ is the harmonic mean of vector x
(20) $t_{max} \leftarrow 30$ minutes
(21) $\hat{O}_2 \leftarrow \min(\hat{O}_2 + \Delta t / t_{max}, 1)$
(22) Find recursively congested neighbours of v in order to assess the spatial propagation of the incident
(23) $\hat{O}_3 \leftarrow$ number of nodes linked to v in a congested state through the graph up to a given maximum
(24) $ID(v) \leftarrow 0.50 \cdot \hat{O}_1 + 0.25 \cdot \hat{O}_2 + 0.25 \cdot \hat{O}_3$
(25) **end**
(26) **return** $ID(v)$
(27) **end**

Figure 8.6: Incident detection using the probabilistic traffic states approach. Part II.

that indicate a severe anomaly. This occurs for both datasets, and the level of anomaly never decreases along the two months because Adarules state is frozen and thus there is no adaptation involved. This is intended to show the anomaly detection with a new situation —swapping out the traffic in the AM and PM periods— using the rules and knowledge gathered from the first year in each dataset. In a real scenario where Adarules would adapt itself automatically after detecting these changes, the level of anomaly would decrease over time. Nevertheless, the key fact is that this method of anomaly detection based on the graph rule relies heavily on the historical traffic conditions and the distribution of flow propagation which has been seen in the scope of every rule pattern independently from the rest of the current network traffic state even if it is consistent. It is also interesting to check how Adarules would progressively adapt to this new data distribution considering it less anomalous over time. This happens in a real scenario where Adarules would receive this data and being adapted to it, as shown in Figure 8.9.

On the other hand, the outlierness scores shown in Figure 8.8 based on the use of the traffic states within the probabilistic spatiotemporal model show a different story. Again, the timeline depicts the aggregated network outlierness using the mean over all the local anomaly scores from every

node v in the graph. In this case, there is no significant difference in the level of anomaly —whose maximum allowable range is in $[0, 1]$ as it is a probabilistic measure— between the real data as part of the first year and the artificial data with the swap AM-PM. This is because even that the method —described in Figure 8.3— makes use of the selected rule pattern, it also makes use of the current network conditions to assess how anomalous is the traffic seen at spot v. Therefore, if the network traffic flow is consistent even if it is not the usual one within the current rule R, it is not seen as an anomaly for this spatiotemporal network model as it is aimed to detect inconsistencies in the traffic flow.

These two outlier detection methods complement each other, as they provide different but valuable output information in the target application for traffic engineers and managers, as more information supports and ease the decision making.

Because a traffic incident can be seen as a special case of anomaly or inconsistency in the traffic flow due to the disruption effect, the spatiotemporal probabilistic model can be used to detect anomalous —i.e. with an expected low probability— congestion spots also called *non-recurrent congestion* in order to raise an alarm and to evaluate the spatiotemporal propagation and its severity. This incident detection will be evaluated in the next subsection.

8.2 Incident detection

As aforementioned, a traffic incident can be seen as a special case of anomaly. This anomaly or inconsistency is presented as a disruption of the traffic flow at a specific point in the network. Therefore, the spatiotemporal probabilistic model can be used to detect such anomalous spot —i.e. with an expected low probability—. However, a traffic incident is characterized by being anomalous —i.e. non-recurrent— but it also must fulfill other requirements as the anomalous spot must be in a congested state —non-recurrent congestion—. Once detected such anomalous spot with the form of an incident, an alarm is raised and the traffic states are kept under observation in order to measure the incident severity through the assessment of the temporal recurrence and the spatial propagation of the incident.

The algorithmic details of the procedure for incident detection using the probabilistic spatiotemporal model are shown in Figures 8.5 and 8.6. More specifically, the first step is to detect those spots in the network which are in a congested state and then evaluate if this congestion is anomalous —non-recurrent— given its surroundings and the current graph pattern. If it is enough anomalous, the incident is scored according to three factors:

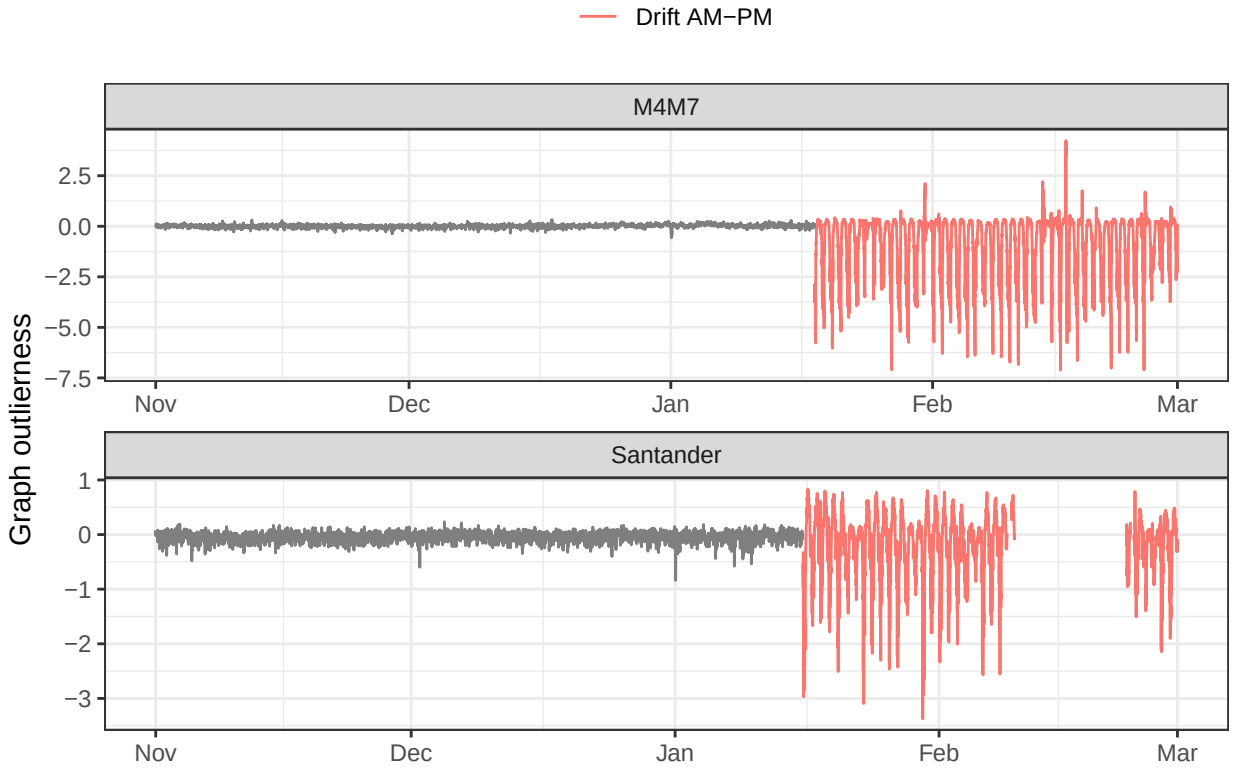


Figure 8.7: Timeline of the outlierness based on the graph anomaly detection —in this case shown as the mean of the anomaly score from all the nodes in the graph at a given time (HH:MM)— for both datasets using Adarules in a *frozen* state after performing the learning of the first year, then facing an artificial drift in the traffic from AM and PM periods.

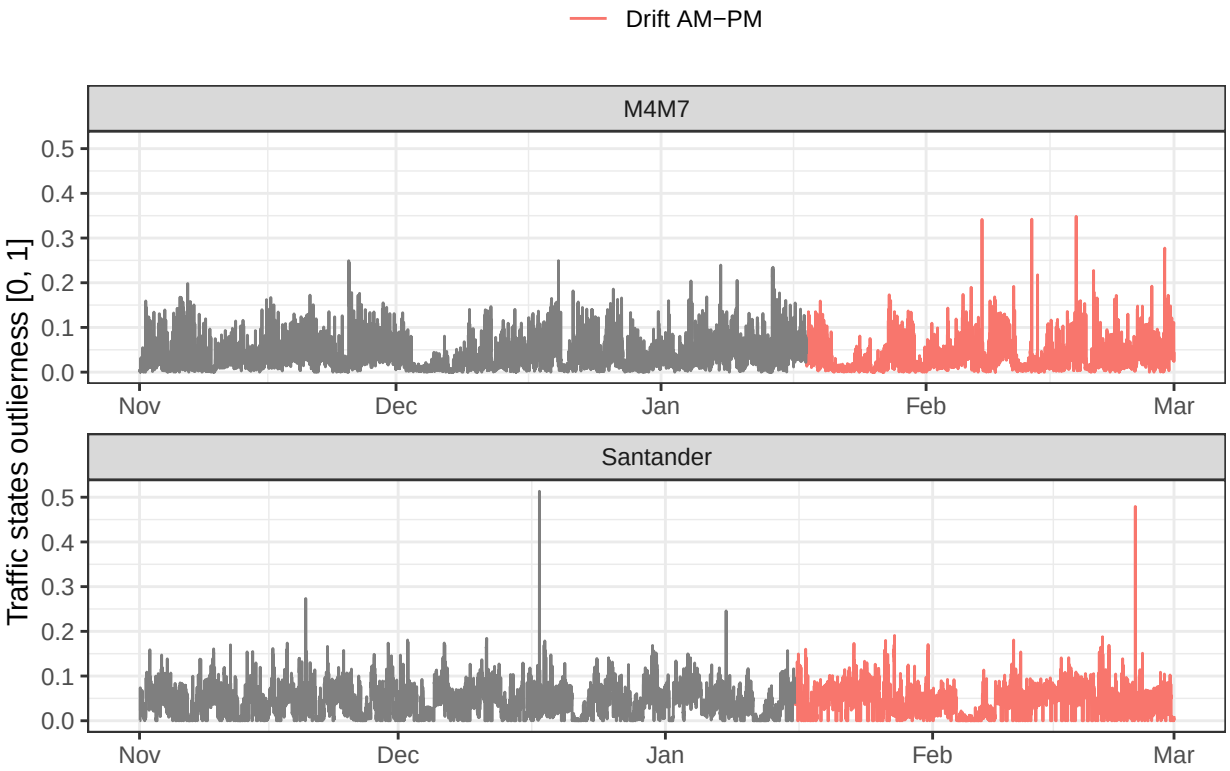


Figure 8.8: Timeline of the outlierness based on traffic states —in this case shown as the mean of the anomaly score from all the nodes in the graph at a given time (HH:MM)— for both datasets using Adarules in a *frozen* state after performing the learning of the first year, then facing an artificial drift in the traffic from AM and PM periods.

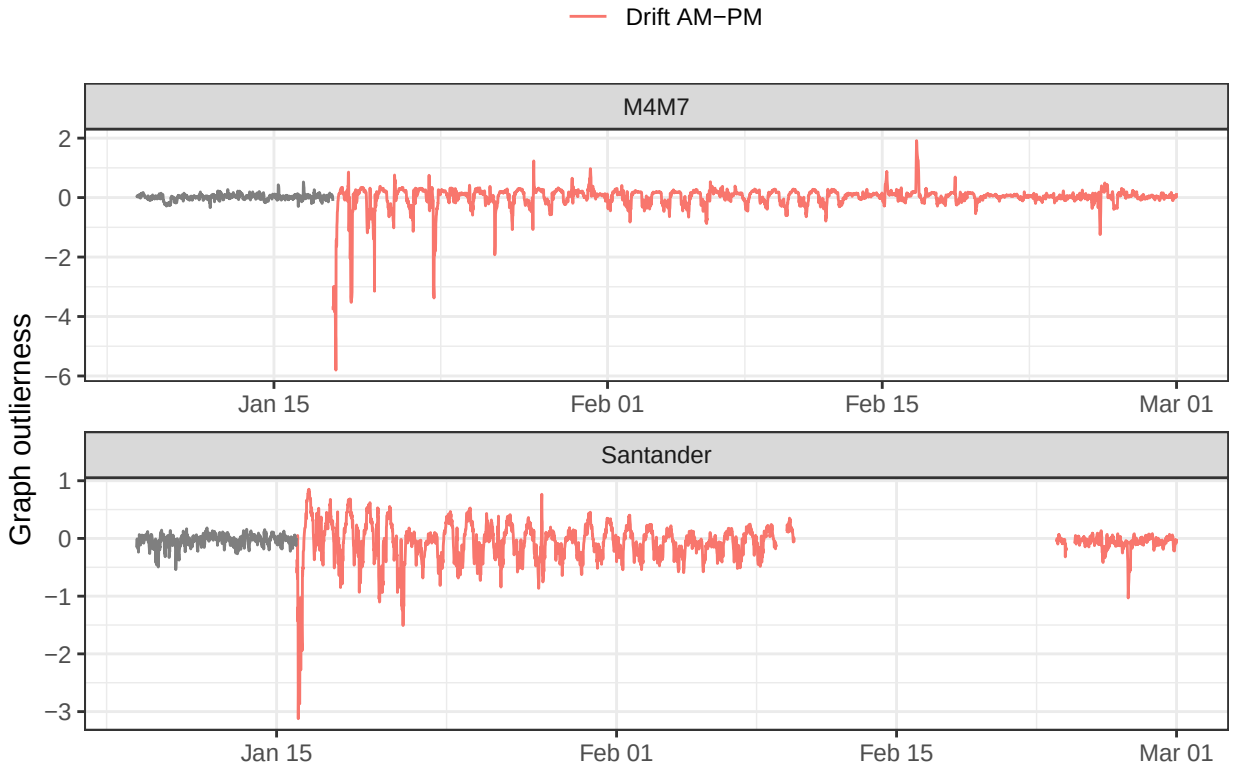


Figure 8.9: Timeline of the outlierness based on the graph anomaly detection —in this case shown as the mean of the anomaly score from all the nodes in the graph at a given time (HH:MM)— for both datasets. In this case, Adarules is receiving and processing streams of this new data with AM-PM drift, and that is the reason why they are progressively considered less anomalous over time.

1. **Severity of the congestion**: This means the level of the measured traffic congestion according to its traffic state at the anomalous spot, as well as the difference with respect to the traffic conditions in its adjacent nodes. This factor represents a 50% of the final score.
2. **Temporal persistence**: It corresponds to the duration in time of the incident and represents a 25% of the final score.
3. **Spatial propagation**: The traffic states of those nodes connected to the incident are recursively analyzed to assess how far the incident has propagated. This represents a 25% of the final score.

In this way, with the joint use of the spatiotemporal information, it is taken into account the propagation effects in space and time of the incident in order to score its severity. It also mitigates the noise effects by taking into account the persistence effect of the traffic incident, potentially reducing the ratio of false positives. The combination of these factors gives a score for the incident severity in the range $[0, 1]$. This score has been scaled to $[0, 5]$ in the different results' charts simply for a matter of ease of viewing.

The evaluation of a traffic incident detection algorithm is always challenging because of the anomalous nature of such events. It is usually difficult to have data with labelled incidents, i.e. knowing exactly when the incidents have occurred, as well as their length, etc. Even more, when this data is labelled —e.g. using reports from local authorities—, the labelling information data is usually noisy and unreliable, as they are usually shifted in time or space, or simply not all incidents are reported. Therefore, this makes difficult to develop and train an incident detection model and that is the reason why an unsupervised and probabilistic approach has been taken to this end in this thesis. This way, the proposed model does not require incident labelled data which makes it more robust and easier to transfer to new areas with unlabelled incident data or even with no incidents observed yet. The potential interpretability of the model by an end-user —such as a traffic engineer or manager— as for example the reasons behind every decision is another strong point.

In addition to the training of the model, the lack of labelled and reliable incident data makes difficult to test the model and even more to evaluate it in a rigorous way as typically happens with classification algorithms relying on the measurement of the ratio of true positives, false positives —e.g. precision and recall are two common model evaluation metrics in classification—.

For this reason, the incident detection algorithm has been evaluated on a set of cases in two road traffic networks. One is the previously described M4/M7 motorways network. To this end, it has been selected a specific time period from January, 2016 – April, 2016, and also a specific network area —the M4 Western Motorway heading east direction. The reason for such focus on a time period and a spatial area is because of the manual work on finding relevant incidents on data while

assuring that they represent a true incident. This work has been carried out in cooperation with Aimsun traffic engineers in order to leverage their expert knowledge and perform the verification of the incidents. The second network corresponds to the city of Bristol (England) in order to also perform the evaluation in an urban network. In this latter case, the data is synthetic as it comes from the output of several simulations using the Aimsun software. The reason for this is the difficulty to find true incidents in the noisy urban data of Santander, as well as to be sure of the time at which such incidents have occurred.

For the aim of showing the incident detection in the following figures, the traffic state —as presented in Chapter 7— is used instead of, for instance, using both the traffic flow and occupancy or the speed because the traffic state already shows a compacted vision of the combination of these former. Moreover, the traffic state is an easy-to-understand latent variable that can be interpreted as an analog to the level-of-service:

1. (Almost) no traffic,
2. Free-flow conditions with low flow,
3. Free-flow conditions with high flow,
4. Maximum capacity,
5. Light traffic congestion,
6. Heavy traffic congestion,
7. Total congestion.

In every figure, there is an orange horizontal line that marks the limit between congested traffic (traffic states 5, 6 and 7) when exceeding the maximum link's capacity and non-congested traffic (traffic states 1, 2, 3 and 4). In addition, there is a shaded area in red for those time periods with a detected incident score for every detector. This incident score is the result of the calculation of three factors —severity of the non-recurrent congestion, temporal persistence, and spatial propagation— as described in the previous section. The incident score ranges originally from 0 to 1 but it is scaled to $[0, 5]$ in order to make easier its visualization within the figures. All this visualization related to the traffic states and the incident scoring is summarized in the legend shown in Figure 8.10.

8.2.1 M4 Western Motorway

The dataset corresponds to the period January, 2016 – April, 2016, and the study area is focused along the M4 Western Motorway heading the east direction. The loop detection identifiers are named MS0040xxA where 'xx' correspond to two digits, so the flow direction moves as the digits increases —i.e. MS004001A → MS004099A—.

The first incident in Figure 8.11 displays an incident occurring between detection sites MS004028A and MS004029A around 15:15. Actually, it can be observed how the incident originates between MS004029A and MS004030A, as can be seen by the congested state (5) at 15:15 in MS004029A while MS004030A remains in the free-flow state (3). The spatiotemporal model identifies as anomalous this transition of traffic states in the time-space. After the incident arises, it is rapidly spread backward to the position of MS004028A. It is in this position where the flow disruption persists longer in time as shown by the congested state as well as the incident score. There are other shorter incident alarms spread in time-space around the main incident site (MS004028A) because of the effect of the traffic flow disruption as the incident propagates. It can be also observed that the severity of the incident is not that high if measured solely by the severity of the congestion (level 5 in MS004028A) compared to the escaping traffic flow level downstream (level 3 in MS004029A and MS004030A). However, the fact that the incident persists in time, and that it is propagating upstream (MS004025A and beyond) is the reason why the incident detection model assigns the incident a higher alarm score.

The incident shown in Figure 8.12 occurs around 17:30 and it has a similar effect to the previous incident. In this case, the incident occurs between the detection site MS004026A and the MS004027A, while the congestion propagates upstream towards the MS004025A. The incident detection model assigns a moderate severity to the incident (about 2 out of 5). Another thing to note is that after 18:15 there is missing data in these detectors so then the level of alarm is gradually decreased.

Figure 8.13 shows a small incident occurring at 12:30 between MS004030A and MS004031A. The incident vanishes rapidly without any effect of congestion propagation.

The incident observed in Figure 8.14 takes place first between detection sites MS004039A and MS004040A at 15:00, but these sites become uncongested and the incident is then observed at the MS004038A site. the congestion severity is not that high, but it lasts around 30 minutes and it is propagated upstream (MS0040436A).

In Figure 8.15, it can be observed many small incidents occurring in the M4 Western Motorway nearby MS004034A and MS004041A at around 17:15. This seems more like a situation where a *phantom* jam occurs because of the small duration of such incidents and how these incident alarms propagate in the upstream direction. This hypothesis is also supported by the fact that traffic sites were in a maximum capacity state (4) before the incidents occur. Therefore, any small abrupt disruption caused by individual drivers —e.g. abruptly braking— could cause such instabilities in the traffic flow.

Figure 8.16 shows an important incident occurring at 19:15 in the M4 Western Motorway between

the detection sites MS004030A and MS004031A. It can be observed how the traffic flow was in a free-flow state with low traffic volume (state 2) and then suddenly it turns out into a heavy traffic congestion (state 6) for about 45 minutes and propagating the congestion upstream (MS004026A and beyond), and thus the incident detection model assigns a high incident score. The incident alarm then moves to an upstream detector (MS004029A) as the incident is vanishing.

Figure 8.10: Color legend for traffic states [1 - 7] and the incident score range [0, 5].

Figure 8.11: An incident occurring in the M4 motorway between MS004028A and MS004029A at 15:15 (6th January 2016). Traffic states [1 - 7] and incident score [0, 5] are shown.

8.2.2 Bristol urban network

For the second set of experiments, a network from the city of Bristol has been used in order to perform several simulations with traffic incidents. The simulations have been performed using the traffic simulation software Aimsun with the support from expert traffic engineers. The geometry

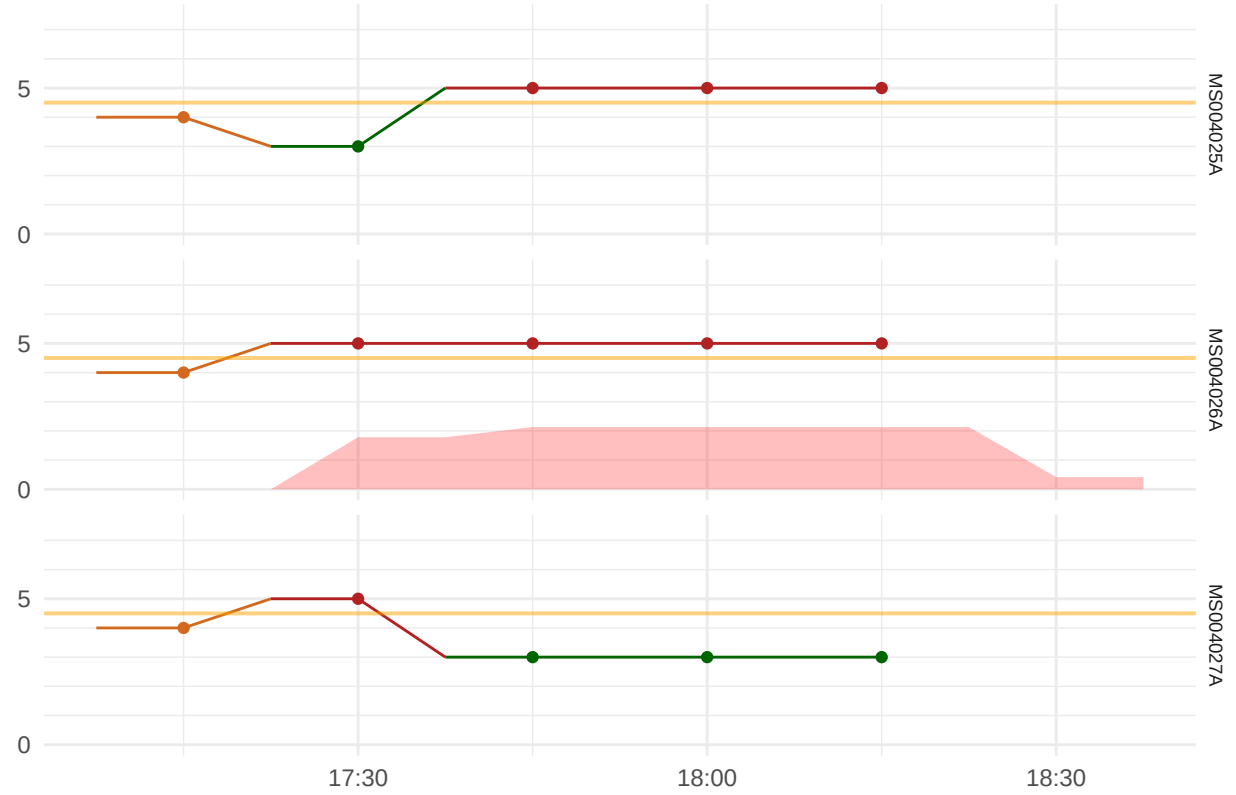


Figure 8.12: An incident occurring in the M4 motorway between MS004026A and MS004027A at 17:30 (13th January 2016). Traffic states [1 - 7] and incident score [0, 5] are shown.

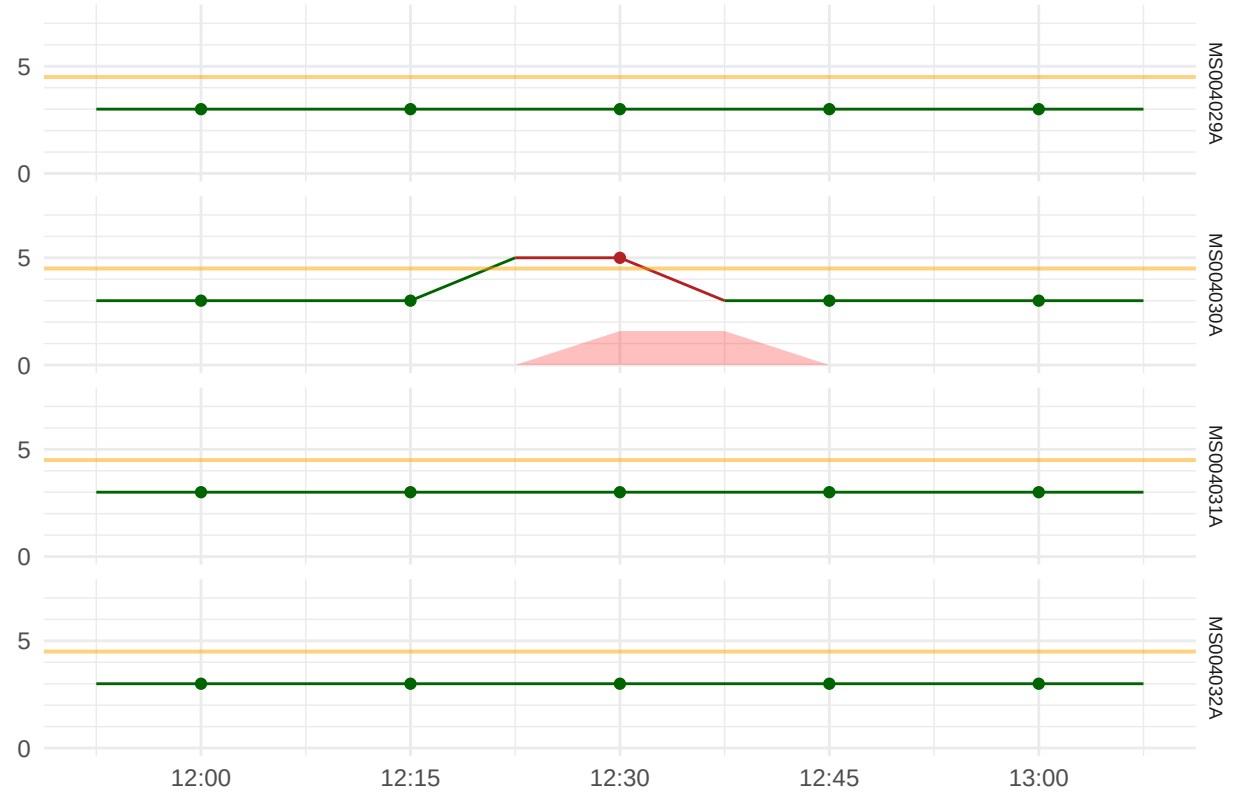


Figure 8.13: An incident occurring in the M4 motorway between MS004030A and MS004031A at 12:30 (12th March 2016). Traffic states [1 - 7] and incident score [0, 5] are shown.

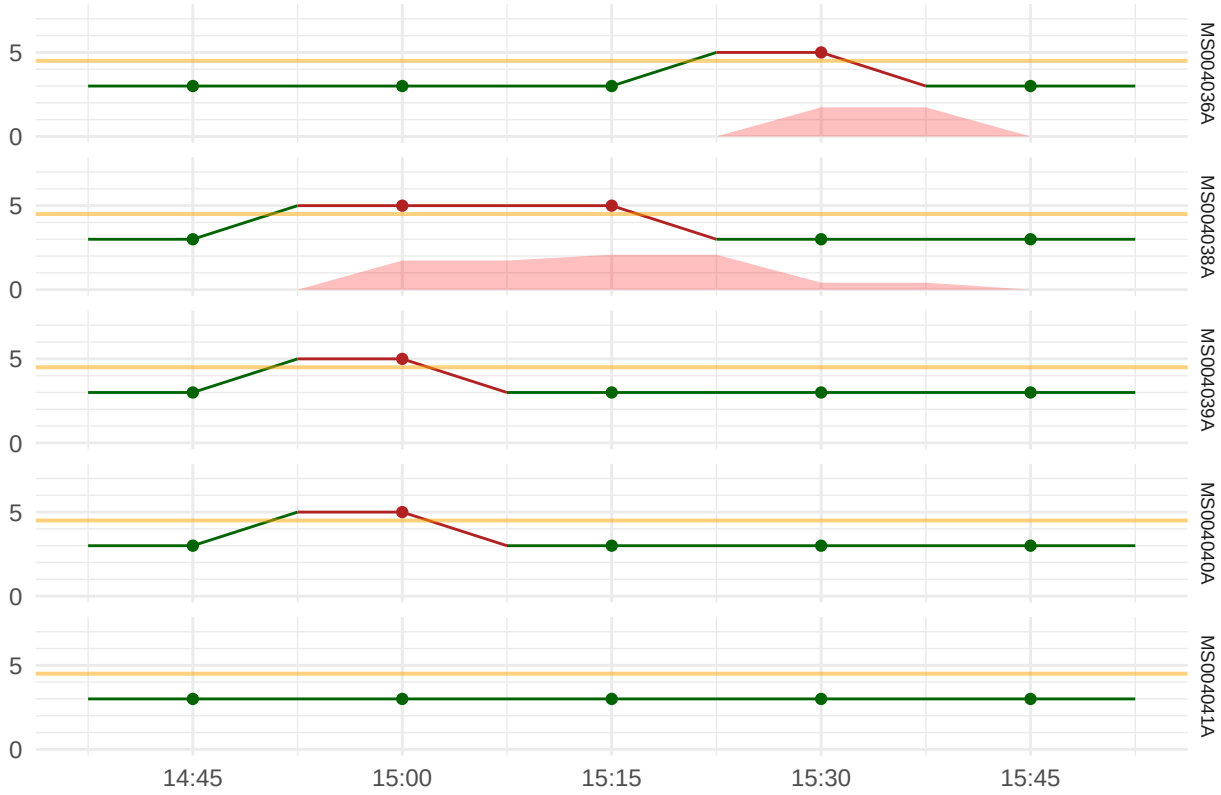


Figure 8.14: An incident occurring in the M4 motorway between MS004040A and MS004041A at 15:00 (29th March 2016). Traffic states [1 - 7] and incident score [0, 5] are shown.

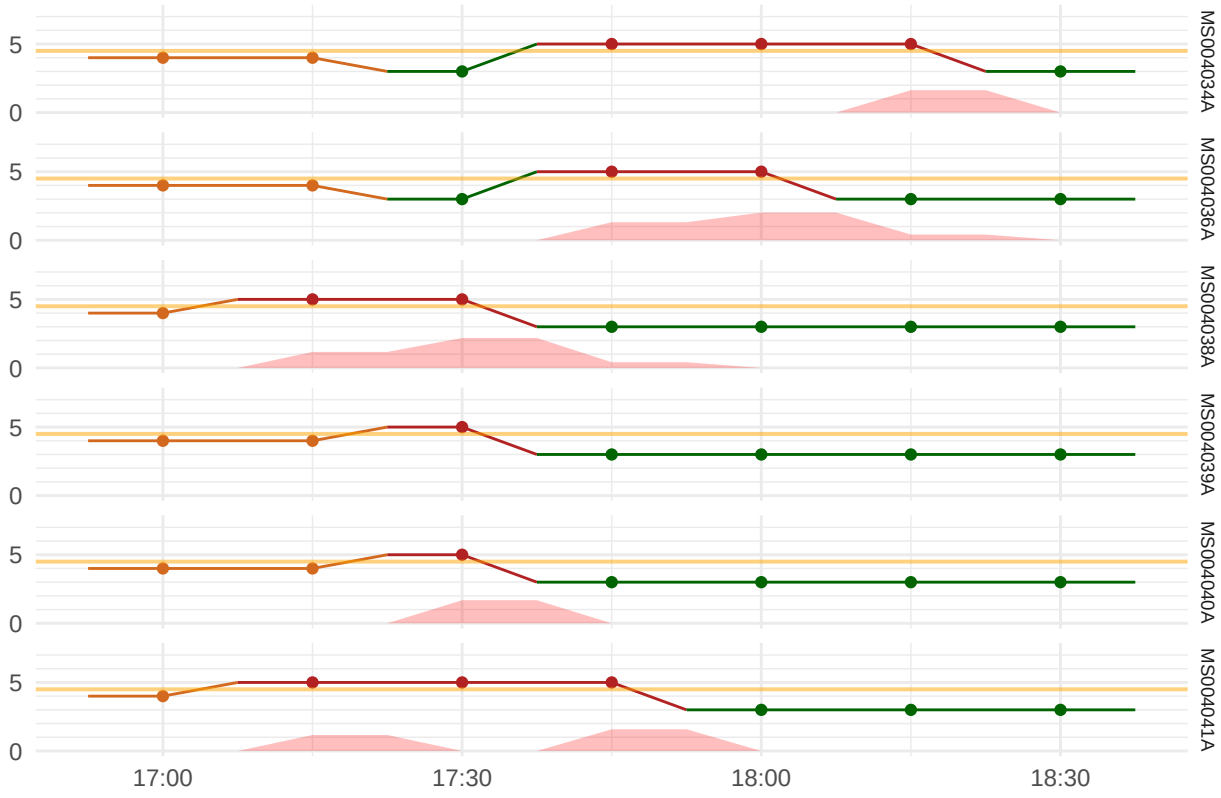


Figure 8.15: An incident occurring in the M4 motorway between MS004040A and MS004041A at 17:15 (28th January 2016). Traffic states [1 - 7] and incident score [0, 5] are shown.

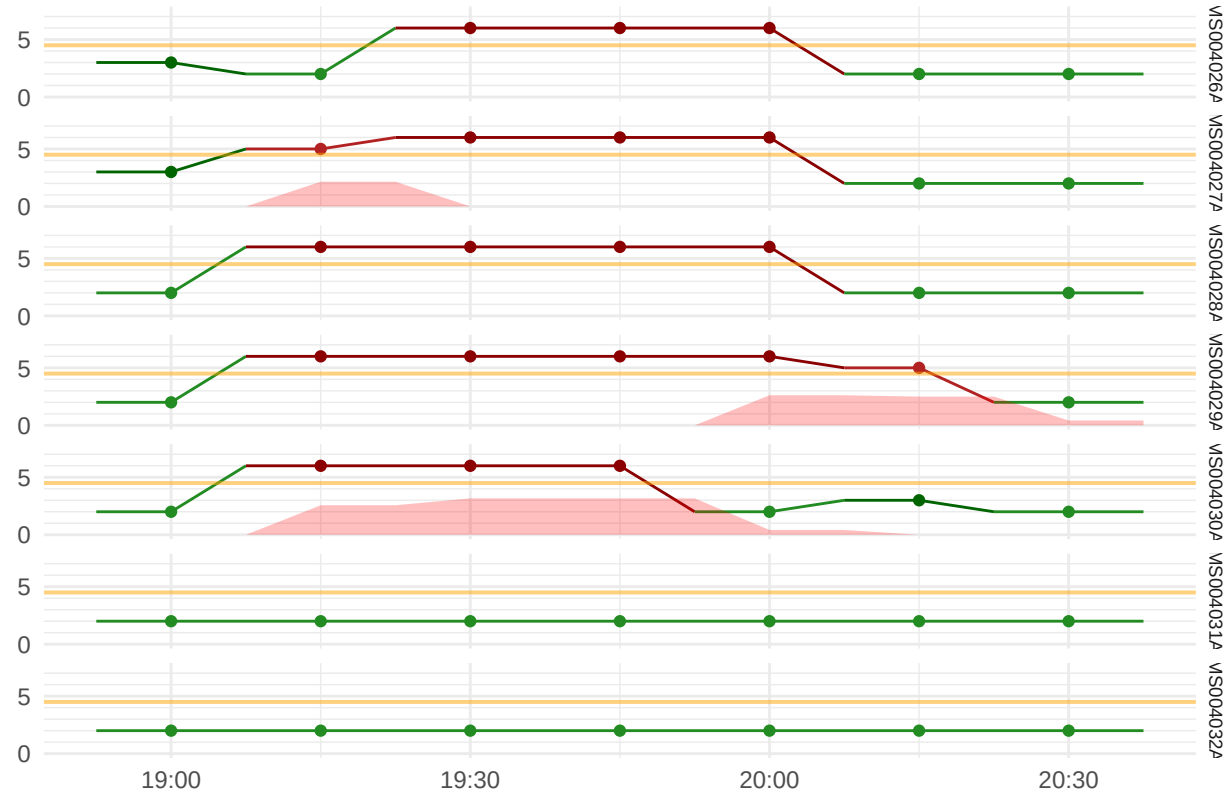


Figure 8.16: An incident occurring in the M4 motorway between MS004030A and MS004031A at 19:15 (31th March 2016). Traffic states [1 - 7] and incident score [0, 5] are shown.

of such road network can be seen in Figures 8.17 and 8.18 focused on those areas where the groups of detector used in the experiments are located.

The first thing to take into consideration is that as the incident detection model is based on a spatiotemporal model that relates the transition of the traffic states in the time-space, it is is possible that some of these traffic state transition could be detected as anomalous. This may be because of the small sample size of the data in spite of the aimed efforts to perform several simulations. More specifically, 4 simulations with normal traffic conditions have been performed, and then 5 different simulations with traffic incidents. Two of these simulations have been run twice in order to change the random seed. Then, the traffic state models from Chapter 7 is trained with this data, and so the spatiotemporal probabilistic model presented in the current Chapter. Data resolution is $\Delta t = 1$ minute per sample.

The first simulation is run from 8:00 to 9:30, performing a full lane closure between 8:30 and 8:40. The affected detector is Semaforo_0. The result is shown in Figures 8.20 and 8.21. There are two figures because the same simulation was run twice changing the random seed, but the result is almost identical. It can be observed the level of maximum congestion (state 7) in the original affected detector Semaforo_0 as well as its upstreams (Semaforo_50, Semaforo_100, N01151O1). Thus the incident score assigned by the incident detection model is high despite the challenging

situation as the detectors are pretty close to a traffic light with the associated noise and the existence of transient traffic states. There are some anomalous detections with a low score out of the incident time that may be due to the small sample size and thus are considered as anomalous transitions of traffic states which have involved a disruption in the flow.

The second simulation is also run from 8:00 to 9:30 with the traffic incident —a full lane closure— occurring between 8:30 and 8:40. This simulation is also carried out with two random seeds. In this case, the studied group of detectors is different and the main affected detector is Exp2_m50. In this case as shown in Figures 8.22 and 8.23, the incident is completely detected at the moment it arises (8:30). The severity assigned to the incident is high and increasing over time because of the spatial propagation of the congestion, the persistence of the incident in time, and also the level of existing congestion (state 7) compared to the level of traffic in the downstream detectors (Exp2_Left, Exp2_50, Exp2_100) which are in the traffic state of no traffic (state 1). After the simulated incident has finished, there is still some effects of the remaining non-recurrent congestion as detected by the model at some spots (Exp2_m50, Exp2_m100).

The third simulation is run from 7:00 to 11:00, with the incident taking place between 8:00 and 8:30. The affected detector is Semaforo_0, performing a full lane closure. The incident is clearly detected as shown in Figure 8.24 with a high severity which is increasing over time. However, the main effects of such incident are detected in the detector N01151O1 instead of the Semaforo_0.

The fourth simulation is also run from 7:00 to 11:00 and the incident occurs between 10:30 and 11:00. The fifth simulation is similar and it is run from 15:00 to 19:00 with the incident taking place between 18:30 and 19:00. In both cases the main affected detector is Semaforo_0, performing a full lane closure. There is an interesting fact here, as shown in Figures 8.25 and 8.26, and it is that even though the incident is timely detected when it arises, there are some moments where the incident score decreases. This is because of the great representation of such transition of traffic states in a small sample size. Because of this, the alarm mainly rises when there are traffic flow disruptions when detectors D01152 or D01155 change from no traffic (state 1) to observe a certain flow level (state 2).

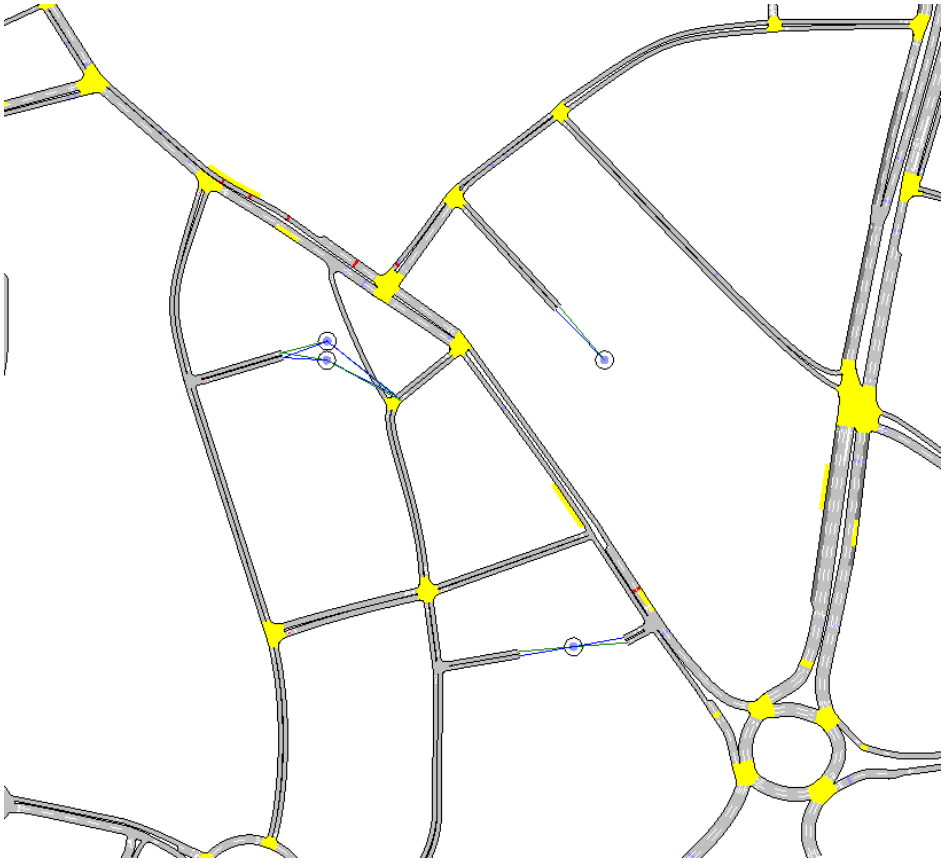


Figure 8.17: Bristol network focused on the area where the detectors *Semaforo_100*, *Semaforo_50*, *Semaforo_0*, *N01151O1*, *D01155*, *D01152* are located.

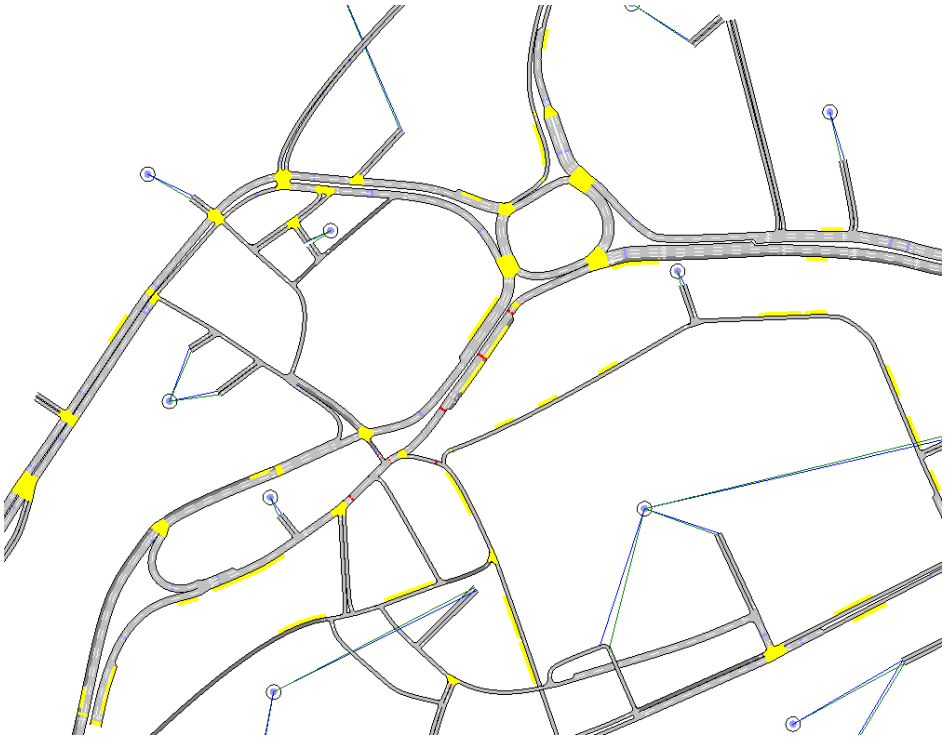


Figure 8.18: Bristol network focused on the area where the detectors *Exp2_m100*, *Exp2_m50*, *Exp2_50*, *Exp2_100*, *Exp2_Left*, *N01331Q1*, *N01331Q2* are located.

Figure 8.19: Color legend for traffic states [1 - 7] and the incident score range [0, 5].

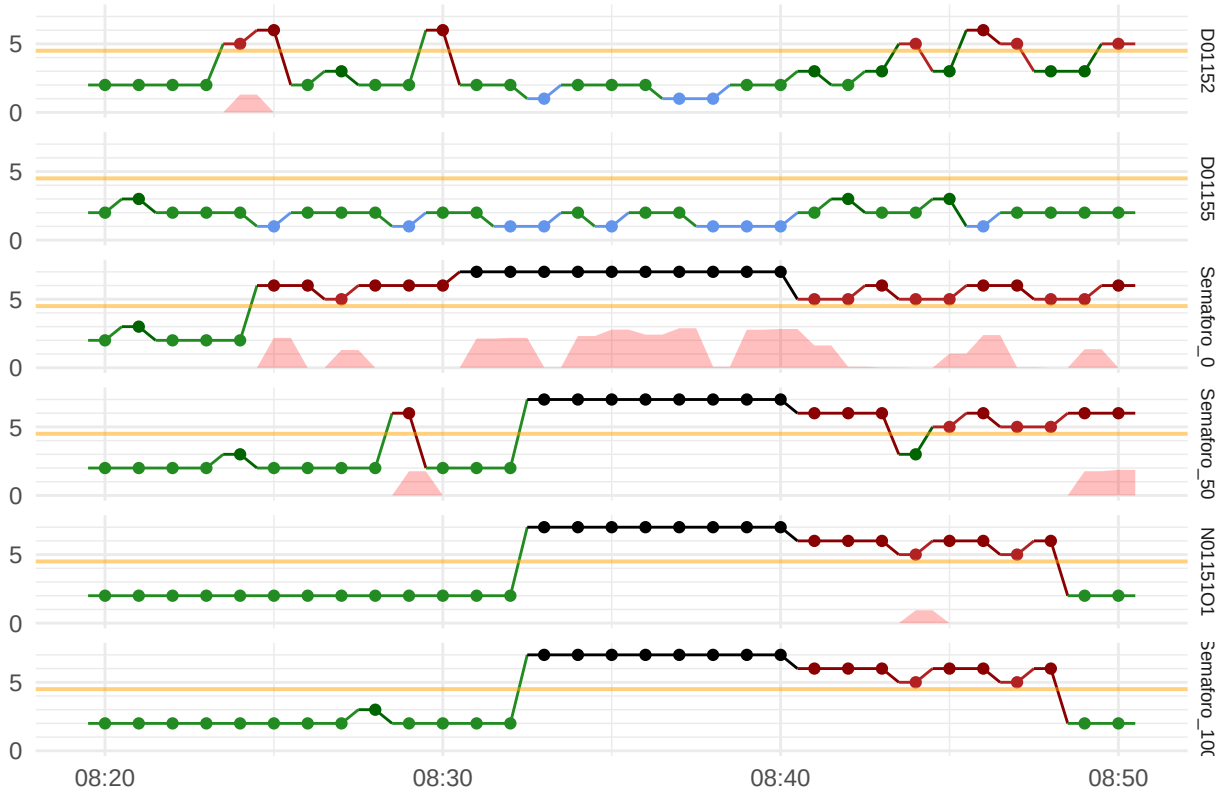


Figure 8.20: An incident occurring in the Bristol urban network nearby Semaforo_0 at 8:30. Random seed *A*. Traffic states [1 - 7] and incident score [0, 5] are shown.

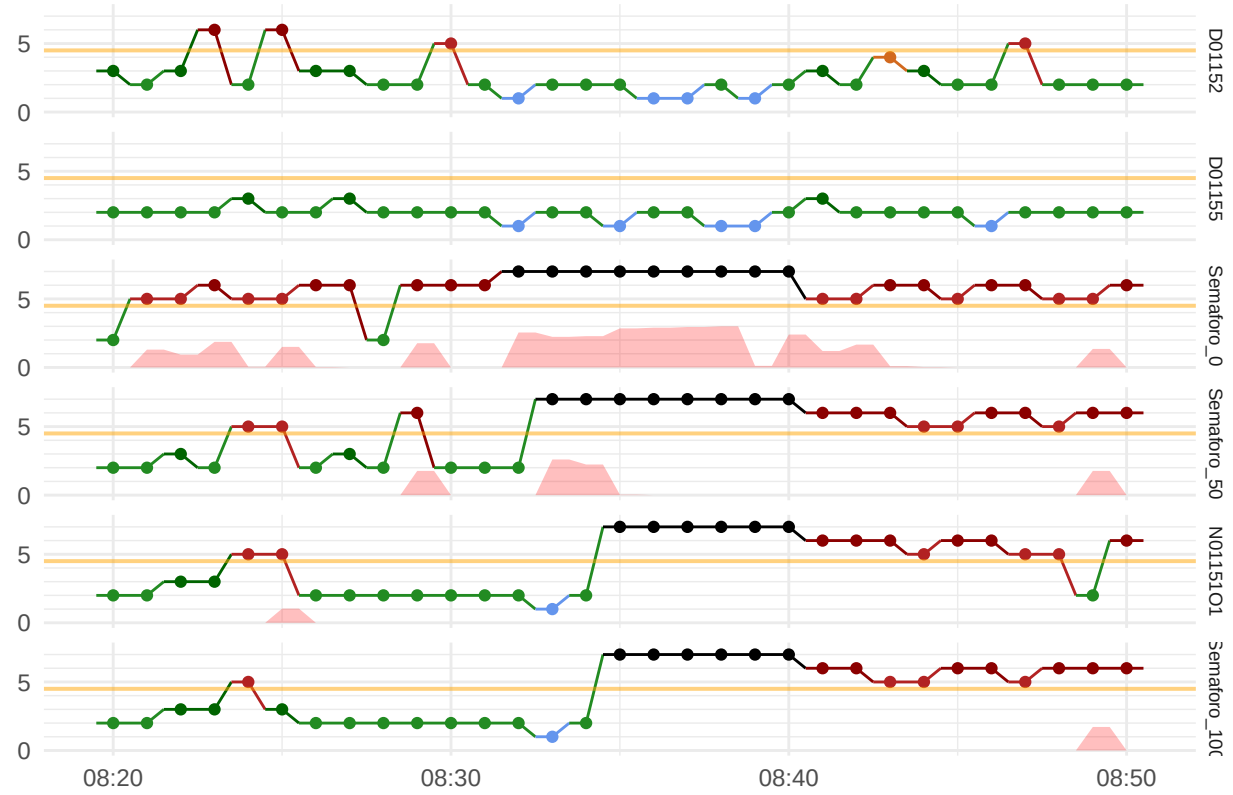


Figure 8.21: An incident occurring in the Bristol urban network nearby Semaforo_0 at 8:30. Random seed *B*. Traffic states [1 - 7] and incident score [0, 5] are shown.

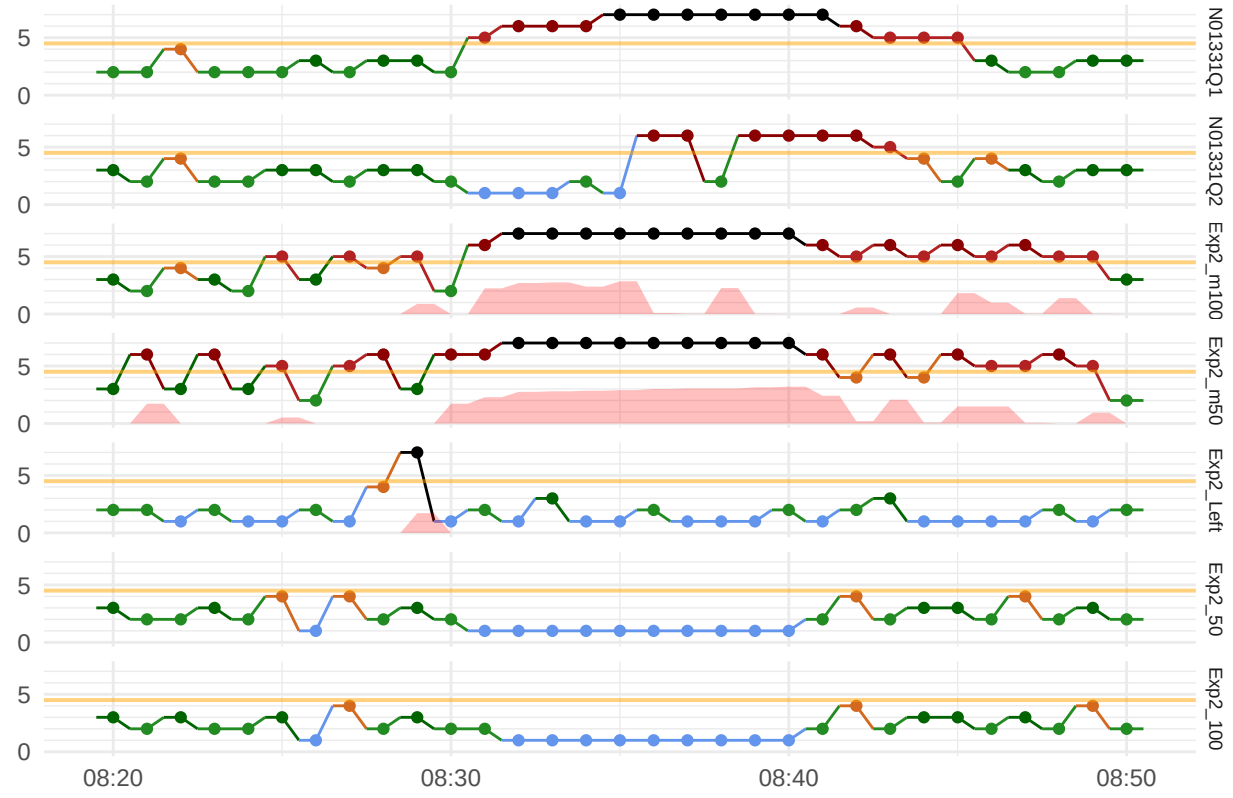


Figure 8.22: An incident occurring in the Bristol urban network nearby Exp2_m50 at 8:30. Random seed *A*. Traffic states [1 - 7] and incident score [0, 5] are shown.

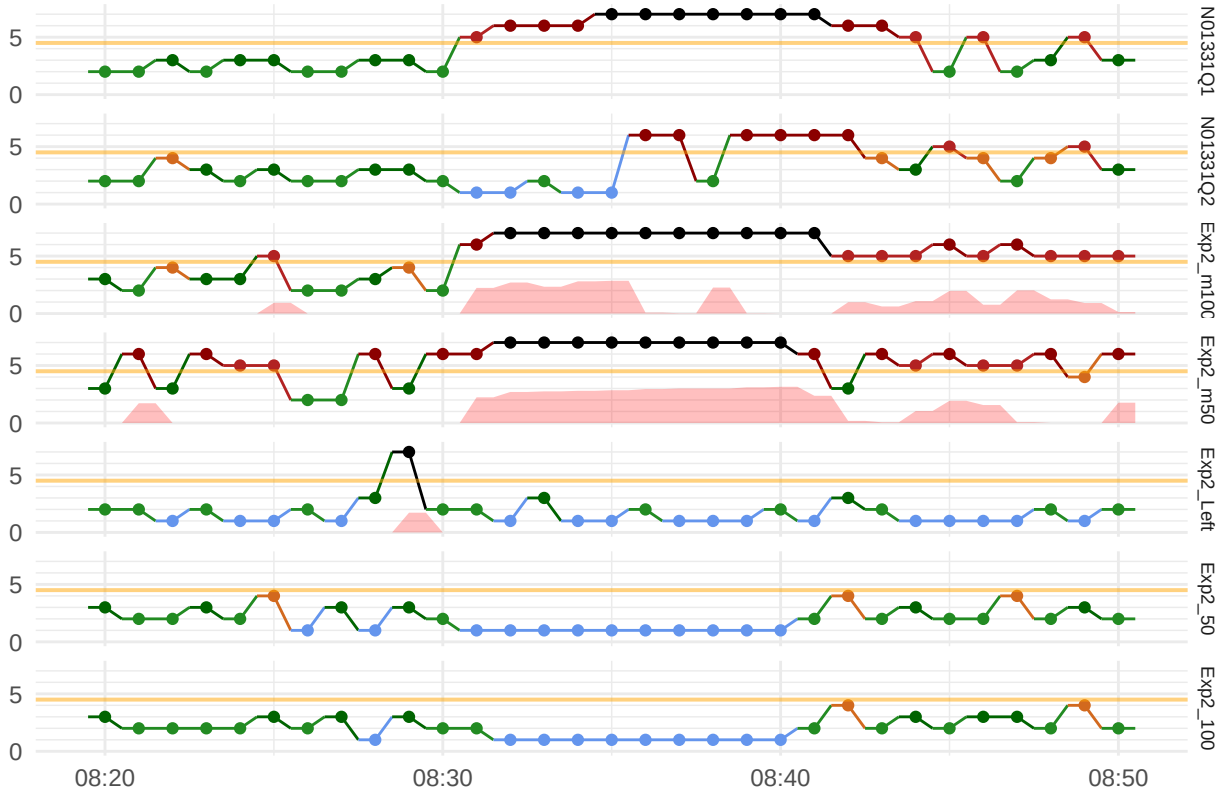


Figure 8.23: An incident occurring in the Bristol urban network nearby Exp2_m50 at 8:30. Random seed *B*. Traffic states [1 - 7] and incident score [0, 5] are shown.

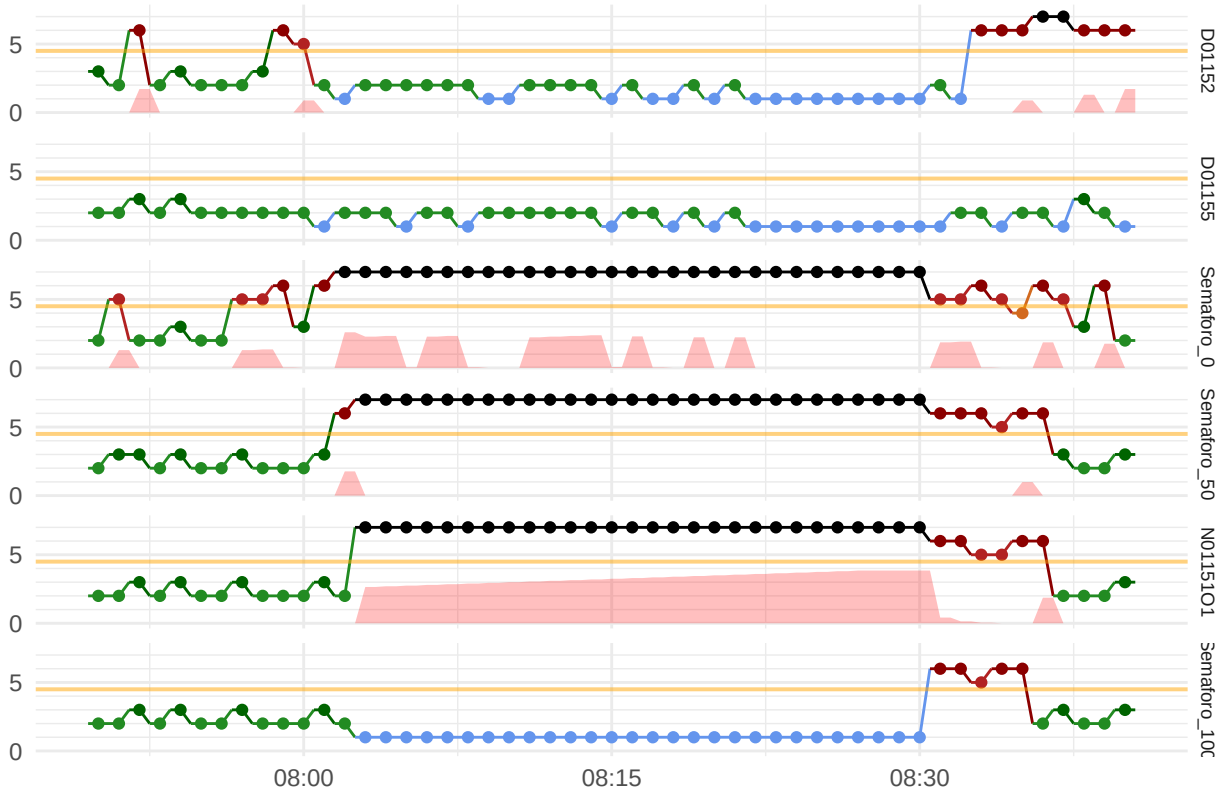


Figure 8.24: An incident occurring in the Bristol urban network nearby Semaforo_0 at 8:00. Traffic states [1 - 7] and incident score [0, 5] are shown.

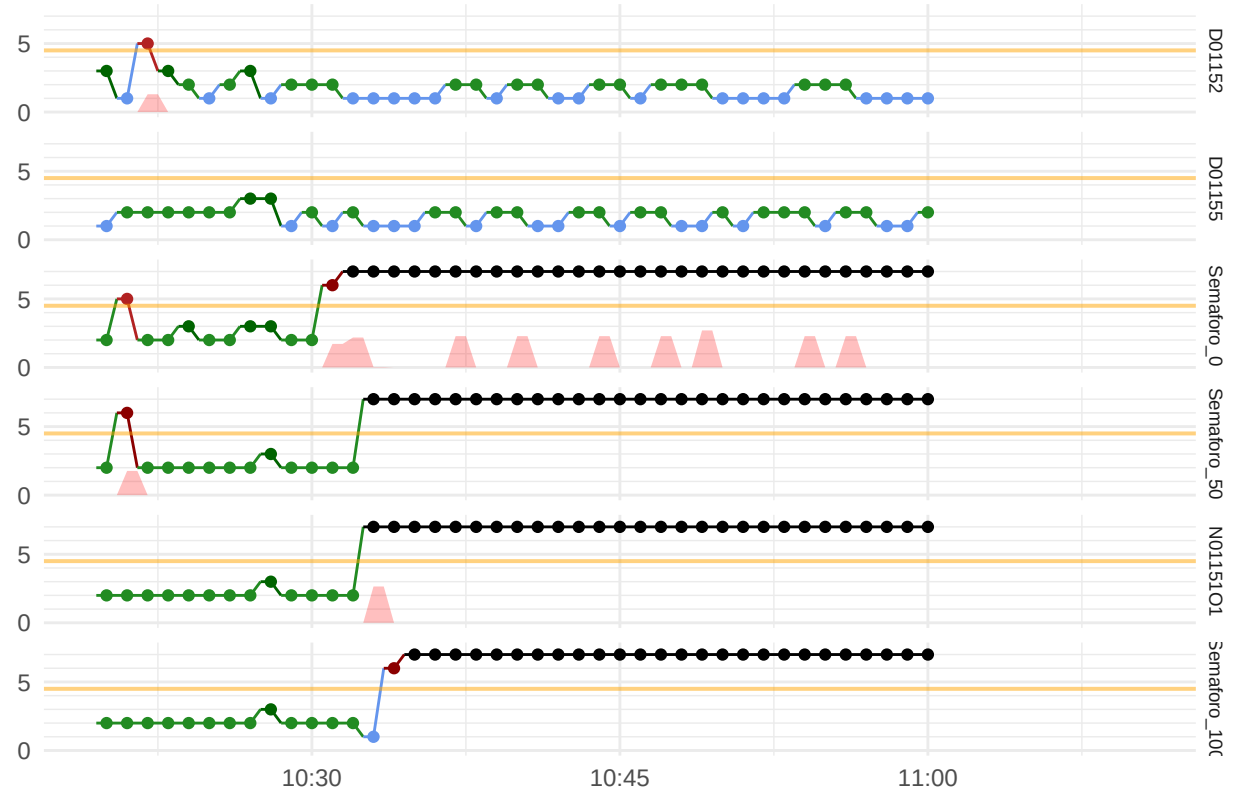


Figure 8.25: An incident occurring in the Bristol urban network nearby Semaforo_0 at 10:30. Traffic states [1 - 7] and incident score [0, 5] are shown.

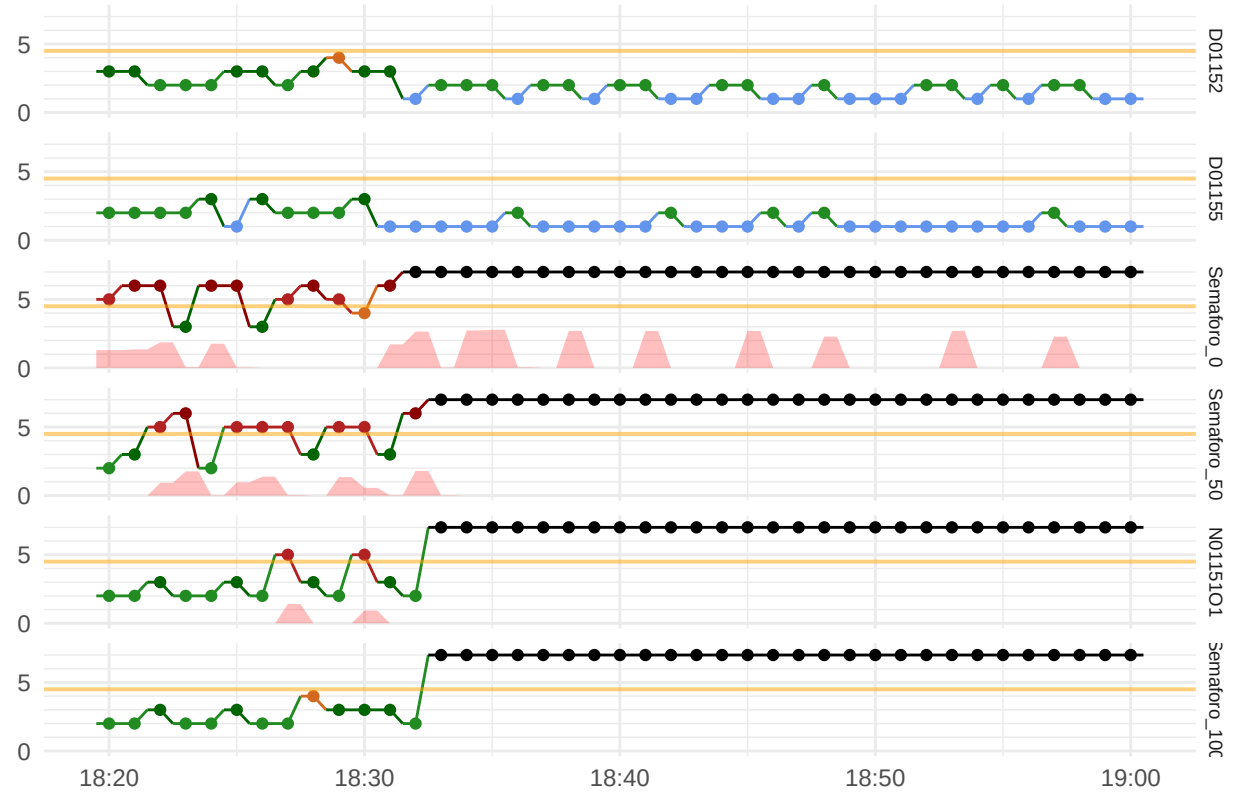


Figure 8.26: An incident occurring in the Bristol urban network nearby Semaforo_0 at 18:30. Traffic states [1 - 7] and incident score [0, 5] are shown.

9 Conclusions and future research

The aim of this thesis has been the design and development of an integrated system for real-time traffic forecasting based on machine learning methods. Nevertheless, despite the fact that traffic prediction has been the driving motivation for the thesis development, the proposed ideas and scientific contributions in this thesis are generic enough to be applied in any other problem where, ideally, their definition is that of the flow of information in a graph-like structure with special interest in environments susceptible to changes in the underlying data generation process. Moreover, the modular architecture of the proposed solution facilitates the adoption of small changes to the components that allow it to be adapted to a broader range of problems.

The focus in this thesis, thus, has been put on a macroscopic perspective of the traffic flow where the individual road traffic flows are correlated to the underlying traffic demand. These short-term forecasts include the road network characterization in terms of the corresponding traffic measurements —traffic flow, density and/or speed—, the traffic state —whether a road is congested or not, and its severity—, and anomalous road conditions —incidents or other non-recurrent events—. The main traffic data used in this thesis is aggregated data coming from inductive-loop detectors installed along the road networks. In spite of their pitfalls, the main reason is that they have become the most widely used sensor in traffic management systems since their introduction in the early 1960s and such the large source of available data. Nevertheless, other kinds of traffic data sources could be equally suitable with the appropriate data preprocessing.

Furthermore, the proposed data-driven forecasting system is planned to be linked to a simulation-based traffic model in a mutually beneficial relationship where they cooperate and assist each other. The simulation-based platform —where multiple traffic models are implemented— is Aimsun Next and the corresponding real-time version for traffic management is Aimsun Live [5]. In this sense, an example is when an incident or non-recurrent event is detected with the proposed methods in this thesis, then the simulation-based forecasting module can simulate different strategies to measure their impact.

Part of this thesis has been also developed in the context of the EU research project “SETA” (H2020-ICT-2015) [61] which is creating a ubiquitous data and service ecosystem for a better metropolitan

mobility and the analysis of how short-term prediction can be improved through the use of multiple, highly diverse sources.

The whole result of this thesis, as shown in Figure 4.1, is an integrated system —*Adarules* described in the Chapter 5— for real-time forecasting which has the ability to make the best of the available historical data, while at the same time it also leverages the theoretical unbounded size of data in a continuously streaming scenario. This is achieved through the online learning and change detection features of the system along with the automatic finding and maintenance of patterns in the network graph. In addition to the Adarules system, another result is the probabilistic model described in the Chapter 7 that characterizes a set of interpretable latent variables related to the traffic state based on the traffic data provided by the sensors along with optional prior knowledge provided by the traffic expert. On top of this traffic state model, it is built the probabilistic spatiotemporal model described in the Chapter 8 that learns the dynamics of the transition of traffic states in the network.

More specifically, as described in Chapter 5, a non-parametric modelling approach has been adopted. This means that patterns related to the existing traffic dynamics in the network are found automatically from the observed data, but the key aspect is that this process of pattern mining is not limited to the collected historical data but it is continuously running in order to unveil patterns on new data, detect abnormalities in the context of such patterns, and also to detect when these become outdated. Therefore, as shown in Figure 4.1, there is no hard requirement in having collected large amounts of data —e.g. a full year to have observed all the seasonality effects as in typical time-series modelling approach— before the beginning of a project. This is of great importance as it is not rare at all to start a new project in a city where the available collected traffic data only extends to three months for example. Obviously, the more collected historical traffic data to use the better, as it can be used to perform a more accurate pattern identification before starting to execute the forecasting system in real-time. Moreover, the system performs an efficient use of that historical data as outdated data would be automatically discarded when appropiate. The historical collected traffic data can also be used to train the traffic states models as well as their interactions in the probabilistic spatiotemporal model.

Another input to the system usually in the offline stage, but that could be carried out in real-time too, is the consideration of the prior knowledge or assumptions in the form of qualitative variables —such as the day of the week, time of the day, public holidays and/or events calendar, weather, etc.—. These priors are the form that the expert user —i.e. a traffic engineer or traffic manager— may express their knowledge or beliefs about the specific road network. Otherwise, there exist some reasonable defaults like using the basic calendar data —time of the day, day of the week and

public holidays calendar—.

Finally, among the inputs, it is important to use the graph of the road network in order to perform an accurate pattern mining —i.e. the rules— as well as building the probabilistic spatiotemporal model. It is not necessary to include a lot of detail about the geometry nor traffic signal control plans. The only required information is the graph describing roads' connections in a topological sense.

Once the system is running in real-time, it is fed with incoming traffic data streams. This streaming data is used in two processes: the Adarules framework and the local traffic state identification. Within the Adarules system, the data is first used to update and check the validity of the identified graph patterns —rules— and perform proper restructuring changes if necessary when a global change is detected on them within the *graph global change detection*. Then, data is also used to unveil new patterns as more evidence is collected in the context of every rule using the *graph pattern mining component*. These graph patterns or rules are the main components within Adarules, as every part of the observed data belongs to a certain rule. This makes the system prone to parallelization and scalable. Besides updating rules and their statistics with new data, detecting if a rule is outdated —i.e. a global change is detected— or discovering new patterns, the streaming data is also used to measure the outlierness of the traffic observations in the context of a given rule using the *graph flow anomaly detection*. Missing observations can be replaced with the statistics collected in every rule, which is called as *basic imputation for missing data*. Furthermore, the data is also used in the local context of every node in the graph for the *forecasting models*. This includes using the models for prediction, updating them with new data and using their outputs in the process of *local change detection*.

On the other hand, the streaming traffic data is also fed to perform the *local traffic state identification* using the model presented in Chapter 7. These traffic states are the input for the *spatiotemporal probabilistic model* presented in Chapter 8, along with the graph of the road network as well as optionally the current rule pattern. The outputs from this spatiotemporal model include the measurement of the outlierness using the *traffic states' anomaly detection*, as well as performing automatic *incident detection* and the quantification of their severity. The spatiotemporal model may also be used to estimate the *most probable network traffic state* using the current traffic observations so far and optionally the current rule pattern too, and this network traffic state can be used to perform an *informed imputation for missing data* that can be used in the context of Adarules for feeding the forecasting models.

Finally, the output generated by Adarules —i.e. the traffic forecasts— and the observed forecasting error is also used to improve the system's overall performance by detecting changes and updating

the forecasting models to adapt them with new data. Another kind of provided output such as the outlierness scores and incident detection is used to raise internal alarms that can be managed to select another rule pattern which fits better the current traffic conditions, and also to raise external alarms, i.e. to the end-user such as the traffic management centre, in order to perform proper actions such as activating the simulation forecasting module or inform the local authorities and/or drivers.

All this has led to the achievement of the proposed goals stated in Chapter 1 and Chapter 2. Namely:

The lack of **autonomy** was a critical point because, for every new Aimsun Live project, the transport modellers and data scientists had to decide —based on data analysis or expert knowledge— the amount of data to be used in order to build the traffic forecasting models for example. This was happening not only at the beginning of every project, but also during their lifetime such as the maintenance timings to update the models with new data. This was limiting very much the ability to scale with an increasing number of projects. In this sense, designing an autonomous system has been a motivation that has guided the development of this thesis. The only prior requirement is to have the graph of the road network at hand which can be retrieved from the geometry of the Aimsun model or, otherwise, fetching it from open collaborative projects such as *OpenStreetMap*. From this, the Adarules system is completely autonomous to unveil patterns of flow propagation in the network as well as discard outdated historical based on the data itself. Therefore, it is an *intelligent* analysis that pursues to make the best of the available historical data so far not simply discarding data older than a specific date but relying on an automatic data analysis to perform the decisions. This data analysis aims to unveil the underlying traffic dynamics in the transport network, conversely to the previous situation at Aimsun where the design criteria for such forecasting models was only time-dependent and did not consider the effects of the traffic dynamics from the specific transport network. However, it is clear that the spatiotemporal correlations in a network are dynamic as they are not just time-dependent, but they are also conditioned on the different movement patterns underlying the transportation system which responds to the existing traffic demand. These mobility patterns —that we associate with *graph patterns* or *rules*— are sought by exploiting the graph structure of the road network following an evidence-based decision making procedure, a process which is carried out by Adarules within the graph mining module.

Furthermore, the autonomous feature of the proposed system is not only aimed during the offline stage for the system preparation but also maintained during the Aimsun Live project's lifetime. Without a doubt, historical patterns are far from being stationary and they must evolve in the same manner traffic demand do because of changes in the needs or behaviour from the users of the

transportation system. In fact, this is equivalent to the level of **adaptation** the system has in order to achieve an automatic operational level in the real-time traffic forecasting task. This feature has been considered of great importance during the development of this thesis, as it is essential for a real-time forecasting system to be able to self-calibrate with new streaming data. This includes correcting the system when the performance is downgrading or a change is detected, and updating the system to incorporate new knowledge. This has been achieved through the different modules in Adarules for change detection both at a local or a global scope within the road network. In the end, this let the system to be more reactive and efficient about the usage of data as it frees the end users from deciding which data size is more appropiate and how often a maintenance must be scheduled to build again the models with new data. Besides, this adaptation in an online learning scenario is performed to both gradual and sudden changes.

Part of the aforementioned autonomy is derived from the non-parametric nature that Adarules has been designed with. The non-parametric approach involves placing as minimum assumptions as possible on the data modelling and, instead, finding and monitoring these data relationships within the graph pattern mining module using evidence-based criteria in an online manner for an accurate adaptation to changes in the traffic demand or supply. On one hand, this makes the Adarules system not dependent on the prior definition of e.g. trend or seasonality components during the modelling stage as occurs in other kinds of data modelling approaches, thus relaxing the requirement to have long data records in order to identify such components. On the other hand, it also makes the proposed system more robust to changes in these assumptions as the found patterns are continuously monitored with new data to verify their validity.

Another important objective achieved is the system's ability to be aware in order to quantify the amount of outlierness present in the transport network in order to being **informative** about such anomalous traffic conditions and react accordingly. This is of great importance as the only measure before this thesis was to rely solely on the forecasting error. Although it is a good indicator of the outlierness, it was only possible to check it after observing the actual values associated with the forecasts. Now, it is possible to measure such outlierness in real-time to react accordingly. For example, when an anomalous network state is observed that do not match the expected pattern behavior but it is recurrent in the context of another identified rule. On the other hand, an anomalous network state whose identification is associated with a non-recurrent event —e.g. an incident— may be better handled by a causal and simulation-based forecasting approach in order to simulate different strategies and response plans. In the end, more information also supports decision making as drivers or autonomous vehicles may take different decisions about routing, and traffic managers and planners can better coordinate emergency responses as well as implement

controls to minimize the traffic flow disruption to other areas of the network.

In the Chapter 6, the **forecasting accuracy** has been demonstrated to be improved with respect to the previous existing methodology (ANA) and with regards to a baseline based on a historical seasonal average. The level of accuracy obtained is appropriate and useful for traffic management purposes.

The **robustness** of the system when facing outliers and missing data has been considerably improved which is of great value because noisy and faulty traffic data is rather common in real-time. The system, as described in Chapter 5, has specific modules to deal with such missing data and to penalize those unreliable data sources such as malfunctioning data devices during real-time. The main motivation behind this design has been to develop a robust solution which makes the most of the observable data.

Interpretability has been another pursued objective during the development of this thesis. And so it is demonstrated by the choices made in performing the different models' selection. For instance, rules identified in Adarules (Chapter 5) are expressed using qualitative variables as it can ease the interpretation for end-users —e.g. traffic engineers or managers— as well as it would allow an easier diagnostic, unlike other black-box modelling techniques in the statistical and machine learning fields. An analysis of the most relevant factors may also be performed. In addition, the forecasting models within Adarules have the form of a sparse model for spatiotemporal correlations, thus seeking only the strongest and most relevant correlations within the network for the sake of better interpretability and robustness. The probabilistic model for traffic states presented in Chapter 7 has also interpretability in mind as its output is a set of latent variables related to the underlying traffic states whose semantic interpretation is very natural. Finally, the modelling performed in Chapter 8 was also aimed to boost the interpretability of both the model itself by using a graphical model as well as the outputs and the intuition behind the results as a consequence of the approach based on probabilities and the interpretable traffic states.

In statistical modelling, the assumptions placed by the analyst during the modelling stage determine which type of conclusions may be extracted and how significant are they. On the other hand, within the machine learning culture, this point is not so crucial as the performance of the forecasting task. Often, this is seen as two different cultures within the statistical modelling field [43, 175]. In this sense, as aforementioned, we have decided not to place very strong assumptions in the data model and thus, approaching the problem from a machine learning perspective. However, although treating the problem with a black-box model is common in many machine learning techniques, we have endeavoured to not sacrifice the human interpretability of the model, especially regarding end-users such as traffic engineers or managers. This end-user of the system has not to be an

expert data analyst to be able to interpret the output as well as have a high-level interpretation of how the system works and what is the reasoning behind it. This is what has motivated our modelling decisions instead of selecting other popular techniques within the machine learning field as the interpretation of their internal workings is more black box.

Finally, the **scalability** has been another objective in mind. For this reason, the proposed solution is prone to be scalable by the design of a modular architecture with emphasis on a parallelizable exploitation of large amounts of data.

All these reasons have motivated the design and development of a self-adapting method which, in addition, has the ability to be interpretable to the end-user —a traffic engineer or manager— about the inference results.

9.1 Future work

A number of ideas for future work are:

The first point concerns the parameterization of the amount of data used in the split evaluation of rules. Currently, rules are updated with new data, but yet they need full historical data —under their scope— when being evaluated to be expanded. This point would make Adarules more computationally efficient in the long-term, ideally with a tiny cost to pay in the accuracy.

Related to this point it is the use of data compression —or summarization— methods such as exponential windows in order to perform compression of old data thus saving costs in storage.

Despite the model is constantly evolving to match the current traffic demand, we are using the identification of recurrent traffic conditions —i.e. patterns— coupled with real-time traffic information to predict the future within a data-driven approach. This means that it can happen at some point that the rule that matches the current qualitative conditions is not actually the best fit in terms of traffic pattern but still, there would be another rule which fits better such traffic conditions. At first, this situation may be handled through graph matching for a better rule selection but in the end, the system should reorganize itself to better accommodate such traffic patterns and it will depend on how fast demand changes occur and the system's ability to detect them and adapt itself. On the other hand, there could also happen that current rule does not match properly the current traffic conditions because there are non-recurrent events in the network —e.g. an incident, traffic rerouting by authorities...— that modify the usual drivers' route choices. This latter case would be an example of non-recurrent traffic conditions which are hard to handle by a predictive data-driven approach —beyond the ability to properly identify them— because of its implicit unpredictability

and is best suited for a simulation-based solution with different case scenarios which works paired with the proposed approach. However, for the former case —i.e. detecting unexpected but recurrent traffic conditions— it is planned to implement a rule pattern matcher that recognizes previously seen situations for faster adaptation.

It is also planned to evaluate the entire proposal of this thesis with new datasets, in order to perform more experimentation. Although the results from the most critical parts have been validated by traffic engineers, the validation of the entire system results to validate their consistence would be very useful.

It would be interesting the use of new data sources —with the corresponding data preprocessing— beyond the induction loop detectors used in this thesis, as for example floating car data or mobile data. Furthermore, it would be very interesting to perform data fusion of these multiple data sources in order to increase the observability of the transport network.

Finally, it would be interesting to study different new split metrics in order to perform the graph pattern mining, as well as test different forecasting models in the context of every rule beyond the proposed sparse model for spatiotemporal correlations.

Appendix I: Taxonomy of traffic modelling

Taxonomy of the types of traffic simulation in transportation may be done according to how time, space and state are represented, namely in a continuous or discrete form.

However, a more common criteria to differentiate between different fields of traffic science is the time scale used and its associated level of detail. As shown in Table 9.1, longer time scales ranging from hours to years refer to the realm of transportation planning where the influence of demographic change over the traffic demand variations is taken into account. These two fields, traffic flow dynamics and transportation planning, complement each other as the output from the transportation planning model —which could be the classical four-step scheme (trip generation, trip distribution, mode choice, and route assignment) or another modern dynamical variant— is the traffic demand (vehicles per hour) on each link of the considered network. On the other side, these variables are externally given for traffic flow simulations typically in the form of boundary conditions. Furthermore, in the last few years there has been a growing overlap between these fields. Some examples are:

- Models for agent-based dynamic traffic assignment combine the route assignment step of classical transportation planning with traffic flow models.
- The new generation of connected navigation systems inside cars couple the dynamics of traffic flow (jam formation) with that of traffic demand (traffic-dependent routing).
- When modelling the effects of driver-assistance systems on traffic flow, it is needed to simultaneously model aspects of vehicular and traffic dynamics.

Travel demand modelling

Transportation planning models operate at a high level, merging land-use models with the socio-economical behaviour of people to link the traffic demand —e.g. the travellers aggregated into spatial zones or centroids— with the traffic supply —i.e. the transportation network infrastructure— . Land-use models seek to explain the growth and layout of urban areas through the mapping of the

Table 9.1: Scales of traffic flow dynamics from vehicular dynamics and transportation planning. Source: Traffic Flow Dynamics.

Models	Time slice	Aspect of traffic (examples)
Vehicle dynamics		
Sub-microscopic	< 0.1 s	Control of engine and brakes
Traffic flow dynamics		
Car-following models	1 s	Reaction time, time gap
	10 s	Acceleration and deceleration
Macroscopic models	1 min	Cycle period of traffic lights
	10 min	Stop-and-go waves
Transportation planning		
Route assignment traffic demand	1 hour	Peak hour
	1 day	Daily demand pattern
	1 year	Building/changing infrastructure
Demography	5 years	Socioeconomic structure
	50 years	Demographic change

activity map —social, economical or cultural activities— with the spatial separations according to where they take place in the road network under study. One of the oldest known models of this type which relates economic markets and spatial distances was published —in a pre-industrial context—- by von Thünen in 1826 [241]. Around 100 years later, other classic land-use models appeared such as the concentric zone model developed by Burguess [50], the sector model by Hoyt [127] or the multiple nuclei model by Harris and Ullman [115]. However, recently the approach has changed as the urban behaviour evolves towards the segregation where activities occur at substantially different spatial locations and the growth of a city is represented as the evolution of a multi-agent system through simulation and coupled with geographical information systems and computer aided design. An schematic view of such transportation planning model can be seen in the Figure 9.1.

Following subsections are dedicated to further explain the two main approaches in the field of transportation planning for modelling the travel demand:

- those based on the handling of the amount of trips and their distribution,
- and those which are individuals' activity-based.

Trip-based approach

The first findings in the relation between activity patterns and the transportation system were published in 1954 [184] providing the basis in transportation analysis which later evolved in a

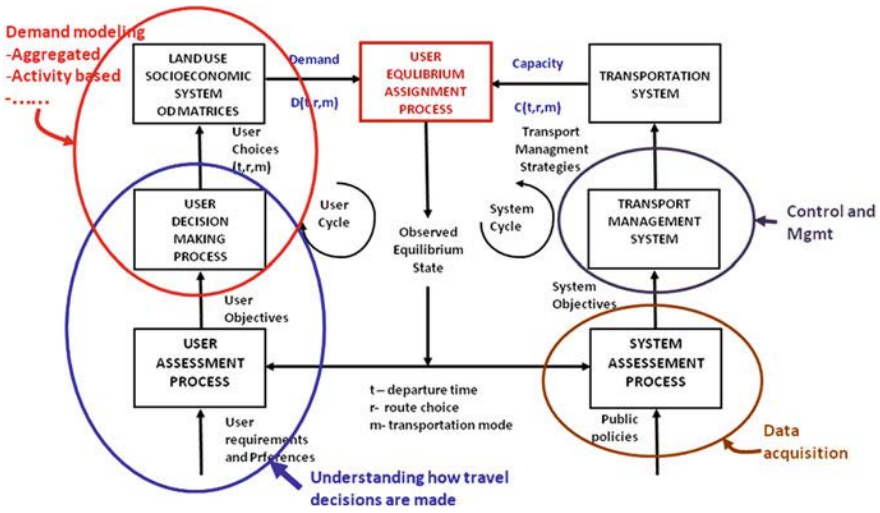


Figure 9.1: Components of the transportation system and their interrelationships. Source: [28] and adapted from [198].

methodology based on four consecutive steps called the *four step model* whose basis element are the trips [70, 179, 220]. In this way, individual travellers are aggregated into bundles of trips going from one point in the transportation network to another, these aggregated spatial zones are typically represented by their centrally located points called centroids.

The four step model is comprised of the following consecutive steps:

1. Trip generation.
2. Trip distribution.
3. Mode choice.
4. Route choice.

The first three steps forms the methodology to set up the travel demand which is expressed as origin-destination (OD) pairs and that relates the amount of trips originating at each centroid —seen as OD matrix rows— and their respective desired centroid destination —seen as OD matrix columns—. The last step aims to load this travel demand into the network supply whereby the trips are assigned to routes. An important note is that these aggregated trips are only considered for a specific time period, so the whole four step process is applied for different time periods although sometimes an iteration is done between the four steps, e.g, using the traffic assignment procedure to calculate link travel times that are fed back as input to traffic distribution and mode choice steps. Currently, this is the most used approach for modelling the traffic demand.

Trip generation

More specifically, the trip generation step takes care of the amount of trips that are originated in certain zones and those that arrive in these zones, i.e. a production and attraction process. This process is based on land use and other socio-economical activities and it is implemented using techniques such as regression analysis, category analysis, or logit models, among others.

Trip distribution

The second step, trip distribution, connects trip originations to their destinations and resulting in the generation of the previously mentioned OD matrix in such a way that a cell O_iD_j at row i and column j represents the total number of trips departing from the centroid O_i and reaching the destination centroid D_j and with diagonal elements denoting intra-zonal trips without any route assignment. Given that the four step process applies to different time periods, the resulting OD matrix should be called time-dependent or dynamic OD matrix. This OD matrix estimation is an under-determined system of equations considering that there are lot of unknown variables and, thus, additional constraints are introduced. Despite this fact, there exists a vast literature of methods to estimate the OD matrices which includes growth factor models when a prior demand model is known, gravity or entropy models [119], methods of least squares [32], Kalman filtering [17, 138], dimensionality reduction approach followed by Kalman filtering [71], also including data fusion from additional sources to the first step [27], among much others [194, 54, 165].

Mode choice

Once the OD matrix has been estimated for the specific network and time period, the next step is to split it among the different modes of transportation. A common tool to solve this problem is discrete choice theory [33].

Route assignment

The last step of route assignment —also known as route choice or traffic assignment— pursues to find out which routes the travellers follow when going from their origins to their destinations, i.e. the sequence of links followed through the transportation network. In this scenario, the logic says that travellers will use the shortest route between their origin and destination, thereby a suitable measure of distance based on shortest path algorithms —such as Dijkstra's algorithm— can provide the possible routes. This procedure is governed by some basic principles published by

Wardrop in 1952 [260] based in Nash equilibrium from game theory, where the concepts of user equilibrium and system optimum appear. The user equilibrium states:

> The journey times on all the routes actually used are equal, and less than those which would be experienced by a single vehicle on any unused route.

While the system optimum states:

> The average journey time is a minimum.

This user equilibrium has several assumptions:

- All individuals' decisions have a negligible effect on the performance of others.
- There is no cooperation between individuals assumed.
- All individuals make their decisions in an egoistic and rational way.

Ideally, it is expected that everybody follows the user equilibrium criteria in such a way that the whole system remains in equilibrium where no one can reach a better state by choosing an alternative route. Nevertheless, this idealistic mathematical scenario is often transgressed, inasmuch as for instance there are situations, mostly in urban centres, where users' travel motivation is to look for parking space and thus incurring in an unexpected congestion; other scenario it is when travellers use their usual route based merely in a routine criteria independently of the quickest or cheapest according to the equilibrium criteria.

There exists two different methodologies in dealing the traffic assignment problem: a static and a dynamic approach. In the static traffic assignment (STA), the time varying congestion effects are neglected, thus assuming constant link flows and travel times where (1) users will choose the same cheapest route between a pair of origins and destinations because users have perfect information about the links' assessments, and (2) these link's assessment values are considered to be constant —i.e. the relation between link's traffic load and its capacity is not considered— thereby congestion generation is not taken into account. According to these assumptions, this implies an all-or-nothing assignment. A refinement to this methodology consists on not assuming that all users would behave in the same manner given that actually they have not perfect information. Thus, a stochastic assignment may be performed where each link route choice is drawn from a probability distribution. Capacity constraints can be introduced in order to modify the link's cost according to the relationship between current load and its capacity instead of keeping them constant. This procedure to evaluate the link's assessment value according to its current load is known in different ways: travel time (loss) functions, congestion functions, volume delay functions, link impedance functions, or link performance functions —e.g. the Bureau of Public Roads (BPR) power function

[49]—. Thus the consideration of this uncertainty related to the travelers' complete knowledge of the routes and if they all behave in the same rational way or not, is what makes the difference between performing a stochastic user equilibrium (SUE) assignment [66] or a deterministic user equilibrium (DUE). Once flows on the network links are in equilibrium, it remains to solve the assignment as an optimization problem —e.g. using Frank-Wolfe algorithm [89] or another approach as the method of successive averages (MSA)—. Thereby, the STA, in spite of its lack of consideration about time varying congestion effects, can produce a good result with these improvements, mostly for the long-term transportation planning.

The main problem is that STA relies on simple travel time functions to determine the link's assessment value and, therefore, it is difficult to recognize the link's capacities because the congestion is a dynamic event where its build-up and dissolution play an important role. In this regard, dynamic traffic assignment (DTA) [206] deals with a DUE or SUE assignment in the same manner that STA but, in addition, DTA can deal with another feature in the route choice regarding the travellers choice of departure time; while in the STA approach all the travel demand is simultaneously assigned to the network, a DTA with departure time choice can spread the departures in time. Another difference respect to the static approach is that, instead of using simple travel time functions, DTA features a dynamic network loading that, through the use of analytical models or detailed simulation, describe the propagation of individual vehicles in the transportation network. Furthermore, in the case of simulation-based traffic assignment, the route choice and dynamic network loading components can be iteratively executed. However, it is important to note that in these cases a good description of the network's links is necessary and that using simulation-based traffic assignment with very large road networks is not always computationally feasible. In addition, there is no unified framework that deals with the convergence and stability issues [205] in contrast to STA.

Activity-based approach

This approach to model the travel demand probably originated in 1970 [113]. The idea is the same as in the trip-based approach, in such a way that travel decisions are originated from a need to participate in social, economical, and cultural activities. But as opposed to the more aggregated trip-based view, the basis unit in this approach are individual activity patterns commonly referred to as household activity patterns [22], which allows transportation demand to be modelled in more detail.

There is no explicit general framework that encapsulates the activity-based modelling scheme, as opposed to the four-step model in trip-based modelling approach. Some common steps can be

identified:

- The generation of activities which is similar to the first step in trip-based modelling (production and attraction).
- The modelling of household choices, including starting time and duration of the activity, its location and modal choice.
- The scheduling of activities and the way that a household performs the tasks.

An interesting technique is the methodology of multi-agent simulations [25], where individual households are represented as agents and the models then allow these agents to make independent decisions about their actions whose time-scale ranges from short-term decisions as in driving behaviour to mid-term of daily activities or long-term decisions.

Traffic flow dynamics

The field of traffic flow dynamics considers traffic flow models that explicitly describe the physical propagation of traffic flows through the network. Research on the subject of traffic flow modelling started in the 1950s when Lighthill and Whitham [167] described a model for the traffic flow of vehicles analogous to particles in a fluid. Since then, mathematical description of traffic flow has been an active research field which has brought a wide diversity of models describing different aspects of traffic flow operations. The field can be classified according to several criteria:

- Model evolution in time: Depending on whether the simulation model evolves synchronously or asynchronously. The former is the most common approach in which the model is advanced at a chosen simulation time step, while the asynchronous or event-based simulations are those in which time advances in variable amounts that correspond to the instants in time at which an event changes the model state.
- Representation of the processes: Whether they are purely deterministic or stochastic.
- Level of detail: Could be macroscopic —highest level of aggregation and lowest level of detail, based on continuum mechanics typically entailing fluid-dynamic models—, mesoscopic (high level of aggregation with low level of detail, typically based on a gas-kinetic analogy in which driver behaviour is explicitly considered), microscopic (low level of aggregation and high level of detail, typically based on models that describe the detailed interactions between vehicles in a traffic stream) or submicroscopic (lowest level of aggregation and highest level of detail, near to microscopic level but with detailed descriptions of vehicles' inner workings).

The most usual classification criteria of traffic simulation models is by the level of detail and, thus, it will be the chosen criteria to describe them thoroughly. A comprehensive tour through the history and state of the art of traffic flow theory can be found in [28, 100, 125, 174, 247].

Macroscopic detail

In the macroscopic level of detail the traffic is described at a high level of aggregation using characteristics as flow-rate, density, and velocity, and individual vehicle dynamics are not explicitly represented. The macroscopic modelling aims at complying with the fundamental diagram [65] and relies on the continuum traffic flow theory for describing the time-space evolution of the variables characterizing the macroscopic traffic flows: volume, speed, and density, which are defined at every instant in time and every point in space. The main equation formally representing this theory is the conservation equation also known as the continuity equation [103, 159]; these equations are supported by the fundamental relation involving flow, density and speed, which determines the equilibrium conditions. The common approach to reach a solution is by numerically integrating the equations once that each traffic model of the road section —in the space dimension— is discretized in time and space, as for instance the Cell Transmission Model, published by Daganzo [64], does.

Macroscopic models —also known as continuum models or fluid-dynamic models in analogy as being based on fluid-dynamic theory and kinematic waves— have its beginning in the work by Lighthill and Whitham [167] which brought the classic first-order LWR model. Since then, several extensions have been introduced [144, 143] including the use of higher-order models [203]. Therefore, macroscopic flow models can be classified according to the number of partial differential equations (PDE) that frequently underlie the model on the one hand, and the order of these PDE on the other hand. However as stated in [63], despite the significant efforts and progress in the field of higher-order macroscopic models, the Berkeley school firmly holds its faith in first-order models and their extensions. The main reason for this it is because of the numerical solution schemes that are well developed and understood which is not the case for higher-order models, as these contain other characteristics that complicate the finite difference schemes and additionally, no analytical solutions exist for the higher-order models. Moreover, they sustain that first-order models are enough to describe the macroscopic traffic flow dynamics.

Within the macroscopic scale, it is worth noting the concept of the fundamental diagram of traffic flow as shown in Figure 9.2 given that its characterization describe the traffic state. A large volume of literature exists on the description of the traffic state [41, 246, 143]. The typical situation is free-flowing conditions when the demand flows are below the capacity of the road network. Here, speeds tend to be near the speed limit, the occupancies are low, and vehicle headways are comfortable. In

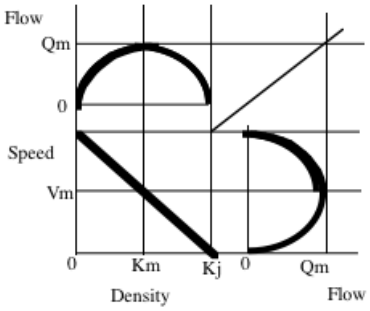


Figure 9.2: Traditional fundamental diagram of traffic: Flow-density and speed-concentration curves. Source: [164].

congested conditions, the actual flows reduce, but the demand flows remain high; the vehicles slow down, the occupancies increase, and vehicles pack more closely together. During congestion the road system is operating in an inefficient manner, with increased vehicle delays, driver frustration, and greater potential for accidents. In addition to these two states, there exist two distinct transition states, where the traffic state changes from free-flowing to congested and from congested to free-flow conditions. These two states may be different from each other in their characteristics. Some properties of the traditional fundamental diagram are the following:

- The variables of flow, density, and space mean speed are related by the definition: $q = k\bar{v_s}$.
- When density on the highway is zero, the flow is also zero because there are no vehicles on the highway.
- As density increases, flow increases.
- When the density reaches a maximum jam density (k_j), flow must be zero because vehicles will line up end to end.
- Flow will also increase to a maximum value (q_m), increases in density beyond that point result in reductions of flow.
- When density is zero, speed is freeflow (v_f). The upper half of the flow curve is uncongested and the lower half is congested.
- The slope of the flow density curve gives speed.

Mesoscopic detail

Mesoscopic modelling usually consists of a simplification that is less demanding of data and more computationally efficient than microscopic models as they operate at the same aggregation scale as the macroscopic models; in this way individuals vehicles are not distinguished nor traced, but its behaviour is specified, e.g. in probabilistic terms.

There are two main approaches for this traffic modelling: those in which individual vehicles are not taken into account because they are grouped in clusters or cells that move along the links, and those in which the modelling is based on simplified dynamics of individual vehicles. In this sense, traffic is represented by small groups of traffic entities, while the activities and interactions of which are described at a low detail level; e.g. a lane-change manoeuvre might be represented as an event, where the decision to perform the action could be based on e.g. relative lane densities, and speed differentials. Other mesoscopic approaches are derived in analogy to gas-kinetic theory [239]. These gas-kinetic models describe the dynamics of velocity distributions.

Another main difference between the mesoscopic approaches lays in the way they deal with time. While the most common approach is based on synchronous timing with a chosen simulation step, other approaches are asynchronous or event based.

Microscopic detail

Microscopic modelling describes both the space-time behaviour of vehicles and drivers as well as their interactions at a high level of detail. The models makes use of characteristics such as vehicle lengths, speeds, accelerations, time and space headways, vehicle and engine capabilities, as well as —occasionally rudimentary— human characteristics that describe the driving behaviour.

The microscopic modelling includes car-following and lane-changing behaviours [103, 42, 211] modelled with optimal velocity models, human behaviourally psychophysiological spacing models, traffic cellular automata models or queueing theory, among other possibilities. All car-following models depend on a number of parameters aimed at mimicking as closely as possible the way in which drivers of follower vehicles adjust their driving to that of leader vehicles. While the increasing number of model parameters could in theory replicate better what is a complex phenomenon that combines components based strictly on the dynamics of the process along with behavioural components, on the other hand, it makes harder to find the right values of these parameters. Thus, the microscopic computational complexity is often a significant disadvantage when compared to meso- or macroscopic models —although there may be some exceptions, e.g. the traffic cellular automata models—. From the point of view of model calibration and validation, this poses an interesting challenge, as in many cases not all parameters are equally influential on the results, requiring, thus, some sensitivity analyses.

As in [28] concludes, car-following models provide reasonable results in uncongested conditions and in some cases in congested conditions as well. But in accordance with the mentioned studies, they fail to provide results of a similar quality in the transitions from uncongested to congested, that is,

when the steady-state hypothesis no longer holds. Maybe one of the reasons could be the lack of empirical evidence of enough quality for these conditions.

There exists two main approaches to feed the traffic demand as input to the simulation:

- The traffic demand input is defined in terms of input flows and turning proportions at intersection and exit sections. In this case there is no intervention of the dynamic traffic assignment (DTA) which was described in the travel demand modelling section. In this case, vehicles travel stochastically in the network, leaving the network occasionally, according to the turning and exit proportions.
- The traffic demand input is defined in terms of OD matrices. Vehicles travel across the network from origins to destinations along the available paths that join them, which has been calculated in the DTA step.

Finally, the most common implementation approach for microscopic models is synchronous timing. In this way at each simulation step, the state of all the network's entities is updated.

Submicroscopic detail

Submicroscopic models describe the characteristics of individual vehicles in the traffic stream, similar to microscopic modelling. However, beyond a detailed description of traveller behaviour, the vehicle control behaviour —e.g. changing gears, engine performance, etc.— in correspondence to prevailing surrounding conditions is modelled in detail. Complementary to the functioning of a vehicle's physical components, submicroscopic models can also describe a human driver's decision taking process in much more detail.

Appendix II: Detailed results for the validation of Adarules under different change scenarios

Adarules: pattern mining using a single-task or a multi-task approach

Figure 9.1 shows the comparison in the **flow** forecasting performance of both pattern mining approches using Adarules: single-task and multi-task. The distribution of $nRMSE$ per detector ($N = 20$) is shown in every time slot of 1 month, along with the median as the line per group.

Figure 9.2 shows the comparison in the **occupancy** forecasting performance of both pattern mining approches using Adarules: single-task and multi-task. The distribution of $nRMSE$ per detector ($N = 20$) is shown in every time slot of 1 month, along with the median as the line per group.

Figure 9.3 shows the comparison in the **speed** forecasting performance of both pattern mining approches using Adarules: single-task and multi-task. The distribution of $nRMSE$ per detector ($N = 20$) is shown in every time slot of 1 month, along with the median as the line per group.

Adarules: forecasting model learning using a single-task or a multi-task approach

Figure 9.4 shows the comparison in the **flow** forecasting performance of both forecasting learning approches using Adarules: single-task and multi-task. The distribution of $nRMSE$ per detector ($N = 20$) is shown in every time slot of 1 month, along with the median as the line per group.

Figure 9.5 shows the comparison in the **occupancy** forecasting performance of both forecasting learning approches using Adarules: single-task and multi-task. The distribution of $nRMSE$ per detector ($N = 20$) is shown in every time slot of 1 month, along with the median as the line per group.

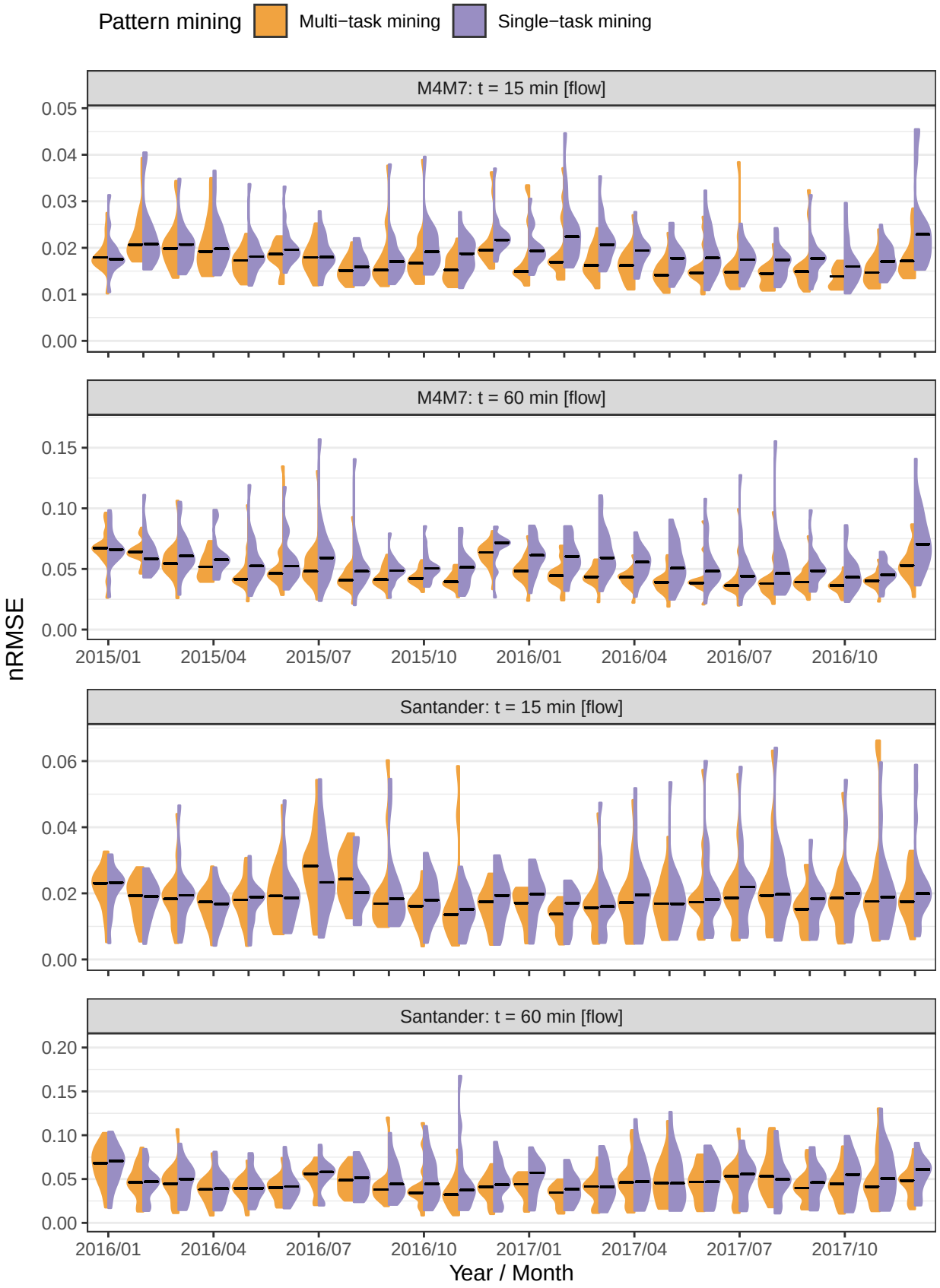


Figure 9.1: Comparison in the **flow** forecasting performance of both pattern mining approches using Adarules: single-task and multi-task. The distribution of $nRMSE$ per detector ($N = 20$) is shown in every time slot of 1 month, along with the median as the line per group.

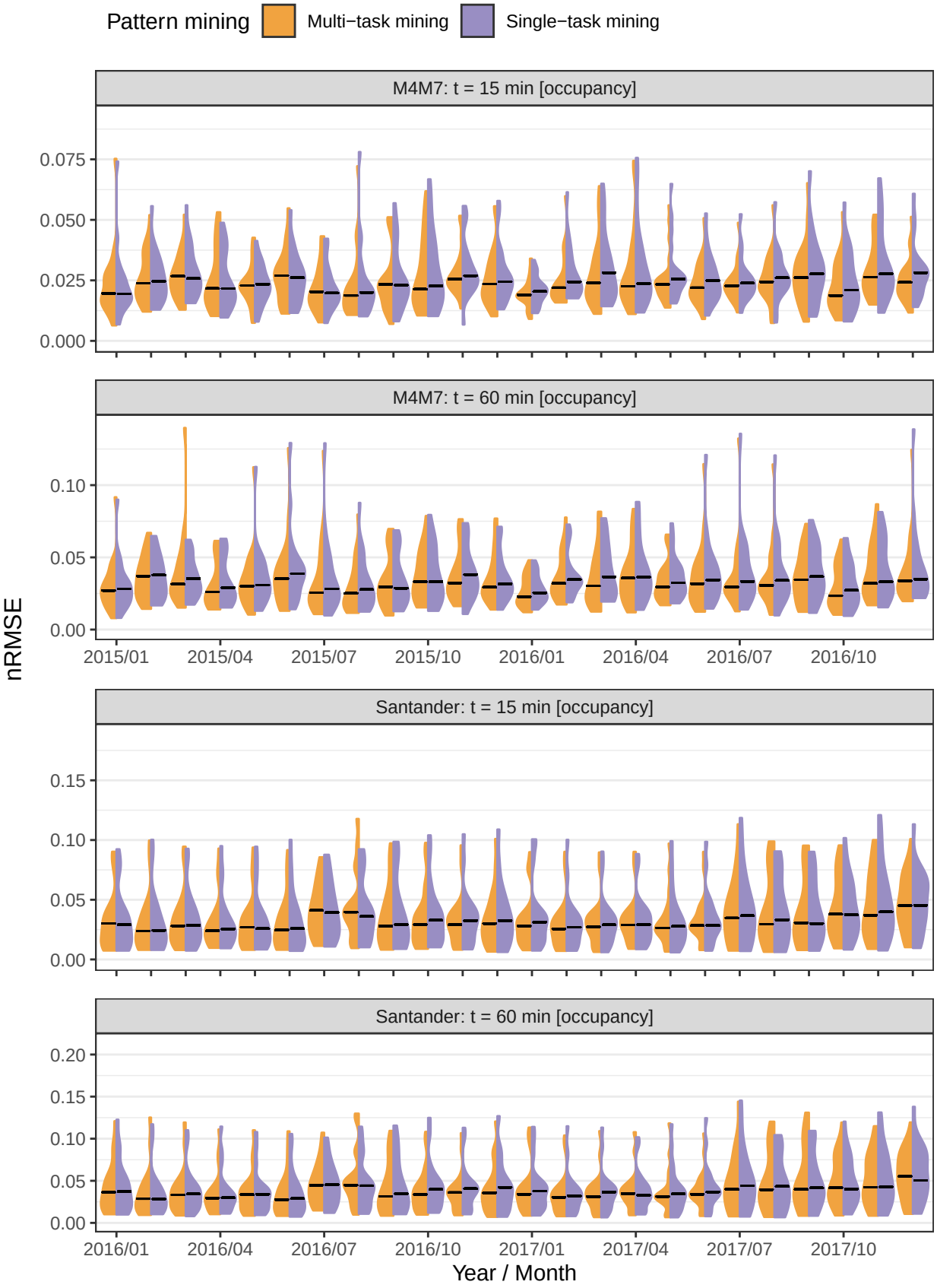


Figure 9.2: Comparison in the **occupancy** forecasting performance of both pattern mining approches using Adarules: single-task and multi-task. The distribution of $nRMSE$ per detector ($N = 20$) is shown in every time slot of 1 month, along with the median as the line per group.

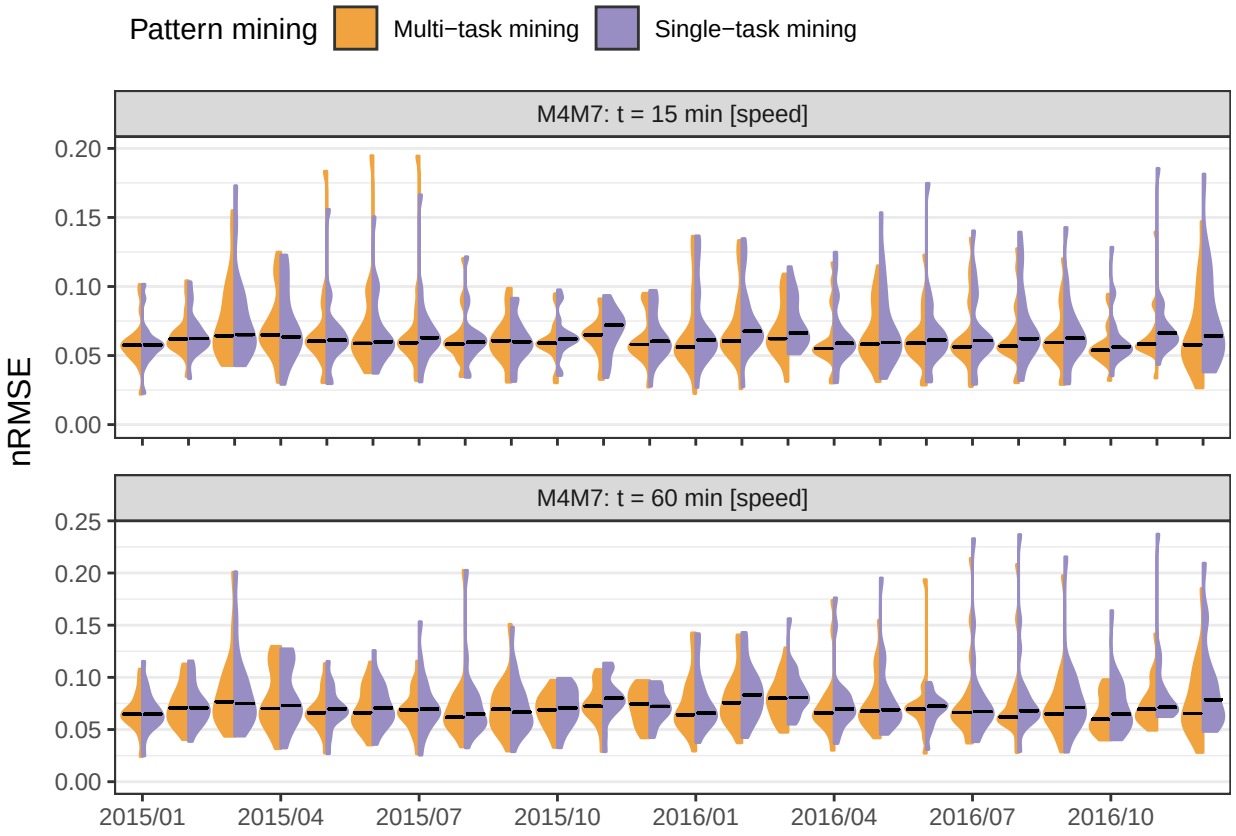


Figure 9.3: Comparison in the **speed** forecasting performance of both pattern mining approches using Adarules: single-task and multi-task. The distribution of $nRMSE$ per detector ($N = 20$) is shown in every time slot of 1 month, along with the median as the line per group.

Figure 9.6 shows the comparison in the **speed** forecasting performance of both forecasting learning approches using Adarules: single-task and multi-task. The distribution of $nRMSE$ per detector ($N = 20$) is shown in every time slot of 1 month, along with the median as the line per group.

Adarules vs baselines: Real data scenario

Figure 9.7 shows the comparison in the **flow** forecasting performance between Adarules and baselines in the *real-data* scenario. The distribution of $nRMSE$ per detector ($N = 20$) is shown in every time slot of 1 month, along with the median as the line per group.

Figure 9.8 shows the comparison in the **occupancy** forecasting performance between Adarules and baselines in the *real-data* scenario. The distribution of $nRMSE$ per detector ($N = 20$) is shown in every time slot of 1 month, along with the median as the line per group.

Figure 9.9 shows the comparison in the **speed** forecasting performance between Adarules and baselines in the *real-data* scenario. The distribution of $nRMSE$ per detector ($N = 20$) is shown in every time slot of 1 month, along with the median as the line per group.

Adarules vs baselines: *Zero* drift scenario

Figure 9.10 shows the comparison in the **flow** forecasting performance between Adarules and baselines in the *zero drift* scenario. The distribution of $nRMSE$ per detector ($N = 20$) is shown in every time slot of 1 month, along with the median as the line per group.

Figure 9.11 shows the comparison in the **occupancy** forecasting performance between Adarules and baselines in the *zero drift* scenario. The distribution of $nRMSE$ per detector ($N = 20$) is shown in every time slot of 1 month, along with the median as the line per group.

Figure 9.12 shows the comparison in the **speed** forecasting performance between Adarules and baselines in the *zero drift* scenario. The distribution of $nRMSE$ per detector ($N = 20$) is shown in every time slot of 1 month, along with the median as the line per group.

Adarules vs baselines: Gradual change scenario

Figure 9.13 shows the comparison in the **flow** forecasting performance between Adarules and baselines in the *gradual change* scenario. The distribution of $nRMSE$ per detector ($N = 20$) is shown in every time slot of 1 month, along with the median as the line per group.

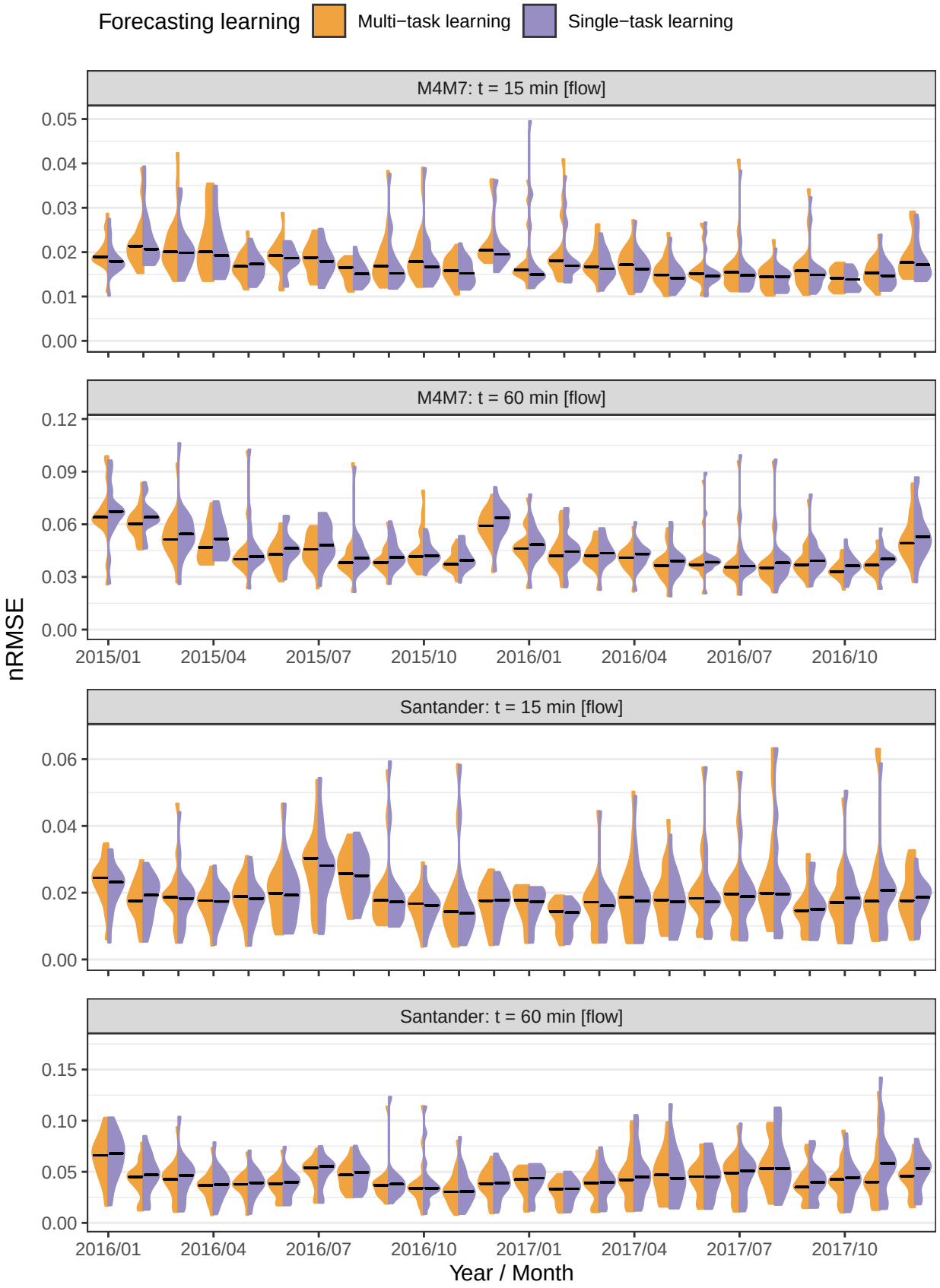


Figure 9.4: Comparison in the **flow** forecasting performance of both forecasting learning approches using Adarules: single-task and multi-task. The distribution of $nRMSE$ per detector ($N = 20$) is shown in every time slot of 1 month, along with the median as the line per group.

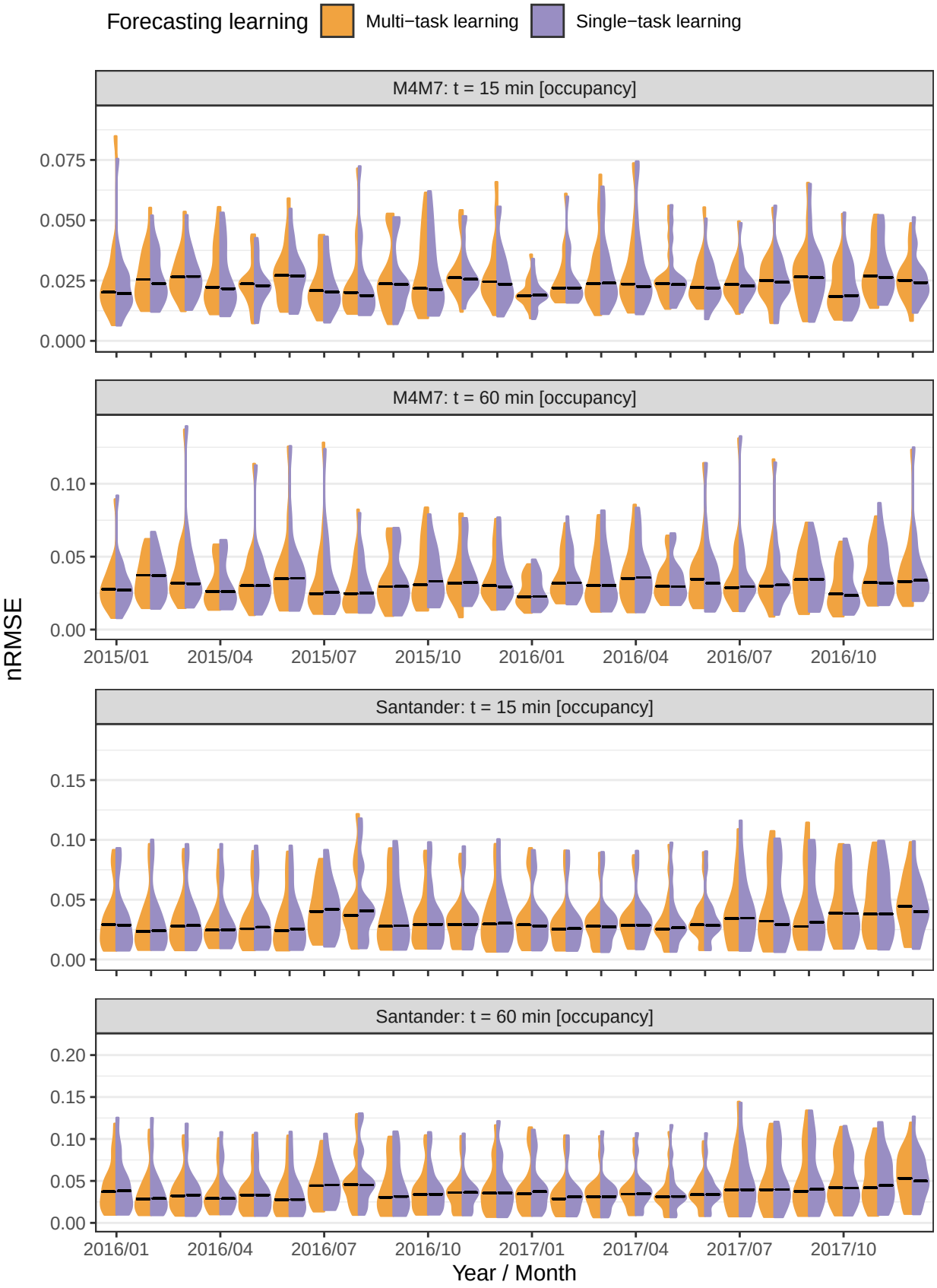


Figure 9.5: Comparison in the **occupancy** forecasting performance of both forecasting learning approches using Adarules: single-task and multi-task. The distribution of $nRMSE$ per detector ($N = 20$) is shown in every time slot of 1 month, along with the median as the line per group.

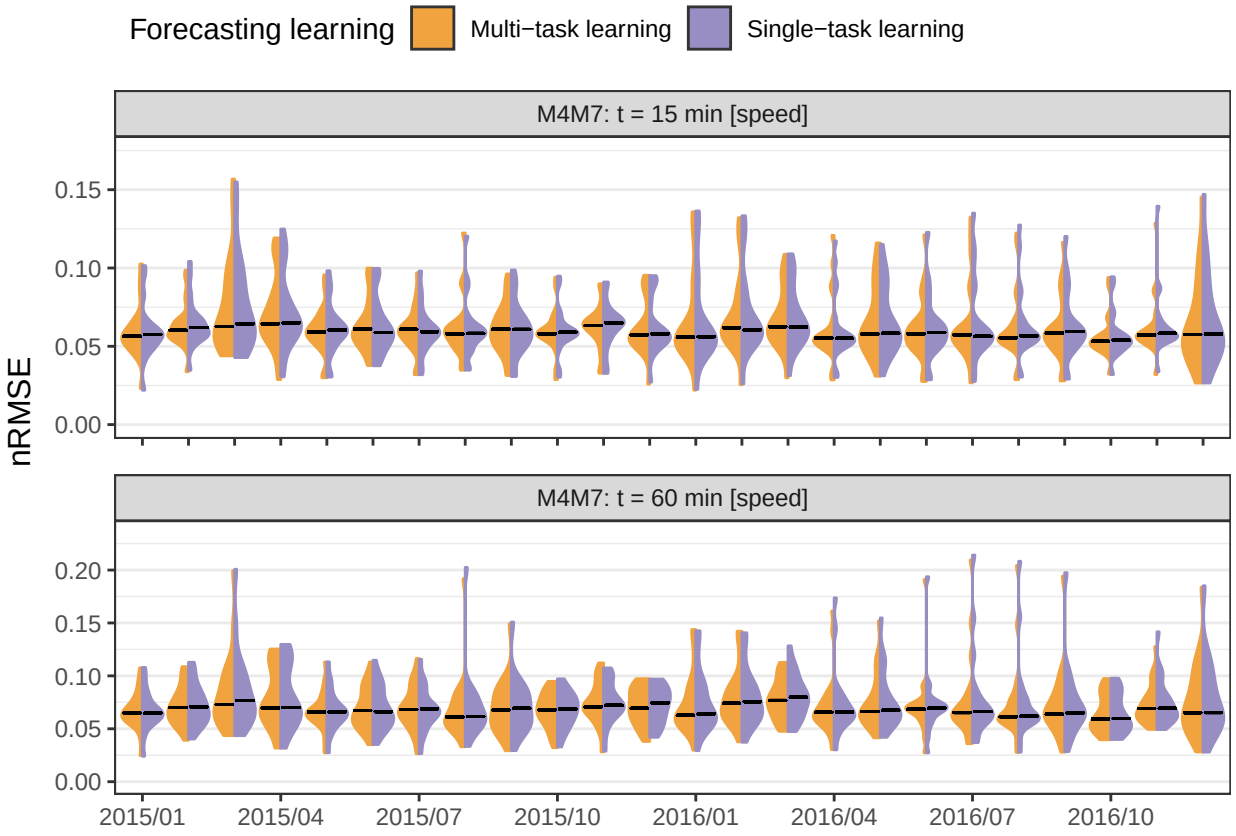


Figure 9.6: Comparison in the **speed** forecasting performance of both forecasting learning approches using Adarules: single-task and multi-task. The distribution of $nRMSE$ per detector ($N = 20$) is shown in every time slot of 1 month, along with the median as the line per group.

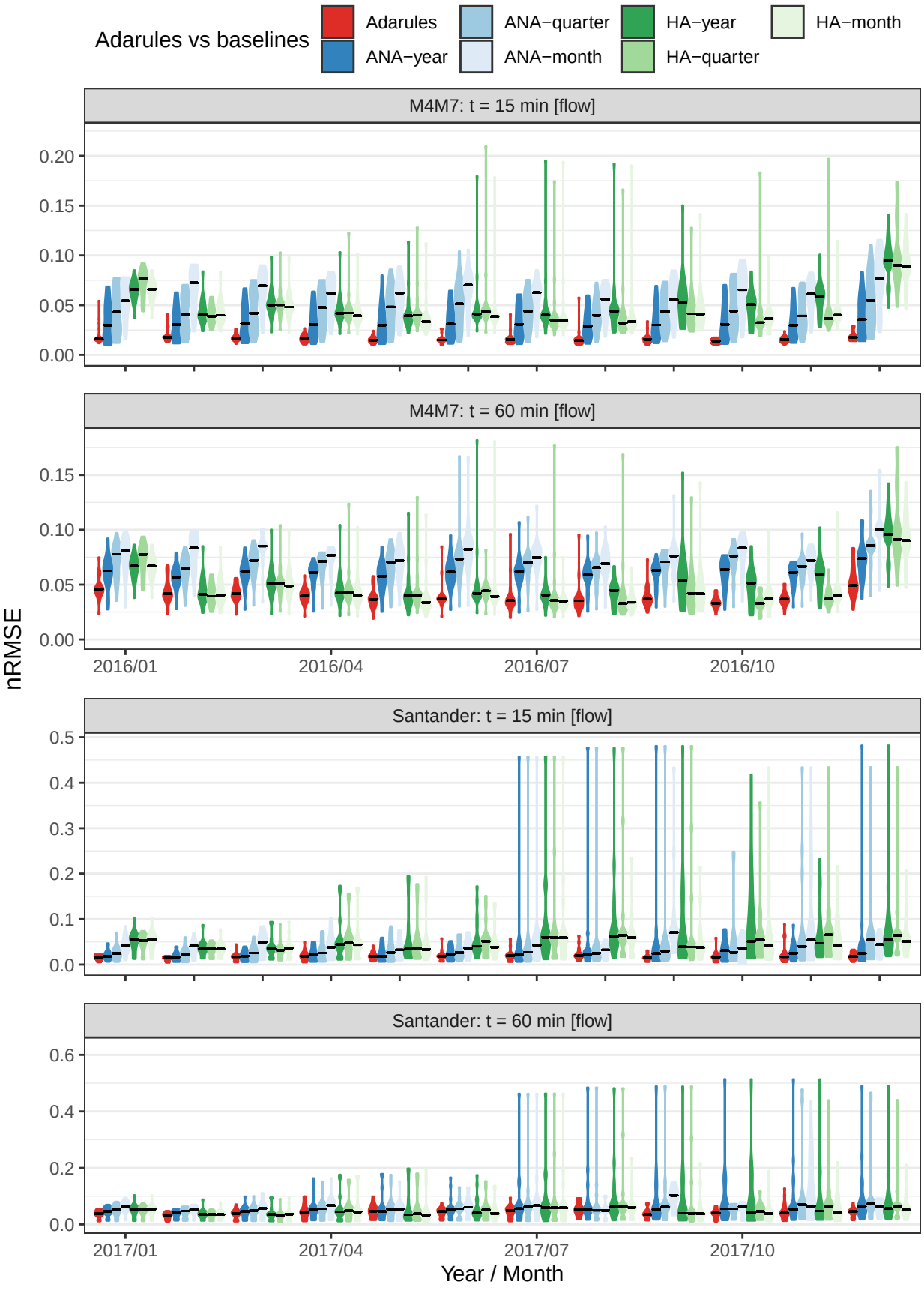


Figure 9.7: Comparison in the **flow** forecasting performance between Adarules and baselines in the *real-data* scenario. The distribution of $nRMSE$ per detector ($N = 20$) is shown in every time slot of 1 month, along with the median as the line per group.

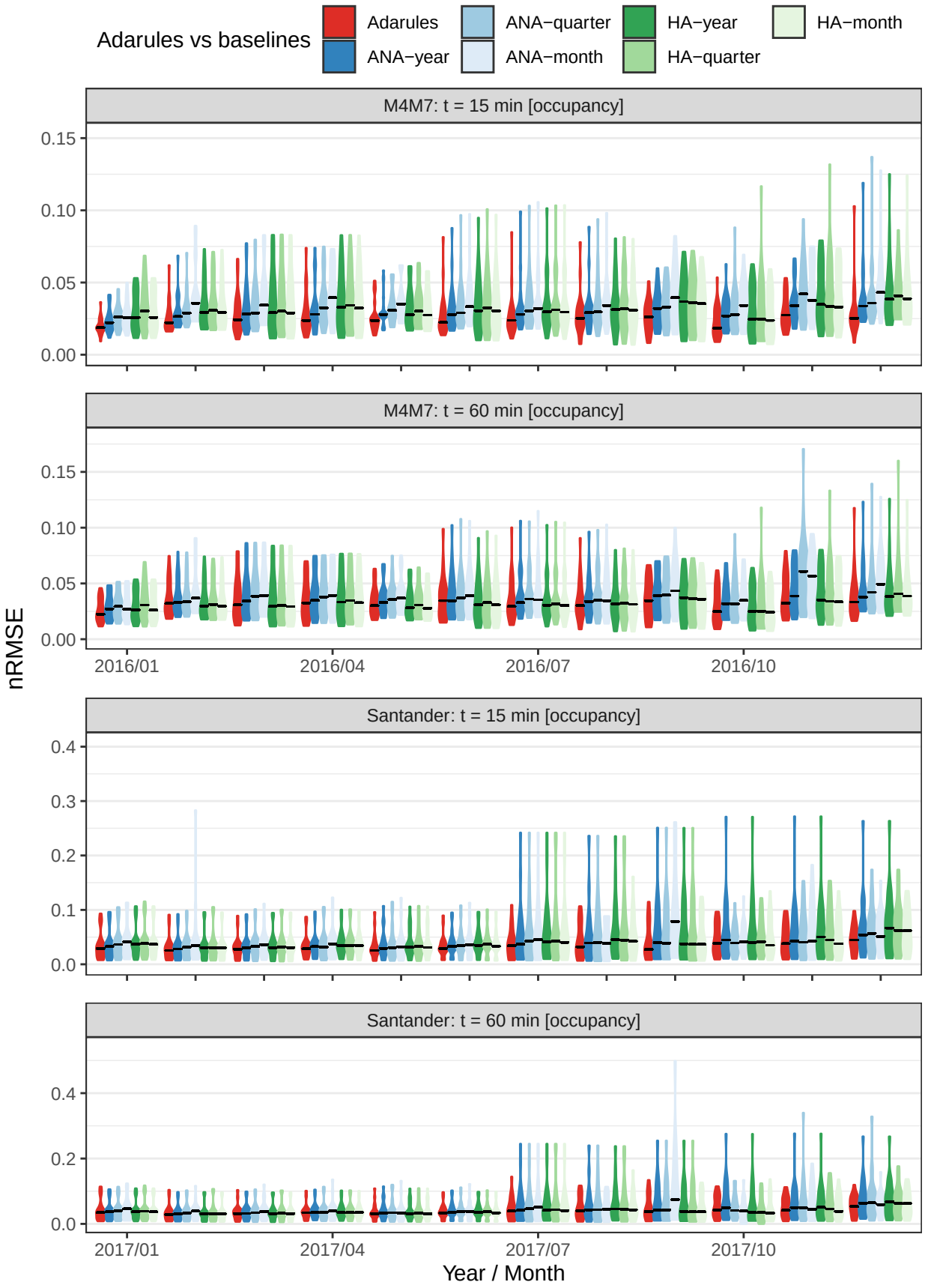


Figure 9.8: Comparison in the **occupancy** forecasting performance between Adarules and baselines in the *real-data* scenario. The distribution of $nRMSE$ per detector ($N = 20$) is shown in every time slot of 1 month, along with the median as the line per group.

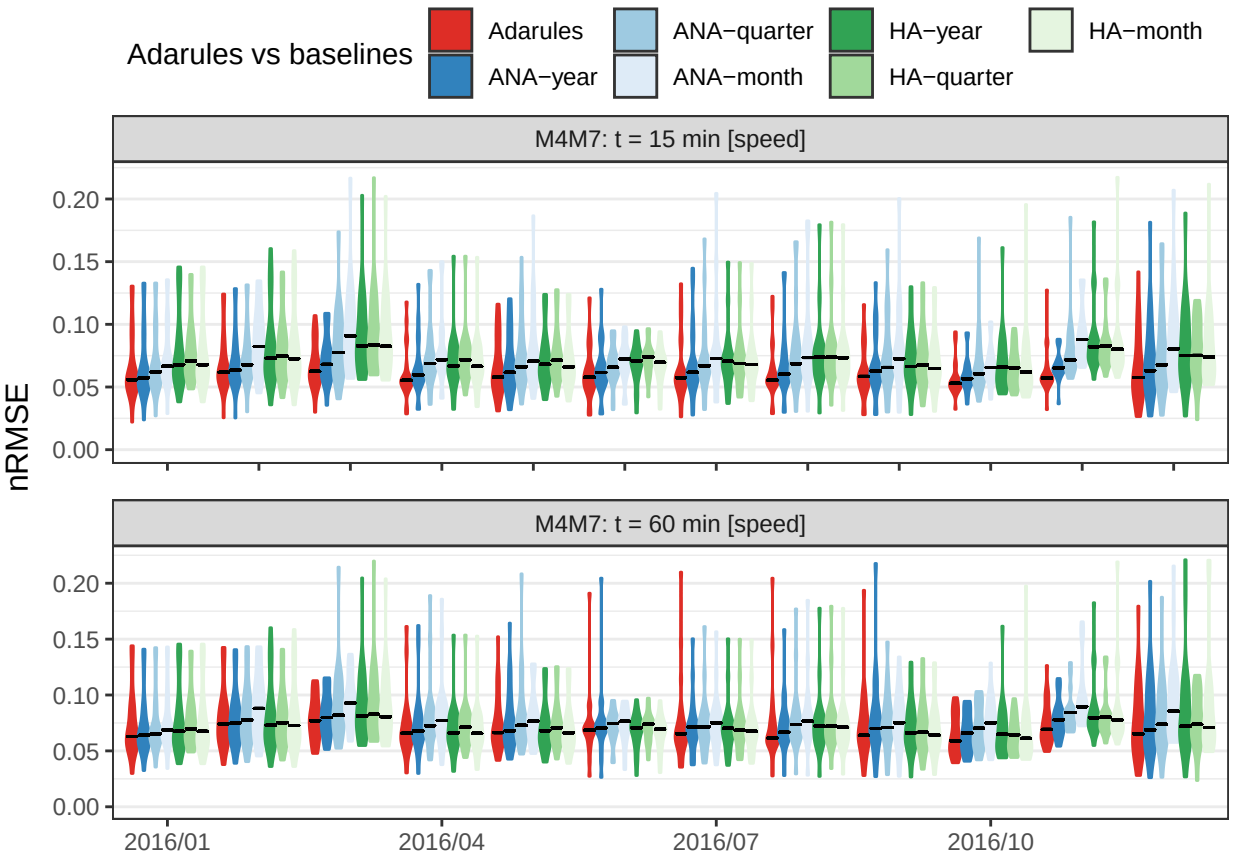


Figure 9.9: Comparison in the **speed** forecasting performance between Adarules and baselines in the *real-data* scenario. The distribution of $nRMSE$ per detector ($N = 20$) is shown in every time slot of 1 month, along with the median as the line per group.

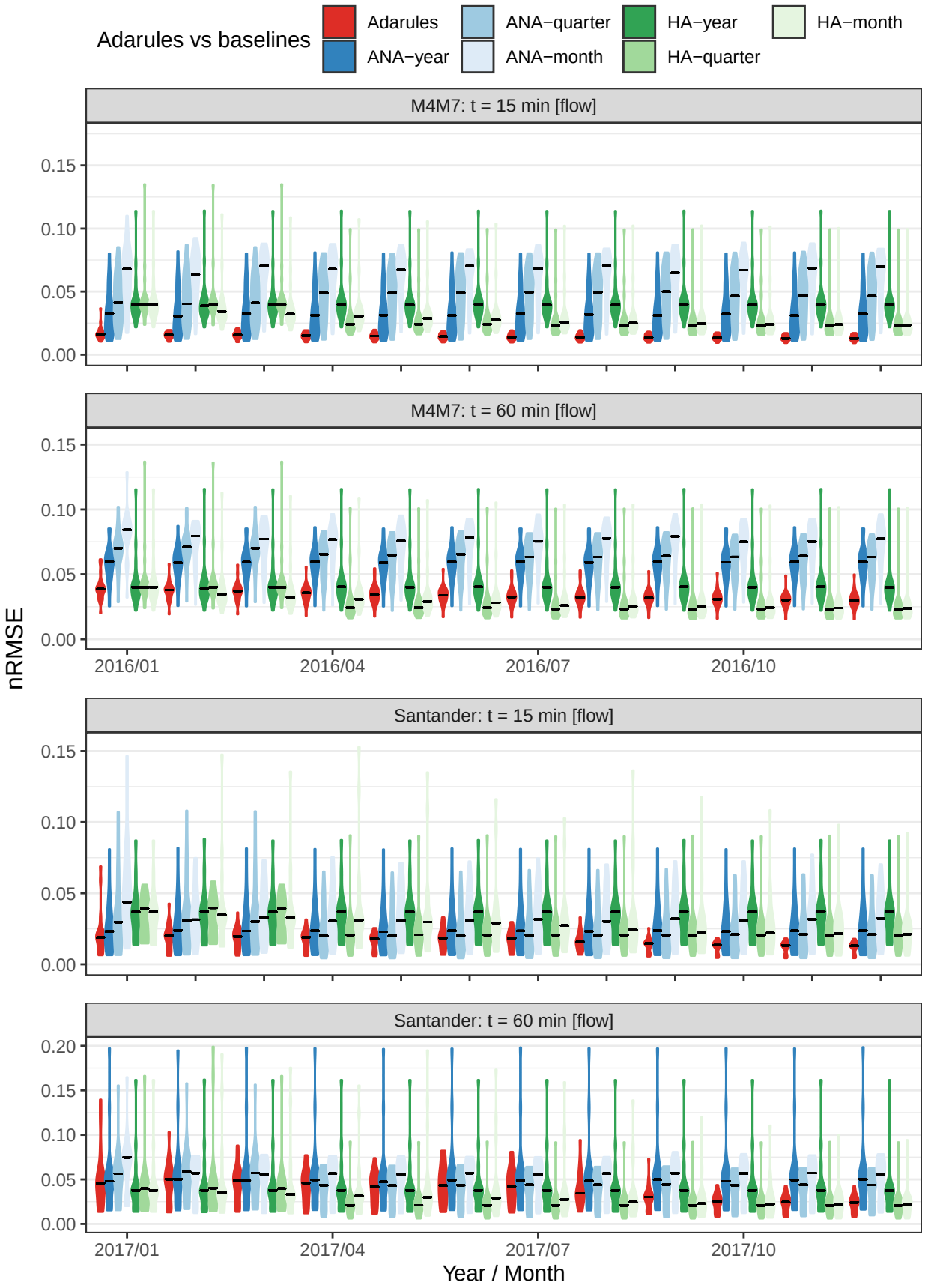


Figure 9.10: Comparison in the **flow** forecasting performance between Adarules and baselines in the *zero drift* scenario. The distribution of $nRMSE$ per detector ($N = 20$) is shown in every time slot of 1 month, along with the median as the line per group.

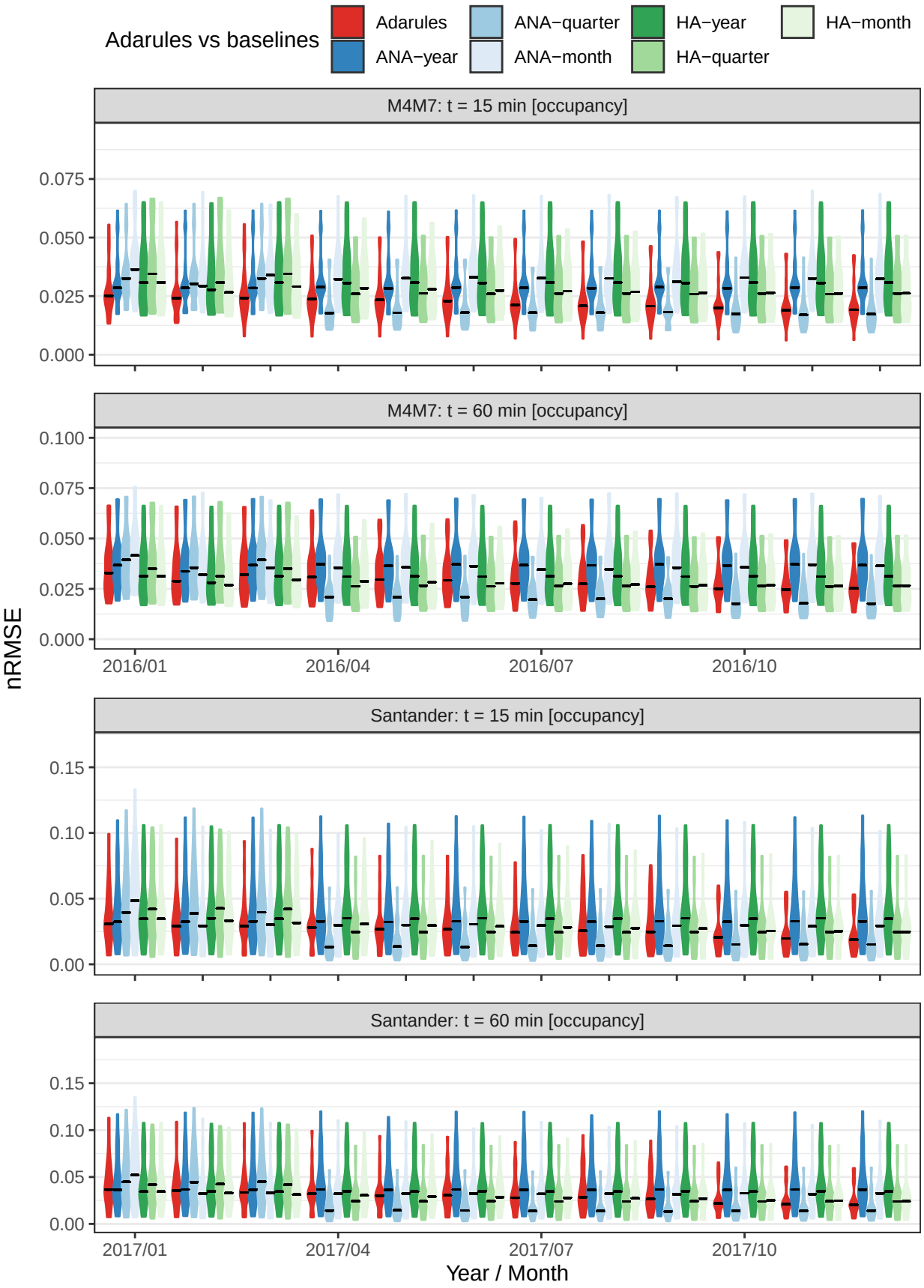


Figure 9.11: Comparison in the **occupancy** forecasting performance between Adarules and baselines in the *zero drift* scenario. The distribution of $nRMSE$ per detector ($N = 20$) is shown in every time slot of 1 month, along with the median as the line per group.

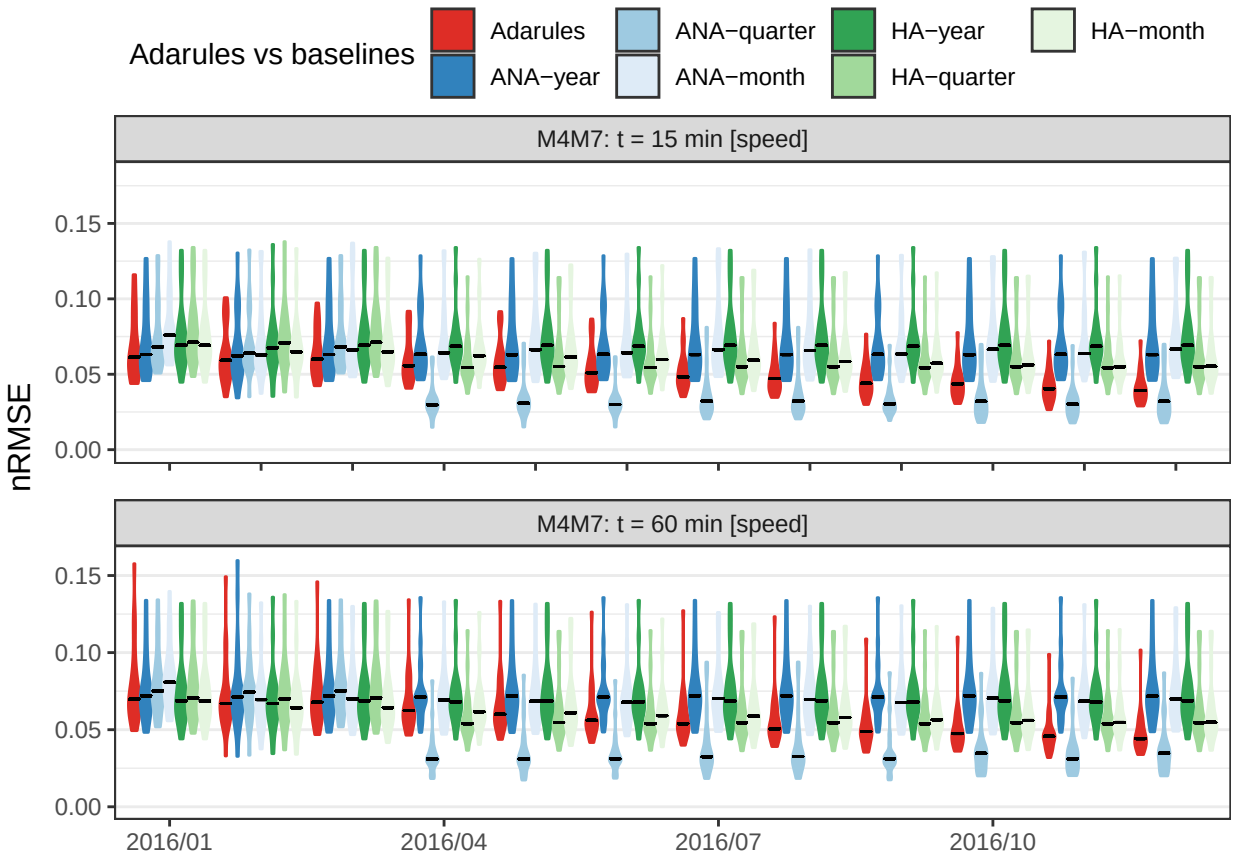


Figure 9.12: Comparison in the **speed** forecasting performance between Adarules and baselines in the *zero drift* scenario. The distribution of $nRMSE$ per detector ($N = 20$) is shown in every time slot of 1 month, along with the median as the line per group.

Figure 9.14 shows the comparison in the **occupancy** forecasting performance between Adarules and baselines in the *gradual change* scenario. The distribution of $nRMSE$ per detector ($N = 20$) is shown in every time slot of 1 month, along with the median as the line per group.

Figure 9.15 shows the comparison in the **speed** forecasting performance between Adarules and baselines in the *gradual change* scenario. The distribution of $nRMSE$ per detector ($N = 20$) is shown in every time slot of 1 month, along with the median as the line per group.

Adarules vs baselines: Abrupt change (AM-PM) scenario

Figure 9.16 shows the comparison in the **flow** forecasting performance between Adarules and baselines in the *abrupt change (AM-PM)* scenario. The distribution of $nRMSE$ per detector ($N = 20$) is shown in every time slot of 1 month, along with the median as the line per group.

Figure 9.17 shwos the comparison in the **occupancy** forecasting performance between Adarules and baselines in the *abrupt change (AM-PM)* scenario. The distribution of $nRMSE$ per detector ($N = 20$) is shown in every time slot of 1 month, along with the median as the line per group.

Figure 9.18 shows the comparison in the **speed** forecasting performance between Adarules and baselines in the *abrupt change (AM-PM)* scenario. The distribution of $nRMSE$ per detector ($N = 20$) is shown in every time slot of 1 month, along with the median as the line per group.

Adarules vs baselines: Abrupt change (IDs) scenario

Figure 9.19 shows the comparison in the **flow** forecasting performance between Adarules and baselines in the *abrupt change (IDs)* scenario. The distribution of $nRMSE$ per detector ($N = 20$) is shown in every time slot of 1 month, along with the median as the line per group.

Figure 9.20 shows the comparison in the **occupancy** forecasting performance between Adarules and baselines in the *abrupt change (IDs)* scenario. The distribution of $nRMSE$ per detector ($N = 20$) is shown in every time slot of 1 month, along with the median as the line per group.

Figure 9.21 shows the comparison in the **speed** forecasting performance between Adarules and baselines in the *abrupt change (IDs)* scenario. The distribution of $nRMSE$ per detector ($N = 20$) is shown in every time slot of 1 month, along with the median as the line per group.

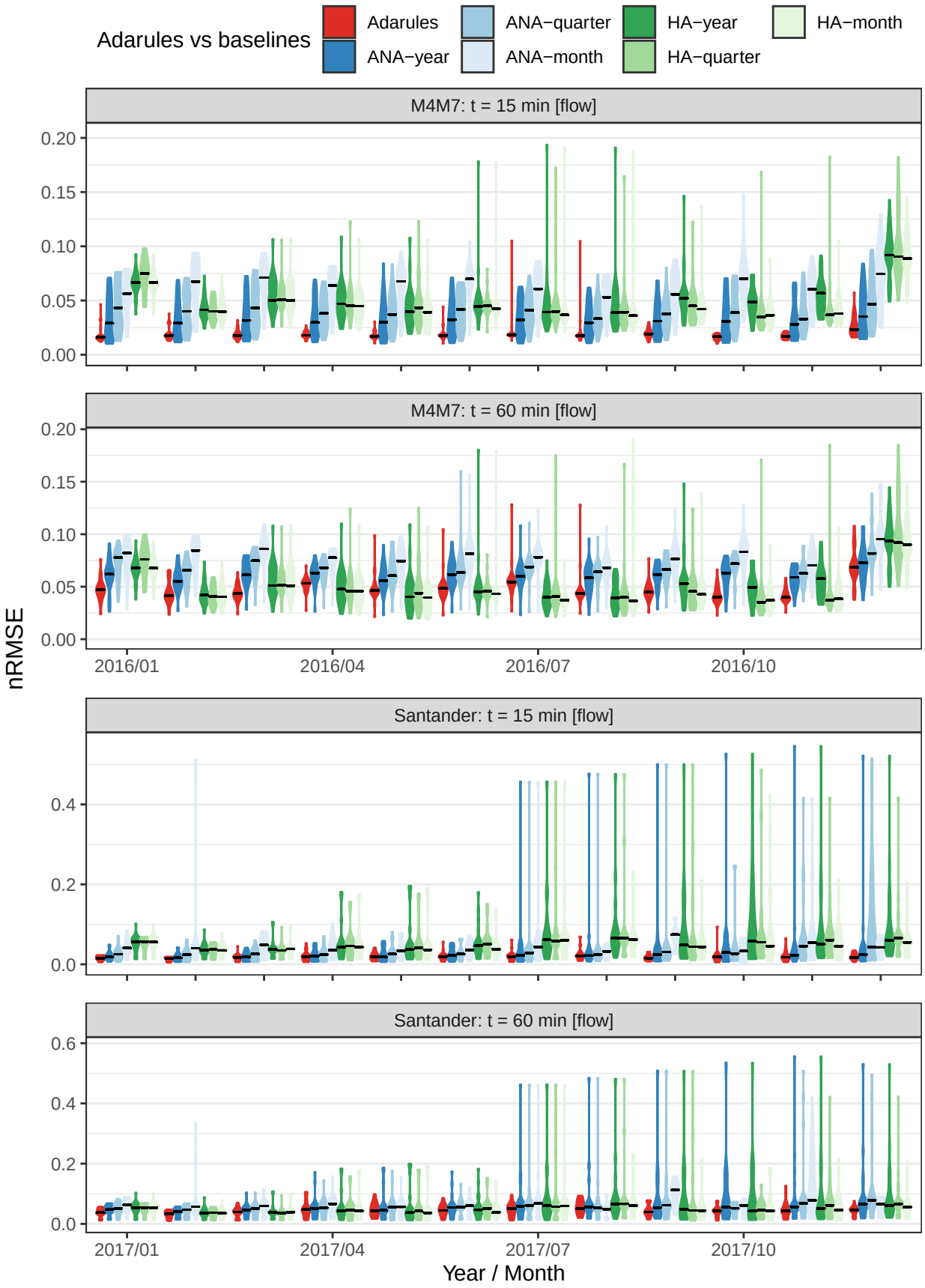


Figure 9.13: Comparison in the **flow** forecasting performance between Adarules and baselines in the *gradual change* scenario. The distribution of $nRMSE$ per detector ($N = 20$) is shown in every time slot of 1 month, along with the median as the line per group.

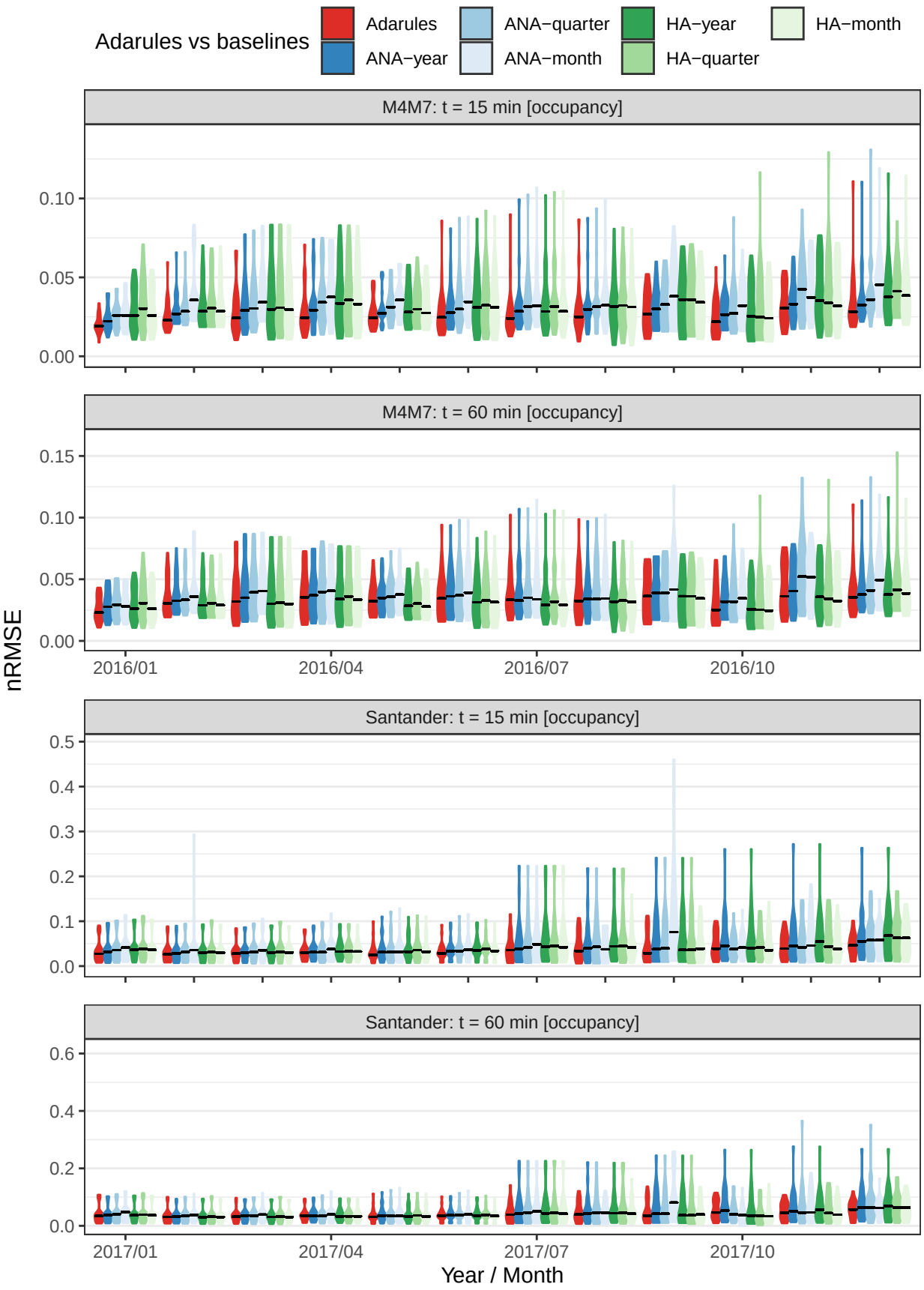


Figure 9.14: Comparison in the **occupancy** forecasting performance between Adarules and baselines in the *gradual change* scenario. The distribution of $nRMSE$ per detector ($N = 20$) is shown in every time slot of 1 month, along with the median as the line per group.

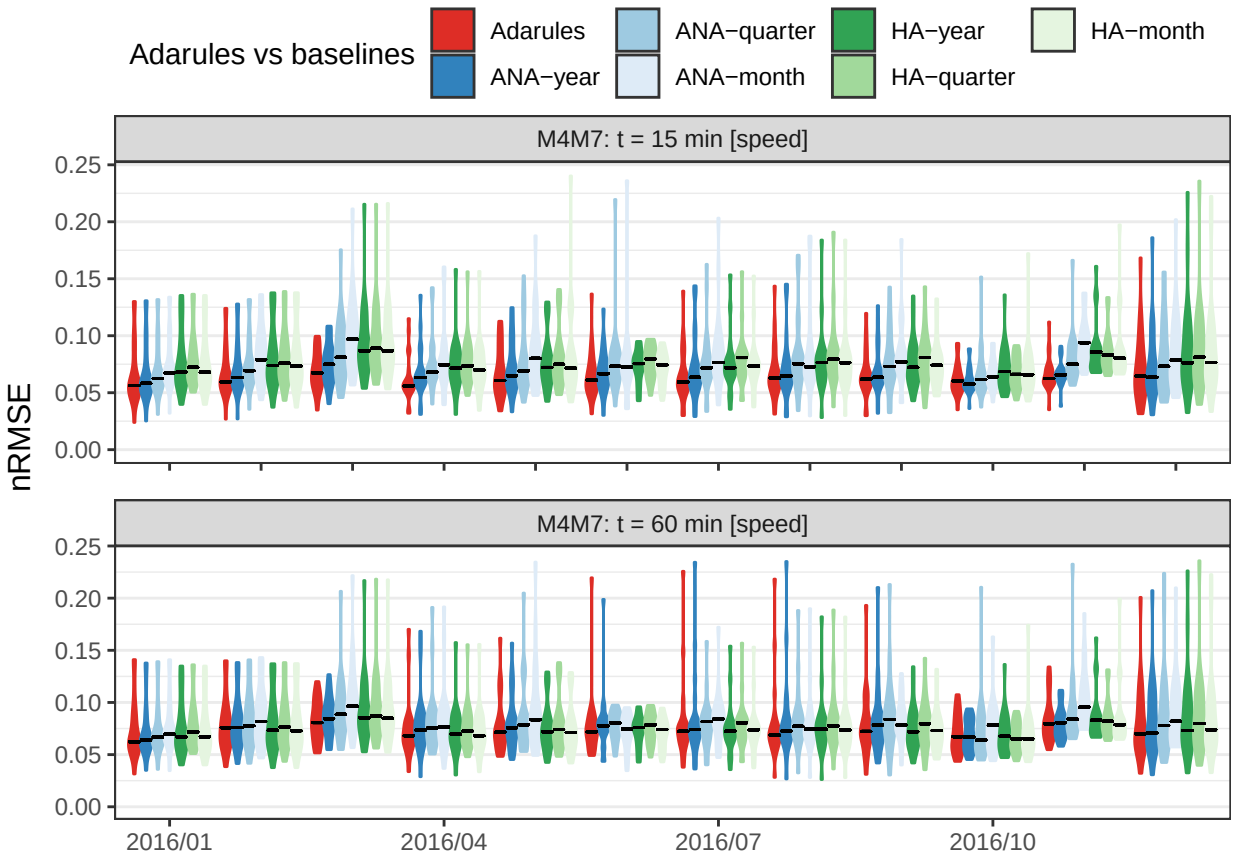


Figure 9.15: Comparison in the **speed** forecasting performance between Adarules and baselines in the *gradual change* scenario. The distribution of $nRMSE$ per detector ($N = 20$) is shown in every time slot of 1 month, along with the median as the line per group.

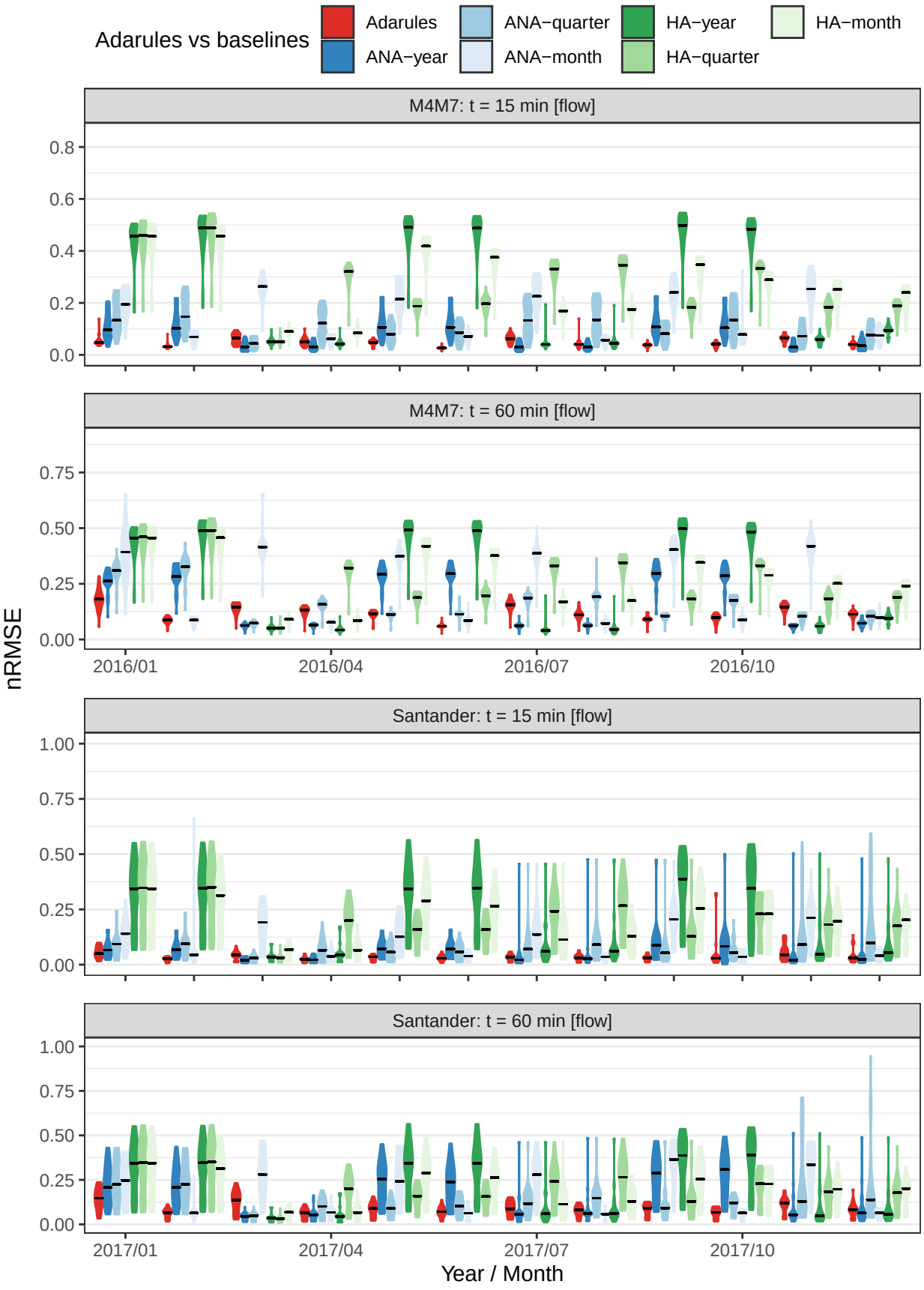


Figure 9.16: Comparison in the **flow** forecasting performance between Adarules and baselines in the *abrupt change (AM-PM)* scenario. The distribution of $nRMSE$ per detector ($N = 20$) is shown in every time slot of 1 month, along with the median as the line per group.

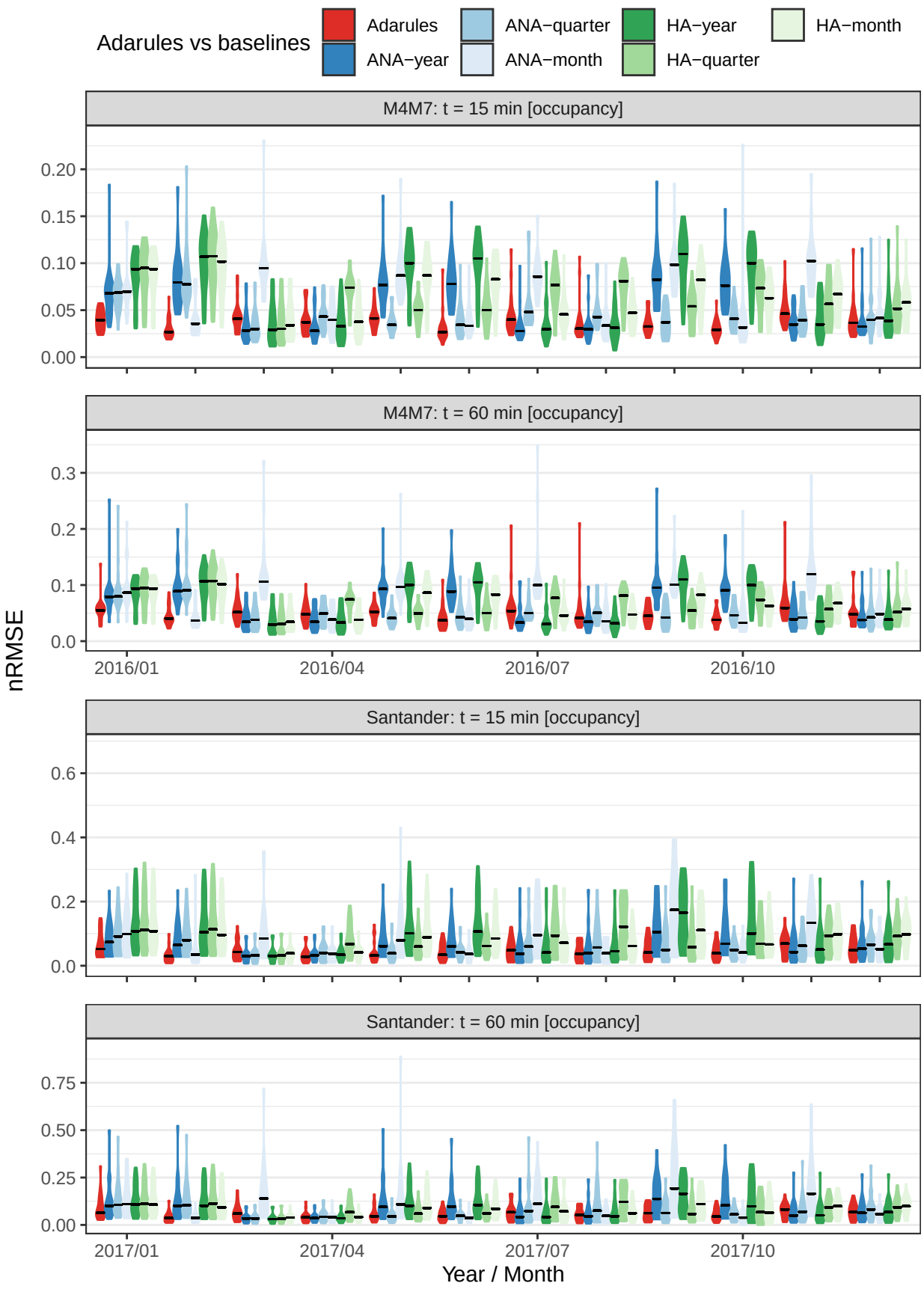


Figure 9.17: Comparison in the **occupancy** forecasting performance between Adarules and baselines in the *abrupt change (AM-PM)* scenario. The distribution of $nRMSE$ per detector ($N = 20$) is shown in every time slot of 1 month, along with the median as the line per group.

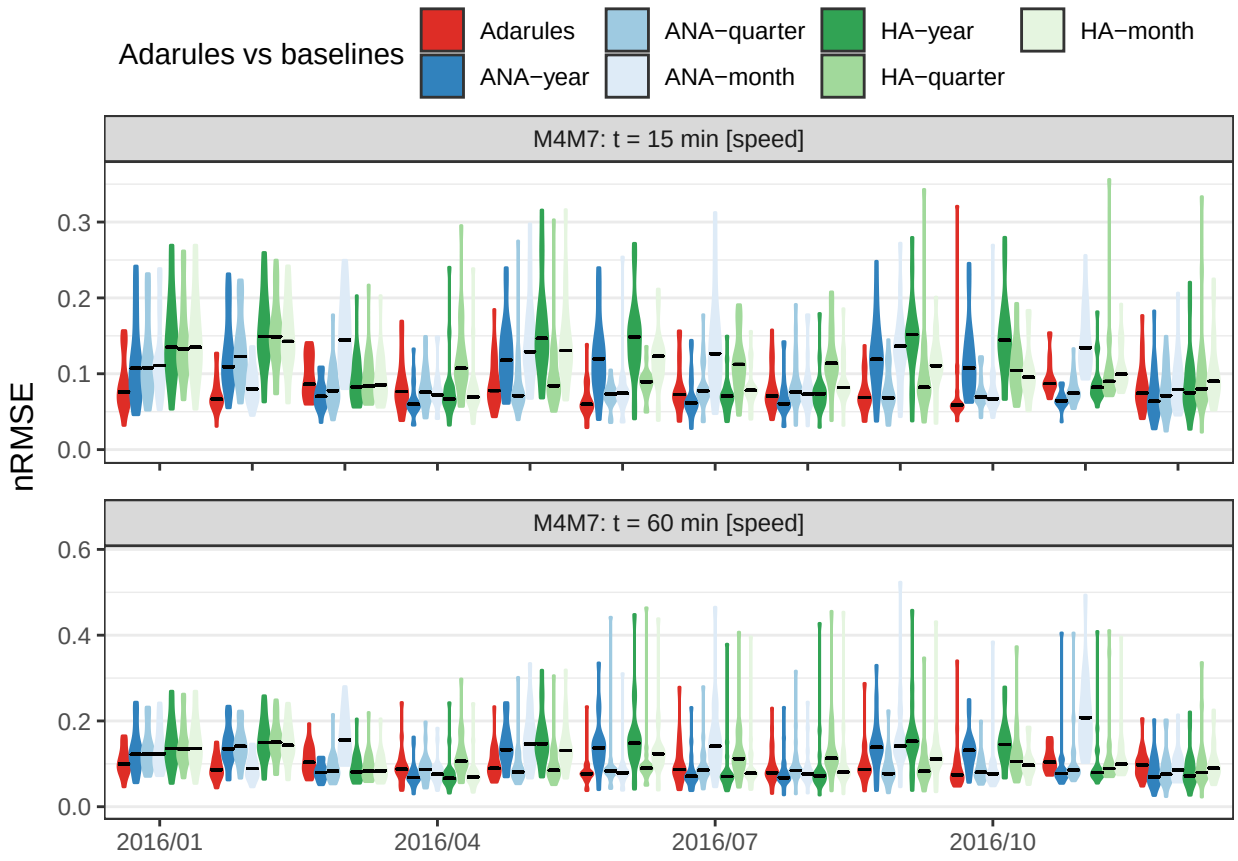


Figure 9.18: Comparison in the **speed** forecasting performance between Adarules and baselines in the *abrupt change (AM-PM)* scenario. The distribution of $nRMSE$ per detector ($N = 20$) is shown in every time slot of 1 month, along with the median as the line per group.

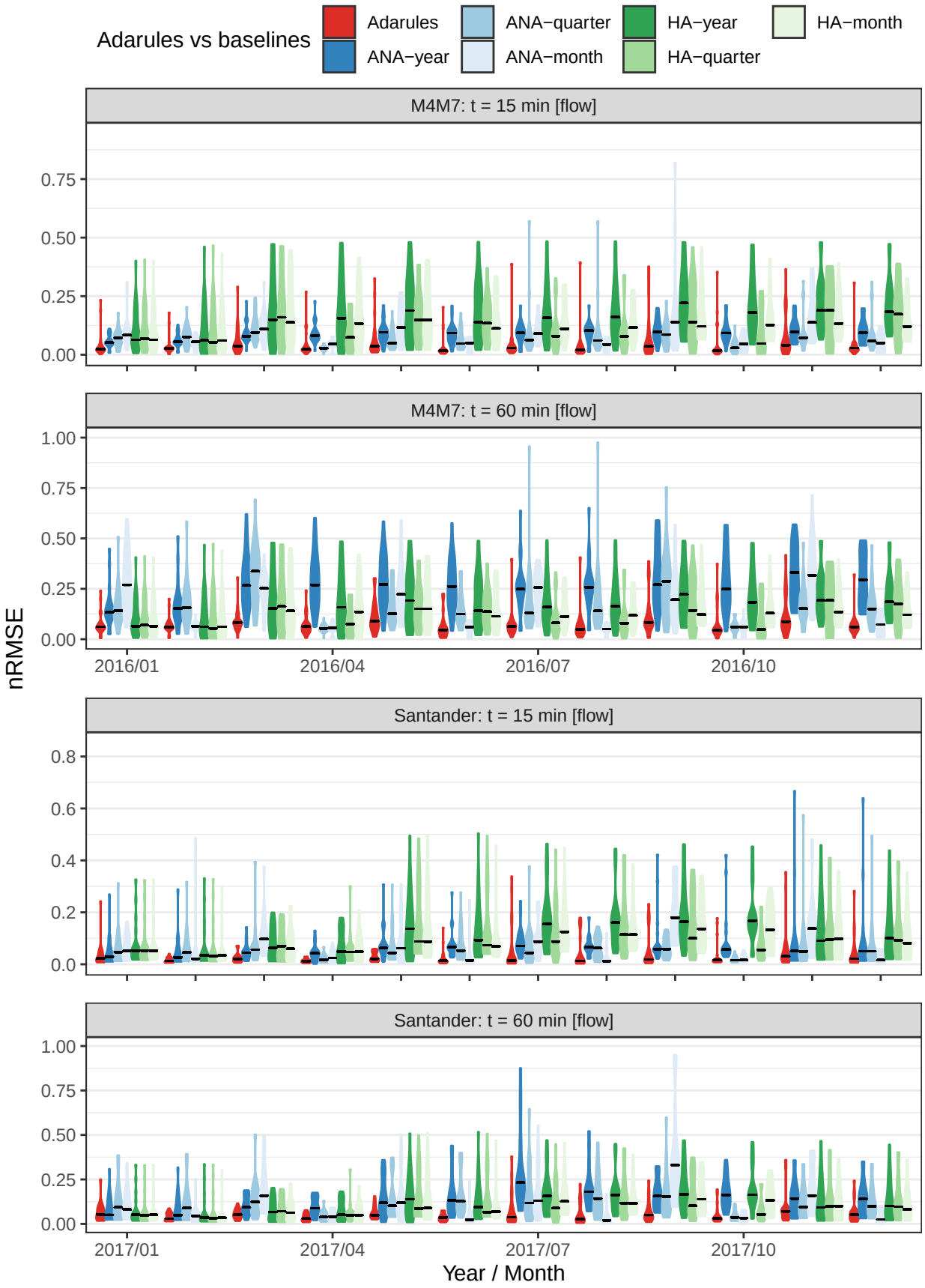


Figure 9.19: Comparison in the **flow** forecasting performance between Adarules and baselines in the *abrupt change (IDs)* scenario. The distribution of $nRMSE$ per detector ($N = 20$) is shown in every time slot of 1 month, along with the median as the line per group.

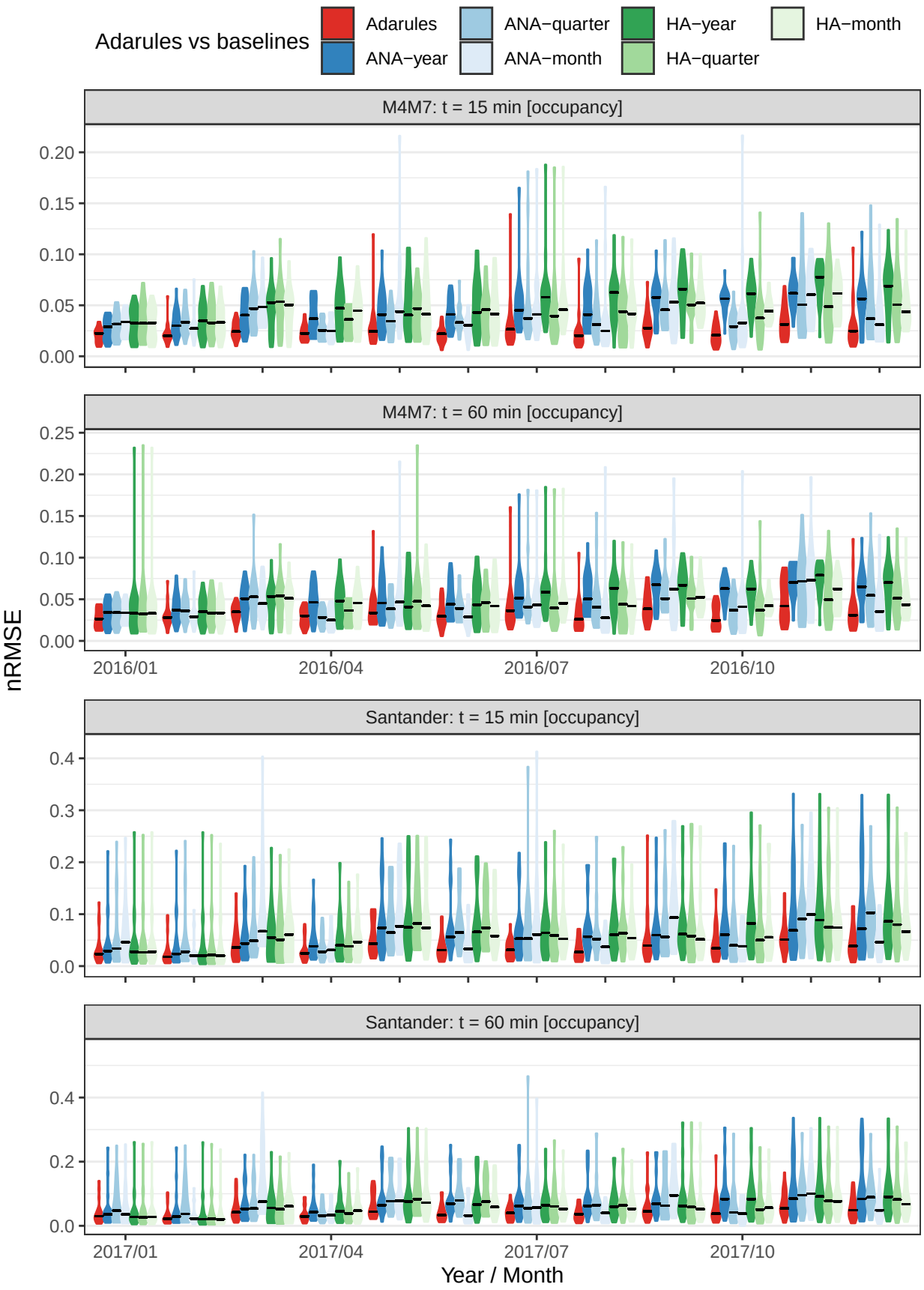


Figure 9.20: Comparison in the **occupancy** forecasting performance between Adarules and baselines in the *abrupt change (IDs)* scenario. The distribution of $nRMSE$ per detector ($N = 20$) is shown in every time slot of 1 month, along with the median as the line per group.

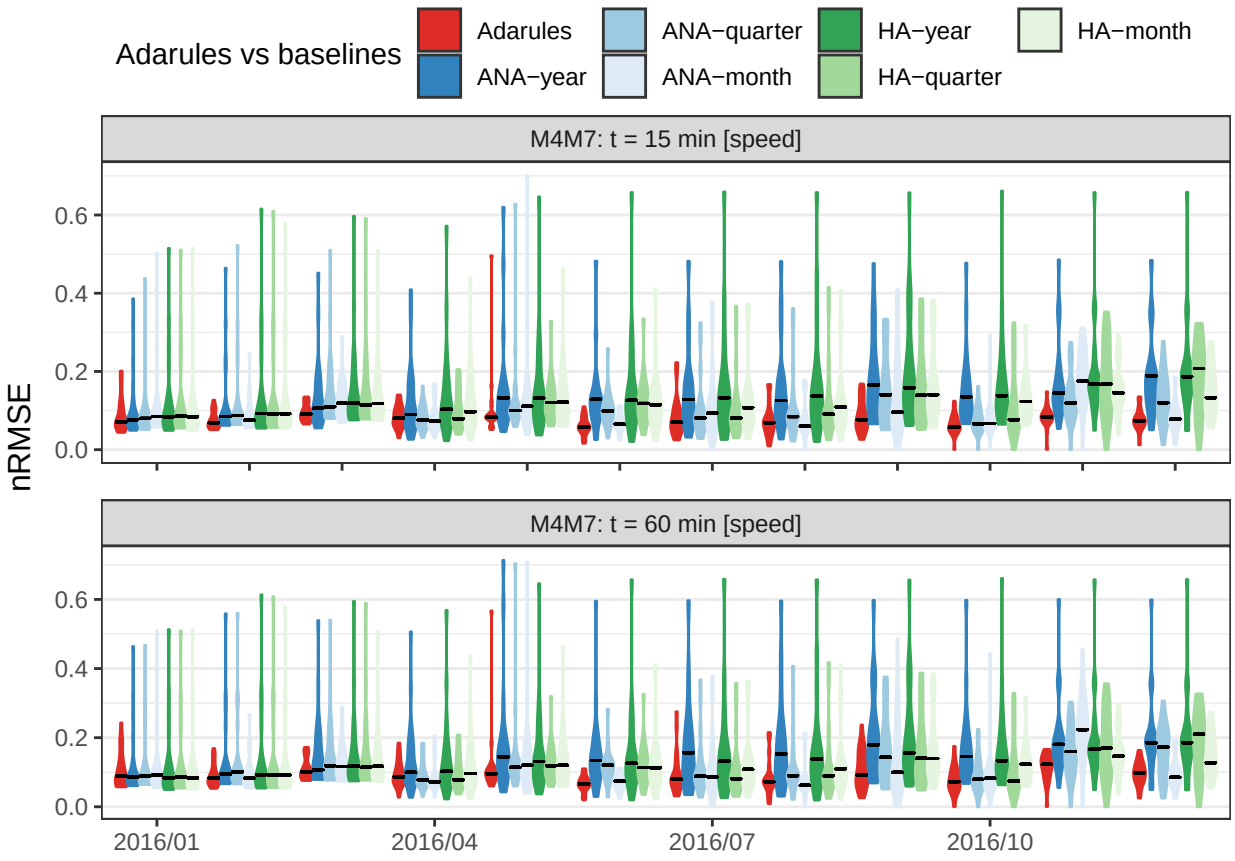


Figure 9.21: Comparison in the **speed** forecasting performance between Adarules and baselines in the *abrupt change (IDs)* scenario. The distribution of $nRMSE$ per detector ($N = 20$) is shown in every time slot of 1 month, along with the median as the line per group.

Bibliography

[1] A. Abadi, T. Rajabioun, and P. A. Ioannou. "Traffic Flow Prediction for Road Transportation Networks With Limited Traffic Data". In: *IEEE Transactions on Intelligent Transportation Systems* 16.2 (Apr. 2015), pp. 653–662. ISSN: 1524-9050. DOI: 10.1109/TITS.2014.2337238.

[2] Baher Abdulhai, Rob Pringle, and Grigoris J Karakoulas. "Reinforcement learning for true adaptive traffic signal control". In: *Journal of Transportation Engineering* 129.3 (2003), pp. 278–285.

[3] J. L Adler. "Investigating the learning effects of route guidance and traffic advisories on route choice behavior". In: *Transportation Research Part C: Emerging Technologies* 9.1 (Feb. 2001), pp. 1–14. ISSN: 0968-090X. DOI: 10.1016/S0968-090X(00)00002-4. URL: http://www.sciencedirect.com/science/article/pii/S0968090X00000024 (visited on 12/11/2015).

[4] Mohammed S. Ahmed and Allen R. Cook. *Analysis of freeway traffic time-series data by using Box-Jenkins techniques*. 722. 1979. URL: http://trid.trb.org/view.aspx?id=148123 (visited on 01/16/2016).

[5] Aimsun, SL. *Aimsun*. 2018. URL: https://www.aimsun.com/ (visited on 02/19/2018).

[6] Aimsun, SL. *Gold Coast predictive solutions trial*. URL: https://www.aimsun.com/gold-coast-predictive-solutions-trial/ (visited on 10/22/2018).

[7] Aimsun, SL. *Madrid: M30 bypass and tunnels*. URL: https://www.aimsun.com/madrid/ (visited on 10/22/2018).

[8] Aimsun, SL. *RMS selects Tyco and Aimsun Live for Sydney M4 Smart Motorway deployment*. URL: https://www.aimsun.com/sydney-m4-smart-motorway-deployment/ (visited on 10/22/2018).

[9] Aimsun, SL. *San Diego: Integrated Corridor Management System*. URL: https://www.aimsun.com/integrated-corridor-management-project-in-san-diego/ (visited on 10/22/2018).

[10] Aimsun, SL. *The Opticities Project, TEC - ITS France, September 2015*. URL: https://www.aimsun.com/the-opticities-project-tec-its-france-september-2015/ (visited on 10/22/2018).

[11] Aimsun, SL. *urban TRaffic management and Air Quality (uTRAQ)*. URL: https://www.aimsun.com/utraq/ (visited on 10/22/2018).

[12] Ezilda Almeida, Carlos Ferreira, and João Gama. "Adaptive Model Rules from Data Streams". In: *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases - Volume 8188*. ECML PKDD 2013. New York, NY, USA: Springer-Verlag New York, Inc., 2013, pp. 480–492. ISBN: 978-3-642-40987-5. URL: http://dx.doi.org/10.1007/978-3-642-40988-2_31.

[13] Osvaldo Anacleto Junior. "Bayesian dynamic graphical models for high-dimensional flow forecasting in road traffic networks". PhD thesis. Open University, 2012.

[14] Osvaldo Anacleto, Catriona Queen, and Casper J. Albers. "Multivariate forecasting of road traffic flows in the presence of heteroscedasticity and measurement errors". In: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 62.2 (2013), pp. 251–270. DOI: 10.1111/j.1467-9876.2012.01059.x. URL: https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9876.2012.01059.x.

[15] Berk Anbaroglu, Benjamin Heydecker, and Tao Cheng. "Spatio-temporal clustering for non-recurrent traffic congestion detection on urban road networks". In: *Transportation Research Part C: Emerging Technologies* 48 (2014), pp. 47–65. ISSN: 0968-090X. DOI: https://doi.org/10.1016/j.trc.2014.08.002. URL: http://www.sciencedirect.com/science/article/pii/S0968090X14002186.

[16] D. Angelosante, J. A. Bazerque, and G. B. Giannakis. "Online Adaptive Estimation of Sparse Signals: Where RLS Meets the L1-Norm". In: *IEEE Transactions on Signal Processing* 58.7 (July 2010), pp. 3436–3447. ISSN: 1053-587X. DOI: 10.1109/TSP.2010.2046897.

[17] C. Antoniou, M. Ben-Akiva, and H.N. Koutsopoulos. "Nonlinear Kalman Filtering Algorithms for On-Line Calibration of Dynamic Traffic Assignment Models". In: *IEEE Transactions on Intelligent Transportation Systems* 8.4 (Dec. 2007), pp. 661–670. ISSN: 1524-9050. DOI: 10.1109/TITS.2007.908569.

[18] Constantinos Antoniou, Haris N. Koutsopoulos, and George Yannis. "Dynamic data-driven local traffic state estimation and prediction". In: *Transportation Research Part C: Emerging Technologies* 34 (Sept. 2013), pp. 89–107. ISSN: 0968090X. DOI: 10.1016/j.trc.2013.05.012. URL: http://linkinghub.elsevier.com/retrieve/pii/S0968090X13001137 (visited on 08/05/2016).

[19] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. "Multi-task feature learning". In: *Advances in neural information processing systems*. 2007, pp. 41–48. URL: http://papers.nips.cc/paper/3143-multi-task-feature-learning.pdf (visited on 06/26/2017).

[20] Yasuo Asakura et al. "Incident detection methods using probe vehicles with on-board GPS equipment". In: *Transportation Research Part C: Emerging Technologies* 81 (Supplement C 2017), pp. 330–341. ISSN: 0968-090X. DOI: https://doi.org/10.1016/j.trc.2016.11.023. URL: http://www.sciencedirect.com/science/article/pii/S0968090X16302479.

[21] Joshua Auld et al. "Impact of Privately-Owned Level 4 CAV Technologies on Travel Demand and Energy". In: *Procedia Computer Science* 130 (C 2018). DOI: 10.1016/j.procs.2018.04.089.

[22] Kay W. Axhausen. "Activity-based modelling: Research directions and possibilities". In: *New Look at Multi-Modal Modelling. Department of Environment, Transport and the Regions, London, Cambridge and Oxford* (2000). URL: http://search.nctcog.org/trans/modeling/nextgeneration/ActivityBasedModelingResearchAndDirections.pdf (visited on 01/09/2016).

[23] Mehdi Azimi and Yunlong Zhang. "Categorizing Freeway Flow Conditions by Using Clustering Methods". In: *Transportation Research Record: Journal of the Transportation Research Board* 2173 (Dec. 2010), pp. 105–114. ISSN: 0361-1981. DOI: 10.3141/2173-13. URL: http://trrjournalonline.trb.org/doi/10.3141/2173-13 (visited on 08/05/2016).

[24] Kevin N Balke. *An evaluation of existing incident detection algorithms*. FHWA/TX-93/1232-20. Texas Transportation Institute, the Texas A&M University System, College Station, TX, 1993.

[25] Michael Balmer. "Travel demand modeling for multi-agent transport simulations: Algorithms and systems". PhD thesis. ETH Zurich, 2007. URL: http://matsim.org/uploads/Balmer2007diss.pdf (visited on 01/09/2016).

[26] Hugo Barbosa et al. "Human mobility: Models and applications". In: *Physics Reports* (2018).

[27] J. Barceló et al. "A Kalman filter approach for dynamic OD estimation in corridors based on Bluetooth and Wifi data collection". In: *12th WCTR, July* (2010), pp. 11–15. URL: http://www.wctrs.leeds.ac.uk/wp/wp-content/uploads/abstracts/lisbon/01387-01.pdf (visited on 01/08/2016).

[28] Jaume Barceló. *Fundamentals of traffic simulation*. Vol. 145. International series in operations research & management science. Springer, 2010. URL: http://www.springer.com/us/book/9781441961419 (visited on 05/05/2015).

[29] James P Barrett. "The coefficient of determination—some limitations". In: *The American Statistician* 28.1 (1974), pp. 19–20.

[30] Joaquim Barros, Miguel Araujo, and Rosaldo JF Rossetti. "Short-term real-time traffic prediction methods: a survey". In: *2015 International Conference on Models and Technologies*

for Intelligent Transportation Systems (MT-ITS). IEEE, 2015, pp. 132–139. URL: http://ieeexplore.ieee.org/abstract/document/7223248/ (visited on 02/23/2017).

[31] Amir Beck and Marc Teboulle. "A fast iterative shrinkage-thresholding algorithm for linear inverse problems". In: *SIAM journal on imaging sciences* 2.1 (2009), pp. 183–202.

[32] Michael GH Bell. "The estimation of origin-destination matrices by constrained generalised least squares". In: *Transportation Research Part B: Methodological* 25.1 (1991), pp. 13–22. URL: http://www.sciencedirect.com/science/article/pii/019126159190010G (visited on 01/08/2016).

[33] Moshe E. Ben-Akiva and Steven R. Lerman. *Discrete choice analysis: theory and application to travel demand*. Vol. 9. MIT press, 1985. URL: https://books.google.es/books?hl=en&lr=&id=oLC6ZYPs9UoC&oi=fnd&pg=PR11&dq=Discrete+Choice+Analysis:+Theory+and+Application+to+Travel+Demand&ots=nMesh-doDi&sig=6YtN5_SxxjB-mURuCwiI5TpZgyE (visited on 01/08/2016).

[34] Eran Ben-Elia and Yoram Shiftan. "Which road do I take? A learning-based model of route-choice behavior with real-time information". In: *Transportation Research Part A: Policy and Practice* 44.4 (2010), pp. 249–264. URL: http://www.sciencedirect.com/science/article/pii/S0965856410000170 (visited on 12/11/2015).

[35] A. Bifet and R. Gavaldà. "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, Apr. 26, 2007, pp. 443–448. ISBN: 978-0-89871-630-6. URL: http://epubs.siam.org/doi/abs/10.1137/1.9781611972771.42 (visited on 07/28/2016).

[36] Albert Bifet et al. *Machine Learning for Data Streams: With Practical Examples in MOA*. MIT Press, 2018.

[37] Ella Bingham. "Reinforcement learning in neurofuzzy traffic signal control". In: *European Journal of Operational Research* 131.2 (2001), pp. 232–241.

[38] Christopher M. Bishop. *Pattern recognition and machine learning*. Information science and statistics. New York: Springer, 2006. 738 pp. ISBN: 978-0-387-31073-2.

[39] Petko Bogdanov, Misael Mongiovì, and Ambuj K. Singh. "Mining heavy subgraphs in time-evolving networks". In: *Data Mining (ICDM), 2011 IEEE 11th International Conference on*. IEEE, 2011, pp. 81–90.

[40] C Bonferroni. "Teoria statistica delle classi e calcolo delle probabilita". In: *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commericiali di Firenze* 8 (1936), pp. 3–62.

[41] H. Botma. "The fundamental diagram: a macroscopic traffic flow model". In: In: Proceedings of the Symposium on methods for determining geometric road design standards, Helsingor, Denmark, May 10-12, 1976, p. 70-71, 3 graph., 1976.

[42] Mark Brackstone and Mike McDonald. "Car-following: a historical review". In: *Transportation Research Part F: Traffic Psychology and Behaviour* 2.4 (Dec. 1999), pp. 181–196. ISSN: 1369-8478. DOI: 10.1016/S1369-8478(00)00005-X. URL: http://www.sciencedirect.com/science/article/pii/S136984780000005X (visited on 01/10/2016).

[43] Leo Breiman. "Statistical Modeling: The Two Cultures (with comments and a rejoinder by the author)". In: *Statistical Science* 16.3 (Aug. 2001), pp. 199–231. ISSN: 0883-4237, 2168-8745. DOI: 10.1214/ss/1009213726. URL: http://projecteuclid.org/euclid.ss/1009213726 (visited on 05/28/2015).

[44] Leo Breiman et al. *Classification and Regression Trees*. Wadsworth, 1984. ISBN: 0-534-98053-8.

[45] D. Brockmann, L. Hufnagel, and T. Geisel. "The scaling laws of human travel". In: *Nature* 439 (Jan. 26, 2006), p. 462. URL: http://dx.doi.org/10.1038/nature04292.

[46] Tamara Broderick et al. "Streaming Variational Bayes". In: *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*. NIPS'13. USA: Curran Associates Inc., 2013, pp. 1727–1735. URL: http://dl.acm.org/citation.cfm?id=2999792.2999805.

[47] Robert D Brook et al. "Particulate matter air pollution and cardiovascular disease: an update to the scientific statement from the American Heart Association". In: *Circulation* 121.21 (2010), pp. 2331–2378.

[48] Christine Buisson and Cyril Ladier. "Exploring the Impact of Homogeneity of Traffic Measurements on the Existence of Macroscopic Fundamental Diagrams". In: *Transportation Research Record: Journal of the Transportation Research Board* 2124 (2009), pp. 127–136. DOI: 10.3141/2124-12. URL: https://doi.org/10.3141/2124-12.

[49] Johannes Martinus Burgers. "A mathematical model illustrating the theory of turbulence". In: *Adv. in Appl. Mech.* 1 (1948), pp. 171–199. URL: http://ci.nii.ac.jp/naid/10004255306/ (visited on 01/15/2016).

[50] E. Burgess. *The growth of the city*. The City. Chicago. Chicago University Press, 1925.

[51] Bob Carpenter et al. "Stan: A Probabilistic Programming Language". In: *Journal of Statistical Software, Articles* 76.1 (2017), pp. 1–32. ISSN: 1548-7660. DOI: 10.18637/jss.v076.i01. URL: https://www.jstatsoft.org/v076/i01.

[52] Rich Caruana. "Multitask Learning". In: *Machine Learning* 28.1 (1997), pp. 41–75. ISSN: 1573-0565. DOI: 10.1023/A:1007379606734. URL: http://dx.doi.org/10.1023/A:1007379606734.

[53] Jordi Casas et al. "Traffic simulation with Aimsun". In: *Fundamentals of traffic simulation*. Springer, 2010, pp. 173–232. URL: http://link.springer.com/chapter/10.1007/978-1-4419-6142-6_5 (visited on 12/11/2015).

[54] Ennio Cascetta and Sang Nguyen. "A unified framework for estimating or updating origin/destination matrices from traffic counts". In: *Transportation Research Part B: Methodological* 22.6 (1988), pp. 437–455. URL: http://www.sciencedirect.com/science/article/pii/0191261588900240 (visited on 01/08/2016).

[55] Enrique Castillo, José María Menéndez, and Santos Sánchez-Cambronero. "Predicting traffic flow using Bayesian networks". In: *Transportation Research Part B: Methodological* 42.5 (June 2008), pp. 482–509. ISSN: 0191-2615. DOI: 10.1016/j.trb.2007.10.003. URL: http://www.sciencedirect.com/science/article/pii/S0191261507001300 (visited on 08/30/2016).

[56] Manoel Castro-Neto et al. "Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions". In: *Expert Systems with Applications* 36.3 (Apr. 2009), pp. 6164–6173. ISSN: 09574174. DOI: 10.1016/j.eswa.2008.07.069. URL: http://linkinghub.elsevier.com/retrieve/pii/S0957417408004740 (visited on 07/26/2016).

[57] Y. Chen and A. O. Hero. "Recursive $1,\infty$ Group Lasso". In: *IEEE Transactions on Signal Processing* 60.8 (Aug. 2012), pp. 3978–3987. ISSN: 1053-587X. DOI: 10.1109/TSP.2012.2192924.

[58] Guillaume Chevillon and David F. Hendry. "Non-parametric direct multi-step estimation for forecasting economic processes". In: *International Journal of Forecasting* 21.2 (2005), pp. 201–218. ISSN: 0169-2070. DOI: https://doi.org/10.1016/j.ijforecast.2004.08.004. URL: http://www.sciencedirect.com/science/article/pii/S016920700400069X.

[59] CIVITAS WIKI consortium and European Commission. *CIVITAS Policy Note: Intelligent Transport Systems and traffic management in urban areas*. 4th CIVITAS WIKI policy note. 2015. URL: http://www.civitas.eu/content/civitas-policy-note-intelligent-transport-systems-and-traffic-management-urban-areas-0 (visited on 10/18/2018).

[60] JF Collins, CM Hopkins, and JA Martin. *Automatic Incident Detection: TRRL Algorithms HIOCC and PATREG*. Transport and Road Research Laboratory, 1979.

[61] CORDIS: Community Research and Development Information Service and European Commission. *SETA: An open, sustainable, ubiquitous data and service ecosystem for efficient, effective, safe, resilient mobility in metropolitan areas*. URL: https://cordis.europa.eu/project/rcn/199852_en.html (visited on 10/22/2018).

[62] Carlos F. Daganzo. *Fundamentals of Transportation and Traffic Operations*. Emerald Group Publishing Limited, Sept. 1997. DOI: 10.1108/9780585475301. URL: https://doi.org/10.1108%2F9780585475301.

[63] Carlos F. Daganzo. "Requiem for second-order fluid approximations of traffic flow". In: *Transportation Research Part B: Methodological* 29.4 (1995), pp. 277–286. URL: http://www.sciencedirect.com/science/article/pii/019126159500007Z (visited on 01/09/2016).

[64] Carlos F. Daganzo. "The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory". In: *Transportation Research Part B: Methodological* 28.4 (1994), pp. 269–287. URL: http://www.sciencedirect.com/science/article/pii/0191261594900027 (visited on 01/09/2016).

[65] Carlos F. Daganzo and Nikolas Geroliminis. "An analytical approximation for the macroscopic fundamental diagram of urban traffic". In: *Transportation Research Part B: Methodological* 42.9 (Nov. 2008), pp. 771–781. ISSN: 0191-2615. DOI: 10.1016/j.trb.2008.06.008. URL: http://www.sciencedirect.com/science/article/pii/S0191261508000799 (visited on 01/10/2016).

[66] Carlos F. Daganzo and Yosef Sheffi. "On stochastic models of traffic assignment". In: *Transportation science* 11.3 (1977), pp. 253–274. URL: http://pubsonline.informs.org/doi/abs/10.1287/trsc.11.3.253 (visited on 01/08/2016).

[67] *Datos Abiertos Santander | Ayuntamiento de Santander*. URL: http://datos.santander.es/ (visited on 08/29/2019).

[68] Onur Deniz and Hilmi Berk Celikoglu. "Overview to some existing incident detection algorithms: a comparative evaluation". In: *Procedia Social and Behavioral Sciences* (2011), pp. 1–13. URL: http://faculty.itu.edu.tr/denizon/DosyaGetir/78706/EWGT2012.pdf (visited on 12/02/2016).

[69] Loukas Dimitriou, Theodore Tsekeris, and Antony Stathopoulos. "Adaptive hybrid fuzzy rule-based system approach for modeling and predicting urban traffic flow". In: *Transportation Research Part C: Emerging Technologies* 16.5 (Oct. 2008), pp. 554–573. ISSN: 0968-090X. DOI: 10.1016/j.trc.2007.11.003. URL: http://www.sciencedirect.com/science/article/pii/S0968090X07000885 (visited on 05/27/2015).

[70] Juan de Dios Ortuzar and Luis G. Willumsen. *Modelling transport*. John Wiley & Sons, 2011. URL: https://books.google.es/books?hl=en&lr=&id=qWa5MyS4CiwC&oi=fnd&pg=PT7&dq=Modelling+Transport&ots=tuYhYg9DRE&sig=8brSuX0YgA8PHbAM7EeRgoiC_EY (visited on 01/08/2016).

[71] Tamara Djukic et al. "Efficient real time OD matrix estimation based on Principal Component Analysis". In: *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*. IEEE, 2012, pp. 115–121. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6338720 (visited on 01/08/2016).

[72] Nemanja Djuric et al. "Travel speed forecasting by means of continuous conditional random fields". In: *Transportation Research Record: Journal of the Transportation Research Board* 2263 (2011), pp. 131–139.

[73] Pedro Domingos. "A unified bias-variance decomposition". In: *Proceedings of 17th International Conference on Machine Learning*. 2000, pp. 231–238.

[74] Pedro Domingos and Geoff Hulten. "Mining high-speed data streams". In: *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2000, pp. 71–80. URL: http://dl.acm.org/citation.cfm?id=347107 (visited on 07/28/2016).

[75] Richard Dowling et al. "Methodology for Measuring Recurrent and Nonrecurrent Traffic Congestion". In: *Transportation Research Record: Journal of the Transportation Research Board* 1867 (2004), pp. 60–68. DOI: 10.3141/1867-08. URL: https://doi.org/10.3141/1867-08.

[76] JL Drake and Joseph L Schofer. "A statistical analysis of speed-density hypotheses". In: *Highway Research Record* 154 (1966), pp. 53–87.

[77] Conrad L. Dudek, Carroll J. Messer, and Nelson B. Nuckles. "Incident detection on urban freeways". In: *Transportation research record* 495 (1974), pp. 12–24. ISSN: 0361-1981. URL: http://dx.doi.org/.

[78] Olive Jean Dunn. "Multiple comparisons among means". In: *Journal of the American statistical association* 56.293 (1961), pp. 52–64.

[79] EasyWay Consortium - European Comission. *A EU project for Europe-wide ITS deployment on the main Trans-European Road Network corridors*. 2015. URL: http://www.its-platform.eu/ (visited on 10/10/2018).

[80] Bradley Efron et al. "Least angle regression". In: *The Annals of statistics* 32.2 (2004), pp. 407–499.

[81] Alireza Ermagun and David Levinson. "Spatiotemporal traffic forecasting: review and proposed directions". In: *Transport Reviews* 38.6 (2018), pp. 786–814. DOI: 10.1080/01441647.2018.1442887. URL: https://doi.org/10.1080/01441647.2018.1442887.

[82] European Commission - Transport. *White Paper 2011: Roadmap to a Single European Transport Area – Towards a competitive and resource efficient transport system*. Mar. 28, 2011.

[83] European Environment Agency, DG Climate Action, and European Commission. *Annual European Union greenhouse gas inventory 1990–2016 and inventory report 2018*. EEA Report No 5/2018. European Environment Agency, May 27, 2018, p. 975. URL: https://www.eea.europa.eu//publications/european-union-greenhouse-gas-inventory-2018.

[84] Jianqing Fan et al. "Statistical sparse online regression: A diffusion approximation perspective". In: *International Conference on Artificial Intelligence and Statistics*. 2018, pp. 1017–1026.

[85] Nour-Eddin El Faouzi, Henry Leung, and Ajeesh Kurian. "Data fusion in intelligent transportation systems: Progress and challenges – A survey". In: *Information Fusion*. Special Issue on Intelligent Transportation Systems 12.1 (Jan. 2011), pp. 4–10. ISSN: 1566-2535. DOI: 10.1016/j.inffus.2010.06.001. URL: http://www.sciencedirect.com/science/article/pii/S1566253510000643 (visited on 05/26/2015).

[86] Olga Feldman. "The GEH measure and quality of the highway assignment models". In: *Association for European Transport and Contributors* (2012), pp. 1–18.

[87] Jaime L. Ferrer and Jaime Barceló. *AIMSUN2: advanced interactive microscopic simulator for urban and non-urban networks*. Research report. Departamento de Estadística e Investigación Operativa, Facultad de Informática, Universidad Politécnica de Cataluña, 1993.

[88] Gerd A. Folberth et al. "Megacities and climate change – A brief overview". In: *Environmental Pollution* 203 (2015), pp. 235–242. ISSN: 0269-7491. DOI: https://doi.org/10.1016/j.envpol.2014.09.004. URL: http://www.sciencedirect.com/science/article/pii/S0269749114003844.

[89] Marguerite Frank and Philip Wolfe. "An algorithm for quadratic programming". In: *Naval Research Logistics Quarterly* 3.1 (Mar. 1, 1956), pp. 95–110. ISSN: 1931-9193. DOI: 10.1002/nav.3800030109. URL: http://onlinelibrary.wiley.com/doi/10.1002/nav.3800030109/abstract (visited on 01/08/2016).

[90] Jerome Friedman, Trevor Hastie, and Rob Tibshirani. "Regularization paths for generalized linear models via coordinate descent". In: *Journal of statistical software* 33.1 (2010), p. 1. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2929880/ (visited on 07/28/2016).

[91] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*. Vol. 1. 10. Springer series in statistics New York, 2001.

[92] Jerome Friedman et al. "Pathwise coordinate optimization". In: *The Annals of Applied Statistics* 1.2 (Dec. 2007), pp. 302–332. ISSN: 1932-6157. DOI: 10.1214/07-AOAS131. URL: http://projecteuclid.org/euclid.aoas/1196438020 (visited on 07/28/2016).

[93] GAETANO Fusco and CHIARA Colombaroni. "An integrated method for short-term prediction of road traffic conditions for Intelligent Transportation Systems Applications". In: *Recent Advances in Information Science, Proc. of the 7th European Computing Conf.(ECC'13), Dubrovnik*. 2013, pp. 339–344.

[94] Joao Gama. *Knowledge Discovery from Data Streams*. Chapman & Hall/CRC, 2010. URL: http://www.liaad.up.pt/area/jgama/DataStreamsCRC.pdf.

[95] João Gama, Ricardo Fernandes, and Ricardo Rocha. "Decision trees for mining data streams". In: *Intelligent Data Analysis* 10.1 (2006), pp. 23–45.

[96] João Gama, Ricardo Rocha, and Pedro Medas. "Accurate Decision Trees for Mining High-speed Data Streams". In: *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '03. event-place: Washington, D.C. New York, NY, USA: ACM, 2003, pp. 523–528. ISBN: 1-58113-737-0. DOI: 10.1145/956750.956813. URL: http://doi.acm.org/10.1145/956750.956813.

[97] João Gama et al. "A Survey on Concept Drift Adaptation". In: *ACM Comput. Surv.* 46.4 (Mar. 2014), 44:1–44:37. ISSN: 0360-0300. DOI: 10.1145/2523813. URL: http://doi.acm.org/10.1145/2523813 (visited on 01/11/2016).

[98] Venkatesh Ganti, Johannes Gehrke, and Raghu Ramakrishnan. "Mining Data Streams Under Block Evolution". In: *SIGKDD Explor. Newsl.* 3.2 (Jan. 2002), pp. 1–10. ISSN: 1931-0145. DOI: 10.1145/507515.507517. URL: http://doi.acm.org/10.1145/507515.507517.

[99] Pierre Garrigues and Laurent E. Ghaoui. "An Homotopy Algorithm for the Lasso with Online Observations". In: *Advances in Neural Information Processing Systems 21*. Ed. by D. Koller et al. Curran Associates, Inc., 2009, pp. 489–496. URL: http://papers.nips.cc/paper/3431-an-homotopy-algorithm-for-the-lasso-with-online-observations.pdf.

[100] Nathan Gartner, Carroll J. Messer, and Aiay Kumar Rathi. "Traffic flow theory: A state-of-the-art report". In: (2001). URL: http://www.researchgate.net/publication/248146380_Traffic_flow_theory_A_state-of-the-art_report (visited on 05/05/2015).

[101] Frank Geels et al. *Automobility in transition?: A socio-technical analysis of sustainable transport*. Routledge, 2011.

[102] Andrew Gelman et al. *Bayesian data analysis*. Vol. 2. Chapman & Hall/CRC Boca Raton, FL, USA, 2014. URL: http://amstat.tandfonline.com/doi/full/10.1080/01621459.2014.963405 (visited on 01/21/2017).

[103] D. L. Gerlough and M. J. Huber. *Traffic flow theory*. 1975. URL: http://trid.trb.org/view.aspx?id=1184296 (visited on 01/09/2016).

[104] N. Geroliminis and A. Skabardonis. "Identification and Analysis of Queue Spillovers in City Street Networks". In: *IEEE Transactions on Intelligent Transportation Systems* 12.4 (Dec. 2011), pp. 1107–1115. ISSN: 1524-9050. DOI: 10.1109/TITS.2011.2141991.

[105] Bidisha Ghosh, Biswajit Basu, and Margaret O'Mahony. "Bayesian time-series model for short-term traffic flow forecasting". In: *Journal of transportation engineering* 133.3 (2007), pp. 180–189.

[106] S. Ghosh, M. T. Asif, and L. Wynter. "Denoising autoencoders for fast real-time traffic estimation on urban road networks". In: *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. Dec. 2017, pp. 6307–6312. DOI: 10.1109/CDC.2017.8264610.

[107] Tilmann Gneiting. "Making and Evaluating Point Forecasts". In: *Journal of the American Statistical Association* 106.494 (2011), pp. 746–762. DOI: 10.1198/jasa.2011.r10138. URL: https://doi.org/10.1198/jasa.2011.r10138.

[108] Marta C. González, César A. Hidalgo, and Albert-László Barabási. "Understanding individual human mobility patterns". In: *Nature* 453 (June 5, 2008), p. 779. URL: http://dx.doi.org/10.1038/nature06958.

[109] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[110] BD Greenshields, Ws Channing, Hh Miller, et al. "A study of traffic capacity". In: *Highway research board proceedings*. Vol. 1935. National Research Council (USA), Highway Research Board, 1935.

[111] Sudipto Guha, Nick Koudas, and Kyuseok Shim. "Data-streams and histograms". In: *Proceedings of the thirty-third annual ACM symposium on Theory of computing*. ACM, 2001, pp. 471–475.

[112] Filmon G. Habtemichael and Mecit Cetin. "Short-term traffic flow rate forecasting based on identifying similar traffic patterns". In: *Transportation Research Part C: Emerging Technologies*. Advanced Network Traffic Management: From dynamic state estimation to traffic control 66 (May 2016), pp. 61–78. ISSN: 0968-090X. DOI: 10.1016/j.trc.2015.08.017. URL: http://www.sciencedirect.com/science/article/pii/S0968090X15003186 (visited on 08/03/2016).

[113] Torsten Hägerstraand. "What about people in regional science?" In: *Papers in regional science* 24.1 (1970), pp. 7–24. URL: http://onlinelibrary.wiley.com/doi/10.1111/j.1435-5597.1970.tb01464.x/abstract (visited on 01/09/2016).

[114] Lee D Han and Adolf Darlington May. *Automatic detection of traffic operational problems on urban arterials*. Institute of Transportation Studies, U.C. Berkeley, 1989.

[115] Chauncy D. Harris and Edward L. Ullman. "The nature of cities". In: *The Annals of the American Academy of Political and Social Science* (1945), pp. 7–17. URL: http://www.jstor.org/stable/1026055 (visited on 01/07/2016).

[116] Trevor Hastie. "Fast regularization paths via coordinate descent". In: *The 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Denver*. Vol. 2009. 2008. URL: http://web.stanford.edu/~hastie/TALKS/glmnet.pdf (visited on 07/28/2016).

[117] Trevor Hastie, Robert Tibshirani, and Martin Wainwright. *Statistical Learning with Sparsity: The Lasso and Generalizations*. Published: Hardcover. Chapman and Hall/CRC, May 2015. URL: http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/1498712169.

[118] Y. E. Hawas and M. S. Mohammad. "A System for Incident Detection in Urban Traffic Networks". In: *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. Sept. 2015, pp. 2405–2411. DOI: 10.1109/ITSC.2015.388.

[119] Dirk Helbing and Kai Nagel. "The physics of traffic and regional development". In: *Contemporary Physics* 45.5 (2004), pp. 405–426. URL: http://www.tandfonline.com/doi/abs/10.1080/00107510410001715944 (visited on 01/08/2016).

[120] David F Hendry and Felix Pretis. "All Change! The Implications of Non-Stationarity for Empirical Modelling, Forecasting and Policy". In: *Oxford Martin Policy Papers* (2016).

[121] Y. Hernandez-Potiomkin et al. "Unsupervised Incident Detection Model in Urban and Freeway Networks". In: *2018 IEEE 21st International Conference on Intelligent Transportation Systems (ITSC)*. Nov. 2018, pp. 1763–1769. DOI: 10.1109/ITSC.2018.8569642.

[122] David C. Hoaglin, Boris Iglewicz, and John W. Tukey. "Performance of Some Resistant Rules for Outlier Labeling". In: *Journal of the American Statistical Association* 81.396 (1986), pp. 991–999. DOI: 10.1080/01621459.1986.10478363. URL: https://amstat.tandfonline.com/doi/abs/10.1080/01621459.1986.10478363.

[123] A. Hofleitner et al. "Learning the Dynamics of Arterial Traffic From Probe Data Using a Dynamic Bayesian Network". In: *IEEE Transactions on Intelligent Transportation Systems* 13.4 (Dec. 2012), pp. 1679–1693. ISSN: 1524-9050. DOI: 10.1109/TITS.2012.2200474.

[124] A Hofleitner, L El Ghaoui, and A Bayen. "Online least-squares estimation of time varying systems with sparse temporal evolution and application to traffic estimation". In: *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*. IEEE, 2011, pp. 2595–2601.

[125] Serge P. Hoogendoorn and Piet HL Bovy. "State of the art of vehicular traffic flow modelling". In: *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering* 215.4 (2001), pp. 283–303. URL: http://pii.sagepub.com/content/215/4/283.short (visited on 05/11/2015).

[126] Eric Horvitz et al. "Prediction, Expectation, and Surprise: Methods, Designs, and Study of a Deployed Traffic Forecasting Service". In: *UAI*. AUAI Press, July 2005, pp. 275–283. ISBN: 0-9749039-1-4. URL: https://www.microsoft.com/en-us/research/publication/prediction-expectation-and-surprise-methods-designs-and-study-of-a-deployed-traffic-forecasting-service-2/.

[127] H. Hoyt. "The structure and growth of residential neighborhoods in American cities". In: (1939). URL: http://trid.trb.org/view.aspx?id=131170 (visited on 01/07/2016).

[128] Ling Huang et al. "Adversarial Machine Learning". In: *Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence*. AISec '11. New York, NY, USA: ACM, 2011, pp. 43–58. ISBN: 978-1-4503-1003-1. DOI: 10.1145/2046684.2046692. URL: http://doi.acm.org/10.1145/2046684.2046692.

[129] Shan Huang and Adel W. Sadek. "A novel forecasting approach inspired by human memory: The example of short-term traffic volume forecasting". In: *Transportation Research Part C: Emerging Technologies* 17.5 (Oct. 2009), pp. 510–525. ISSN: 0968090X. DOI: 10.1016/j.trc.2009.04.006. URL: http://linkinghub.elsevier.com/retrieve/pii/S0968090X09000333 (visited on 07/26/2016).

[130] Wenhao Huang et al. "Deep architecture for traffic flow prediction: deep belief networks with multitask learning". In: *IEEE Transactions on Intelligent Transportation Systems* 15.5 (Oct. 2014), pp. 2191–2201. ISSN: 1524-9050, 1558-0016. DOI: 10.1109/TITS.2014.2311123. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6786503 (visited on 07/25/2016).

[131] Elena Ikonomovska, João Gama, and Sašo Džeroski. "Learning model trees from evolving data streams". In: *Data Mining and Knowledge Discovery* 23.1 (2011), pp. 128–168. ISSN: 1384-5810, 1573-756X. DOI: 10.1007/s10618-010-0201-y. URL: http://link.springer.com/article/10.1007/s10618-010-0201-y (visited on 08/30/2016).

[132] Intergovernmental Panel on Climate Change (IPCC). *IPCC Fifth Assessment Report (AR5) Observed Climate Change Impacts Database, Version 2.01*. NASA Socioeconomic Data and Applications Center (SEDAC), 2017. URL: https://doi.org/10.7927/H4FT8J0X.

[133] Sherif Ishak, Prashanth Kotha, and Ciprian Alecsandru. "Optimization of dynamic neural network performance for short-term traffic prediction". In: *Transportation Research Record: Journal of the Transportation Research Board* 1836.1 (2003), pp. 45–56. URL: http://trb.metapress.com/index/901451745921351g.pdf (visited on 05/04/2015).

[134] J. W. Wedel, B. Schünemann, and I. Radusch. "V2X-Based Traffic Congestion Recognition and Avoidance". In: *2009 10th International Symposium on Pervasive Systems, Algorithms, and Networks*. 2009 10th International Symposium on Pervasive Systems, Algorithms, and Networks. Dec. 14, 2009, pp. 637–641. ISBN: 1087-4089. DOI: 10.1109/I-SPAN.2009.71.

[135] Ali Jalali et al. "A dirty model for multi-task learning". In: *Advances in Neural Information Processing Systems*. 2010, pp. 964–972. URL: http://papers.nips.cc/paper/4125-a-dirty-model-for-multi-task-learning (visited on 07/06/2017).

[136] Feng Jin and Shiliang Sun. "Neural network multitask learning for traffic flow forecasting". In: *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. IEEE, 2008, pp. 1897–1901. URL: http://ieeexplore.ieee.org/abstract/document/4634057/ (visited on 07/10/2017).

[137] Rong-Chang Jou. "Modeling the impact of pre-trip information on commuter departure time and route choice". In: *Transportation Research Part B: Methodological* 35.10 (2001), pp. 887–902. URL: http://www.sciencedirect.com/science/article/pii/S019126150000028X (visited on 12/11/2015).

[138] Pushkin Kachroo, Kaan Ozbay, and Arvind Narayanan. "Investigating the use of kalman filtering approaches for dynamic origin-destination trip table estimation". In: *Proceedings of Southeastcon '97: Engineering the New Century* (Apr. 1, 1997), pp. 138–142. URL: http://digitalscholarship.unlv.edu/ece_fac_articles/83.

[139] Yiannis Kamarianakis, H. Oliver Gao, and Poulicos Prastacos. "Characterizing regimes in daily cycles of urban traffic using smooth-transition regressions". In: *Transportation Research Part C: Emerging Technologies* 18.5 (2010), pp. 821–840. ISSN: 0968-090X. DOI: https://doi.

org/10.1016/j.trc.2009.11.001. URL: http://www.sciencedirect.com/science/article/pii/S0968090X09001442.

[140] Yiannis Kamarianakis, Wei Shen, and Laura Wynter. "Real-time road traffic forecasting using regime-switching space-time models and adaptive LASSO". In: *Applied Stochastic Models in Business and Industry* 28.4 (July 2012), pp. 297–315. ISSN: 15241904. DOI: 10.1002/asmb.1937. URL: http://doi.wiley.com/10.1002/asmb.1937 (visited on 07/25/2016).

[141] Zhuoliang Kang, Kristen Grauman, and Fei Sha. "Learning with whom to share in multi-task feature learning". In: *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*. 2011, pp. 521–528.

[142] Matthew G. Karlaftis and Eleni I. Vlahogianni. "Statistical methods versus neural networks in transportation research: Differences, similarities and some insights". In: *Transportation Research Part C: Emerging Technologies* 19.3 (2011), pp. 387–399. URL: http://www.sciencedirect.com/science/article/pii/S0968090X10001610 (visited on 09/13/2016).

[143] Boris S. Kerner. *Introduction to modern traffic flow theory and control: the long road to three-phase traffic theory*. Springer Science & Business Media, 2009. URL: https://books.google.es/books?hl=es&lr=&id=4g7f1h4BfYsC&oi=fnd&pg=PA1&dq=three+phase+flow+theory&ots=v6wGSntDh3&sig=lgY7D9fcvhzQSkSSRvCNzQXskVs (visited on 01/10/2016).

[144] Boris S. Kerner. "The physics of traffic". In: *Physics World* 12.8 (1999). URL: http://trid.trb.org/view.aspx?id=645081 (visited on 01/10/2016).

[145] Boris S. Kerner. "Three-phase traffic theory and highway capacity". In: *Physica A: Statistical Mechanics and its Applications* 333 (2004), pp. 379–440. ISSN: 0378-4371. DOI: https://doi.org/10.1016/j.physa.2003.10.017. URL: http://www.sciencedirect.com/science/article/pii/S0378437103009221.

[146] Jalil Kianfar and Praveen Edara. "A Data Mining Approach to Creating Fundamental Traffic Flow Diagram". In: *Procedia - Social and Behavioral Sciences* 104 (Supplement C 2013), pp. 430–439. ISSN: 1877-0428. DOI: https://doi.org/10.1016/j.sbspro.2013.11.136. URL: http://www.sciencedirect.com/science/article/pii/S1877042813045278.

[147] Daniel Kifer, Shai Ben-David, and Johannes Gehrke. "Detecting Change in Data Streams". In: *Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30*. VLDB '04. event-place: Toronto, Canada. VLDB Endowment, 2004, pp. 180–191. ISBN: 0-12-088469-0. URL: http://dl.acm.org/citation.cfm?id=1316689.1316707.

[148] Jiwon Kim and Guangxing Wang. "Diagnosis and Prediction of Traffic Congestion on Urban Road Networks Using Bayesian Networks". In: *Transportation Research Record: Journal of the Transportation Research Board* 2595 (Jan. 2016), pp. 108–118. ISSN: 0361-1981. DOI: 10.3141/2595-12. URL: http://trrjournalonline.trb.org/doi/10.3141/2595-12 (visited on 11/05/2017).

[149] Youngho Kim, Woojin Kang, and Minju Park. "Application of Traffic State Prediction Methods to Urban Expressway Network in the City of Seoul". In: *Journal of the Eastern Asia Society for Transportation Studies* 11 (2015), pp. 1885–1898. DOI: 10.11175/easts.11.1885.

[150] Kin-Pong Chan and Ada Wai-Chee Fu. "Efficient time series matching by wavelets". In: *Proceedings 15th International Conference on Data Engineering (Cat. No.99CB36337)*. Mar. 1999, pp. 126–133. DOI: 10.1109/ICDE.1999.754915.

[151] Akira Kinoshita, Atsuhiro Takasu, and Jun Adachi. "Real-time traffic incident detection using a probabilistic topic model". In: *Information Systems* 54 (Supplement C 2015), pp. 169–188. ISSN: 0306-4379. DOI: https://doi.org/10.1016/j.is.2015.07.002. URL: http://www.sciencedirect.com/science/article/pii/S0306437915001301.

[152] Andreas Kleinert. "Der messende Luchs". In: *NTM International Journal of History & Ethics of Natural Sciences Technology & Medicine* 17 (2009), pp. 199–206. DOI: 10.1007/s00048-009-0335-4.

[153] Ralf Klinkenberg. "Learning drifting concepts: Example selection vs. example weighting". In: *Intelligent data analysis* 8.3 (2004), pp. 281–300.

[154] Daphne Koller, Nir Friedman, and Francis Bach. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[155] Andreĭ Nikolaevich Kolmogorov. *Foundations of the theory of probability*. Oxford, England: Chelsea Publishing Co, 1950.

[156] M Koshi, M Iwasaki, and I Ohkura. "Some findings and overview on vehicle flow characteristics". In: *Proceedings of the 8th International Symposium on Transportation and Traffic Theory*. Toronto: University of Toronto Press, 1981, pp. 403–426.

[157] Ivan Koychev. "Gradual Forgetting for Adaptation to Concept Drift". In: *In Proceedings of ECAI 2000 Workshop Current Issues in Spatio-Temporal Reasoning*. 2000, pp. 101–106.

[158] Michal Krzyżanowski, Birgit Kuna-Dibbert, and Jürgen Schneider. *Health effects of transport-related air pollution*. WHO Regional Office Europe, 2005.

[159] Reinhart Kuhne and Panos Michalopoulos. "Continuum flow models". In: *Traffic Flow Theory: A State of the Art Report—Revised Monograph on Traffic Flow Theory, Oak Ridge National Laboratory, Oak Ridge, TN* (1997), p. 432. URL: http://www.licejus.lt/~fizmat/rytis/traffic.pdf (visited on 01/09/2016).

[160] Abhishek Kumar and Hal Daume III. "Learning task grouping and overlap in multi-task learning". In: *arXiv preprint arXiv:1206.6417* (2012).

[161] Jaimyoung Kwon, Michael Mauch, and Pravin Varaiya. "Components of Congestion: Delay from Incidents, Special Events, Lane Closures, Weather, Potential Ramp Metering Gain, and Excess Demand". In: *Transportation Research Record* 1959.1 (2006), pp. 84–91. DOI: 10.1177/0361198106195900110. URL: https://doi.org/10.1177/0361198106195900110.

[162] John Langford, Lihong Li, and Tong Zhang. "Sparse online learning via truncated gradient". In: *Journal of Machine Learning Research* 10 (Mar 2009), pp. 777–801. URL: http://www.jmlr.org/papers/volume10/langford09a/langford09a.pdf (visited on 07/28/2016).

[163] Ludovic Leclercq. "Calibration of Flow-Density Relationships on Urban Streets". In: *Transportation Research Record: Journal of the Transportation Research Board* 1934 (2005), pp. 226–234. DOI: 10.3141/1934-24. URL: https://doi.org/10.3141/1934-24.

[164] David Levinson et al. *Fundamentals of Transportation*. 2019. URL: https://en.wikibooks.org/wiki/Fundamentals_of_Transportation.

[165] Baibing Li and Bart De Moor. "Dynamic identification of origin–destination matrices in the presence of incomplete observations". In: *Transportation Research Part B: Methodological* 36.1 (2002), pp. 37–57. URL: http://www.sciencedirect.com/science/article/pii/S0191261500000370 (visited on 01/08/2016).

[166] Li Li et al. "Robust causal dependence mining in big data network and its application to traffic flow predictions". In: *Transportation Research Part C: Emerging Technologies* 58 (Sept. 2015), pp. 292–307. ISSN: 0968090X. DOI: 10.1016/j.trc.2015.03.003. URL: http://linkinghub.elsevier.com/retrieve/pii/S0968090X15000820 (visited on 07/25/2016).

[167] M. J. Lighthill and G. B. Whitham. "On Kinematic Waves. II. A Theory of Traffic Flow on Long Crowded Roads". In: *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 229.1178 (May 10, 1955), pp. 317–345. ISSN: 1364-5021, 1471-2946. DOI: 10.1098/rspa.1955.0089. URL: http://rspa.royalsocietypublishing.org/content/229/1178/317 (visited on 01/07/2016).

[168] Lei Lin, Zhengbing He, and Srinivas Peeta. "Predicting station-level hourly demand in a large-scale bike-sharing network: A graph convolutional neural network approach". In: *Transportation Research Part C: Emerging Technologies* 97 (2018), pp. 258–276.

[169] Wei-Hua Lin and Carlos F. Daganzo. "A simple detection scheme for delay-inducing freeway incidents". In: *Transportation Research Part A: Policy and Practice* 31.2 (1997), pp. 141–155. ISSN: 0965-8564. DOI: https://doi.org/10.1016/S0965-8564(96)00009-2. URL: http://www.sciencedirect.com/science/article/pii/S0965856496000092.

[170] Hans van Lint and Chris van Hinsbergen. "Short-Term Traffic and Travel Time Prediction Models". In: *Transportation Research E-Circular* (E-C168 Nov. 2012). ISSN: 0097-8515. URL: https://trid.trb.org/view.aspx?id=1225153 (visited on 07/29/2016).

[171] Hao Liu et al. "Predicting urban arterial travel time with state-space neural networks and Kalman filters". In: *Transportation Research Record* 1968.1 (2006), pp. 99–108.

[172] Yisheng Lv et al. "Traffic Flow Prediction With Big Data: A Deep Learning Approach". In: *IEEE Transactions on Intelligent Transportation Systems* (2014), pp. 1–9. ISSN: 1524-9050, 1558-0016. DOI: 10.1109/TITS.2014.2345663. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6894591 (visited on 07/25/2016).

[173] Yuting Ma and Tian Zheng. "Stabilized Sparse Online Learning for Sparse Data". In: *J. Mach. Learn. Res.* 18.1 (Jan. 2017), pp. 4773–4808. ISSN: 1532-4435. URL: http://dl.acm.org/citation.cfm?id=3122009.3208012.

[174] Sven Maerivoet and Bart De Moor. "Transportation planning and traffic flow models". In: *arXiv preprint physics/0507127* (2005). URL: http://arxiv.org/abs/physics/0507127 (visited on 01/07/2016).

[175] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. "Statistical and Machine Learning forecasting methods: Concerns and ways forward". In: *PLOS ONE* 13.3 (2018), pp. 1–26. DOI: 10.1371/journal.pone.0194889. URL: https://doi.org/10.1371/journal.pone.0194889.

[176] Frank J Massey Jr. "The Kolmogorov-Smirnov test for goodness of fit". In: *Journal of the American statistical Association* 46.253 (1951), pp. 68–78.

[177] P. H. Masters, J. K. Lam, and Kam Wong. "Incident detection algorithms for COMPASS - An advanced traffic management system". In: *Vehicle Navigation and Information Systems Conference, 1991*. Vol. 2. Oct. 1991, pp. 295–310. DOI: 10.1109/VNIS.1991.205776.

[178] A. Mazloumian, N. Geroliminis, and D. Helbing. "The spatial variability of vehicle densities as determinant of urban network capacity". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 368.1928 (Oct. 13, 2010), pp. 4627–4647. ISSN: 1364-503X, 1471-2962. DOI: 10.1098/rsta.2010.0099. URL: http://rsta.royalsocietypublishing.org/cgi/doi/10.1098/rsta.2010.0099 (visited on 11/10/2017).

[179] Michael G. McNally. "The four step model". In: *Center for Activity Systems Analysis* (2008). URL: http://escholarship.org/uc/item/0r75311t.pdf (visited on 01/08/2016).

[180] Rafael Mena-Yedra, Jordi Casas, and Ricard Gavaldà. "Probabilistic model for robust traffic state identification in urban networks". In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 1934–1940.

[181] Rafael Mena-Yedra, Ricard Gavaldà, and Jordi Casas. "Adarules: Learning rules for real-time road-traffic prediction". In: *Transportation Research Procedia* 27 (2017), pp. 11–18. ISSN: 2352-1465. DOI: https://doi.org/10.1016/j.trpro.2017.12.106. URL: http://www.sciencedirect.com/science/article/pii/S2352146517310037.

[182] Ryszard S. Michalski, Jaime G. Carbonell, and Tom M. Mitchell. *Machine learning: An artificial intelligence approach*. Springer Science & Business Media, 2013. URL: https://books.google.es/books?hl=es&lr=&id=-eqpCAAAQBAJ&oi=fnd&pg=PA2&dq=Tom+M.+Mitchell&ots=Wk4NOA8Fl2&sig=KdQKOU25ZRS9qUdQOCvteV2aEK8 (visited on 01/10/2016).

[183] Wanli Min and Laura Wynter. "Real-time road traffic prediction with spatio-temporal correlations". In: *Transportation Research Part C: Emerging Technologies* 19.4 (2011), pp. 606–616. URL: http://www.sciencedirect.com/science/article/pii/S0968090X10001592 (visited on 04/30/2015).

[184] R. B. Mitchell and C. Rapkin. "Urban traffic - A function of land use". In: (1954). URL: http://trid.trb.org/view.aspx?id=131510 (visited on 01/08/2016).

[185] Misael Mongiovi et al. "Netspot: Spotting significant anomalous regions on dynamic networks". In: *Proceedings of the 2013 SIAM International Conference on Data Mining*. SIAM, 2013, pp. 28–36.

[186] Ajith Muralidharan and Roberto Horowitz. "Imputation of ramp flow data for freeway traffic simulation". In: *Transportation Research Record* 2099.1 (2009), pp. 58–64.

[187] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012. ISBN: 0-262-01802-0 978-0-262-01802-9.

[188] Shanmugavelayutham Muthukrishnan et al. "Data streams: Algorithms and applications". In: *Foundations and Trends® in Theoretical Computer Science* 1.2 (2005), pp. 117–236.

[189] National Academies of Sciences, Engineering, and Medicine and Transportation Research Board - Artificial Intelligence and Advanced Computing Committee. *Artificial Intelligence Applications to Critical Transportation Issues*. Transportation Research Board, 2012.

[190] Sahand Negahban and Martin J Wainwright. "Joint support recovery under high-dimensional scaling: Benefits and perils of $_{1,\infty}$-regularization". In: *Proceedings of the 21st International Conference on Neural Information Processing Systems*. Curran Associates Inc., 2008, pp. 1161–1168.

[191] Yu. Nesterov. "Efficiency of Coordinate Descent Methods on Huge-Scale Optimization Problems". In: *SIAM Journal on Optimization* 22.2 (Jan. 2012), pp. 341–362. ISSN: 1052-6234, 1095-7189. DOI: 10.1137/100802001. URL: http://epubs.siam.org/doi/abs/10.1137/100802001 (visited on 07/28/2016).

[192] Andrew Y Ng and Michael I Jordan. "On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes". In: *Advances in neural information processing systems*. 2002, pp. 841–848.

[193] Paul Nieuwenhuis. *Sustainable Automobility: Understanding the Car as a Natural System*. Edward Elgar Publishing, 2014.

[194] Nancy L. Nihan and Gary A. Davis. "Recursive estimation of origin-destination matrices from input/output counts". In: *Transportation Research Part B: Methodological* 21.2 (1987), pp. 149–163. URL: http://www.sciencedirect.com/science/article/pii/0191261587900130 (visited on 01/08/2016).

[195] Robert B. Noland and John W. Polak. "Travel time variability: A review of theoretical and empirical issues". In: *Transport Reviews* 22.1 (2002), pp. 39–54. DOI: 10.1080/01441640010022456. URL: https://doi.org/10.1080/01441640010022456.

[196] Iwao Okutani and Yorgos J. Stephanedes. "Dynamic prediction of traffic volume through Kalman filtering theory". In: *Transportation Research Part B: Methodological* 18.1 (1984), pp. 1–11. URL: http://www.sciencedirect.com/science/article/pii/019126158490002X (visited on 04/30/2015).

[197] *OpenStreetMap*. URL: https://www.openstreetmap.org/ (visited on 05/27/2019).

[198] Organisation for Economic Co-operation and Development. *Dynamic traffic management in urban and suburban road systems: report*. Paris: Organisation for Economic Co-operation and Development, Apr. 1987. 108 pp.

[199] ES Page. "Continuous inspection schemes". In: *Biometrika* 41.1 (1954), pp. 100–115.

[200] T. L. Pan et al. "Short-Term Traffic State Prediction Based on Temporal-Spatial Correlation". In: *IEEE Transactions on Intelligent Transportation Systems* 14.3 (Sept. 2013), pp. 1242–1254. ISSN: 1524-9050, 1558-0016. DOI: 10.1109/TITS.2013.2258916. URL:

http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6517239 (visited on 07/25/2016).

[201] Trevor Park and George Casella. "The bayesian lasso". In: *Journal of the American Statistical Association* 103.482 (2008), pp. 681–686.

[202] Emily Parkany and Chi Xie. *A complete review of incident detection algorithms & their deployment: what works and what doesn't*. 2005. URL: https://trid.trb.org/view.aspx?id=754914 (visited on 12/02/2016).

[203] Harold J. Payne. "Models of freeway traffic and control." In: *Mathematical models of public systems* (1971). URL: http://trid.trb.org/view.aspx?id=531574 (visited on 01/09/2016).

[204] Howard J. Payne and Samuel C. Tignor. "Freeway incident-detection algorithms based on decision trees with states". In: *Transportation Research Record* 682 (1978). URL: https://trid.trb.org/view.aspx?id=91809 (visited on 12/20/2016).

[205] Srinivas Peeta and T.-H. Yang. "Stability issues for dynamic traffic assignment". In: *Automatica* 39.1 (2003), pp. 21–34. URL: http://www.sciencedirect.com/science/article/pii/S0005109802001796 (visited on 01/08/2016).

[206] Srinivas Peeta and Athanasios K. Ziliaskopoulos. "Foundations of Dynamic Traffic Assignment: The Past, the Present and the Future". In: *Networks and Spatial Economics* 1.3 (Sept. 2001), pp. 233–265. ISSN: 1566-113X, 1572-9427. DOI: 10.1023/A:1012827724856. URL: http://link.springer.com/article/10.1023/A%3A1012827724856 (visited on 01/08/2016).

[207] Bhagwant N Persaud, Fred L Hall, and Lisa M Hall. "Congestion identification aspects of the McMaster incident detection algorithm". In: *Transportation Research Record* 1287 (1990), pp. 167–175.

[208] Nicholas G. Polson and Vadim O. Sokolov. "Deep learning for short-term traffic flow prediction". In: *Transportation Research Part C: Emerging Technologies* 79 (2017), pp. 1–17. ISSN: 0968-090X. DOI: https://doi.org/10.1016/j.trc.2017.02.024. URL: http://www.sciencedirect.com/science/article/pii/S0968090X17300633.

[209] Fengxiang Qiao, Hai Yang, and William HK Lam. "Intelligent simulation and prediction of traffic flow dispersion". In: *Transportation Research Part B: Methodological* 35.9 (2001), pp. 843–863. URL: http://www.sciencedirect.com/science/article/pii/S0191261500000242 (visited on 01/16/2016).

[210] Injong Rhee et al. "On the Levy-walk Nature of Human Mobility". In: *IEEE/ACM Trans. Netw.* 19.3 (June 2011), pp. 630–643. ISSN: 1063-6692. DOI: 10.1109/TNET.2011.2120618. URL: http://dx.doi.org/10.1109/TNET.2011.2120618.

[211] Richard W. Rothery. "Car following models". In: *Trac Flow Theory* (1992). URL: http://live.iugaza.edu/NR/rdonlyres/Civil-and-Environmental-Engineering/1-225JFall2002/C8DCAE43-FEE6-4DFD-8F81-6D7F47E72135/0/carfollowinga.pdf (visited on 01/10/2016).

[212] Sebastian Ruder. "An Overview of Multi-Task Learning in Deep Neural Networks". In: *arXiv preprint arXiv:1706.05098* (2017). http://sebastianruder.com/multi-task/. URL: https://arxiv.org/abs/1706.05098 (visited on 06/27/2017).

[213] Issam S. Strub and Alexandre M. Bayen. "Weak formulation of boundary conditions for scalar conservation laws: An application to highway traffic modelling". In: *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal* 16.16 (2006), pp. 733–748.

[214] A. W. Sadek. "Artificial intelligence in transportation: information for application". In: *Transportation Research Board Circular (E-C113), TRB, National Research Council, Washington, DC. http://onlinepubs. trb. org/onlinepubs/circulars/ec113. pdf* (2007).

[215] Avishek Saha et al. "Online learning of multiple tasks and their relationships". In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. 2011, pp. 643–651.

[216] Mohammad Saifuzzaman et al. "Understanding incident impact on traffic variables to reduce false incident detection". In: *Proceedings of the 40th Australasian Transport Research Forum (ATRF)*. Darwin, Northern Territory, Australia, Oct. 30, 2018.

[217] Christian M. Schneider et al. "Unravelling daily human mobility motifs". In: *Journal of The Royal Society Interface* 10.84 (2013). ISSN: 1742-5689. DOI: 10.1098/rsif.2013.0246. URL: http://rsif.royalsocietypublishing.org/content/10/84/20130246.

[218] Behrooz Shahsavari and Pieter Abbeel. "Short-Term Traffic Forecasting: Modeling and Learning Spatio-Temporal Relations in Transportation Networks Using Graph Neural Networks". Master's Thesis. EECS Department, University of California, Berkeley, Dec. 2015. URL: http://www2.eecs.berkeley.edu/Pubs/TechRpts/2015/EECS-2015-243.html.

[219] Shai Shalev-Shwartz and Ambuj Tewari. "Stochastic methods for 1-regularized loss minimization". In: *Journal of Machine Learning Research* 12 (Jun 2011), pp. 1865–1892. URL: http://www.jmlr.org/papers/v12/shalev-shwartz11a.html (visited on 07/28/2016).

[220] Y. Sheffi. *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods*. Prentice-Hall, Englewood Cliffs, NJ, 1985.

[221] Yoram Shiftan, Shlomo Bekhor, and Gila Albert. "Route choice behaviour with pre-trip travel time information". In: *IET intelligent transport systems* 5.3 (2011), pp. 183–189. URL: http://digital-library.theiet.org/content/journals/10.1049/iet-its.2010.0062 (visited on 12/11/2015).

[222] Noah Simon et al. "A sparse-group lasso". In: *Journal of Computational and Graphical Statistics* 22.2 (2013), pp. 231–245.

[223] Tomas Singliar. "Machine Learning Solutions for Transportation Networks". PhD thesis. Pittsburgh, PA, USA: University of Pittsburgh, 2008.

[224] C. Siripanpornchana, S. Panichpapiboon, and P. Chaovalit. "Effective variables for urban traffic incident detection". In: *2015 IEEE Vehicular Networking Conference (VNC)*. Dec. 2015, pp. 190–195. DOI: 10.1109/VNC.2015.7385576.

[225] Alexander Skabardonis, Pravin Varaiya, and Karl Petty. "Measuring Recurrent and Non-recurrent Traffic Congestion". In: *Transportation Research Record: Journal of the Transportation Research Board* 1856 (2003), pp. 118–124. DOI: 10.3141/1856-12. URL: https://doi.org/10.3141/1856-12.

[226] Nikolai V Smirnov. "On the estimation of the discrepancy between empirical curves of distribution for two independent samples". In: *Bulletin Mathématique de l'Université de Moscou* 2.2 (1939), pp. 3–14.

[227] Brian L. Smith, Billy M. Williams, and R. Keith Oswald. "Comparison of parametric and nonparametric models for traffic flow forecasting". In: *Transportation Research Part C: Emerging Technologies* 10.4 (2002), pp. 303–321. URL: http://www.sciencedirect.com/science/article/pii/S0968090X02000098 (visited on 04/30/2015).

[228] Chaoming Song et al. "Limits of predictability in human mobility". In: *Science* 327.5968 (2010), pp. 1018–1021. ISSN: 0036-8075. DOI: 10.1126/science.1177170. URL: http://science.sciencemag.org/content/327/5968/1018.

[229] Daniel Sperling. *Three Revolutions: Steering Automated, Shared, and Electric Vehicles to a Better Future*. Island Press, 2018.

[230] Dipti Srinivasan, Min Chee Choy, and Ruey Long Cheu. "Neural networks for real-time traffic signal control". In: *IEEE Transactions on Intelligent Transportation Systems* 7.3 (2006), pp. 261–272.

[231] Linda Steg and Robert Gifford. "Sustainable transportation and quality of life". In: *Journal of Transport Geography* 13.1 (2005), pp. 59–69. ISSN: 0966-6923. DOI: https://doi.org/10.

1016/j.jtrangeo.2004.11.003. URL: http://www.sciencedirect.com/science/article/pii/S0966692304000870.

[232] Yorgos Stephanedes, A.P. Chassiakos, and Panos Michalopoulos. "Comparative performance evaluation of incident detection algorithms". In: *Transportation Research Record* 1360 (1992), pp. 50–57.

[233] Hongyu Sun et al. "Use of Local Linear Regression Model for Short-Term Traffic Forecasting". In: *Transportation Research Record: Journal of the Transportation Research Board* 1836 (Jan. 1, 2003), pp. 143–150. ISSN: 0361-1981. DOI: 10.3141/1836-18. URL: http://trrjournalonline.trb.org/doi/abs/10.3141/1836-18 (visited on 07/26/2016).

[234] S. Sun, C. Zhang, and G. Yu. "A Bayesian Network Approach to Traffic Flow Forecasting". In: *IEEE Transactions on Intelligent Transportation Systems* 7.1 (Mar. 2006), pp. 124–132. ISSN: 1524-9050. DOI: 10.1109/TITS.2006.869623. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1603558 (visited on 07/25/2016).

[235] Shiliang Sun. "Traffic flow forecasting based on multitask ensemble learning". In: *Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation*. ACM, 2009, pp. 961–964.

[236] Shiliang Sun, Rongqing Huang, and Ya Gao. "Network-Scale Traffic Modeling and Forecasting with Graphical Lasso and Neural Networks". In: *Journal of Transportation Engineering* 138.11 (2012), pp. 1358–1367. DOI: 10.1061/(ASCE)TE.1943-5436.0000435. URL: https://ascelibrary.org/doi/abs/10.1061/%28ASCE%29TE.1943-5436.0000435.

[237] Souhaib Ben Taieb et al. "A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition". In: *Expert Systems with Applications* 39.8 (2012), pp. 7067–7083. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2012.01.039. URL: http://www.sciencedirect.com/science/article/pii/S0957417412000528.

[238] Alireza Talebpour and Hani S. Mahmassani. "Influence of connected and autonomous vehicles on traffic flow stability and throughput". In: *Transportation Research Part C: Emerging Technologies* 71 (2016), pp. 143–163. ISSN: 0968-090X. DOI: https://doi.org/10.1016/j.trc.2016.07.007. URL: http://www.sciencedirect.com/science/article/pii/S0968090X16301140.

[239] Chris Tampère, Serge P. Hoogendoorn, and Bart Van Arem. "Capacity funnel explained using the human-kinetic traffic flow model". In: *Traffic and Granular Flow'03*. Springer, 2005, pp. 189–197. URL: http://link.springer.com/chapter/10.1007/3-540-28091-X_15 (visited on 01/10/2016).

[240] The European Parliament and the Council of the European Union. *Directive 2010/40/EU*. The European Parliament and the Council of the European Union, July 7, 2010. URL: http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2010:207:0001:0013:EN:PDF.

[241] Johann Heinrich von Thünen. "Der isolierte Staat in Beziehung auf Nationalökonomie und Landwirtschaft". In: *Gustav Fisher, Jena, Germany* (1826).

[242] Robert Tibshirani. "Regression Shrinkage and Selection via the Lasso". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 58.1 (1996), pp. 267–288. ISSN: 0035-9246. URL: http://www.jstor.org/stable/2346178 (visited on 01/11/2016).

[243] Robert Tibshirani et al. "Strong rules for discarding predictors in lasso-type problems". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 74.2 (2012), pp. 245–266.

[244] Alexandre Torday. "Simulation-based decision support system for real time traffic management". In: *89th Transportation Research Board Annual Meeting*. Transportation Research Board-TRB, 2010, 13–p.

[245] Transportation Research Board. *75 Years of the Fundamental Diagram for Traffic Flow Theory: Greenshields Symposium*. Transportation research circular. Woods Hole, Massachusetts: Transportation Research Board, 2011. URL: http://www.trb.org/Publications/Blurbs/165625.aspx.

[246] TRB Transportation Research Board. *75 Years of the Fundamental Diagram for Traffic Flow Theory: Greenshields Symposium | Blurbs | Main*. URL: http://www.trb.org/Main/Blurbs/165625.aspx (visited on 01/16/2016).

[247] Martin Treiber and Arne Kesting. "Traffic flow dynamics". In: *Traffic Flow Dynamics: Data, Models and Simulation, Springer-Verlag Berlin Heidelberg* (2013). URL: https://link.springer.com/book/10.1007%2F978-3-642-32460-4 (visited on 05/05/2015).

[248] Yoshimasa Tsuruoka, Jun'ichi Tsujii, and Sophia Ananiadou. "Stochastic gradient descent training for l1-regularized log-linear models with cumulative penalty". In: *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*. Association for Computational Linguistics, 2009, pp. 477–485. URL: http://dl.acm.org/citation.cfm?id=1687946 (visited on 07/28/2016).

[249] United Nations Department of Economic and Social Affairs. *2018 Revision of World Urbanization Prospects*. 2018. URL: https://www.un.org/development/desa/publications/2018-revision-of-world-urbanization-prospects.html (visited on 10/19/2018).

[250] CP Van Hinsbergen, JW Van Lint, and FM Sanders. "Short term traffic prediction models". In: *Proceedings of the 14th World Congress on Intelligent Transport Systems (ITS)*. Beijing, Oct. 2007.

[251] V. Vapnik. "Principles of Risk Minimization for Learning Theory". In: *Proceedings of the 4th International Conference on Neural Information Processing Systems*. NIPS'91. event-place: Denver, Colorado. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1991, pp. 831–838. ISBN: 1-55860-222-4. URL: http://dl.acm.org/citation.cfm?id=2986916.2987018.

[252] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Berlin, Heidelberg: Springer-Verlag, 1995. ISBN: 0-387-94559-8.

[253] Jeffrey S Vitter. "Random sampling with a reservoir". In: *ACM Transactions on Mathematical Software (TOMS)* 11.1 (1985), pp. 37–57.

[254] Eleni I. Vlahogianni, John C. Golias, and Matthew G. Karlaftis. "Short-term traffic forecasting: Overview of objectives and methods". In: *Transport reviews* 24.5 (2004), pp. 533–557. URL: http://www.tandfonline.com/doi/abs/10.1080/0144164042000195072 (visited on 01/15/2016).

[255] Eleni I. Vlahogianni, Matthew G. Karlaftis, and John C. Golias. "Optimized and meta-optimized neural networks for short-term traffic flow prediction: A genetic approach". In: *Transportation Research Part C: Emerging Technologies* 13.3 (June 2005), pp. 211–234. ISSN: 0968090X. DOI: 10.1016/j.trc.2005.04.007. URL: http://linkinghub.elsevier.com/retrieve/pii/S0968090X05000276 (visited on 07/25/2016).

[256] Eleni I. Vlahogianni, Matthew G. Karlaftis, and John C. Golias. "Short-term traffic forecasting: Where we are and where we're going". In: *Transportation Research Part C: Emerging Technologies*. Special Issue on Short-term Traffic Flow Forecasting 43, Part 1 (June 2014), pp. 3–19. ISSN: 0968-090X. DOI: 10.1016/j.trc.2014.01.005. URL: http://www.sciencedirect.com/science/article/pii/S0968090X14000096 (visited on 05/22/2015).

[257] Peter Wagner et al. "Fundamental Diagram of Traffic Flows on Urban Roads". In: *Transportation Research Record: Journal of the Transportation Research Board* 2124 (2009), pp. 213–221. DOI: 10.3141/2124-21. URL: https://doi.org/10.3141/2124-21.

[258] Haizhong Wang et al. "Stochastic modeling of the equilibrium speed–density relationship". In: *Journal of Advanced Transportation* 47.1 (May 2011), pp. 126–150. DOI: 10.1002/atr.172. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/atr.172.

[259] Jie Wang et al. "Lasso Screening Rules via Dual Polytope Projection". In: *Advances in Neural Information Processing Systems 26*. Ed. by C. J. C. Burges et al. Curran Associates, Inc., 2013, pp. 1070–1078. URL: http://papers.nips.cc/paper/4892-lasso-screening-rules-via-dual-polytope-projection.pdf.

[260] J G Wardrop. "Some theoretical aspects of road traffic research." In: *Proceedings of the Institution of Civil Engineers* 1.3 (May 1, 1952), pp. 325–362. DOI: 10.1680/ipeds.1952.11259. URL: http://www.icevirtuallibrary.com/doi/abs/10.1680/ipeds.1952.11259 (visited on 01/08/2016).

[261] Tao Wen et al. "Integrated Incident Decision-Support using Traffic Simulation and Data-Driven Models". In: *Transportation Research Record* 2672.42 (2018), pp. 247–256. DOI: 10.1177/0361198118782270. URL: https://doi.org/10.1177/0361198118782270.

[262] Gerhard Widmer and Miroslav Kubat. "Learning in the presence of concept drift and hidden contexts". In: *Machine learning* 23.1 (1996), pp. 69–101.

[263] Billy M. Williams and Lester A. Hoel. "Modeling and Forecasting Vehicular Traffic Flow as a Seasonal ARIMA Process: Theoretical Basis and Empirical Results". In: *Journal of Transportation Engineering* 129.6 (2003), pp. 664–672. ISSN: 0733-947X. DOI: 10.1061/(ASCE)0733-947X(2003)129:6(664). URL: http://dx.doi.org/10.1061/(ASCE)0733-947X(2003)129:6(664) (visited on 07/26/2016).

[264] A. Willsky et al. "Dynamic model-based techniques for the detection of incidents on freeways". In: *IEEE Transactions on Automatic Control* 25.3 (June 1980), pp. 347–360. ISSN: 0018-9286. DOI: 10.1109/TAC.1980.1102392.

[265] World Health Organization. *Save lives: a road safety technical package*. World Health Organization, 2017. URL: http://www.who.int/iris/handle/10665/255199.

[266] Ning Wu. "A new approach for modeling of Fundamental Diagrams". In: *Transportation Research Part A: Policy and Practice* 36.10 (2002), pp. 867–884. ISSN: 0965-8564. DOI: https://doi.org/10.1016/S0965-8564(01)00043-X. URL: http://www.sciencedirect.com/science/article/pii/S096585640100043X.

[267] Yuankai Wu et al. "A hybrid deep learning based traffic flow prediction method and its understanding". In: *Transportation Research Part C: Emerging Technologies* 90 (2018), pp. 166–180. ISSN: 0968-090X. DOI: https://doi.org/10.1016/j.trc.2018.03.001. URL: http://www.sciencedirect.com/science/article/pii/S0968090X18302651.

[268] Jingxin Xia and Mei Chen. "A Nested Clustering Technique for Freeway Operating Condition Classification". In: *Computer-Aided Civil and Infrastructure Engineering* 22.6 (2007), pp. 430–437. ISSN: 1467-8667. DOI: 10.1111/j.1467-8667.2007.00498.x. URL: http://dx.doi.org/10.1111/j.1467-8667.2007.00498.x.

[269] Jingxin Xia, Wei Huang, and Jianhua Guo. "A clustering approach to online freeway traffic state identification using ITS data". In: *KSCE Journal of Civil Engineering* 16.3 (Feb. 29, 2012), pp. 426–432. ISSN: 1226-7988, 1976-3808. DOI: 10.1007/s12205-012-1233-1. URL: http://link.springer.com/article/10.1007/s12205-012-1233-1 (visited on 08/03/2016).

[270] Yuanchang Xie, Yunlong Zhang, and Zhirui Ye. "Short-Term Traffic Volume Forecasting Using Kalman Filter with Discrete Wavelet Decomposition". In: *Computer-Aided Civil and Infrastructure Engineering* 22.5 (2007), pp. 326–334. URL: http://onlinelibrary.wiley.com/doi/10.1111/j.1467-8667.2007.00489.x/pdf (visited on 01/14/2016).

[271] Haiqin Yang et al. "Online learning for group lasso". In: *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*. 2010, pp. 1191–1198. URL: http://machinelearning.wustl.edu/mlpapers/paper_files/icml2010_YangXKL10.pdf (visited on 07/09/2015).

[272] Ming Yuan and Yi Lin. "Model selection and estimation in regression with grouped variables". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68.1 (2006), pp. 49–67.

[273] Zheng Zhao et al. "LSTM network: a deep learning approach for short-term traffic forecast". In: *IET Intelligent Transport Systems* 11.2 (Mar. 2017), 68–75(7). ISSN: 1751-956X. URL: http://digital-library.theiet.org/content/journals/10.1049/iet-its.2016.0208.

[274] Hui Zou. "The Adaptive Lasso and Its Oracle Properties". In: *Journal of the American Statistical Association* 101.476 (2006), pp. 1418–1429. DOI: 10.1198/016214506000000735. URL: https://doi.org/10.1198/016214506000000735.