MODELAGEM, PREVISÃO E CONTROLE DE CONGESTIONAMENTO DE TRÂNSITO

ANNA IZABEL JOÃO TOSTES RIBEIRO

MODELAGEM, PREVISÃO E CONTROLE DE CONGESTIONAMENTO DE TRÂNSITO

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Doutor em Ciência da Computação.

Orientador: Antonio Alfredo Ferreira Loureiro
Coorientadora: Fátima de Lima Procópio Duarte Figueiredo

Belo Horizonte
Setembro de 2015

ANNA IZABEL JOÃO TOSTES RIBEIRO

MODELING, PREDICTING AND CONTROLLING TRAFFIC JAM

Thesis presented to the Graduate Program in Computer Science of the Federal University of Minas Gerais in partial fulfillment of the requirements for the degree of Doctor in Computer Science.

Advisor: Antonio Alfredo Ferreira Loureiro
Co-Advisor: Fátima de Lima Procópio Duarte Figueiredo

Belo Horizonte
September 2015

Ficha catalográfica elaborada pela Biblioteca do ICEx - UFMG

Ribeiro, Anna Izabel João Tostes.

R484m Modelagem, previsão e controle de congestionamento de trânsito em redes veiculares / Anna Izabel João Tostes Ribeiro. – Belo Horizonte, 2015.
xxv, 177 f.: il.; 29 cm.

Tese (doutorado) - Universidade Federal de Minas Gerais – Departamento de Ciência da Computação.

Orientador: Antônio Alfredo Ferreira Loureiro
Coorientador: Fátima de Lima Procópio de Figueiredo

1. Computação - Teses. 2. Computação ubíqua. 3. Redes de sensores sem fio. 4. Redes veiculares. 5. Trânsito – Congestionamento. 6. Redes bayesianas. I. Orientador. II. Coorientador. III. Título.

CDU 519.6*22(043)

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Modelagem, previsão e controle de congestionamento de trânsito em redes veiculares

ANNA IZABEL JOÃO TOSTES RIBEIRO

Tese defendida e aprovada pela banca examinadora constituída pelos Senhores:

Antonio

Prof. Antonio Alfredo Ferreira Loureiro - Orientador
Departamento de Ciência da Computação - UFMG

Fátima de [illegible] Figueiredo

Profa. Fátima de Lima Procópio D. Figueiredo - Coorientador
Departamento de Ciência da Computação - PUC-MG

Prof. Felipe Maia Galvão França
COPPE - UFRJ

Leandro Villas

Prof. Leandro Aparecido Villas
Instituto de Computação - UNICAMP

Prof. Pedro Olmo Stancioli Vaz de Melo
Departamento de Ciência da Computação - UFMG

Raquel Mini

Profa. Raquel Aparecida de Freitas Mini
Departamento de Ciência da Computação - PUC-MG

Prof. Renato Martins Assunção
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 17 de setembro de 2015.

Ao amor da minha vida, Ana Gabriela,
minha esposa e fiel companheira, do meu lado,
ontem, hoje e sempre.

Resumo

Congestionamento é um problema comum dos centros urbanos, que ocasiona perdas econômicas e de tempo, maior consumo de combustível e maiores emissões de CO_2. A literatura indica que os Sistemas de Transporte Inteligentes (ITS) vão influenciar no trânsito. As redes veiculares, por sua vez, surgem como um componente do ITS, possibilitando a comunicação cooperativa entre veículos e de veículos com a infraestrutura. Além do congestionamento, a maioria das aplicações depende de localização e mobilidade das informações de veículos, que é difícil de estimar.

Para superar esses problemas, esta tese aborda congestionamento de trânsito explorando seu comportamento através de dados históricos, incidentes de trânsito, transições de fase, modelo analítico e de previsão, e serviços. A afirmação da tese é modelar, prever e controlar o congestionamento de trânsito usando redes veiculares. Os objetivos são: (1) coletar diferentes fontes de dados sobre as condições de trânsito; (2) caracterizar condições de trânsito reais e correlacionar com essas fontes de dados; (3) prever as condições de trânsito e seus efeitos; (4) propor serviços de sugestão de rotas para fazer um balanceamento de carga do fluxo de veículos na cidade. Todos esses objetivos fazem parte de um Sistema de Gerenciamento de Trânsito (SGT). Além das contribuições em todas os módulos de um SGT, a diferença para outras abordagens é a coleta e a análise de informações em tempo real sobre as condições e os incidentes de trânsito por meio de serviços de mapas.

Os resultados mostraram que os modelos de previsão melhoram a acurácia em até 80% para prever o congestionamento com base em dados históricos e de tempo real. Adicionalmente, a partir da correlação dos dados de sistemas de sensores participativos com as condições de tráfego intenso, a acurácia dos modelos de previsão foi melhorada de 80% para até 90%, em geral. Essas soluções reduzem o tempo de viagem em até 56%, e o consumo de combustível e a emissão de CO_2 em 18%, em média.

Palavras-chave: Rede veicular, engarrafamento, congestionamento de trânsito, serviços de mapa, controle de congestionamento, rede bayesiana, regressão logística.

Abstract

Traffic jam is a common contemporary society problem in urban areas, leading to economic losses, waste of time in traffic, higher fuel consumption and major CO_2 emissions. The literature indicates that Intelligent Transportation Systems (ITS) will influence the traffic. In this context, vehicular networks emerge as a component of ITS, allowing the cooperative communication between vehicles and vehicles with infrastructure. Besides the congestion problem, most applications rely on location and mobility information of vehicles, which are hard to estimate.

To overcome such matters, this thesis tackles traffic jam by exploiting the traffic behavior through historical data, traffic incidents, phase transitions, analytical and prediction models, and services. The statement is to model, to predict, and to control traffic jams using vehicular networks. Our goals are: (i) to collect different data sources related to traffic conditions; (ii) to characterize real traffic conditions and to correlate with these data sources; (iii) to predict the traffic flow and its effects; (iv) to propose a suggestion of routes service that makes the load balance of vehicles flow in the city. All of these goals are part of a Traffic Management System (TMS). Besides the contributions in all modules of a TMS, the difference to other approaches is the collection and analysis of real-time data about traffic conditions through map services.

Our results show that our prediction models improve the accuracy in up to 80% to predict the traffic jam based on historical and real-time data. In addition, by correlating data from participatory sensing systems with the intense traffic conditions, the accuracy of our prediction models was improved from 80% up to 90%, in general. These solutions reduce the travel time in up to 56%, and the fuel consumed and the CO_2 emissions in up to 18%, in average.

Palavras-chave: Vehicular network, traffic jam, traffic congestion, map services, congestion control, bayesian network, logistic regression.

List of Figures

List of Tables

List of Algorithms

Contents

Chapter 1

Introduction

At the beginning there were not as many cars nor big roads as nowadays. After a while, there were more cars, but no infrastructure improvement. The consequence to this "evolution" is traffic jam: a common contemporaneous problem that leads to economic losses, waste of time, more fuel consumption, and more environmental issues. In the context of a smart city, dealing with the traffic jam in urban areas is a challenge, especially regarding traffic prediction, when the goal is to perform short-term forecasting. Although we can find in the literature some advances in algorithms and techniques to tackle this problem [Araújo et al., 2014; Endarnoto et al., 2011; Fortz and Thorup, 2000; Google, 2009; Krumm, 2010; Microsoft Research, 2013b; Niu et al., 2014; Schougaard, 2007; Silva et al., 2013b; Soares et al., 2014b; Su et al., 2000], we still need innovative solutions, such as those that consider different information sources about the city dynamics and urban social behavior, leading to richer possibilities. This problem is explored in this thesis.

1.1 Motivation

A disturbing issue of large urban centers is traffic jam. Despite the technological advances in vehicles improving the experience of drivers and passengers, traffic jam leads to economic losses, decreases the overall productivity and impacts negatively in the environment [Bauza et al., 2010]. Consequently, drivers waste too much time in traffic, increasing both fuel consumption and CO_2 emissions [Capone and Martignon, 2007].

The problem is that, no matter the proposed solution, traffic jam is still getting worst. One example of "congested city" in United States is Chicago, which was the number one in road congestion, according to the Urban Mobility Report, issued by the Texas Transportation Institute in 2011 [Tribune, 2011]. Beyond the time a driver normally takes to travel without delays, the national average for traffic delays in the U.S. was 34 hours while commuters in the Chicago area spent an additional 70 hours behind the wheels in 2009, and this is just increasing (55 hours of wasted time in 1999, and 18 hours in 1982).

Another example is New York City that, according to the NYMTC report [Council, 2013], has the third lowest daily vehicle miles traveled[1] (VMT) per capita. The reasons for that include the high population density and the high proportion of transit use, among others. Notice that Manhattan region has 10,702,575 VMT daily, and this is not exclusive to New York. As an illustration, Figure 1.1 shows that New York and Los Angeles have the highest travel volumes in comparison with other cities and a very large area (a metropolitan statistical area with over three million residents).

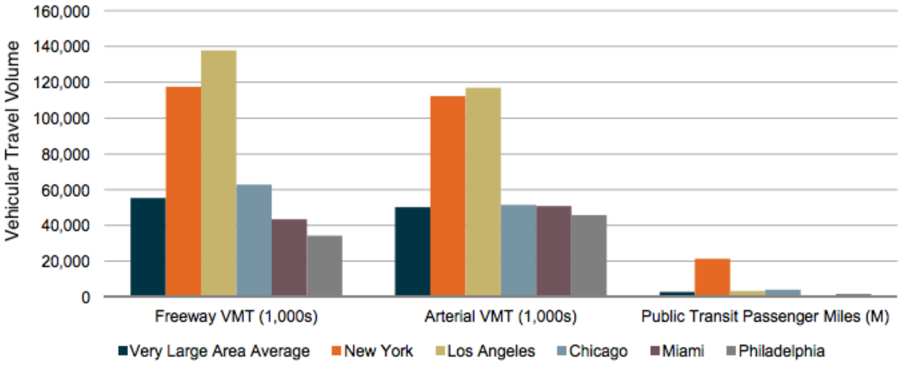


Figure 1.1: Vehicular travel volumes in New York and other areas [Council, 2013].

In this context, the literature indicates four important areas that will influence the traffic [Wang et al., 2013]: (i) road infrastructure; (ii) urban planning and design; (iii) supply and demand; and (iv) traffic management system. The first area consists in making physical infrastructure modifications such as junction improvements, building tunnels, viaducts, bridges, local-express lanes, reversible lanes, and others. The second area has more impact in future congestion, but it is also the hardest to implement in big cities. It relies on applying some practices of city planning and urban design, such as zone laws, grid plans, car-free cities, and having a better urban transportation system. The third area aims to increase the road capacity, which can happen either increasing the infrastructure of roads, or reducing the demand (of traffic) by introducing park restrictions, or road pricing and policy approaches (public transportation and cycling

[1] Vehicle Miles Traveled (VMT) is the sum of distances traveled by all motor vehicles in a specified region [Council, 2013].

promotions, for instance). Whereas other areas are less practical and might have more impact on our society (political, economic and culturally speaking), the fourth area looks more attractive considering smart cities: the development of Intelligent Transportation Systems (ITS) [Boukerche et al., 2011].

ITS use infrastructure sensors to monitor the traffic condition in a vehicular environment and provide applications, such as collision detection systems, with traffic-related information and ubiquitous connectivity to the Internet [Briesemeister et al., 2000; Hartenstein and Laberteaux, 2008; Li and Wang, 2007]. An important component of ITS is the Vehicular Ad Hoc Network (VANET), which focuses on the cooperative communication between vehicles and roadside units. Its main characteristics are the high mobility of nodes, intermittent links, and stringent latency requirements. Unlike traditional mobile ad hoc networks, a VANET does not have energy constraints nor limited processing power. Its dynamic nature, i.e. the constant change in the network topology, raises challenges about communication. These challenges are summarized in [Domingos Da Cunha et al., 2014].

1.2 Statement and Goals

The statement of this thesis is to model, predict and control traffic jam. The following are the goals of this thesis:

1. Collect different data sources about traffic conditions;

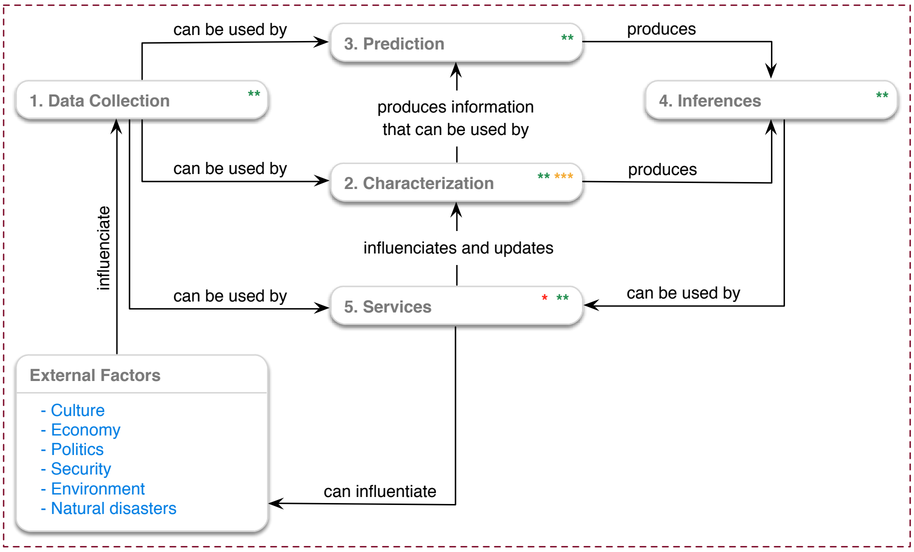
2. Characterize real traffic conditions and correlate them with other data sources;

3. Predict the traffic flow and its effects;

4. Propose a suggestion of routes in order to achieve the load balance of vehicles' flow in the city.

All these goals can be seen as part of a Traffic Management System (TMS). Figure 1.2 illustrates the modules in TMS and the logical interconnection among them. We can observe the following modules: (i) data collection; (ii) characterization; (iii) prediction; (iv) inferences; and (v) services. In addition, Figure 1.3 shows more details about the thesis proposal and contributions to these different TMS modules.

In module 1, we have the heterogeneous data collection, in which different data sources related with traffic conditions are gathered by using traditional Application Programming Interfaces (APIs), new methodologies or streams of data. In this module, we have challenges in terms of clock synchronization worldwide, data quality,



* Thesis collaboration: producing bachelor and master thesis.

** This PhD thesis contribution.

*** Correlation with traffic jam.

Figure 1.2: The architecture of a Traffic Management System.

representativeness, variations in the availability of data, and big data questions such as *"is it really necessary to store all data gathered?"*

In module 2, we observe data characterization. The main goal of this module is to understand how data gathered in module 1 are correlated and how they influence the traffic conditions. It is important to understand their spatio-temporal distributions, identifying traffic patterns and producing metrics for a better detection and to control traffic jams. The challenge is how to deal with the distinct granularity of data, that is, we have different time intervals of data collection with different amount of data per time interval. *"How to correlate such huge amount of data, with different granularities?"*

In module 3, we have prediction models. Here, the goal is not just to identify when and where traffic jams will occur, but more important is to answer to the question: *how does it spread in the city?* Therefore, we have two challenges: (a) when and where traffic jams will occur, and (b) how the data source spreads in the city. Examples of prediction models are short-term traffic forecasting and mobility prediction models.

In addition to module 3, there is the module 4, where we have the inferences of the prediction models. Recall that we can vary the time horizon for the traffic forecasted in a near future. This time horizon is the fixed point of time in the future at which the prediction will occur. About the application of this module, inferred data can be used as new data sources in the services (module 5).

Finally, in the module 5, we have services that take advantage of data sources

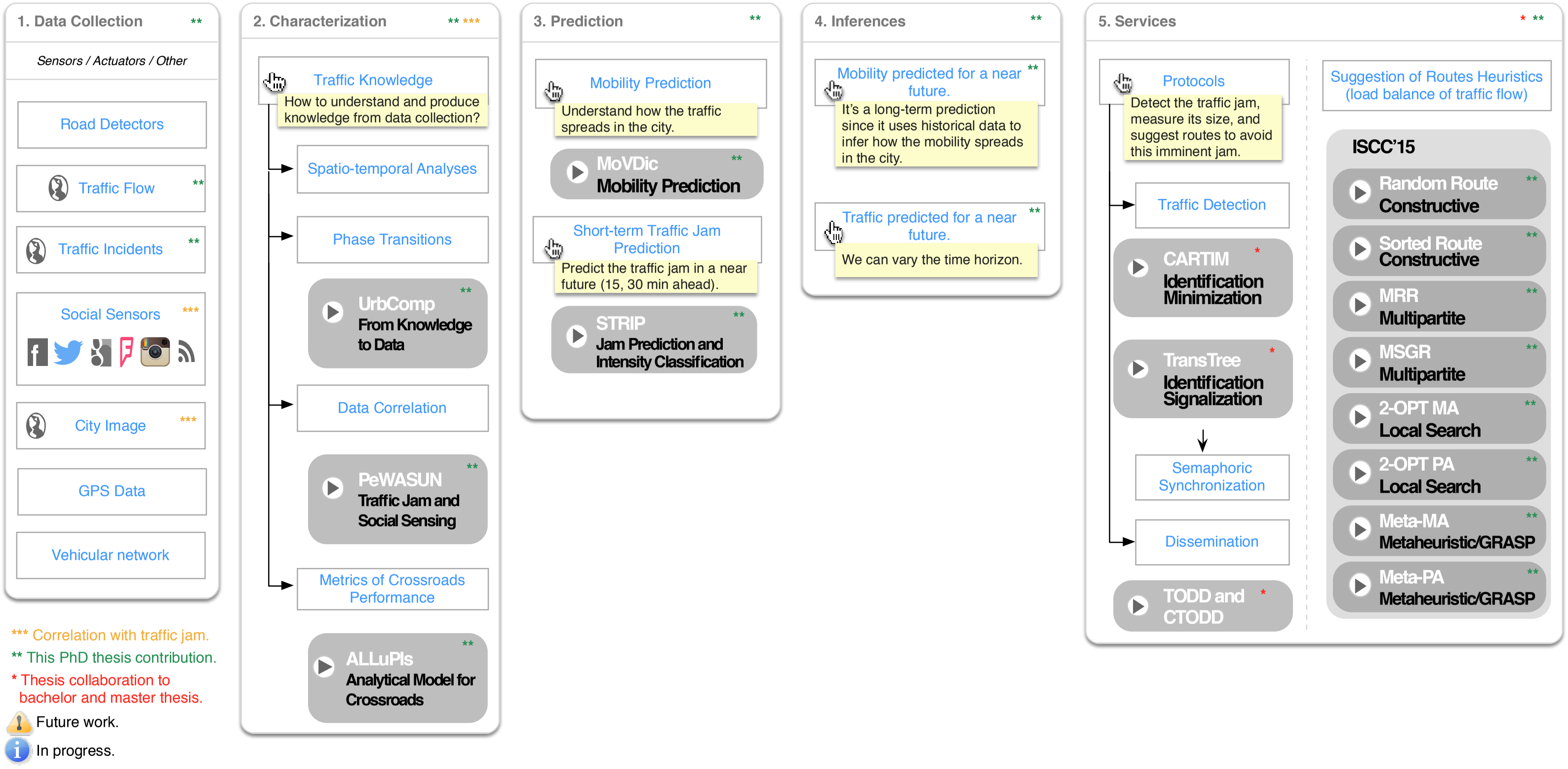

1. Data Collection **
Sensors / Actuators / Other
Road Detectors
Traffic Flow **
Traffic Incidents **
Social Sensors ***
City Image ***
GPS Data
Vehicular network
2. Characterization ** ***
Traffic Knowledge
How to understand and produce knowledge from data collection?
Spatio-temporal Analyses
Phase Transitions
UrbComp From Knowledge to Data **
Data Correlation
PeWASUN Traffic Jam and Social Sensing **
Metrics of Crossroads Performance
ALLuPIs Analytical Model for Crossroads **
3. Prediction **
Mobility Prediction
Understand how the traffic spreads in the city.
MoVDic Mobility Prediction **
Short-term Traffic Jam Prediction
Predict the traffic jam in a near future (15, 30 min ahead).
STRIP Jam Prediction and Intensity Classification **
4. Inferences **
Mobility predicted for a near future. **
It's a long-term prediction since it uses historical data to infer how the mobility spreads in the city.
Traffic predicted for a near future. **
We can vary the time horizon.
5. Services * **
Protocols
Detect the traffic jam, measure its size, and suggest routes to avoid this imminent jam.
Traffic Detection
CARTIM Identification Minimization *
TransTree Identification Signalization *
Semaphoric Synchronization
Dissemination
TODD and CTODD *
Suggestion of Routes Heuristics (load balance of traffic flow)
ISCC'15
Random Route Constructive **
Sorted Route Constructive **
MRR Multipartite **
MSGR Multipartite **
2-OPT MA Local Search **
2-OPT PA Local Search **
Meta-MA Metaheuristic/GRASP **
Meta-PA Metaheuristic/GRASP **
*** Correlation with traffic jam.
** This PhD thesis contribution.
* Thesis collaboration to bachelor and master thesis.
Future work.
In progress.

Figure 1.3: Proposal details and contributions of the thesis in different modules.

(module 1) and inferences (module 4) to manage traffic jams. Examples of services are: (i) VANET protocols to detect and control intense traffic conditions; and (ii) mechanisms of route suggestions that make the load balance of vehicle flows in the city. In the first service, there are protocols to detect and control traffic jam, others to make the semaphore synchronization, and another one to disseminate the information of traffic jams as an alert of congestion.

1.3 Contributions

In this thesis, we have contributions in all modules of the traffic management system. More specifically, we have the following:

- We exploit underexploited sources of real-time traffic flow from map services, from which we have analyzed the traffic behavior through phase transitions and traffic patterns [Tostes et al., 2013].

- We analyze the traffic conditions according to social sensing from Foursquare and Instagram [Tostes et al., 2014], showing that both sources are correlated with intense traffic flow for Manhattan region, New York.

- We introduce and validate an analytical model, named ALLuPIs [Tostes et al., 2012] that provides a real-time measure of performance for road intersections.

- We introduce and validate two prediction models: (i) MoVDic (Mobility Vehicular preDiction model) [Tostes et al., 2015a], which predicts how the mobility of vehicles spreads in the city; and (ii) STRIP (Short-term TRaffIc jam Prediction model) [Tostes et al., 2013, 2015b], which forecast the future traffic flow in each road based on historical traffic flows in that same road. MoVDic has been validated through a realistic mobility trace from Cologne, Germany, while STRIP has been evaluated for different cities, such as Chicago and New York.

- We propose and evaluate several route suggestion mechanisms while making the load balance of flows in the city in Tostes et al. [2015].

1.4 Origins of the Material

Partial results of this thesis have been previously presented in conferences (or are under submission). We now give an overview of published research, organized by chapters.

The red asterisk represents contribution of this thesis, while unmarked present the collaboration with other students.

- **Chapter 2:** we review the state of the art and compare our proposed protocols TransTree, CARTIM, and TODD with the related work.

  – Silva, T. H., Vaz de Melo, P. O. S., Almeida, J. M., Borges Neto, J., Tostes, A. I. J., Celes, C. S. F. S., Mota, S., V. F., Cunha, F. D., Ferreira, A. P. G., Machado, K. L. S., and Loureiro, A. A. F. (2015b). Redes de sensoriamento participativo: Desafios e oportunidades. In *Minicursos / XXXIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 266--315, Porto Alegre. Sociedade Brasileira de Computação*

  – Silva, T. H., da Cunha, F. D., Tostes, A. I. J., Neto, J. B. B., de S Celes, C. S. F., Mota, V. F. S., Ferreira, A. P. G., de Melo, P. O. S. V., Almeida, J. M., and Loureiro, A. A. F. (2015a). Users in the urban sensing process: Challenges and research opportunities (accepted). In *Pervasive Computing: Next Generation Platforms for Intelligent Data Collection**

  – Araújo, G., Queiroz, M., de Lima Procópio Duarte-Figueiredo, F., Tostes, A. I. J., and A.F. Loureiro, A. (2014). CARTIM: a proposal toward identification and minimization of vehicular traffic congestion for VANET. In *19th IEEE Symposium on Computers and Communications (IEEE ISCC 2014)*, Madeira, Portugal*

  – Soares, R. B., Tostes, A. I. J., Nakamura, E. F., and A.F. Loureiro, A. (2014b). An adaptive data dissemination protocol with dynamic next hop selection for vehicular networks. In *19th IEEE Symposium on Computers and Communications (IEEE ISCC 2014)*, Madeira, Portugal*

  – Araújo, G., Duarte-Figueiredo, F., Tostes, A. I. J., and Loureiro, A. A. F. (2014). Um protocolo de identificação e minimização de congestionamentos de tráfego para redes veiculares. In *SBRC 2014**

  – de Brito, M. R., Tostes, A. I. J., Duarte-Figueiredo, F., and Loureiro, A. A. F. (2014b). Simulação e análise de métodos de detecção de congestionamento de veículos em vanet. In *SBRC 2014 - WGRS**

  – de Brito, M. R., Tostes, A. I. J., Duarte-Figueiredo, F., and Loureiro, A. A. F. (2014a). Simulação e análise de congestionamento em redes veiculares. In *CTIC 2014**

  - de Brito, M. R., Silva, B., Tostes, A. I. J., Duarte-Figueiredo, F., and Loureiro, A. A. F. (2015). Transtree: Detecção de congestionamento utilizando redes veiculares. In *SBRC 2015 - WGRS**
  - de Castro, M. S., Tostes, A. I., Duarte-Figueiredo, F., and Loureiro, A. A. F. (2013). Disseminação de mensagens de acidente em redes veiculares. In *SEMISH 2013**

- **Chapter 3:** we describe our proposed methodology to collect traffic flow and incidents from Bing Maps, and social data from Twitter, Foursquare and Instagram.
  - Tostes, A. I. J., de L. P. Duarte-Figueiredo, F., Assunção, R., Salles, J., and Loureiro, A. A. F. (2013). From data to knowledge: City-wide traffic flows analysis and prediction using bing maps. In *Proc. of ACM SIGKDD UrbComp'13*, pages 12:1--12:8, Chicago, Illinois*
  - Tostes, A. I. J., Silva, T. H., Duarte-Figueiredo, F., and A.F. Loureiro, A. (2014). Studying traffic conditions by analyzing foursquare and instagram data. In *PeWaSUN'14)**

- **Chapter 4:** we analyze the traffic flow and traffic incidents in real time from online map, such as Bing Maps and Google Maps. We also analyze social data (e.g., check-ins, when users share their location) and presented a correlation between them and intense traffic flow. Finally, we propose the analytical model of crossroads performance named ALLuPIs. Notice that this chapter also considers the already mentioned publications [Tostes et al., 2013] and [Tostes et al., 2014].
  - Tostes, A. I., Duarte-Figueiredo, F., Almeida, J., and Loureiro, A. A. F. (2012). Modelo analítico de contenção de tráfego em vanet usando dados reais de mobilidade. In *CSBC 2012 - WPerformance*, Curitiba-PR*

- **Chapter 5:** we introduce our proposed prediction model of mobility MoVDic, submitted to [Tostes et al., 2015a], and our prediction model of traffic flow STRIP, submitted to [Tostes et al., 2015b].
  - Tostes, A. I. J., Maia, G., ao, R. A., Duarte-Figueiredo, F., and Loureiro, A. A. (2015a). MoVDic: predicting the mobility of vehicles through information of street detectors [submitted]. In *Ad Hoc Network Journal**
  - Tostes, A. I. J., Silva, T. H., ao, R. A., Figueiredo, F. D., and Loureiro, A. A. (2015b). Strip: A short-term traffic jam prediction model based on online

  maps and social sensors (submitted). In *IEEE 82nd Vehicular Technology Conference (VTC2015-Fall)**

- **Chapter 6:** we present several heuristics to suggest routes that avoid traffic congestion and that makes the load balance of vehicles flow.

  - Tostes, A. I., Maia, G., Duarte-Figueiredo, F., and A.F. Loureiro, A. (2015). Suggestion of routes for vehicles in vehicular networks using the multicommodity flow model. In *20th IEEE Symposium on Computers and Communications (ISCC2015)*, Larnaca, Cyprus*

1.5 Outline

The remainder of this thesis is organized as follows. Chapter 2 outlines the background in vehicular networks as well as a review of the related literature about prediction models, social sensing, and services in TMS. Chapter 3 describes the data collection module. We present our proposed methodology to collect and analyze the traffic flow from online maps. Chapter 4 describes our contributions to the characterization module. We present spatial and temporal analyses of the collected data. Within the scope of the later, we introduce the concept of phase transitions and, then, we analyze the traffic flow for several cities. In addition, we propose our analytical model ALLuPIs. In the following, Chapter 5 introduces our prediction models MoVDic and STRIP. Chapter 6 describes our solutions in the service module of the TMS, i.e., the suggestion of routes with the load balance of flows. Finally, Chapter 7 shows our final remarks and future directions.

Chapter 2

Background and Related Work

Vehicular networks, also called VANETs (Vehicular Ad hoc Networks) are networks formed by vehicles and stationary equipment (Road Side Units - RSUs) located on the roadsides. These networks are characterized by the high mobility of its nodes (vehicles), intermittent links and stringent latency requirements. According to [Karagiannis et al., 2011], these attributes prevent some protocols used in ad hoc networks to be used in vehicular networks, because they do not have satisfactory performance.

Although having another concepts, vehicular networks have five important topics for the understanding of this thesis: (i) architecture; (ii) social sensing; (iii) prediction models; (iv) services to deal with traffic congestion, such as the load balance of traffic flow and protocols (congestion control mechanisms and data dissemination). Next sections presents their main concepts while we review the most correlated approaches.

2.1 Architecture

The architecture of vehicular networks is organized depending on the adopted communication models. There are three distinct communication models as illustrates Figure 2.1. Depending on how the network nodes organize themselves and communicate, the architecture can be: (i) pure ad hoc, (ii) infrastructured, and (iii) hybrid.

Notice that, in the communication model (i), vehicles communicate among themselves without any external support or centralized element, which is known as vehicle-to-vehicle (V2V) communication. In this model, the protocols and applications must consider the high mobility of vehicles, which causes a high frequency of disconnections in the network. Model (ii), in the other hand, is the V2I (Vehicle-to-Infrastructure) communication. It deals with the communication between vehicles and nearby fixed equipment among the road. This model can increase the network connectivity if it has

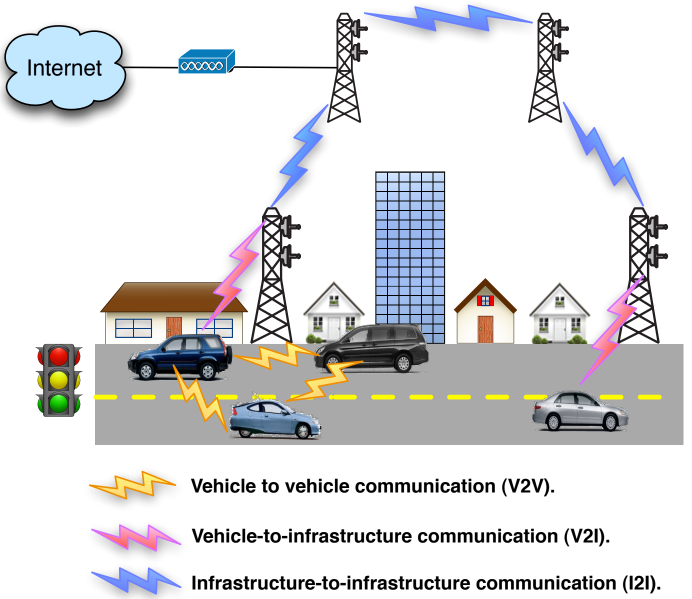

Internet
Vehicle to vehicle communication (V2V).
Vehicle-to-infrastructure communication (V2I).
Infrastructure-to-infrastructure communication (I2I).

Figure 2.1: The architecture of vehicular networks.

a suitable amount of static nodes, although the network cost can be increased. Finally, model (iii) combines both models (i) and (ii) [Hartenstein and Laberteaux, 2008].

In vehicular networks, due to the special nature of the environment, protocols and applications must consider some aspects that characterize this type of network [Domingos Da Cunha et al., 2014]: (i) highly dynamic topology; (ii) the frequently disconnection; (ii) geographical communication; (iv) constrained mobility and prediction; and (v) propagation model to highways, rural and cities. In order to deal with such issues, most protocols and applications rely on location and mobility information of vehicles, which is very difficult to predict keeping in view the nature and pattern of movement of each vehicle [Su et al., 2001]. For instance, the challenges in dissemination protocols, that is, dealing with the broadcast storm and network fragmentation, can be more easily tackled with the knowledge of future directions that vehicles will take when they arrive at an intersection. However, how to forecast this mobility on the fly on such a highly dynamic environment is still a challenge.

In this context, distinct routing protocols have been proposed and evaluated for VANETs [Li and Wang, 2007] concerning whether its performance can satisfy the throughput and the delay requirements of safety and emergency applications. Traditional routing protocols, such as the protocols ad hoc on-demand distance vector (AODV) and the dynamic source routing (DSR) [Nzouonta et al., 2009], can not be used due the route instability that causes packet loss and high overhead. Notwithstanding, geographical routing protocols, e.g. the greedy perimeter stateless routing

(GPSR) protocol, have better path stability, but the matter is how to forward the packet through the next hop. To solve this, the main routing protocols use road information to choose the crossroad to forward the packet. There are works that use prediction models for route prediction, but they do not use real-time information of crossroads behavior or traffic flow inference. This is the main difference for this thesis.

2.2 Social Sensing

Participatory Sensing Systems (PSSs) provides a mobile interface that allows people to share data about the environment (context information) they are in any time and place using any mobile device. Examples of PSSs deployed and functioning at global scale are location-sharing services, such as Foursquare, and photo sharing services, such as Instagram. They can provide valuable information about an aspect of a given city or society in almost real-time, such as its traffic and weather conditions, local parties and festivals, riots, among others integrate user interactions [Silva et al., 2013b].

Data shared in participatory sensing systems have the active participation of users. These systems can be seen as a sort of sensor network, also known as participatory sensor network (PSN) [Silva et al., 2014]. In this network, users plus his/her mobile devices can be considered a sensor, because users carry mobile devices that are able to sense the environment and make relevant observations at a personal level.

PSNs are an example of the interplay between technological networks and social networks, since a key element in a participatory sensor network is the human being. As shown in [Silva et al., 2014] data shared by the sensors in a PSN are associated with users' habits and routines. Thus, these sensors can be considered social sensors, which provide valuable to better understand city dynamics and the urban behavioral patterns of their inhabitants. Many questions emerge when using this new type of data, for example: *Can we use them to better understand traffic conditions?* In fact, this is a very interesting question that is discussed in Chapter 4.

There are several studies about participatory sensing, obtained, for example, from Twitter, in order to better understand city dynamics. The authors in [Frias-Martinez et al., 2012] used a dataset from Twitter and proposed a technique to determine the type of activities that is most common in a city by studying tweeting patterns. [Bollen et al., 2011] studied whether collective mood states derived from Twitter feeds are correlated to the value of the Down Jones Industrial Average over time. [Sakaki et al., 2010] studied the real-time interaction of events in Twitter (e.g. earthquakes), and propose an algorithm to monitor tweets to detect a target event. [Lee and Sumiya, 2010] present a geo-social event detection system to identity local events (e.g., local festivals) by monitoring crowd behaviors indirectly via Twitter.

In the same direction [Silva et al., 2013b] considers the traffic alert system Waze, another example or participatory sensing source, as a sensor network in order to verify its properties. Among the results, the authors show that data sharing is correlated with users' routines, and that Waze data can improve traffic condition understanding. Correlated with [Silva et al., 2013b], in [Endarnoto et al., 2011], the authors used a dataset from Twitter and proposed an information extraction technique to get the data of traffic. The traffic data is presented in map view as a mobile application of Android.

In the health direction, in [Gomide et al., 2011], the authors have analyzed how Dengue epidemic is reflected on Twitter and to what extent that information can be used for the sake of surveillance. They showed that Twitter can be used to predict, spatially and temporally, dengue epidemics by means of clustering.

Although much has been done, no previous effort studied the correlation of information from participatory sensing systems and real traffic conditions data. This is the point that differentiates our work from the previous ones. In Section 4.3 (Chapter 4), we have investigated whether we use can use participatory sensing data (or social data) to better understand traffic conditions.

2.3 Prediction Models

In VANETs, prediction models are used to mainly predict the location and mobility of vehicles, which is difficult to predict in such dynamic environment [Su et al., 2001]. Most of researches use Markov Models, Hidden Markov Models, Variable Order Markov Models and/or Bayes Theory [Uma Nagaraj, 2011]. A Markov Model predicts the future route of the vehicle based on its near term past route, not in its entire trajectory. A Hidden Markov Model is a Markov model with unobservable state, which is well known as hidden states. A Variable-Order Markov Model captures longer regularities while avoiding the size explosion caused by increasing the order of the model. That is, the number of conditioning random variables may vary based on context information. The past states that are independent from the future states can be removed, reducing the number of model parameters.

In this section, we divided the related works in two categories: (i) the prediction of mobility; and (ii) the prediction of traffic flow.

2.3.1 Prediction of Mobility

The knowledge of how the traffic spreads in the city can help agencies or department of transportation to manage the traffic in order to avoid jams. If we know where and when people moves, and which routes they follow, we could know where to focus our strengths and investments. Researchers all over the world tried to predict the traffic mobility using several sources of information and distinct models, some with a good accuracy, but as they do not scale well, they can not be applied in real-time, and some may not perform very well with temporary changes of the surrounding infrastructure, or with lack of data.

Recent years have seen a considerable amount of work done on mobility prediction models, mainly about vehicles. The motivation behind such studies is to support short-term prediction models of jams and/or the suggestion of route mechanisms that make a load balancing. If we knew the directions that a vehicle will take, how the traffic would spread, we could use such information to make improve the suggestion of route algorithm in order to avoid/prevent current/future jams. This could be an easy task if we could assume that all drivers will strictly follow the routes suggested by a GPS. As stated in [Xue et al., 2009], this is not true: drivers who request GPS routes do not always follow the indicated route. Given a source and destination points, drivers may prefer to drive through different routes, depending on traffic condition. If an avenue is congested during rush hour, probably most of the drivers will follow alternative routes, created based on their common knowledge. There is still one source of information that captures this mobility: the road detector sensor, which counts the number of vehicles on road in a time-interval.

Although much have been done, as we can see in Table 2.1, there is not much models that were developed based on real-time sensing. For instance, the model proposed in [Su et al., 2000] depends on speed and moving direction information. Same happen in [Hayashi and Yamada, 2009], for speed, acceleration and direction angle, and [Son et al., 2013] for average encounter rate and degree node. Solutions from [Schougaard, 2007], [Krumm, 2008], [Kaaniche and Kamoun, 2010a] and [Manasseh and Sengupta, 2013], they all depend on the vehicle location, some depending also on historical data of the vehicle's route. The model proposed in [Manasseh and Sengupta, 2013] predicts the most likely origin-destination routes, which is different from this topic. In this work, we want to forecast the mobility of traffic, its route, and not from where and to where it is more likely the vehicle go.

Therefore, the only approaches that would be feasible to apply in practice are [Civilis, 2006], [Krumm, 2010] and our approach MoVDic [Tostes et al., 2015a]. The

Table 2.1: Predicting mobility in cities: models and its characteristics

Model	**Technology**	**Variables**	**History**
Wireless mobility prediction [Su et al., 2000]	Gauss-markov random process	Speed and moving direction.	Yes
Prediction of crossroad passing [Civilis, 2006]	Artificial neural network	Booleans for each crossroad segment.	Yes
Vehicular mobility model [Schougaard, 2007]	Bayesian network	Location and direction.	No
Random guess [Krumm, 2008]	–	–	No
Simple markov model [Krumm, 2008]*	Markov model	The most recent 3 segments into the past.	Yes
		The most recent 10 segments into the past.	Yes
Predicting unusual right-turns [Hayashi and Yamada, 2009]	Hidden markov model	Speed, acceleration and heading direction angle.	No
Basic [Krumm, 2010]*	Markov model	Number of destinations associated with that turn direction.	No
Triangles [Krumm, 2010]*		Number of triangles associated with that turn direction.	
Trip time probabilities [Krumm, 2010]*		Probabilities of remaining trip times.	
Trip time weights [Krumm, 2010]*		Down-weighting more distant destinations.	
Ad hoc mobility prediction [Kaaniche and Kamoun, 2010a]	Recurrent neural networks	XYZ location	Yes
Bayesian model [Son et al., 2013]*	Bayesian network / Maximum a posteriori	Average encounter rate and node degree.	No
Predicting driver destination [Manasseh and Sengupta, 2013]	Decision tree with pruning	The current position of the driver, the position at which the driver was 5 min prior, the time of day, and the day of the week.	Yes
MoVDic [Tostes et al., 2015]	Bayesian model and Markov Chain Monte Carlo	Street Detectors.	No

* Important references

model proposed in [Civilis, 2006] depends on historical data to construct a training dataset of crossroads passing to be used in an artificial neural network. For that, all vehicles should have GPS systems uploading to a database in real-time. Supposing that this is feasible with cloud computing and sensing, there is still the problem of scalability since the ANN training takes too much time and might not capture all the patterns considering the time dimension (weekdays, weekends, seasons, events). By the other hand, in [Krumm, 2010], Krumm presents four algorithms, in which the best one in terms of mean error is Trip Time Weights, based on the drivers' turn proportions at road intersections and trip times, but not all cities present turn count sensors, which difficult the usage of this approach in practice. In our approach, we overcome such issue by using only detectors information, which is feasible to obtain in real time and that are present in cities as São Paulo, Rio de Janeiro and Belo Horizonte.

Another important characteristic when comparing models is their accuracy/mean error, which can be seen in Table 2.2. Notice that there are some models that does not

Table 2.2: Predicting mobility in cities: accuracy and experiments details

Model	Accuracy	Mean Error	Evaluated City	Segments
Wireless mobility prediction [Su et al., 2000]	NA	NA	Simulation	–
Prediction of crossroad passing [Civilis, 2006]	NA	NA	Simulation and GPS dataset (city unknown)	–
Random guess [Krumm, 2008]*	50%	–	Seattle, USA	237.5 m
Simple markov model [Krumm, 2008]*	76%	–		
	90%	–		
Predicting unusual right-turns [Hayashi and Yamada, 2009]	–	< 0.16	–	–
Basic [Krumm, 2010]*	–	0.192	Seattle, USA	237.5 m
Triangles [Krumm, 2010]*	–	0.183		
Trip time probabilities [Krumm, 2010]*	–	0.183		
Trip time weights [Krumm, 2010]*	–	0.142		
Ad hoc mobility prediction [Kaaniche and Kamoun, 2010a]	NA	NA	Simulation	–
Bayesian model [Son et al., 2013]*	–	< 0.15	Simulation	–
Predicting driver destination [Manasseh and Sengupta, 2013]	96%	1.72% of confidence interval error	San Francisco, USA	–
Vehicular mobility model [Schougaard, 2007]**	80.54%	–	Aalborg, Denmark	200 m
MoVDic [Tostes et al., 2015]	S125***: (87.82%, 83.22%, 100%, 58.06%)	S2216***: (94.52%, 93.24%, 91.17%, 96.34%)	Realistic trace in Cologne, Germany	–

*** Prediction accuracy for different types of crossroads (2, 3, 4 and 5 exit-streets).
** Most related reference.
* Important references.

present their accuracy nor even their mean error, such as [Su et al., 2000] and [Civilis, 2006]. Indeed, since it is difficult to obtain real time data, their evaluation were made only for a small controlled scenario via simulation. Though in [Schougaard, 2007] real data have been used, the model's evaluation was only for a small city (Aalborg in Denmark), with a population of approximately 120,000 residents, where does not present that much traffic and mobility patterns such as big cities, with more than 1,000,000 residents (Cologne, New York, Los Angeles). With that said, the best approach in term of accuracy/mean error is the Trip Time Weights algorithm [Krumm, 2010], with a mean error of 0.142. The drawback of this approach is that, due to limited data, Krumm only evaluated for a small area in Seattle, with only 40 intersections in a fairly limited area, with only a shopping mall and a software maker dominant destinations. In our approach MoVDic [Tostes et al., 2015a], we have evaluated for two sub-scenarios of Cologne, with more dominant destinations.

Another approach was proposed in [Schougaard, 2007], in which a Bayesian network has been projected to predict the mobility of vehicles in a grid map. The model is based on location information and the direction that vehicles take when they arrive at an intersection. On such approach, the common sense knowledge about how vehicles move feeds into the network and the possibility of an accuracy of 80.54%. The map is divided into hexagonal cells. The model predicts which cell the vehicle will go into it. One difference to the proposed MoVDic [Tostes et al., 2015a] is the model adopted, which is quite different from the research of Schougaard [2007], although they are both within the class of Bayesian models. Another more relevant difference is the granularity of the prediction. MoVDic [Tostes et al., 2015a] considers street segments themselves rather than large cells. The idea is to predict what is the probability of a vehicle to follow in the next possible directions when he arrives at an intersection.

Additionally, in [Uma Nagaraj, 2011], there is also a prediction solution using Markov Model, Hidden Markov Model and Variable-Order Markov Model. The goal is to predict vehicle's future path, elevation, and turns based on the observations of the path that was driven by commuters, recovered from GPS traces. Notice that the idea is not to predict where on the road segment a vehicle will be, or when it will arrive in a road segment, and this is the difference from our work. In [Tostes et al., 2015a], we predicted which road segment the vehicle will be in a near future.

Finally, we summarize the limitations of existing approaches as follows:

- Only [Manasseh and Sengupta, 2013] considers the temporal context (e.g., day-of-week and time-of-day), which can improve the accuracy of other models.

- They need inputs that are not available in real-time [Hayashi and Yamada, 2009; Kaaniche and Kamoun, 2010a; Krumm, 2008; Manasseh and Sengupta, 2013; Son et al., 2013; Su et al., 2000].

- They are limited to short-term (e.g., next cell/road) mobility prediction [Civilis, 2006; Hayashi and Yamada, 2009; Krumm, 2008, 2010; Manasseh and Sengupta, 2013; Schougaard, 2007].

- They do not consider inputs related to the most common destinations of commutes in a time dimension (all of them).

- They require historical data storage space [Hayashi and Yamada, 2009; Kaaniche and Kamoun, 2010b; Krumm, 2008; Manasseh and Sengupta, 2013; Su et al., 2000]; (6) they solely rely on the history of individual users' movement [Civilis, 2006; Hayashi and Yamada, 2009; Krumm, 2008, 2010; Manasseh and Sengupta, 2013; Schougaard, 2007].

- Most of them do not scale well due to a high processing overhead.

2.3.2 Prediction of Traffic Flow

Short-term traffic forecasting models are well studied in the literature. Some have good accuracy, but they differ in the use of information sources. Some do not scale well, others can not be applied to real time, or they might just not perform well with temporary changes of the surrounding infrastructure nor with the lack of data. Thus, this is still an open problem.

Figure 2.2 shows a classification of the prediction models of traffic jam. We can see that the models can be classified by type (long-term or short-term), by the source of information that they use (univariate or multivariate), by the used technique (theoretical or empirical – parametric or nonparametric), and by their performance evaluation (with real data, traces or simulations, i.e. mobility models). In addition to this figure, Table 2.3 shows a summary about the classification of related work.

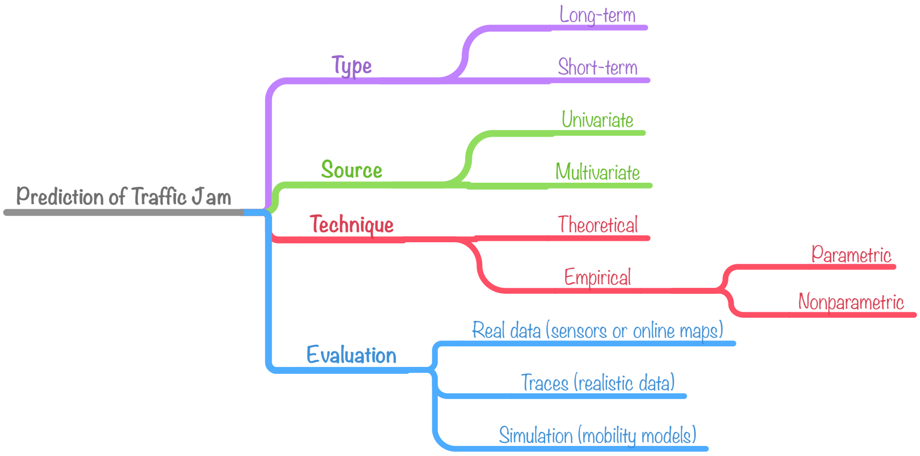


Figure 2.2: Taxonomy about the prediction models of traffic jam.

In this context, Table 2.4 summarizes the main prediction models of traffic jam with a comparison between the sources of information that they use and the following characteristics: traffic patterns, real-time sensing, real-time forecasting, historical data, time series, and customization. Notice that there are 11 models that do detection of traffic jam, 20 models that predict the traffic jam and only 2 models that do both.

The models use different sources of information to make the prediction, and most of them use GPS traces, cellular updates, road data and weather. Only four of them use seasons, events and traffic incidents information ([Ghosh et al., 2009; Inrix, 2006; Microsoft Research, 2013b; Stutz and Runkler, 2002; Tostes et al., 2015b]). Although, the only models that use community sensing are the ones described in [Google, 2009; Tostes et al., 2015b]. The one presented in [Google, 2009] only uses information reported from

Table 2.3: Classification of the prediction models of traffic jam.

Model	Technique	Theoretical	Empirical	Parametric	Nonparametric	Long-term	Short-term	Univariate	Multivariate	Real data	Freeway	Urban	Cities	Error
[Okutani and Stephanedes, 1984]	Kalman filter	●					●		●	●	●		Nagoya	< 0.09
[Davis and Nihan, 1991]	k-Nearest Neighbor		●		●		●	●			●		–	–
[Williams and Hoel, 2003]	SARIMA	●	●	●			●	●		●	●		Atlanta and London	0.086
[Stutz and Runkler, 2002]	Fuzzy clustering		●		●	●		●		●	●		German autobahn	–
[Yu et al., 2003]	Gaussian mixture model and expectation maximum	●					●	●		●	●		Beijing	0.0895
[Sun et al., 2004]	Sampling markov chain	●	●				●	●		●	●		Beijing	–
[Yoon et al., 2007]	Traffic patterns and cluster		●	●		●		●		●	●	●	Michigan	< 0.10
[Yuecong et al., 2007]	Genetic algorithm		●		●		●	●					–	0.057
[Ye et al., 2008]	Multivariate Adaptive Regression Splines (MARS)		●		●		●		●				–	0.167
[Jun and Ying, 2008]	Artificial neural network (ANN)		●		●	●			●				–	0.083
[Horvitz et al., 2005]	Bayesian network		●		●		●		●	●	●	●	Seattle	0.05[1]
[Ghosh et al., 2009]	Time series		●	●			●		●	●		●	Dublin	0.0581[1]
[Tan et al., 2009]	Combination of models		●	●	●		●	●		●	●		Guangzhou, China	0.06[1]
[Min et al., 2009]	Dynamic-STARIMA		●	●			●	●		●		●	Beijing	0.0818
[Chunmei et al., 2010]	Chaos theory and ANN		●		●		●	●					–	–
[Thomas et al., 2010]	ARIMA and kalman filter		●	●		●	●	●		●		●	Almelo, Netherlands	0.05
[Duan et al., 2011]	Support vector machine		●		●		●	●					Minnesota	–
[Marfia and Roccetti, 2011]	Mathematical model		●	●			●	●		●		●	Los Angeles and Pisa, Italy	–
[Chan et al., 2012]	Taguchi method and ANN		●		●		●	●		●	●		Australia	0.13
[Kong et al., 2013]	Urban traffic network	●					●	●				●	–	0.7263[2]
[Li et al., 2013]	Support vector machine		●		●		●		●				–	0.5[1]
[Kurihara, 2013]	Pheromone communication model		●		●		●	●					–	0.14
[Niu et al., 2014]	Deep learning and support vector machine		●		●		●		●	●		●	Wuhan, China	0.15
[Tostes et al., 2015b]	Logistic regression	●			●		●		●	●	●	●	Chicago and New York[4]	0.07[1]

[1] With variations.
[2] Mean absolute scaled error – this model is better than ARIMA, Kalman Filter, k-NN and ANN.
[3] Mean absolute relative error.
[4] Extension to other cities such as Los Angeles, Seattle, London, Paris, Ottawa, Helsinki, São Paulo, and Belo Horizonte.

a participatory sensing system, which is different from [Tostes et al., 2015b] that uses information about the mobility of users captured from social networks characterization.

The prediction models of traffic jam can be classified as univariate or multivariate [Ghosh et al., 2009]. Univariate and multivariate define whether the model uses information sources from a single location point (univariate [Kurihara, 2013; Li et al., 2013; Microsoft Research, 2013b; Niu et al., 2014; Thomas et al., 2010]), or from several sites (multivariate [Ghosh et al., 2009; Kamarianakis and Prastakos, 2003; Y.Kamarianakis and P.Prastacos, 2005]). Multivariate approaches have the advantage of capturing the

Table 2.4: Predicting traffic jam in cities: models and its characteristics

Model	Detection	Forecasting	GPS trace	Cellular updates	Online maps	Road data	Weather	Seasons	Events	Community sensing	Social sensing	Construction reports	Traffic incidents	Others	Traffic patterns	Real-time Sensing	Real-time Forecasting	Historical data	Time series	Customization
[Parkinson and Gilbert, 1983]			●																	
[Okutani and Stephanedes, 1984]		●				●												●		
[Davis and Nihan, 1991]		●				●												●		
[Williams and Hoel, 2003]		●				●												●	●	
[Stutz and Runkler, 2002]	●	●				●		●	●					●				●		
[Yu et al., 2003]		●				●										●	●	●		
[Sun et al., 2004]		●				●												●		
[Intellione, 2006]	●			●												●				
[Inrix, 2006]	●		●	●		●	●	●				●	●			●				
[Yoon et al., 2007]	●		●												●	●				
[Yuecong et al., 2007]		●												●				●		●
[Jun and Ying, 2008]		●																		
[Google, 2008] Maps	●		●	●												●		●		
[Microsoft Research, 2013b]	●	●	●	●		●	●	●	●			●	●		●	●		●		
[Google, 2009] Waze	●									●						●				
[Ghosh et al., 2009]		●				●		●	●										●	
[Tan et al., 2009]		●				●												●	●	●
[Min et al., 2009]		●				●										●	●	●	●	
[Thomas et al., 2010]		●				●												●	●	
[Duan et al., 2011]		●				●	●										●	●		
[Marfia and Roccetti, 2011]	●	●	●												●				●	
[Jain and Sethi, 2012]	●					●										●				
[Wisitpongphan et al., 2012]	●		●															●		
[Chan et al., 2012]		●				●												●		
[Kong et al., 2013]		●				●													●	
[Li et al., 2013]		●				●	●	●						●				●		
[Kurihara, 2013]		●				●								●		●	●			
[Roess and Prassas, 2014]	●					●														
[Niu et al., 2014]		●	●															●		
[Tostes et al., 2015b]		●			●		●			●	●	●	●		●	●	●	●	●	●

temporal and spatial evolutions of the traffic conditions over the time, although it demands more computational effort to estimate more parameters [Ghosh et al., 2009].

The empirical or theoretical classifications explicit the used technique. Empirical approaches use a statistical methodology or heuristic without referring to the actual traffic dynamics. It can be parametric – time-series models and autoregressive linear processes (e.g., [Y.Kamarianakis and P.Prastacos, 2005]), or nonparametric – nonparametric regressions are neural networks (e.g., [Kurihara, 2013; Li et al., 2013; Microsoft Research, 2013b; Niu et al., 2014; Thomas et al., 2010]). Most solutions are univariate-empirical-parametric solutions due to the ease of computation and good accuracy.

Most approaches differ in terms of the used information source in the prediction. While most solutions use only road data (e.g., [Kong et al., 2013; Kurihara, 2013; Li et al., 2013]) and GPS traces (e.g., [Niu et al., 2014]), some approaches use additional distinct inputs such as weather (e.g., [Inrix, 2006; Li et al., 2013; Microsoft Research, 2013b]), events (e.g., [Ghosh et al., 2009; Microsoft Research, 2013b]), and traffic incidents (e.g., [Inrix, 2006; Microsoft Research, 2013b]). What is quite interesting is that none of them uses SSs data, and that is exactly what we did in [Ribeiro et al., 2014].

Urban computing deals with a massive amount of data, gathered by ubiquitous mobile sensors from personal GPS devices to mobile phone. In [Toole et al., 2012], the authors measure spatiotemporal changes in the population, identifying clusters of locations with similar zones used and mobile phone activity patterns. Beyond characterizing human mobility patterns and measuring traffic congestion, Ban and Gruteser [2012] show how mobile sensing can reveal details about intersection performance statistics. None of them use Bing maps information, which is the difference to our research. We have used AJAX/JavaScript APIs to visualize traffic layer over the city map.

From Microsoft Research to Bing Maps, in [Apacible et al., 2005], the authors describe the JamBayes project, started in 2002, which provides estimates of flows inferences about current and future traffic flows. The challenge was to predict the future of traffic flow in Seattle area. *When does the highway system would become clogged?* The authors developed a probabilistic traffic forecasting system based on Bayes Theory, and predict future surprises about traffic congestion and flow. Results have shown an accuracy ranging from 84% up to 98% for Seattle for some bottlenecks.

Afterwards, following on JamBayes effort, the Clearflow project focused on applying machine learning to learn how to predict the flows on all street segments of a greater city area [Microsoft Research, 2013b]. It was based on GPS data collected from volunteers, buses, and vehicles for over five years. Clearflow considers all flows on all roads via predictive models in addition to real-time sensing while directions are provided based on best guesses about flows over all roads. Its main contribution is high coverage of traffic flow for 72 major cities in North America, inferring on over 60 million road segments in North America [Microsoft Research, 2013b].

Bing Maps is a web mapping platform that can provide business intelligence and data visualization solutions [Microsoft Research, 2013a]. Bing Maps services can be used to accurately pinpoint locations from geo-coding address (latitude and longitude), base maps and imagery, overlay customer locations, and data analysis. It is similar to GIS systems but without its complexity. This is done using Bing Maps APIs[1], including JavaScript/AJAX or Silverlight Controls. Data from SQL Server or other BI data sources can be easily visualized without the complexity of traditional GIS systems. Also, base maps and imagery can be manipulated. Bing map cloud platform infrastructure is divided in consumer offering and AJAX, Silverlight Control and web services APIs [Microsoft Research, 2013a].

In [Sera, 2007], a traffic jam prediction device has been developed. The prediction

[1] A full overview of Bing Maps API can be found at `www.bingmapsportal.com`.

method correlates traffic jam with the traffic jam pattern and predicts the current traffic jam degree based on the up-to-the-minute traffic jam information and the current traffic state. Results have shown that the device can be used in a conventional navigation method and so as to plot driving routes for a vehicle.

2.4 Services

Here we present a brief overview about the services that deal with traffic congestion. These services can use different data sources, such as sensors, social networks, detectors, online maps, and also inferences produced by the prediction models. In this section, we focus on: (i) the load balance of traffic flow; (ii) protocols.

2.4.1 Suggestion of Routes

There are several mechanisms to suggest routes in order to avoid or reduce traffic jams. Table 2.5 presents a summary of the most recent works that suggests routes. Some makes suggestion of routes to avoid worst traffic conditions, or traffic incidents, while there are others that suggest beautiful routes to improve people's quality of life.

Table 2.5: Related work about suggestion of routes mechanisms.

Model	Characteristics	Taxi Dataset	Simulation	Freeway	Urban Area	Visualization
[Liu and Ozguner, 2003]	Traffic throughput		•			
[Hu and Wang, 2006]	Frank-Wolfe algorithm with Gradient Projection		•	•		
[Shen et al., 2008]	Game		•			
[Shen et al., 2009]	Particle swarm optimization		•			
[Liu et al., 2011]	Visualization aspects	•				•
[Wang et al., 2014]	Unexpected congestion		•	•	•	
[Quercia et al., 2014]	Happy and beautiful routes		•		•	
[Liang and Wakahara, 2014]	Personalized routing		•		•	
[de Souza et al., 2014]	Shortest-path without accident roads			•		
[Salnikov et al., 2015]	Comparison between prices of cabs	•			•	
[Brennand et al., 2015]	k-shortest-paths selected based on the Boltzmann probability distribution		•	•	•	
[Meneguette et al., 2015]	UCONDES		•	•	•	
[Tostes et al., 2015]	Model as the multicommodity flow problem		•	•	•	

Besides, we can observe in Table 2.5 that most researches use simulations to evaluate their mechanisms, and only two have used a real dataset (taxi dataset) [Liu et al., 2011; Salnikov et al., 2015]. They are evaluated in urban areas or in freeways. The best evaluation was made in [Wang et al., 2014] and [Tostes et al., 2015], part of this thesis, since both have used a realistic dataset (TAPAS Cologne scenario). However, the evaluation that was made in this thesis stands out, since we used metrics beyond the average trip time, such as fuel consumed and CO_2 emissions.

The interesting aspect is that we can compare [Wang et al., 2014] and [Tostes et al., 2015], since they use the same evaluation scenario: the TAPAS Cologne trace. Although, in this thesis, there were more evaluations in terms of CO_2 emissions and fuel consumption. In [Wang et al., 2014], the results have shown the following average trip time: MNTR with 58.09s, ModRe with 63.55s, ConRe with 78.04s, ERE with 65.94, and ORG with 56.26s. In this thesis, the average travel time was less than 20s.

Following the modeling of this thesis, described in Chapter 6, the problem of traffic congestion minimization can be modeled as the MFP, which is well study since the decade of 1960. Its origin is in the contributions of maximum flow [Fulkerson and Ford, 1962] and multicommodity flow [Hu, 1963]. The flow of products, that is, of vehicles, is distributed through the network edges, homogeneously, without any arc overload. Considering a cost in each arc, the goal is to minimize the sum of all these costs.

The literature presents heuristic-based and metaheuristic-based strategies. Typically, solutions uses two phases [Mourão, 2009]: (i) elaborate a viable initial solution by a constructive heuristic; (ii) improve this initial solution by using other heuristics/metaheuristics, moving closer to optimal solution. We can quote as heuristics: approximate algorithms (greedy), local search, randomized algorithms, linear programming and integer programming. As metaheuristics: tabu search, genetic algorithms, iterative local search and simulated annealing.

Approximation heuristics are also quite used. For instance, a greedy heuristic has been proposed in [Awerbuch and Khandekar, 2007], which achieves a $1 + \epsilon-$approximation. In this case, multiple cooperative agents, but uncoordinated, obtained distributed solutions. The advantage is that the heuristic requires a more aggressive increase in the flow rate on a link than a decrease in the rate [Awerbuch and Khandekar, 2007]. In the greedy heuristics, we can also quote [Gabrel et al., 2003], in which they implement the Benders heuristic. Remarkably, it produces the best heuristic solutions in comparison with classical greedy approaches.

Another category of MFP works uses metaheuristics, such as tabu search, genetic algorithms, iterative local search and simulated annealing. Indeed, examples of iterative local search and genetic algorithms for MFP are explained in [Silva, 2007] and evaluated in [Mourão, 2009]. The different between both works is that, in [Mourão, 2009], they evaluate first genetic algorithms and then iterative local search, while the opposite occurred in [Silva, 2007].

Briefly, several studies reduce network problems to MFP. In [Buriol et al., 2003], the authors have reduced MFP to the network routing optimization problem. The authors propose heuristics and metaheuristics, standing out the tabu search and genetic algorithms, to minimize the packet-based traffic congestion through the Open Shortest Path First protocol. Another example can be found in [Fortz and Thorup, 2000], in which the minimum congestion problem is studied using the real backbone data from AT&T. Although, there is no exhaustive proposal and evaluation of heuristics and metaheuristics in the context of VANETs, which is our focus.

Finally, as motivation we can quote examples of ITS applications. Some works use RSUs to suggest routes for vehicles, such as [Doolan and Muntean, 2013]. In the distributed version, each RSU is responsible to suggest routes for vehicles inside its coverage area. When the vehicle arrives at another coverage area, it requests another route for that specific area. So, our heuristics proposed could be (and were – see Section 6.3) developed in these RSUs so as to keep the load balancing of vehicles in each coverage area.

2.4.2 Protocols

There are several protocols in vehicular networks that can take advantage of prediction model such as the one we described in Section 2.3. According with [Chen et al., 2009], location information has a direct impact on the behavior of several models and algorithms in distinct areas of VANET, such as routing, resources management and security. Algorithms that use such information usually have better performance. However, the performance gains depend on the location precision [Son et al., 2004], which actually culminates in the assumption that the GPS error tends to zero. In [Son et al., 2004], the authors study the effects of errors in the location estimation used in routing protocols for MANETs and wireless sensor networks. Results showed an improvement of up to 27% in packet delivery and 37% reduction in network resource wastage, on average.

In addition to such protocols, here we present related works about protocols in VANET applied in traffic management systems, focusing on: (i) congestion control mechanisms; and (ii) dissemination protocols.

2.4.2.1 Congestion Control Mechanisms

Adequate congestion control mechanisms are essential to optimize the flow of vehicles in vehicular networks. Congestion control applications in vehicular networks can be broken down into identifying, minimizing and preventing congestion. The identification of traffic congestion is the effectiveness in characterizing congestion in traffic. The minimization corresponds to an attempt to reduce detected congestion. Congestion avoidance is a complex approach, since it needs to manage vehicular flows on the roads, to prevent the onset of congestion. The problem of preventing traffic congestion can be reduced to a minimum-cost flow problem in flow networks, which falls into the category of NP-Complete problems.

The works of [Fukumoto et al., 2007] and [Fahmy and Ranasinghe, 2008] present proposals for monitoring and detection of traffic conditions, using V2V (Vehicle-to-Vehicle) communication. To accomplish this, beacon-type messages are transmitted. Such messages are simple, have low overhead and are used to announce the presence of a vehicle to its neighbors. Other approaches, such as ConProVa [Silva et al., 2013a] use only V2I (Vehicle-to-Infrastructure) communication, which limits the proposal. In ConProVa, mechanisms to deal with issues relating to the provision of contexts in vehicular networks are implemented in the infrastructure. For this, ConProVa implements a middleware in order to assist in decision making and resolution of conflicts of interest regarding the actual traffic situation reported by vehicles, since vehicles cannot collaboratively validate conditions detected on the road.

There also exist approaches such as ECODE [Younes and Boukerche, 2013], which use both V2I and V2V communication. However, ECODE depends on the infrastructure (RSU) in monitoring areas, and this restrains the proposal to specific scenarios.

As the approaches in the literature have distinct characteristics, a rank encompassing the proposals for monitoring vehicular traffic was adapted for this work, in order to enable analyzed techniques to be compared. Table 2.6 presents a comparative information regarding congestion protocols in vehicular networks.

The adopted classification distinguishes the analyzed works according to the following aspects: (i) cooperative V2V validation refers to the proposal's ability to use cooperative communication between vehicles to reach a consensual decision regarding the traffic situation in a region. (ii) congestion identification portrays the effectiveness in measuring traffic conditions on the road. (iii) congestion size indicates the effectiveness in estimating congestion length. (iv) the overhead is the ability to avoid overloading the communication channel. (v) dissemination refers to whether or not the proposal transmits traffic information in the network. (vi) the architecture shows how

Table 2.6: Evaluated proposals for monitoring vehicular traffic

Proposal	**V2V Cooperative Validation**	**Congestion Identification**	**Congestion Levels Estimation**	**Congestion Size**	**Overhead**	**Dissemination**	**Architecture(s)**	**Rerouting**
COC [Fukumoto et al., 2007]	●	●		●	●		V2V	
[Fahmy and Ranasinghe, 2008]		●					V2V	
ConProVa [Silva et al., 2013a]		●				●	V2I	●
ECODE [Younes and Boukerche, 2013]		●				●	V2X$^{(1)}$	●
[Lu and Cao, 2003]		●	●				V2I	
[Binglei et al., 2008]		●	●				V2I	
COTEC [Bauza et al., 2010]	●	●	●	●		●	V2V	
TransTree [de Brito et al., 2014b]		●	●	●			V2V	
[Souza and Villas, 2015]						●	V2V	●
SCORPION [de Souza et al., 2015]	●	●	●			●	V2X$^{(1)}$	●
CARTIM [Araújo et al., 2014]	●	●	●	●		●	V2X$^{(1)}$	●

1 - V2V and V2I architectures

the vehicular network is organized. (vii) rerouting indicates that vehicles can change their initial routes.

In [Lu and Cao, 2003], an algorithm for automatic congestion level classification, using computational intelligence such as fuzzy logic, have been proposed. In [Binglei et al., 2008] and [Bauza et al., 2010], the idea proposed in [Lu and Cao, 2003] was used to increase the efficiency in detecting congestion. As portrayed in [Binglei et al., 2008], the use of fixed boundaries between categories of vehicular traffic leads to high error rates, especially when a category is close to the classification border between two categories. This is due to the characteristics of vehicular traffic flow, which has no clear boundaries between different types of traffic. In this case, for non-deterministic decision problems, fuzzy logic gets good results by allowing input variables of the fuzzy system to contain elements with partial degree of membership in the border regions. This characteristic of the system helps ensure the accuracy of the results.

COTEC, presented in [Bauza et al., 2010], uses fuzzy logic to detect the level of congestion. The technique also provides a cooperative approach to validate traffic conditions detected by the system. The concepts addressed by the authors in [Bauza et al., 2010] were incorporated into this work. However, COTEC has some shortcomings such as: (i) the inefficiency in preventing a vehicle to reach a congested area and (ii) the lack of a policy to allow route modification. (iii) New settings in the fuzzy system are also necessary so this technique is not confined to the ideal scenario (freeway) proposed by the authors.

2.4.2.2 Dissemination Protocols

Data dissemination is concerned about how to send data from a source to a destination [Soares et al., 2014b]. The matter is how to send packets while meeting application requirements, such as high delivery rate and low delivery delay. The source and destination could be a set of vehicles, vehicles located in a given geographic area or roadside units. There are three categories of dissemination protocols: (i) unicast dissemination, which consists in one source to one destination; (ii) multicast dissemination, in which the data is sent to a set of nodes, and (iii) broadcast dissemination, which is sending data to all neighbor nodes.

In VANETs scenarios, protocols need to deal with different issues caused by their dynamic nature. For instance, the network topology is temporary and short-lived and network partition leads to increase in delivery delay. Moreover, broadcast-based protocols could cause broadcast storm problem, which leads to high packet traffic and packet losses. Therefore, the most important challenge is to develop a protocol suitable to different vehicular environments, as city and highway scenarios. For instance, in city scenarios the vehicle density along the streets varies a lot during the day, while in rural scenarios, the density tends to be low during most of the time.

Routing and data dissemination protocols developed for traditional ad hoc network [Karp and Kung, 2000; Perkins and Royer, 1999b] suffer a critical drop in performance when applied to vehicular environments [Li and Wang, 2007]. This is caused by high mobility, speed and disconnection rate among the network nodes. However, VANETs present some factors that can help the dissemination process, as vehicles mobility pattern, limited by road layout, [Ranjan and Ahirwar, 2011] the tendency of vehicles to move in groups, the integration with in-built vehicle sensors (GPS), and the use of real-time and historical traffic information to infer network behavior. TODD intends to use all these factors in order to obtain good performance of data dissemination in different scenarios.

In recent years, researchers proposed several protocols and algorithms that take advantage of VANETs main characteristics seeking to improve dissemination performance. We have separated some key aspects related to data dissemination and an overview of the state of the art in dissemination protocols.

Summary. Table 2.7 compares the main routing protocols in terms of technique, category, maps and GPS (Global Positioning System), and buffer [de Castro et al., 2013; Soares et al., 2014b]. The category can be the following: (i) based on topology, (ii) geographical position; (iii) opportunistic; and (iv) dissemination. Some protocols use digital maps with route data (maximum speed, length and geographic coordination) and GPS. Others also use buffer in the vehicle to store messages during a specific time interval until the message's time to live is exceeded, or to carry the message until the next hop is selected. Finally, some protocols were designed or adapted to ad hoc networks, while others were specifically designed for VANETs.

Table 2.7: Evaluated proposals for routing and dissemination

	Technique			**Category**							
Proposal	**Shortest Path**	**Multicast Routing**	**Best Moment**	**Topology**	**Positioning**	**Dissemination**	**Opportunistic**	**Maps**	**GPS**	**Buffer**	**Specific for VANET**
AODV [Perkins and Royer, 1999a]	•			•							
New-AODV [Ding et al., 2011]	•			•							•
[Atechian and Brunie, 2008]		•			•				•		•
Flooding [Banzi et al., 2011]		•			•			•			
GFG/GFPG [Seada and Helmy, 2006]		•			•				•		
Gossiping[Banzi et al., 2011]		•			•			•			
ROVER [Kihl et al., 2008]		•			•			•		•	•
UMB [Korkmaz et al., 2004a]		•				•		•	•	•	•
VADD [Zhao and Cao, 2008]			•			•	•	•	•	•	•
SADV [Ding and Xiao, 2010]			•				•	•	•	•	•
GyTAR [Jerbi et al., 2009]	•			•					•	•	•
CBF [Füßler et al., 2004]	•			•				•	•		
GEDDAI-NP [Villas et al., 2013]		•			•			•	•		•
DRIVE [Villas et al., 2014]		•			•			•	•		•
TODD/CTODD [Soares et al., 2014b]			•			•	•	•	•	•	•

1 - Only detects a jam, does not rank or estimate congestion levels.
2 - V2V and V2I architectures

Usage of traffic information. As vehicular environments are constantly changing, using traffic information can help to treat or even avoid problems caused by these changes. Nowadays, data dissemination and routing protocols use different sources of traffic information. Historical traces can be generated by GPS [Xue et al., 2009], gathered by roadside sensors or induction loops. Statistical information can be derived from road map and traffic environment characteristics, e.g., number of lanes, maximum speed and period of the day [Skordylis and Trigoni, 2008]. Also, with the increase in usage of social networks, researchers are studying ways to extract traffic information from published messages [Huzita et al., 2012].

Some dissemination protocols gather traffic information by broadcasting control packets. In this case, distributed algorithms use special nodes located within intersections of the scenario to store traffic information about the nearby roads. These special nodes could be passing vehicles or roadside units. HTAR protocol [Lee et al., 2011] uses junc-trackers, vehicles located in intersections, responsible to periodically collect and disseminate up-to-date traffic information from adjacent connected roads. When a vehicle that carries a packet arrives at an intersection, it requests the traffic information from the junc-tracker in order to choose the next road of the dissemination path. Due to the mobility of vehicles, junc-trackers need to be elected very often, which could lead to increase in overhead and delivery delay rates.

One of the main drawbacks of using control packets to evaluate traffic behavior is the overhead caused by broadcasts. Online traffic information resources already exist and can be used to assist data dissemination. Besides, when an online resource is not available, traffic behavior can be predicted by historical traces and traffic statistics.

Topology-based dissemination paths. Several works propose topology-based protocols, i.e., they calculate dissemination paths before send packets [Chang et al., 2011; Chou et al., 2011; Nzouonta et al., 2008; Wu et al., 2004]. In these works, the road network is abstracted as a directed graph, where vertices represent intersections and edges represent road segments. Protocols can thereby assign weights to the edges according to traffic information. The IG-based architecture [Chang et al., 2011] uses historical traffic statistics to compute average inter-vehicle distance. Then, it creates an intersection graph consisting of all connected road segments and uses Dijkstra algorithm to find the path to destination that presents shorter inter-vehicle distance. In HTAR [Lee et al., 2011], the edges weights are adjusted in real-time based on monitored traffic information and routing paths are computed using Dijkstra.

RBVT [Nzouonta et al., 2008] is a topology-based protocol that seeks to disseminate packets through paths that present highest connectivity rate among the network nodes. When a vehicle wants to send a packet, it broadcasts a route discovery packet (RD) using a controlled flooding approach to discover a connected path to packet destination. When the destination vehicle receives the RD packet, it responds with a route reply packet (RR), in which stores the connected path. As soon as the sender receives the RR packet, it starts the data dissemination. When a route is broken, route error (RE) and update (RU) packets are disseminated. One of main problems of this approach is that packets are disseminated only through connected paths. Therefore, the protocol does not work well on sparse networks, where it is very difficult to have such paths.

The most critical issues of topology-based dissemination approaches are caused by the dynamic behavior of vehicular scenarios. The topology of VANETs is constantly changing. At one time, we have few vehicles interconnected because they are near each other. After few seconds, these interconnections (network topology) can change, since the vehicles are constantly moving and they may be not near each other anymore. Thus, a route path that existed before may not exists in the near future. In this case, dissemination protocols seek to overcome this problem by recalculating the topology every time a new event is detected, e.g. broken dissemination paths. This process could lead to packet losses and an increase in overhead rate, caused by the broadcasting of control packets to fix broken paths. Also, packets need to wait until a new path to the destination is calculated, which contributes to an increase in delivery delay.

Next hop selection. One of the most used next hop selection methods is to maintain tables to store neighbor information [Lee et al., 2011; Wu et al., 2004; Zhao et al., 2007]. When a vehicle wants to send a packet, it chooses the best next hop based on context information stored in the neighbor table, e.g., position, speed and distance to packet destination. In order to keep tables updated, the vehicles must broadcast control packets to request neighbors' information. The main problem of this approach is the overhead caused by the broadcasting of control packets. In vehicular scenarios, this problem is more critical, because the tables become outdated very quickly, due to the dynamic behavior of VANETs. Thus, vehicles must broadcast control packets more often, generating more overhead.

In order to reduce the overhead, some protocols use RTS (Request-To-Send)/CTS (Clear-To-Send) method [Fasolo et al., 2006; Korkmaz et al., 2004b; Nzouonta et al., 2008]. When a vehicle wants to send a packet, it starts a neighbor selection phase by broadcasting a RTS packet. When their neighbors receive the RTS, they choose whether to reply or not based on their current information. For instance, a neighbor could reply only if it is closer to packet destination than the RTS sender. If a vehicle chooses to reply, it computes a waiting time to reply with a CTS packet. This approach leads to reduce in overhead rate in comparison to the use of neighbor tables, but contributes to an increase in delivery delay.

RBVT proposes a new function to calculate the waiting time, which is based on three factors: forward progress, optimal transmission area and received power. Also, it proposes the use of factor weights that can be dynamically set based on network and traffic conditions. However, RBVT does not take into account important context information like speed and traffic information. Besides that, it does not evaluate the use of different weights and how they impact in different vehicular scenarios.

2.5 Chapter Remarks

This chapter provided a comprehensive and up-to-date background on vehicular network, social sensing, prediction models, and TMS services. Starting with an overview of vehicular network, in Section 2.1, we described its architecture and presented its main characteristics. Within the scope of the later, we provided a contextualized background on social sensing and the main related approaches in Section 2.2. Then, Section 2.3 summarized the prediction models that have been used and proposed in VANETs. Moreover, in Section 2.4, we extended our discussion about vehicular networks by presenting the main challenges and related work in services, such as the suggestion of routes in Section 2.4.1 and protocols in Section 2.4.2. In the next chapter, we introduce the data collection module of our traffic management system.

Chapter 3

Data Collection

Although ITS use infrastructure sensors to monitor the traffic condition in a vehicular environment, there are many kinds of sensors that can be used such as road detectors (inductive loop detection), in-road reflectors, infrastructure-to-vehicle and vehicle-to-infrastructure electronic beacons. These are some examples of “traditional sensors”. However, today we have many other emerging source of data that can be very useful for ITS. For instance, there are information about traffic conditions in the Web, such as Bing Maps[1] and Google Maps[2]. This system presents real-time information about the traffic conditions (e.g., free or congested).

In addition to Bing and Google Maps, there are crowdsourcing geodata in projects such as OpenStreetMap[3], which is a collaborative project to create a free editable world map. Although not presenting traffic jam information as well as Bing and Google Maps, there are several ways to download data from OpenStreetMap and export a file to a simulator of urban mobility as SUMO. Thus, junctions, traffic lights, and road network can be a VANET scenario, allowing a more realistic evaluation of VANET protocols. The only missing information is traffic jam, which is available at Bing or Google Maps. As [Horvitz and Mitchell, 2010] said, methods for learning automated driving competencies from data will be crucial in the development of autonomous vehicles that drive without human intervention. First, how to acquire and predict the traffic flow are the matters to build safer cars that employ collision warning and avoidance systems.

Besides that, the use of participatory sensing systems, such as Foursquare[4], In-

[1] `www.bing.com/maps`

[2] `www.google.com/maps`

[3] Available at `www.openstreetmap.org`.

[4] http://www.foursquare.com.

stagram[5], and Waze[6], are becoming very popular. For example, in 2014 Foursquare registered 45 million users, Instagram 200 million users, and Waze 50 million users. Data shared in these systems have the active participation of users. In this case, these systems can be seen as a sort of sensor network, also known as participatory sensor network (PSN) [Silva et al., 2014]. In this network users plus his/her mobile devices can be considered a sensor, because users carry mobile devices that are able to sense the environment and make relevant observations at a personal level.

This chapter describes our contributions in the Data Collection module, illustrated by Figure 3.1. We describe which data we have collect, how and for how long. Details about our methodology, as well as the challenges that we faced, are explained.

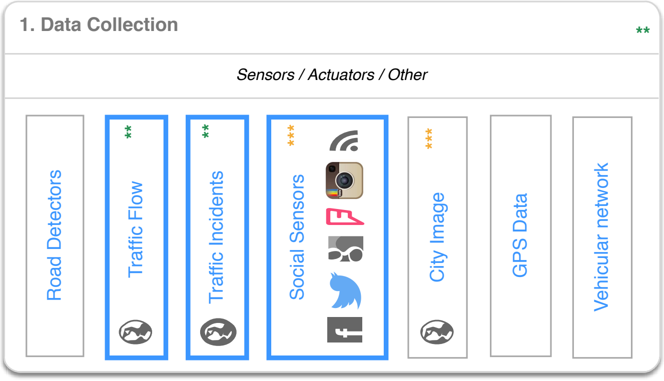


Figure 3.1: Description of the Data Collection chapter.

First, we present our methodology to collect the traffic flow from Bing Maps as well as our traffic flow dataset [Tostes et al., 2013]. Second, we have collected data about traffic incidents, from Bing maps. Third, we describe our collected data about check-ins, i.e. when users share their location in the Web [Tostes et al., 2014]. In this case, we have used Twitter, Foursquare and Instagram. We have collected data from several cities, such as Chicago, New York, London, Paris, Belo Horizonte and São Paulo, besides others. Recall that, in this work, we have focused in data from Bing Maps due to data quality. Although, it is important to notice that a natural extension is to apply this thesis' methodology with Google Maps, as future directions.

This chapter is organized as follows. Section 3.1 describes details about the elaboration of our traffic flow datasets. Section 3.2 presents our dataset of traffic incidents. Section 3.3 shows details about our social dataset, i.e. check-ins data (social data). Finally, Section 3.4 concludes this chapter.

[5] http://www.instagram.com.
[6] http://www.waze.com.

3.1 Traffic Flow Dataset

3.1.1 Methodology

Aiming a city-wide traffic flow information to establish inferences and big data analysis for patterns discovery, a methodology for acquiring traffic flow data from distinct sources has been designed [Tostes et al., 2013]. Any GIS map service can be input of this methodology. This was a necessary step, because we required an API that allowed us unlimited requisitions to obtain the traffic flow data, which is not available.

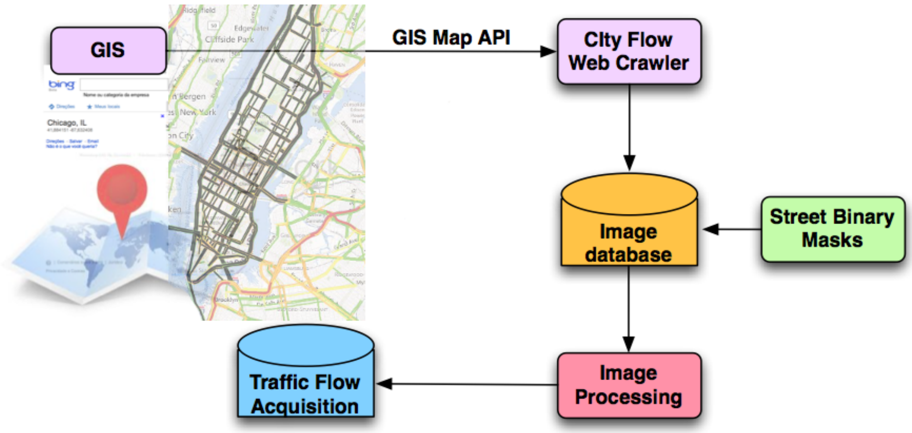


Figure 3.2: Traffic flow acquisition methodology through a GIS map web crawler.

Figure 3.2 presents the process of traffic flow acquisition. Through the API map service, a city flow web crawler has been developed. Then, a script was designed in order to collect the traffic flow image from the selected city. We used PhatomJS[7], a JavaScript library to take a screenshot of the city map with the traffic flow layer on and save the image into the database. Next, an image processing software was developed for extracting each road traffic intensity, saving the percentage of green pixels, yellow pixels, and red pixels, which correspond to the flow intensity (green is free while red is congested). Each image from the image database was processed and its flow intensity was saved to a specific date and time into the Traffic Flow Acquisition.

In this work, Bing Map has been used as input. Algorithm 1 presents the procedure for loading Bing Map in a web page, with the traffic layer on, and without the illustrations over the map. First the map is set to the specific city center location (the geo-code of New York is 40.783094, -73.980324), with the specific zoom level (12). Then, the traffic layer is enabled and its opacity is turned off (1). We display only the map, disregarding everything else. To know when the maps and the traffic layer

[7] Available at http://phantomjs.org/

are loaded on the browser, we start two event handlers. When the map and the traffic layer are loaded, two events *tiledownloadcomplete* are triggered.

Algorithm 1 Bing Map Traffic Module Load

```
setMapView()
showTrafficLayer()
opacityTrafficLayer()
eventHandlers()
```

The web crawler algorithm is presented by Algorithm 2. For each HTML scenario established as Algorithm 1, we get current hour, open the web browser with the created HTML file, and wait for the event signals indicating that all map layers were loaded. Then, we get a screenshot. This process is repeated every second. This approach was implemented with PhantomJS, an open-source headless browsers.

Algorithm 2 Traffic Flow Web Crawler

```
for each scenario $s$ do
    get current hour
    open web browser with the code from Algorithm 1
    print the screen
    kill web browser
    m ← 60 - current minutes in the time
    sleep(m)
end for
```

After collecting image data, the next step is the processing. As input, the algorithm needs the image data and the masks for each road. The mask is a binary image with white background and black street line. Figure 3.3 illustrates a street mask for one street segment and a translucent image of Chicago map overlapped, which were used in this work. Notice that, in this work, all street masks were manually drawn. Although not being in the scope of this thesis, we can automatize such procedure by applying Harris Corner detector to find the crossroads and, then, the Canny edge detector algorithm to isolate the street borders.

Algorithm 3 presents the steps followed to process one map image. For each black pixel in the street mask, the counter for each flow category (green, yellow, red, or no category – error) was increased according to its color. To establish a band for each flow, HSL (Hue, Saturation, and Lightness) was used. As HSV (Hue, Saturation, Value), HSL is one of the most common cylindrical-coordinate representations of points in an RGB color model. The variation of the hue corresponds to the values 0–360, in which

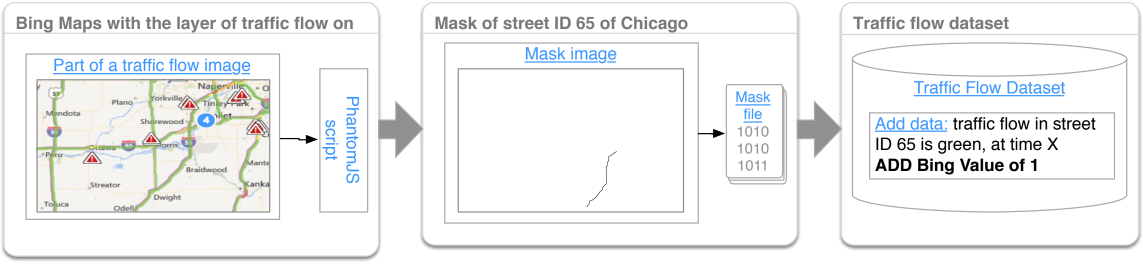

Bing Maps with the layer of traffic flow on
Part of a traffic flow image
PhantomJS script
Mask of street ID 65 of Chicago
Mask image
Mask file
1010
1010
1011
Traffic flow dataset
Traffic Flow Dataset
Add data: traffic flow in street ID 65 is green, at time X
ADD Bing Value of 1

Figure 3.3: Street mask example, considering the city of Chicago.

Algorithm 3 Traffic Flow Image Processing

```
Require: Image file $i$, Set of Road Masks $k_r$
procedure ImageConversion ($i$, $k_r$)
    GreenPixels = 0
    YellowPixels = 0
    RedPixels = 0
    NoCategoryPixels = 0
    for each Road Mask $k_r$ do
        for each Pixel $p$ in the $k_r$ image do
            if $p$ is black then // Increase the counter of its respective color
                if $hue(p) < 30$ or $hue(p) \geq 330$ then RedPixels++
                else
                    if $hue(p) < 70$ then YellowPixels++
                    else
                        if $hue(p) < 150$ then GreenPixels++
                        elseNoCategoryPixels++
                        end if
                    end if
                end if
            end if
        end for
    end for
end procedure
```

0 is a red band, followed by a yellow band, and other band colors, and finally a red band again. So it is possible to identify color bands to Bing's traffic flow intensity.

Therefore, considering a street ID α, we can define the named Bing value for α as only one of the following definitions.

Definition 1. Bing value = 1: if the traffic flow in the street α is free (green color).

Definition 2. Bing value = 2: if the traffic flow in the street α is congested (yellow color).

Definition 3. Bing value = 3: if the traffic flow in the street α is jammed (red color).

Definition 4. Bing value = 0: if there is no traffic flow in the street α, that is, the image processing mechanism have matched the street background in the map without the traffic layer on. This can be considered as either an error or the absence of data.

3.1.2 Description

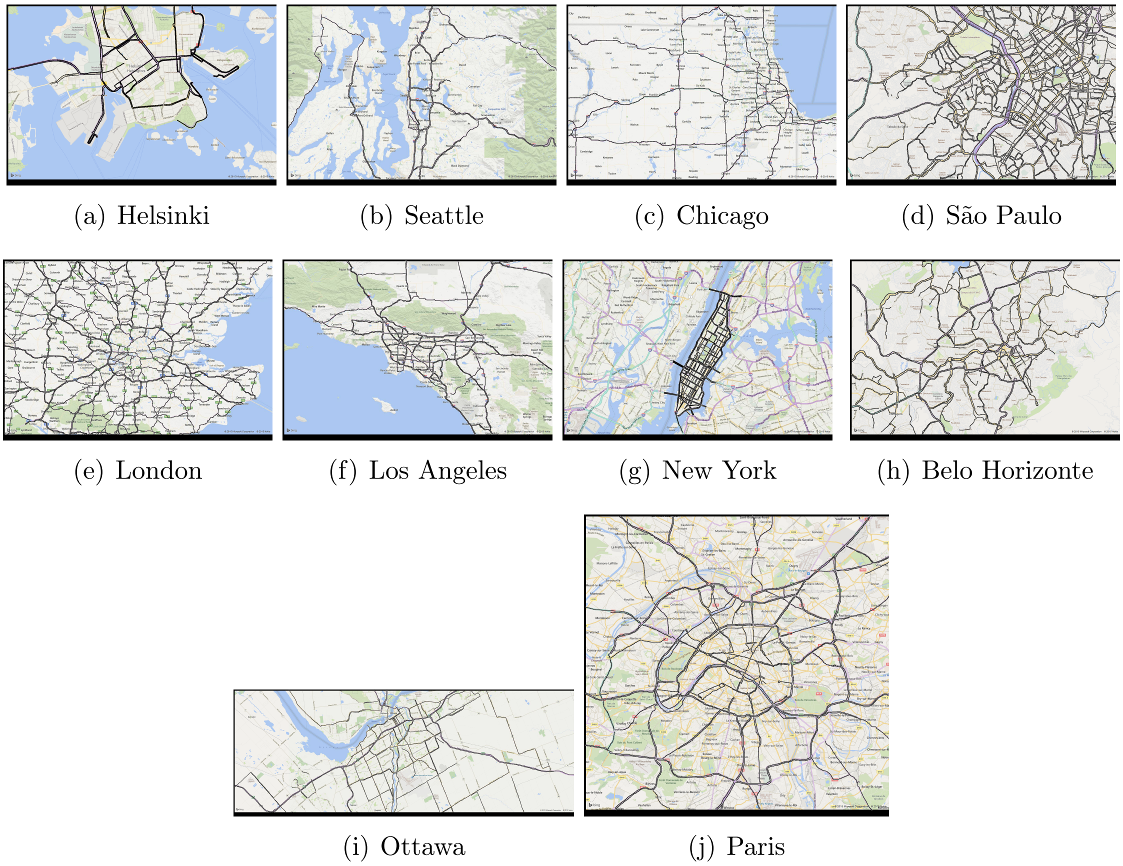
This section describes the dataset of traffic flow collected from Bing Maps for different cities, median ones (e.g., Ottawa, Helsinki, Seattle), and large ones (e.g., New York, Los Angeles, London). In this work, we present the results for datasets containing approximately two months of data. Table 3.1 shows the main characteristics of the datasets from 10 cities: Helsinki, Seattle, Chicago, Ottawa (Canada), London, Los Angeles, Paris, New York, Belo Horizonte (Brazil) and São Paulo (Brazil).

Table 3.1: Description of datasets used in this work.

City	**Start**	**End**	**# Days**	**# Streets**	**Free Flow (%)**	**Congestion (%)**	**Traffic Jam (%)**	**No Data (%)**
1. Helsinki	02-13	04-08	53	91	15.71	5.94	0.04	78.32
2. Seattle	02-13	04-15	60	38	70.11	1.53	0.57	27.79
3. Chicago	02-13	04-17	62	51	88.74	1.20	0.61	9.45
4. Ottawa	02-13	04-09	54	141	36.85	1.02	0.18	61.95
5. London	02-13	04-09	54	273	76.99	4.33	0.29	18.39
6. Los Angeles	02-13	04-09	54	172	86.63	7.39	3.80	2.18
7. Paris	02-13	04-09	54	160	65.66	21.29	1.65	11.4
8. New York	02-13	04-16	62	175	76.27	6.51	1.48	15.73
9. Belo Horizonte	02-13	04-17	62	169	34.11	7.55	0.23	58.11
10. São Paulo	02-13	04-17	62	996	65.23	20.77	1.56	12.45

Additionally, Figure 3.4 shows a map of each city with the masks overlapped. The masks indicate streets that have available traffic conditions. We have chosen streets that presented traffic data on Bing Maps (colored lines over the map). The absence of the other streets in our datasets does not affect the prediction model itself, since a street is a parameter to the model. Although, the absence of traffic flow on streets can indicate for which streets we need other sources of information about traffic flow.

We can observe that cities with more congestion data, based on Table 3.1, are Paris and São Paulo (Brazil), followed by Los Angeles, New York and Belo Horizonte (Brazil). The city with more mapped streets is São Paulo (996 streets), London (273 streets), Belo Horizonte (175 streets) and Los Angeles (172 streets). We mapped the main streets, such as avenues and highways. All these datasets are from 2015. The *no data* column indicates the absence of data in the database, because there was no data about the traffic conditions in the Bing for some streets. This happens for other traffic maps as well such as Google, and, possibly, we will have more information in the future.



(a) Helsinki

(b) Seattle

(c) Chicago

(d) São Paulo

(e) London

(f) Los Angeles

(g) New York

(h) Belo Horizonte

(i) Ottawa

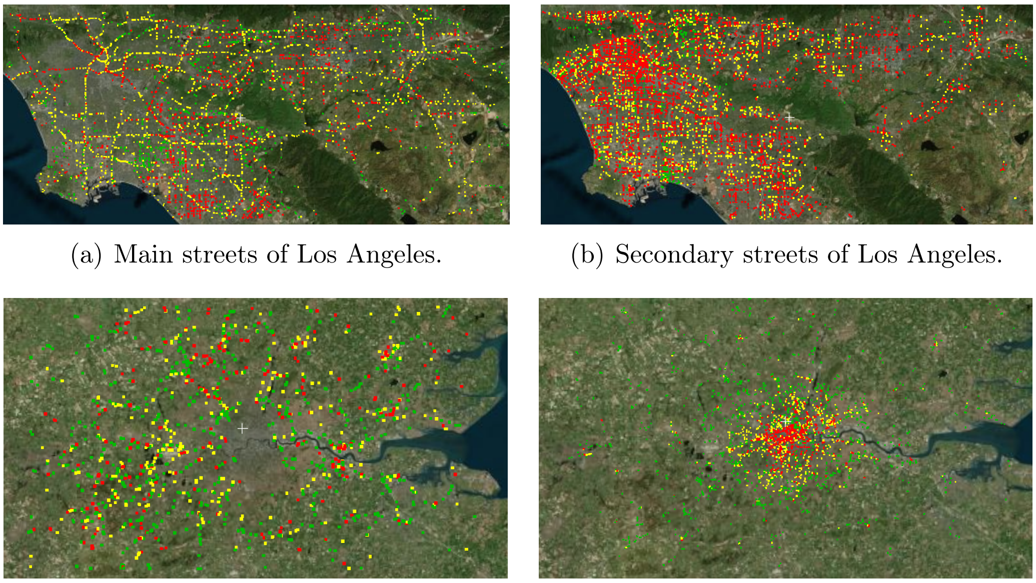
(j) Paris

Figure 3.4: Masks of the crawled cities.

3.1.3 Dataset From Microsoft Research

Here we present a dataset obtained by a collaboration with Bing Maps team at Microsoft Research, when the author of this thesis were doing an internship. We extracted one week data from Los Angeles and London, from June 28th 2014 to July 6th 2014.

The reason to choose just one week data is because the traffic flow dataset is huge, around some terabytes.



(a) Main streets of Los Angeles.

(b) Secondary streets of Los Angeles.

(c) Main streets of London.

(d) Secondary streets of London.

Figure 3.5: Overview of the traffic flow on street sectors.

This dataset is composed by date time, street sector, and speed flow. Each street sector has its own latitude and longitude. Figure 3.5 presents the traffic flow for the main and secondary streets. The classification was made based on the speed limit of the street, based on limited thresholds.

3.2 Traffic Incidents Dataset

In the context of smart cities, a challenge relies on not only how to deal with traffic jam, but why they occur. Several reasons can occur, even the named invisible jams, which are traffic congestion caused by no known reason. Some aspects can involve changes in infrastructure, constructions, accidents, weather, road hazards, and others.

Nowadays, Bing Maps system offers the RESTful API of Traffic Incidents[8]. In this work, we used this API in order to collect real-time traffic incidents. With it, we can store data of the traffic incidents from a rectangular area with a maximum size of 500 km × 500 km. We can also recovery real-time data from different types of incidents: (1) accident; (2) congestion; (3) disabled vehicle; (4) mass transit; (5) miscellaneous (incidents that do not fit in another category); (6) other news; (7) planned event; (8)

[8] http://msdn.microsoft.com/en-us/library/hh441726.aspx

road hazard; (9) construction; (10) alert; and (11) weather. Each traffic incident can have different levels, classified as low, minor, moderate and serious impact.

We have collected one year of traffic incidents data from several cities (LosAngeles, Chicago, Seattle, New York, London, Ottawa, Paris), since May 1st of 2014 until May 8th of 2015. Table 3.2 summarizes the total of incidents per type for each city. They are normalized by the maximum amount of incidents in each category.

Table 3.2: Details about the collected dataset about traffic incidents world wide.

City	**Total (%)**	**Accident**	**Congestion**	**Disabled Vehicle**	**Mass Transit**	**Miscellaneous**	**Other News**	**Planned Event**	**Road Hazard**	**Construction**	**Alert**	**Weather**
Los Angeles	96	100	3	100	57	100	6	27	37	0	67	
Chicago	62	68	7	39	0	17	27	8	11	42	0	0
New York	27	16	1	26	0	26	22	1	7	19	100	100
Seattle	23	16	1	39	0	9.25	26	3	9	17	0	33
London	100	18	100	39	0	100	95	100	100	100	0	33
Ottawa	2	2	0	1	0	1	0	0	2	1	0	0
Max	48311	23434	3301	5833	0	15211	415	366	1436	21091	1	3

- All data are in percentage.

We can observe that the city with more reported traffic incidents in Bing maps is London, followed by Los Angeles and Chicago. Although having more traffic incidents, most of them are not accidents, such as in Los Angeles, but of road hazards (21,091 out of 48,311 incidents).In New York, most of the incidents are accidents, followed by road hazards. Actually, this also happens in both Los Angeles and Chicago, although Los Angeles has more miscellaneous incidents than Chicago (18.53% of all incidents in Los Angeles in comparison with 8.5% of all incidents in Chicago). Notice that alert and weather incident are not well used as a traffic incident alerts, since it has few occurrences in over 150,614 incidents (1 and 7 respectively).

3.3 Social Dataset

In this study we focus on users' check-ins given on Foursquare and Instagram. Foursquare is a very popular location sharing service, class of system that enables users to share their location with friends. To give an idea of the popularity of Foursquare, this year it gathered over 45 million users worldwide, which shares millions of locations every day [Foursquare, 2014].

In location sharing services the main object are venues, which represent any physical location, such as a restaurant, a university, or a gas station, that a user can be at. Thus, the basic activity users can perform in location sharing services is called check-in, which is an action to announce in the system the venue you are at a certain moment. Foursquare also maintains a set of eight pre-defined venue categories, namely, “Arts & Entertainment”, “Colleges & Universities”, “Food”, “Great Outdoors”, “Nightlife Spots”, “Travel Spots”, “Shops”, “Home, Work and Others”. Other actions could also be allowed. For instance, in Foursquare users can post tips in specific places aiming at sharing information on any aspect related to the venue.

Instagram, created in 2010, is a photo sharing service that allows users to take pictures, and share them on a several social networking services, such as Twitter. Currently, Instagram users can create Web profiles featuring recently shared pictures, biographical information, and other personal details. Instagram is a very popular photo-sharing service. In February 2013 Instagram announced that they had 100 million users, and in 2014 this number reached 200 million users [Instagram, 2014]. In photo sharing services the main object are photos. Photos could be taken anywhere, e.g., at a restaurant or at home. Thus, the basic action is the sharing of a photo. In Intagram the user can also associate a location with each photo. The action of sharing a photo with a location is also called check-in, because, as in location sharing services, the user is announcing where he/she is at a certain moment.

We have collected data from Foursquare and Instagram directly from Twitter, since Foursquare and Instagram check-ins are not publicly available, by default. Some tweets provide a URL to the Foursquare or Instagram website, where information about the geographic location of the venue was acquired. We have datasets for three different cities: (i) New York – Manhattan region; (ii) Los Angeles; and (iii) London. This cities were chosen because they present different patterns of traffic jams and a high usage of participatory systems.

Our datasets are described by table 3.3. Each line in each dataset represents a check-in, with its latitude, longitude, timestamp and the participatory system id (0 – Twitter, 1 – Foursquare, 2 – Instagram). For New York, we have extracted 65,293 geo-tagged tweets containing check-ins shared on Foursquare in 18,896 unique venues. The Instagram dataset is composed of 12,7185 geo-tagged tweets containing check-ins shared by 41,205 users in 30,738 unique venues and 269 streets segments (we will show how this was collected from Bing Maps in next section). For Los Angeles, we extracted 1,231,222 geo-tagged tweets from Twitter, being 15,156 from Foursquare, and 65,175 from Instagram for 12,374 street segments. For London, we extracted 812,106 geo-tagged tweets from Twitter, 11,652 from Foursquare, and 35,077 from Instagram for

5,046 street segments. The number of street segments is higher in London and Los Angeles because we have used a different dataset from Bing Maps itself.

Table 3.3: Dataset description.

City	Participatory System			Street Segments	Period	
	Twitter	Foursquare	Instagram		From	To
New York	–	65,293	127,185	269	June, 24th, 2014	August, 22th, 2014
Los Angeles	1,231,222	15,156	65,175	12,374	June, 29th, 2014	July, 6th, 2014
London	812,106	11,652	35,077	5,046	June, 29th, 2014	July, 6th, 2014

In summary, since data from sensors in a PSN can provide valuable to better understand city dynamics and the urban behavioral patterns of their inhabitants, an important question emerges: *are they related to with traffic conditions?* It is known that they are associated with users' habits and routines Silva et al. [2014], therefore, maybe we can use data from social sensors, specifically from Twitter, Foursquare and Instagram, to better understand traffic conditions. In fact, one goal of next chapter is to answer this question.

3.4 Chapter Remarks

People typically use map services and social networks in their life in order to avoid traffic jams and to share with others information such as his/her location, photos, videos, or simply messages. In the other hand, there are many kinds of sensors used by ITS, e.g., the traditional sensors as road detectors, and the unconventional sensors, such as map services and social sensors. In this context, this chapter showed how and why can we collect data from map services, presenting several datasets about traffic flow, incidents and social data (check-ins).

The main challenge in the data collection is where to store such huge amount of data and how to extract the traffic flow information from online maps, which are not freely available. For the first, we have used a storage in the Cloud through Microsoft Azure[9]. For the second, we have presented a methodology to acquire flow intensities from map services such as Bing maps and Google maps. Data from several cities have been collected regarding traffic flow and incidents, such as Los Angeles, Chicago, New York, Seattle, London, Paris, Ottawa, Belo Horizonte, São Paulo, and others. These datasets will be analyzed in the following chapters.

[9] We had an award in Microsoft Azure with the Traffic Prediction project. More information are available at `http://research.microsoft.com/apps/video/?id=241812`

Chapter 4

Characterization

In this chapter, we analyze the data that we have collected in the previous chapter in order to understand its effects in different cities. It's a common sense that different cities have different characteristics. Some can be similar, while others are very different. However, *how can we identify such information so as to use as inferred data in prediction models?* Understanding these needs helps to calibrate prediction models to each city, improving its performance. Besides, there are other questions such as: *how do they evolve spatially and temporally?; which data can influence in the other?; how are they correlated?; can social data, from participatory sensor networks (PSNs), be a reflect of the traffic flow behavior?* These are all an example of questions made for the Characterization module that will be answered in this chapter.

In fact, as mentioned in [Silva et al., 2014], data from sensors in a PSN are associated with users' habits and routines. These sensors can be considered social sensors, which provide valuable to better understand city dynamics and the urban behavioral patterns of their inhabitants. Since this is a new type of data many questions emerge. For instance, as it was mentioned, *can we use data from social sensors, specifically from Twitter, Foursquare and Instagram, to better understand traffic conditions?* In fact, answer this question is one goal of this chapter.

Here we describe our contributions in terms of the Characterization module. They are summarized in Figure 4.1. First, we can observe that they were published in [Tostes et al., 2013], [Tostes et al., 2015b] and [Tostes et al., 2012]. Second, we can see that our characterizations are based on data that were collected in the first module of TMS, discussed in the previous chapter, that is, traffic flow, traffic incidents, and check-ins.

We have made the following analyses: (i) spatial and temporal analyses; (ii) the identification of phase transitions; (iii) heterogeneous correlation; and (iv) the design of metrics to analyze the performance of crossroads. The spatial and temporal analy-

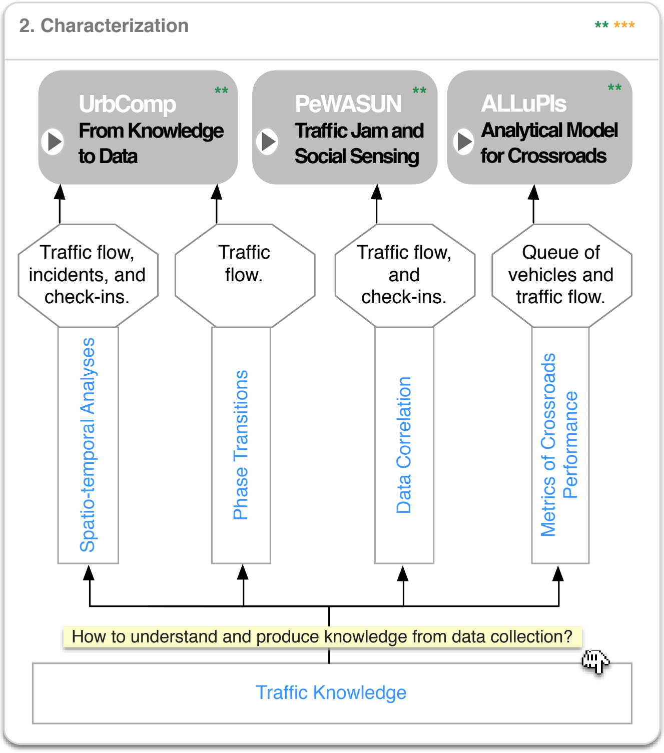

2. Characterization
UrbComp
From Knowledge to Data
PeWASUN
Traffic Jam and Social Sensing
ALLuPIs
Analytical Model for Crossroads
Traffic flow, incidents, and check-ins.
Traffic flow.
Traffic flow, and check-ins.
Queue of vehicles and traffic flow.
Spatio-temporal Analyses
Phase Transitions
Data Correlation
Metrics of Crossroads Performance
How to understand and produce knowledge from data collection?
Traffic Knowledge

Figure 4.1: Description of the Characterization chapter.

ses have been made with the several cities datasets, mentioned previously. Then, the traffic behavior was analyzed based on phase transitions (shocks in the system). In addition, we have correlated data from social sensors (check-ins from Twitter, Foursquare and Instagram) and intense traffic conditions (traffic jam from Bing maps). Finally, we proposed ALLuPIs: an Analytical modeL to evaLUate the Perfomance of IntersectionS [Tostes et al., 2012], which presents some metrics to evaluate the performance of crossroads in accordance with the current traffic flow from the incident roads.

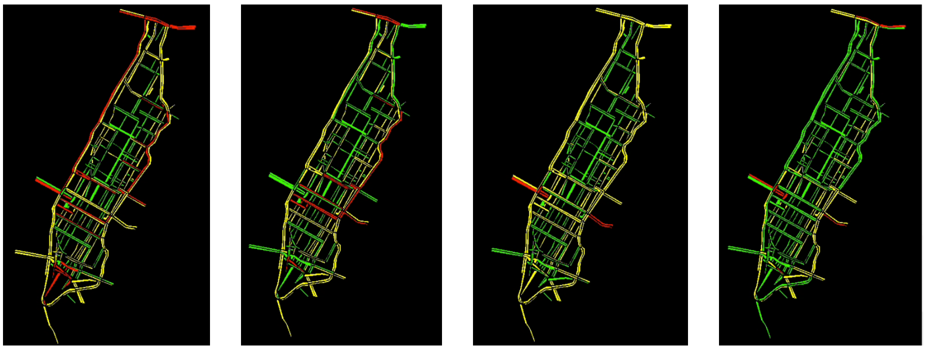
This chapter is organized as follows. Section 4.1 presents the spatial and temporal analyses of our data collected previously. Section 4.2 defines the concept of phase transitions, with some evaluations. Section 4.3 correlates heterogeneous data, that is, intense traffic flow (traffic jam) and check-ins. Section 4.4 explains and evaluates our analytical model ALLuPIs.

4.1 Spatial and Temporal Analyses

In this section, we analyze the traffic flow and the traffic incidents from several cities, focusing on Chicago and New York (Manhattan). As it was mentioned, these cities have been chosen due to data quality in Bing Maps.

4.1.1 About the Traffic Flow

Spatial Analysis. To make a spatial analysis, we generated videos presenting the most likely traffic flow for Manhattan region considering weekdays and weekends. These videos are available online[1]. Intervals of 5 minutes have been used in order to consider most of the traffic flow transitions on streets. Figure 4.2(a) and 4.2(b) show two snapshots during the instant 7AM and 4PM, respectively. We can see which street segments are more congested and which streets have free flow. Comparing with Figures 4.2(c) and 4.2(d), we can see that at 7AM the yellow traffic flow is more likely than at 4PM during weekdays. In weekends, the free flow is more likely in such hours than during weekdays.



(a) Monday, 7AM (b) Monday, 4PM (c) Weekend, 7AM (d) Weekend, 4PM

Figure 4.2: Most likely traffic flow in time intervals of 5 minutes.

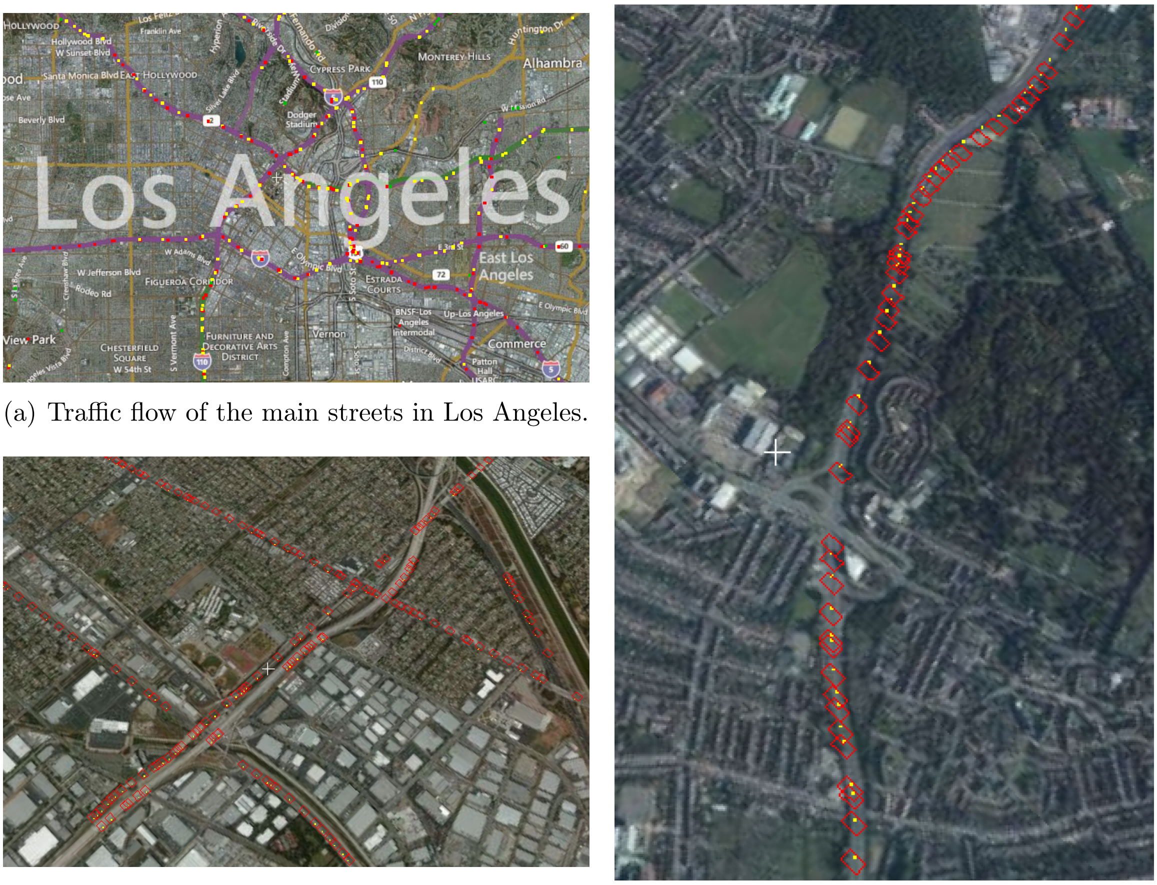
Dataset from Microsoft Research. Here we have analyzed the dataset obtained from Microsoft Research about the traffic flow from Bing Maps. We have considered the check-ins dataset in order to consider only street regions with check-ins data.

Figure 4.3 illustrates a closer look on the main streets for Los Angeles and London. We can observe that it is a good classification. Also, recall that each street segment has several sectors that can be combined to generate the traffic flow for the street segment, as related in Section 3.1.1, in Chapter 3. Future works may combine the flow of sectors on the same street segment to generate a street traffic flow. In this study, we consider the lowest granularity that is each sector as a street segment.

In addition, Figures 4.3(b) and 4.3(c) show a closer look to street sectors in Los Angeles and London, respectively, when we consider also the check-ins dataset. Recall that yellow squares represents the street segment and the red squares represent

[1] `http://annatostes.azurewebsites.net/bing-maps-videos/`

the polygon that we use to count the check-ins on each street segment. We consider 0.0002 as the latitude/longitude error from the center of the square, which represents approximately 20 meters from the center.



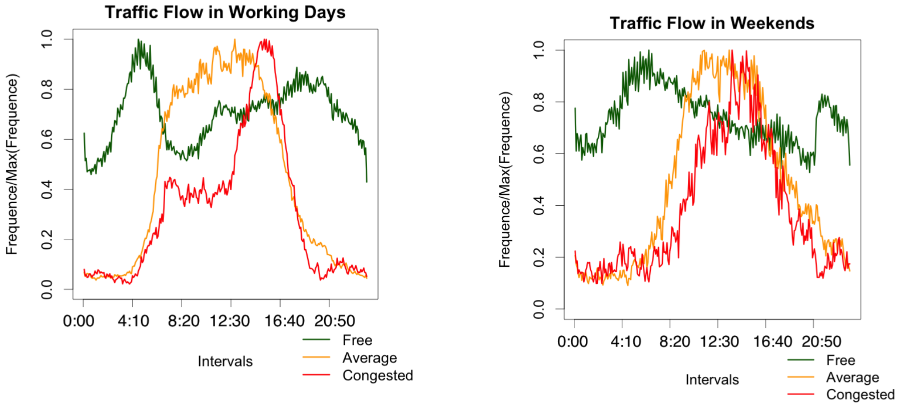
(a) Traffic flow of the main streets in Los Angeles.

(b) Sectors with check-ins box as street segments in Los Angeles.

(c) Sectors with check-ins box as street segments in London.

Figure 4.3: Closer look on the main streets in cities, considering the check-ins.

Temporal Analysis. To make a temporal analysis of the traffic flow for Manhattan region we designed Figures 4.4(a) and 4.4(b), which presents the frequency of green, yellow and red traffic flow during weekdays (Monday to Friday) and during weekend (Saturday and Sunday). One can notice that the traffic congestion has two peaks of congested traffic flow, at $x = 85$ (approximately 7am) and at $x = 192$ (4PM), corresponding to rush hours (when people drive to work and to home). During weekends, one can notice that we have only one peak when $x = 175$ (approximately 3PM), but the free traffic is more usual that does not occurs on weekdays in which the congested traffic is more usual (yellow and red flows).



(a) Weekdays

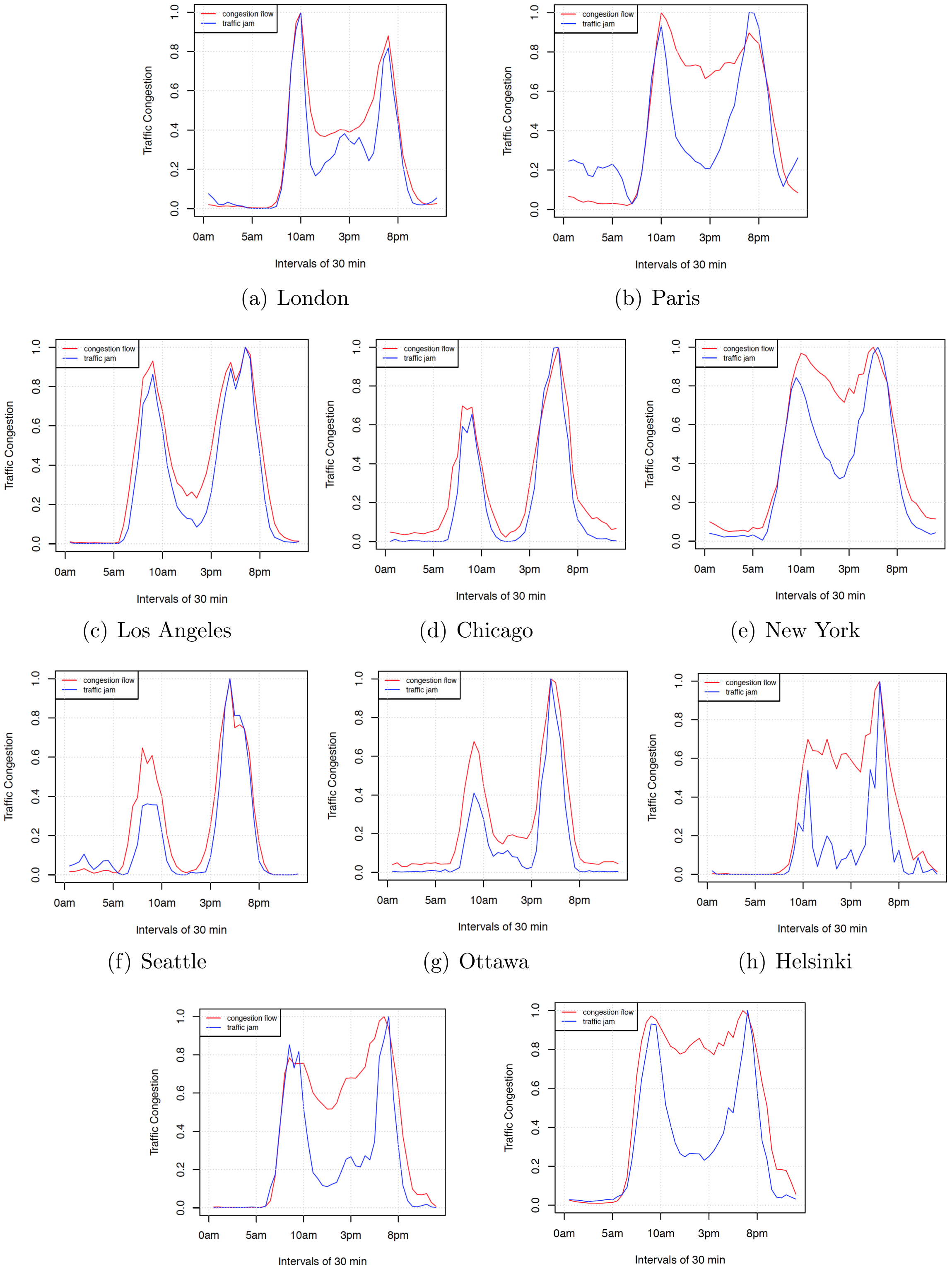
(b) Weekends

Figure 4.4: Frequency of traffic flow aggregate in time intervals of 5 minutes.

Comparison Between All Cities. Here we analyze how the traffic flow temporally evolves in different cities world wide. Figure 4.5 illustrates the hourly variations of intense traffic conditions in the following cities: London, Paris, Los Angeles, Chicago, New York, Seattle, Ottawa, Helsinki, Belo Horizonte, and São Paulo. We have two series: congestion flow and traffic jam. The series of congestion flow is calculated as the frequency of Bing values 2 (yellow – congested) and 3 (red – jams), while the series of traffic jam consists only of the frequency of Bing value 3 (red – jams). Note that the curve of each city follows almost the same pattern observed for all locations: two peaks of intense traffic conditions, one in the morning and other in the evening.

Notice that these peaks reflect distinct rush times that are related to the common working hours in different cities. For instance, in London (Figure 4.5(a)) the morning peak is around 10 AM, as in Paris (Figure 4.5(b)) and in Helsinki (Figure 4.5(h), with smaller intensity since it is a smaller city). In contrast, in New York, Los Angeles, Chicago and in the Brazilian cities (Figures 4.5(i) and 4.5(j)), the morning peak is usually two hours later, suggesting that people tend to leave later to work in those cities. The second most expressive peaks is at 6PM in London, Paris, New York, Chicago, Helsinki, Chicago, while in Belo Horizonte and São Paulo this time is at 7PM, being one hour later. It is also interesting to recall that in cities such as Seattle and Ottawa, which are smaller and have a colder weather, people tend to leave later to work, at 9PM, and they also tend to arrive earlier, at 5PM.

In addition, we can observe in Paris, New York, Belo Horizonte and São Paulo that the congestion flow starts in the morning, but it stays higher during all day until the other rush hour peak in the evening. This does not happen in other cities, such as London, Los Angeles, Chicago, Seattle and Ottawa, where we have two clear peaks of intense traffic flow and a decrease in the afternoon.



(a) London (b) Paris

(c) Los Angeles (d) Chicago (e) New York

(f) Seattle (g) Ottawa (h) Helsinki

(i) Belo Horizonte (j) São Paulo

Figure 4.5: Evaluation of the traffic flow in several cities, considering a typical weekday.

4.1.2 About the Traffic Incidents

Traffic jam is always caused by some reason, even when we can not explain (invisible jams). In order to understand why traffic jam occur in our previous dataset, we have collected data regarding traffic incidents, also from Bing Maps, as we mentioned in Chapter 3. We have chosen London, Los Angeles, Chicago, and New York, since they had more traffic incidents than the others.

Q1. What are the most common types of traffic incidents according with the days of the week? Are they caused more due to accidents, or construction, or weather, or other type?

Figure 4.6 illustrates the amount of traffic incidents per type and day of the week, from Sunday up Saturday, in the fours cities that we have analyzed. We can observe that accident, construction and miscellaneous are the most reported types of incidents.

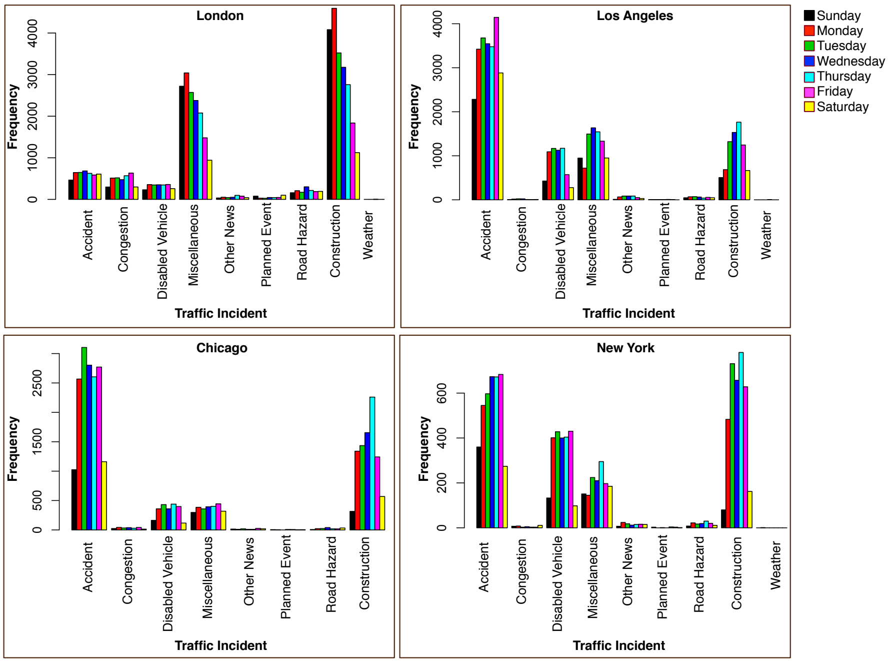


Figure 4.6: Amount of traffic incidents per type and weekday.

In London, there are more miscellaneous and construction incidents. These incidents are more reported on Monday and Sunday. Accidents tend to occur more often in Wednesday, while congestion incidents are more common on Friday.

In Los Angeles, we have more accidents than others, with similar occurrences of disabled vehicles, miscellaneous and construction. Notice that accidents occur more often on Friday. Incidents of miscellaneous occur more often on Wednesday. Construction and disabled vehicles are more common on Thursday.

In Chicago, there are more accidents and construction. Most accidents occur on Tuesday, while most constructions are reported on Thursday. Regarding disabled vehicles and miscellaneous, we can observe that they occur more often on weekdays.

Similar as in Chicago, New York has more accidents and construction incidents. Accidents usually occur on Friday, while construction incidents are more often reported on Tuesday and on Thursday.

Q2. How traffic incidents evolve spatially and temporally?

Here we analyze the distribution of traffic incidents spatially. Figure 4.7 presents the spatial distribution of incidents, being represented in blue the slow congestion, in red the road closed, and in black incidents that did not cause road closure.

According with our results, we can observe that London has more slow congestion than others being reported. In Los Angeles, Chicago and New York, there are more road closed incidents in comparison with London. Also, we can notice that most of them happen in the downtown.

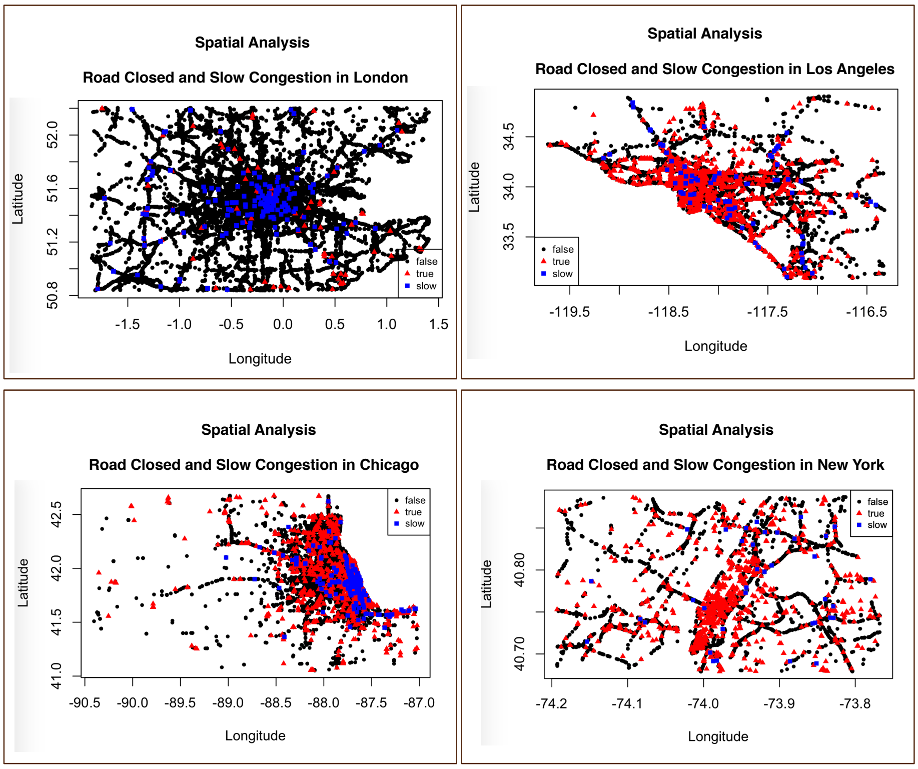


Figure 4.7: Spatial analysis of traffic incidents per city.

Figure 4.8 presents a temporal analysis for accidents in Los Angeles, Chicago and New York. We can observe that the frequency of accidents in Los Angeles achieves peaks at 1PM, 3PM and 9PM. In Chicago, the peaks are at noon and 8PM. In New York, the peaks are at 10AM and 6PM.

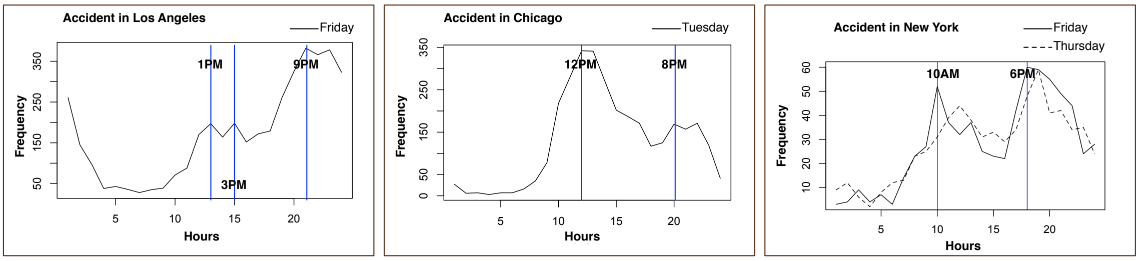


Figure 4.8: Temporal analysis of accidents in different weekdays.

In addition, Figure 4.9 illustrates how do the traffic incidents of several types evolve temporally in these different cities. In London, we can observe that the construction incidents on Tuesday have different peaks than on Thursday (6AM, 6PM and 11PM in comparison with 10AM and 9PM), but they are concentrated to occur mostly in the evening. This does not happen with Los Angeles, where the highest peak for construction and disabled vehicle is at 2PM, for miscellaneous is at 3AM, 2PM and 4PM and for construction is at 4AM, 2PM and 4PM. In Chicago, disabled vehicles and miscellaneous are not as expressive as accident and construction, which have higher peaks at 1PM and 4PM. The latter has peaks at 2AM, 9AM, 1PM and 7PM.

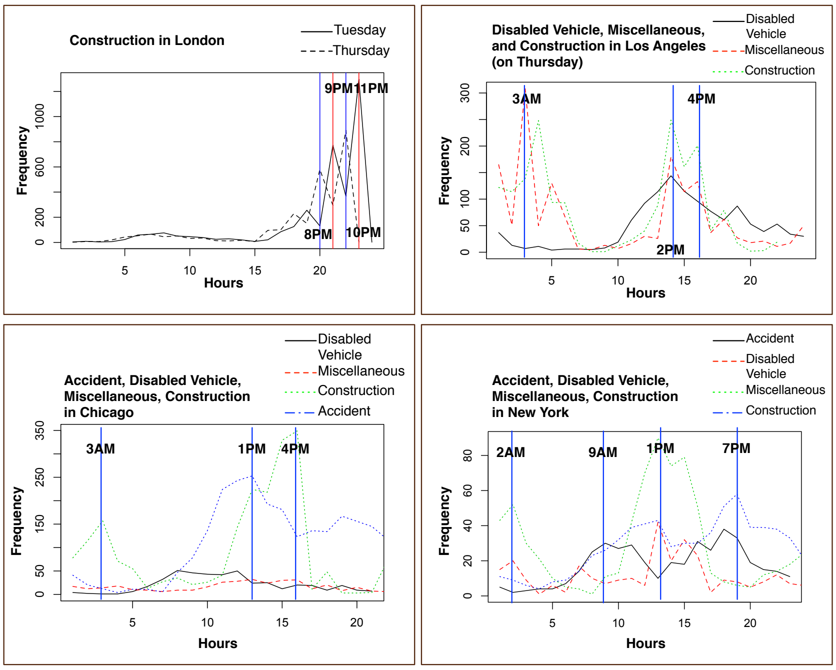


Figure 4.9: Temporal analysis of traffic incidents per type and weekday.

Q3. How the severity of traffic incident impacts in the traffic jam?

Here we analyze the severity of traffic incidents per day of the week. Figure 4.10 illustrates the results for London, Los Angeles, Chicago and New York. Recall that, in London, most traffic incidents with lower and minor severity occur on Sunday or Monday, while the more severe (serious) occur on Thursday and Friday. Regarding to Los Angeles, we can see that, despite the severity of the traffic incident, they are concentrated on weekdays, specially on Tuesday to Thursday. The same happens for Chicago, concentrating its incidents on Tuesday to Thursday, without much difference on the levels of severity. Similar to London, New York concentrates the more severe incidents (moderate and serious) specially on Friday, while the minor incidents are concentrated on Monday and Tuesday.

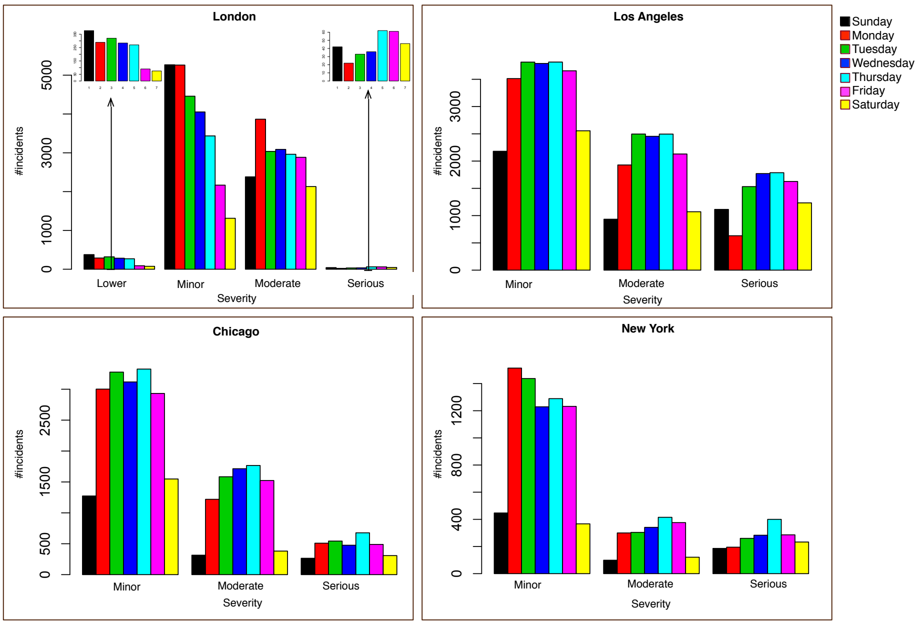


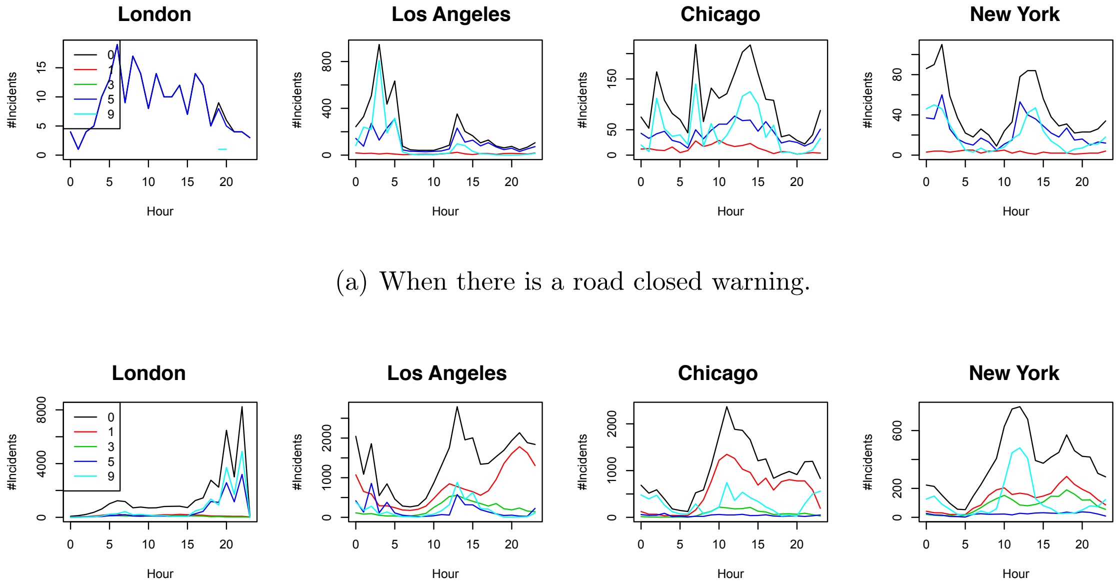
Figure 4.10: How and when less and more severe traffic incidents occur.

Q4. How a road closure impacts in the traffic congestion?

One interesting variable related to traffic incidents is the road closed, which indicates is the road was closed due to the traffic incident or not. In this context, the question that emerges is how this road closed information impacts the distribution of traffic incidents temporally, that is: *"does the temporal evolution of traffic incidents changes when the road is closed?"*

Figure 4.11 presents the results, being Figure 4.11(a) related to incidents when the road is closed and Figure 4.11(b) when the road is not closed. Notice that, in

London, we have only traffic incidents of accidents, disabled vehicles and construction when the road is not closed. In Los Angeles, Chicago and New York, we can observe that the peaks of incidents for each type change when we consider the traffic incidents. For instance, in Los Angeles, we have more incidents (0 - black) in the night and afternoon when there is a road closed warning, but when there is not, the incidents are more likely to occur in the evening and specially in the afternoon.



(a) When there is a road closed warning.

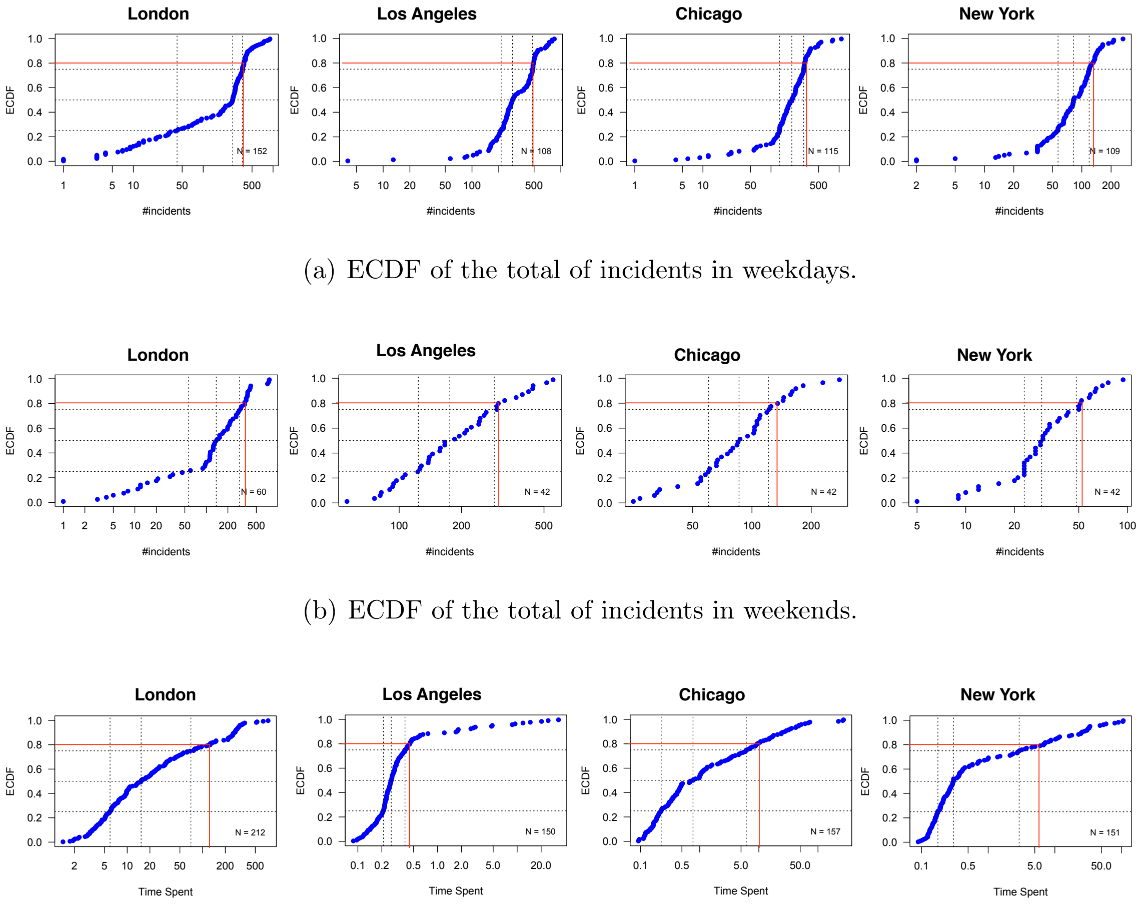
(b) When there is not a road closed warning.

Figure 4.11: Analysis of the impact of road closure in the report of traffic incidents.

Q5. Do the traffic incidents vary in weekdays and weekends? And what is the time duration of most traffic incidents in the city?

Figure 4.12 illustrates the empirical cumulative distribution of accidents in each city, considering weekdays (Figure 4.12(a)), weekends (Figure 4.12(b)), and of the time duration, in hours, of such traffic incidents (Figure 4.12(c)). The x-axis and y-axis are both presented in log-scale. Notice that we have lines indicating 80% of probability to occur more than a certain amount of accidents in each city. The grids indicate the 25th, 50th and 75th quartiles.

We can observe how the probability of incidents evolve during a typical day. Regarding the time duration, we can notice that most traffic incidents in London have a higher duration than in other cities (0 up to days), since it happens more on highways. As for Los Angeles, Chicago and New York, incidents with lower duration are more likely to occur than others (respectively 30min, 6 hours and 5 hours of time duration



(a) ECDF of the total of incidents in weekdays.

(b) ECDF of the total of incidents in weekends.

(c) ECDF of the time duration (in hours).

Figure 4.12: Empirical cumulative distribution function of incidents and their duration.

with 80% of probability). Finally, we can see that it follows a power law, indicating that places with less incidents are more common than with more incidents.

Q6. Does traffic accidents have an impact in the traffic congestion?

Figure 4.13 analyzes the impact of traffic accidents in the traffic congestion. We have calculated the frequency of accidents in streets with at least 1,000 incidents that were considered as moderate or severe. Then, we have calculated the frequency of intense traffic flow (Bing values of 2– yellow or 3 – red) in streets that have not achieved 1,000 accidents. Notice that we did not want to classify streets. Instead, we can have different streets in distinct time intervals.

Despite the different x-axis scales, the important thing is to notice these two series are different. In London, for instance, most traffic congestion that occur in the morning might be due to accident, but not in the evening. In Los Angeles, the opposite occur. The curves are similar in the evening, but not in the morning as in London. In Chicago we have a similar behavior than in London. Finally, in New York, we have a similar shape than in Los Angeles, being similar in the evening.

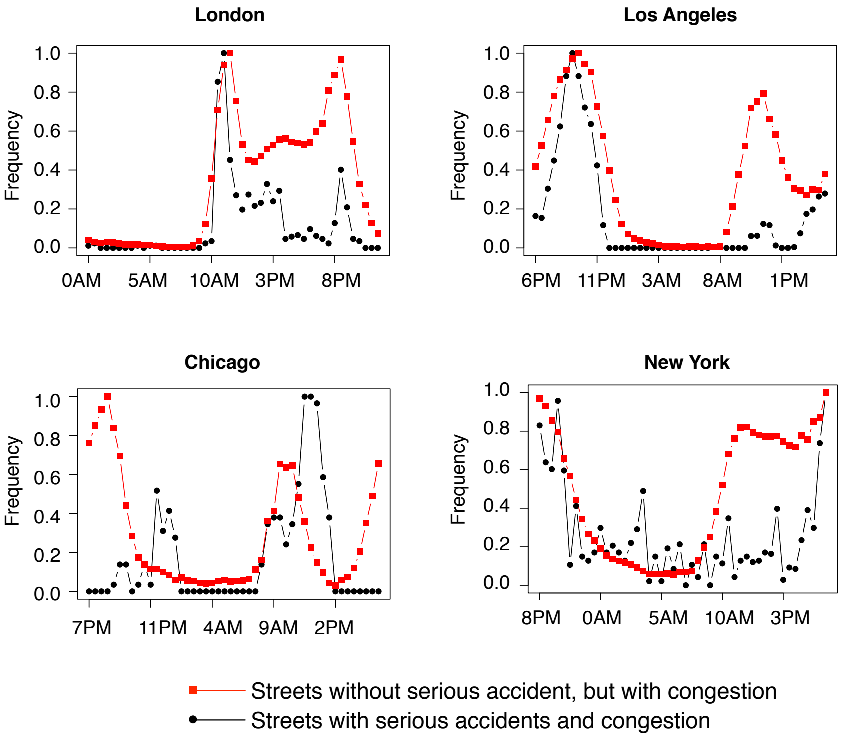


Figure 4.13: How much do accidents impact in the traffic congestion?

4.1.3 About the Social Dataset

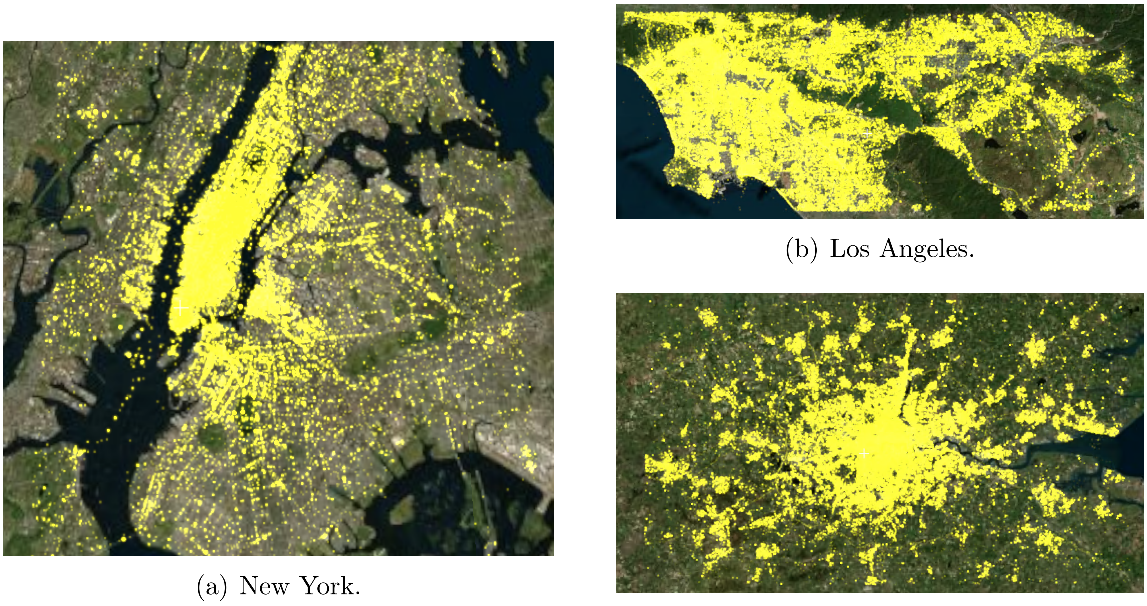
In this section, we analyze two aspects of our social datasets presented in Section 3.3, in Chapter 3: (i) network coverage; (ii) seasonality.

Network Coverage. Figure 4.14 shows the occurrence of check-ins in New York, Los Angeles and London using Worldwide Telescope[2], a Microsoft Research tool. Videos were also produced and they are available online[3]. They show how the amount of check-ins varies temporally and spatially. About the spatial aspect, notice that check-ins occur more often in downtown for all cities.

Since we are interested in the check-ins nearby or over a street segment, we have to filter the dataset. So we have created a polygon around each street segment and we

[2] http://www.worldwidetelescope.org/

[3] http://annatostes.azurewebsites.net/



(a) New York.

(b) Los Angeles.

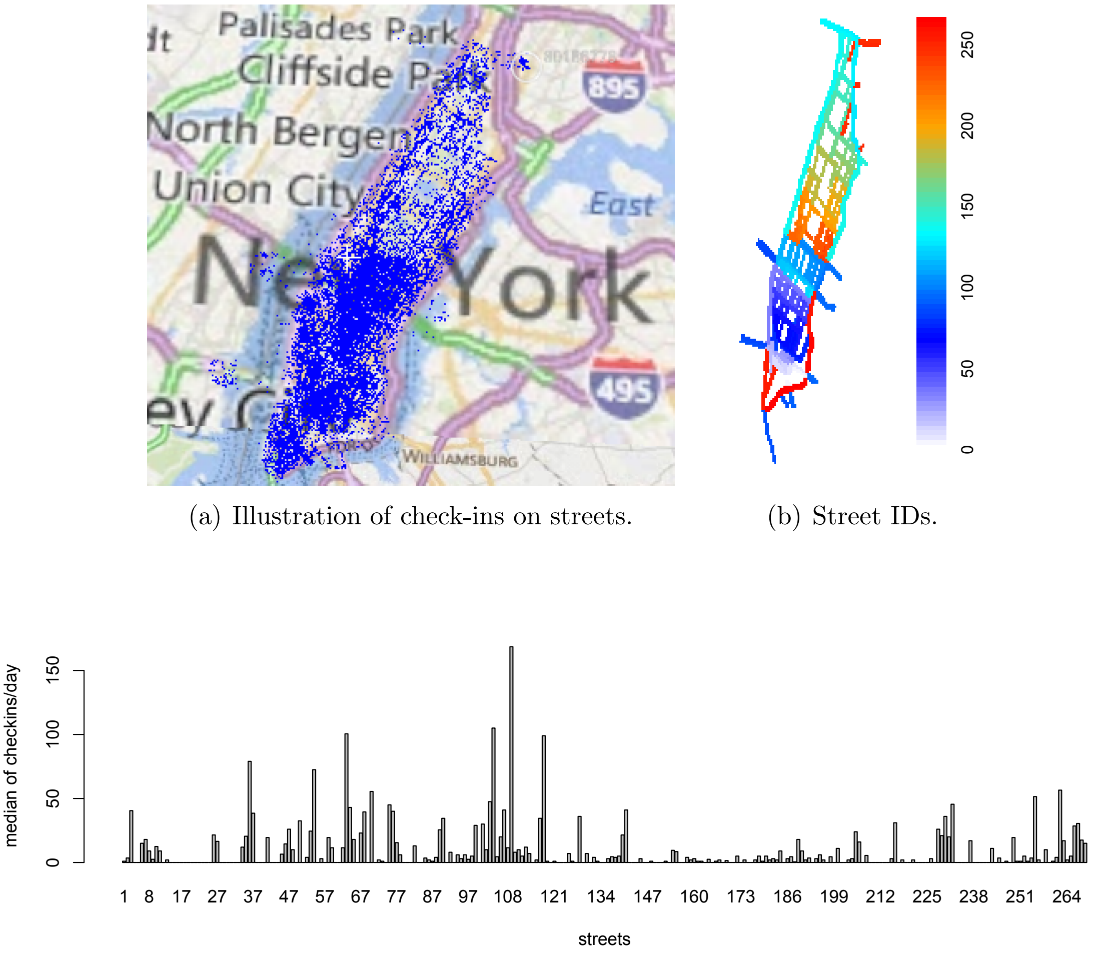
(c) London.

Figure 4.14: Check-ins over cities.

have counted the number of check-ins that occur inside it. Figure 4.15(a) illustrates the check-ins on streets. Additionally, in Figure 4.15(b), we can see the street id. Notice that the street IDS for downtown avenues are less than 120, despite Broadway which is around 220. Figure 4.15(c) shows the median of the amount of check-ins per day in Manhattan. We can observe that there are more check-ins over downtown avenues, such as Time Square, and 5th-9th avenues. Counterwise, we have less check-ins in the north of Manhattan and some streets have no check-ins at all.

Although there are some streets with more check-ins than others, it is important to notice that we have almost the same peaks for all streets. To demonstrate that, Figure 4.16(a) represents a heatmap of the relative check-ins in a time interval, that is, the amount of check-ins in a time interval divided by the maximum amount of check-ins in each street. The color indicates the relative amount of check-ins, from blue (less checkins) to red (more check-ins). Notice that the white columns represent streets with no check-ins. Recall that streets have more check-ins in the evening.

Additionally, it is also important to verify if the check-ins are well distributed on it, depending on the area of this polygon. For that, we calculated the percentage of the polygon area that is occupied by check-ins over the entire New York dataset, which is shown by Figure 4.16. In the y-axis we have intervals of 30 minutes during one typical weekday (a total of 48 intervals of 30 minutes). The color in Figure 4.16(b) means how well the check-ins over a street segment are distributed, from blue (lower coverage distribution) to red (higher coverage distributed).



(a) Illustration of check-ins on streets.

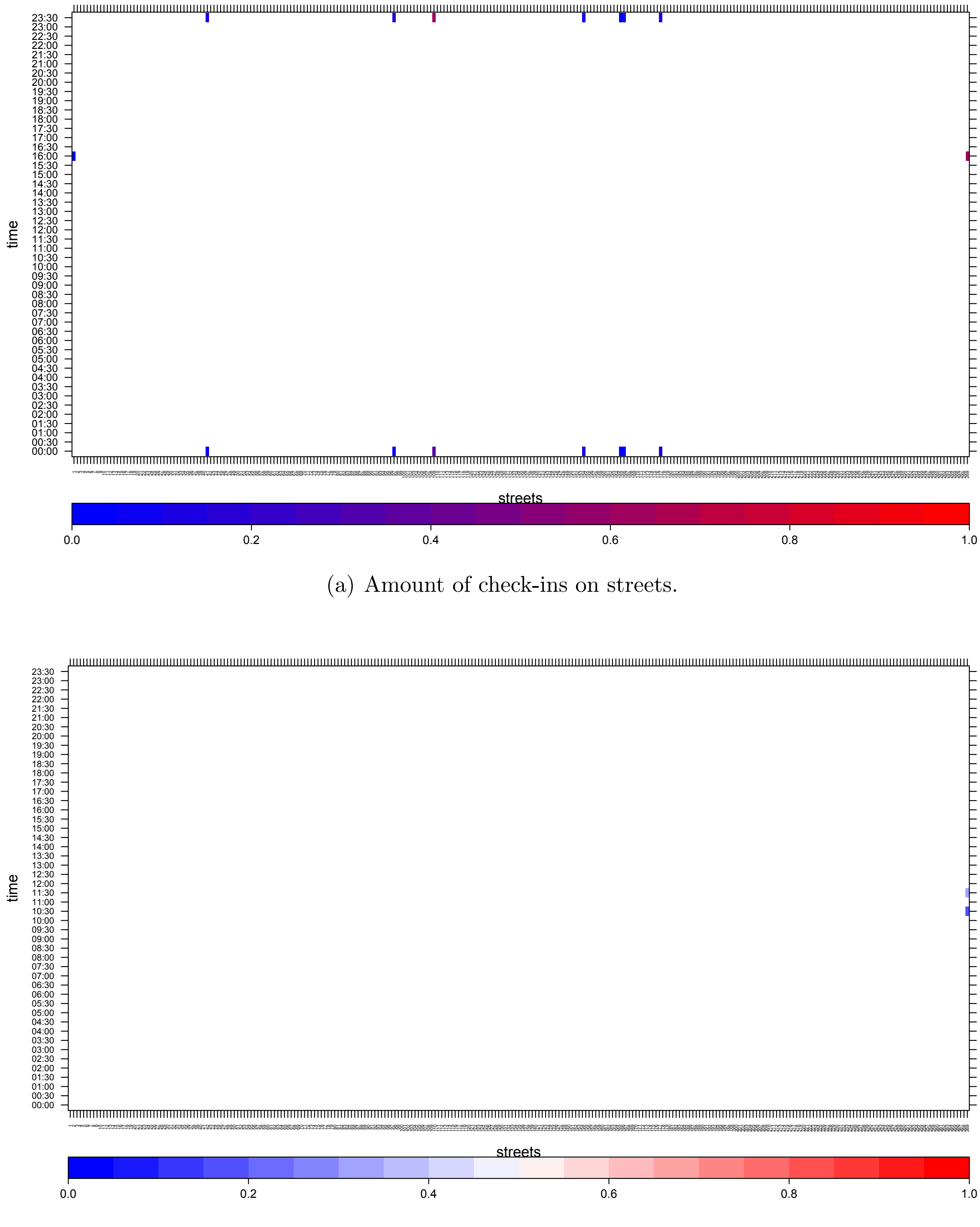
(b) Street IDs.

(c) Median of check-ins per day.

Figure 4.15: Coverage of check-ins per day.

Recall that streets with highest coverage are more frequent from 9AM to 8PM, with the highest values in the evening. The streets with lower coverage tends to stay in the same behavior during all day long. There are also streets with no check-ins at all. Thus, if we compare both heatmaps at the same time, we can see that there are five categories of streets: (i) with a lot of check-ins that are well distributed; (ii) with a lot of check-ins that are not well distributed, so they are concentrated in some specific part of the street; (iii) a lower number of check-ins well distributed; (iv) a lower number of check-ins concentrated in a region; and (v) no check-ins.

Figure 4.17 summarizes a spatial representation of these categories. We can observe that most of downtown avenues are on categories (i) and (ii). Categories (i) and (ii) are more interesting, because we have to consider streets with more check-ins than a certain minimum threshold to be able to use in traffic jam forecasting models.



(a) Amount of check-ins on streets.

(b) Street coverage distribution.

Figure 4.16: Analysis of how well check-ins are distributed on the street.

That is why we do not consider categories (iii) and (iv). Besides, categories (i) and (ii) might have a different influence in the accuracy of prediction models. Similar to categories (iii) and (iv), category (v) has to be not considered due to the minimum

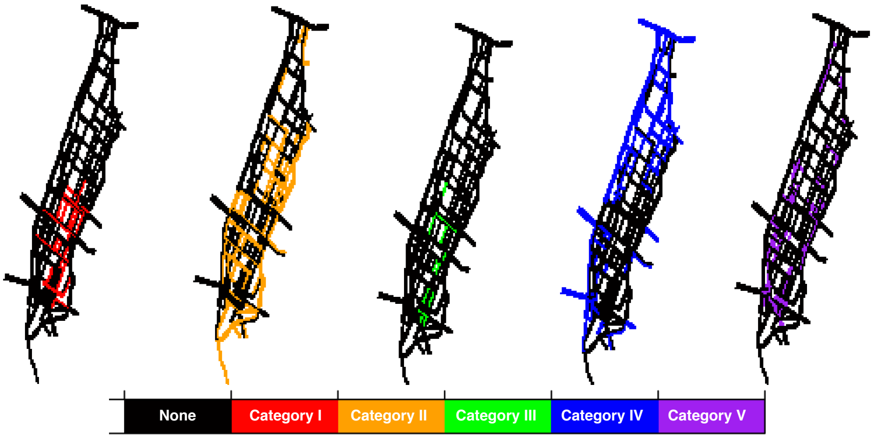


Figure 4.17: Categories of streets in New York.

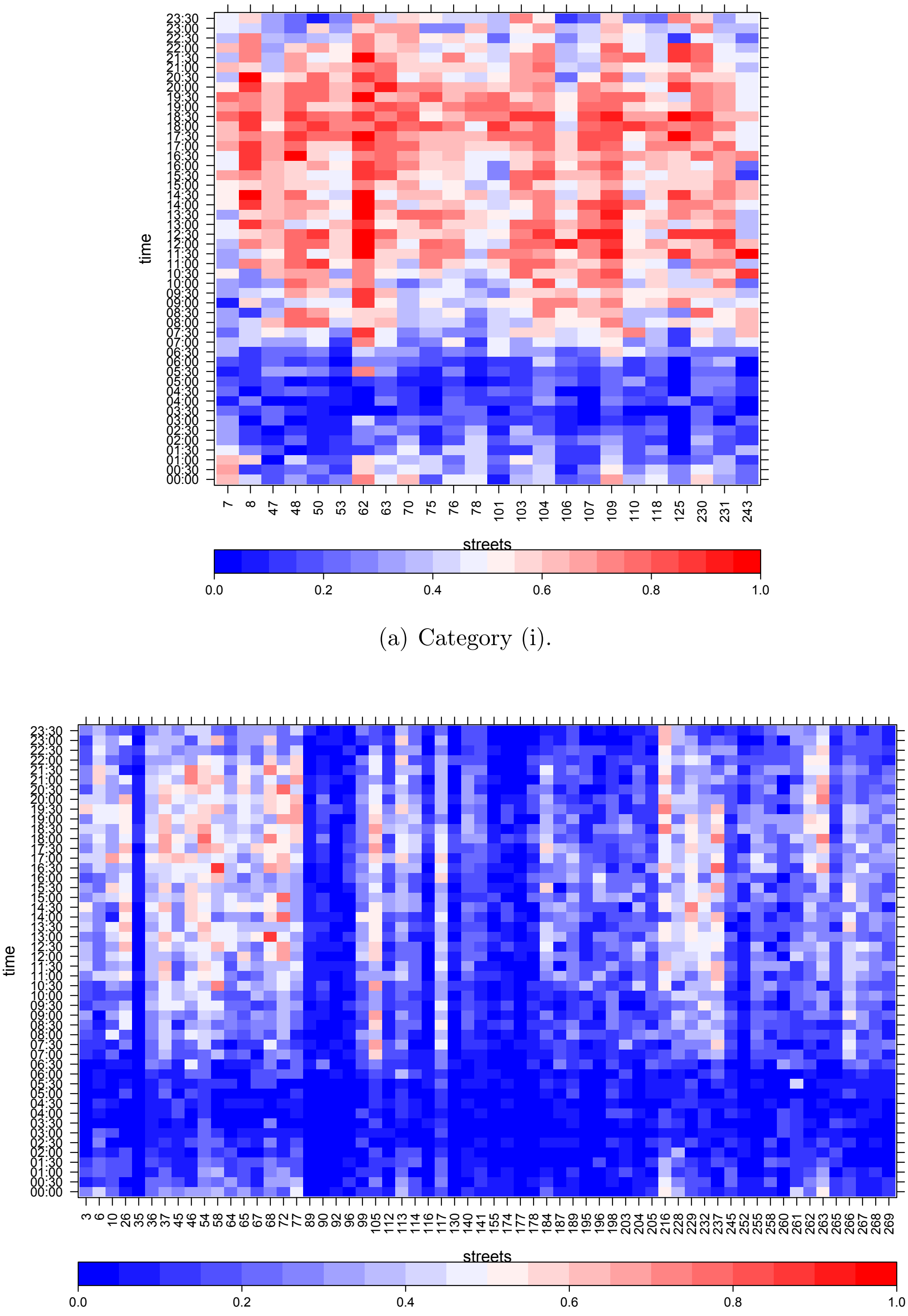
amount of check-ins requirement.

Since we are interested in categories (i) and (ii), Figure 4.18 remarks the street coverage distribution for streets in category (i) and in category (ii). Recall that category (i) is more well distributed than category (ii), as mentioned.

Seasonality. We now analyze how the seasonal behavior affects the location-sharing data. Timestamp were normalized for each city local time.

Figure 4.19 illustrates the number of check-ins throughout the hours of the day in different cities, considering all check-ins and just the check-ins on streets. Recall that we have two curves, one from the entire dataset, including all the geolocated tweets for Los Angeles and London, and another from Foursquare–Instagram tweets only. For New York, we have collected only check-ins from Foursquare/Instagram. Notice that we have three peaks of check-ins, one in the morning at 8AM, other at 12PM, and another in the evening at 6PM. From Los Angeles and London, we have collected all geolocated tweets. For the Los Angeles *all* dataset, we have one peak in the evening at 8PM and two peaks in the morning, at 8AM and at 12PM. Notice that in this case we have a little drawback at 1PM. Considering only tweets from Foursquare/Instagram, we have two peaks, one at 10AM and another at 4PM. For London, the shape of the curve is similar to New York for the *all* dataset, but with three peaks, one at 10AM, other at 5PM and another at 9PM. For the Foursquare–Instagram dataset, we have only two peaks, similar to Los Angeles, one at 12PM and another at 6PM.

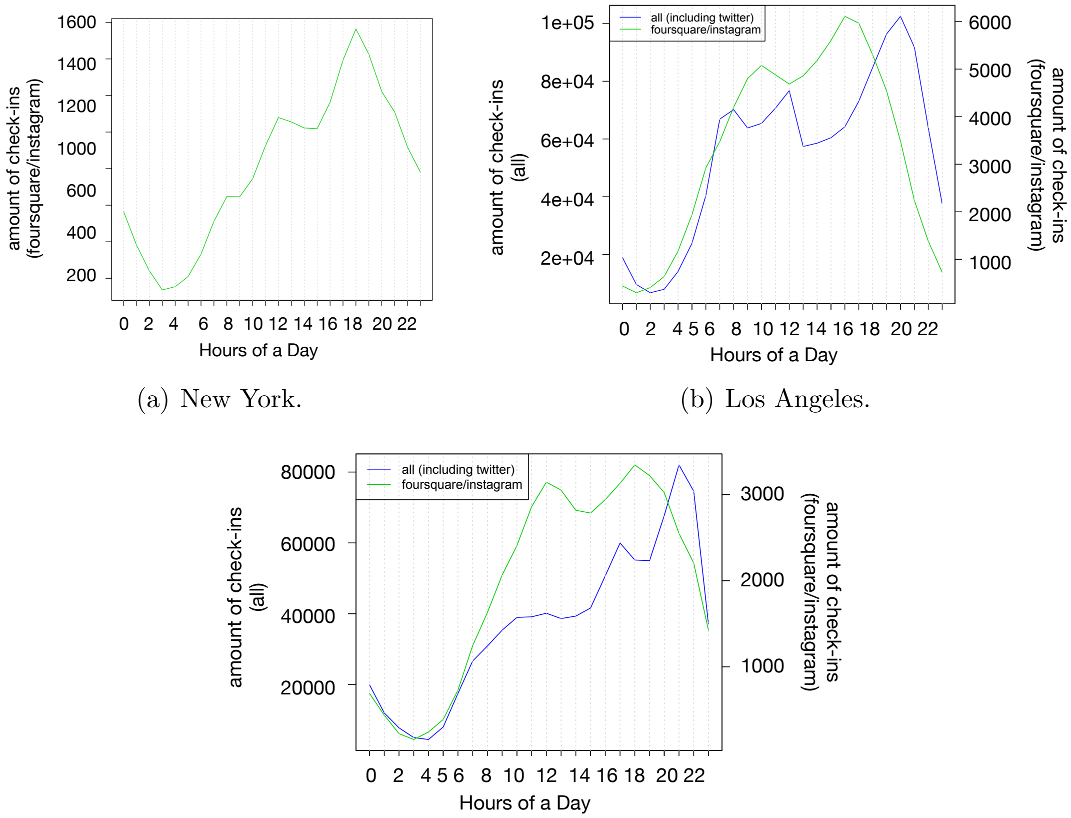
Figure 4.20 presents the time series of accumulated check-ins on streets per days of the week for New York, considering check-ins on Foursquare and Instagram (pink), only on Foursquare (green) and only on Instagram (blue). Notice that there are more



(a) Category (i).

(b) Category (ii).

Figure 4.18: Heatmap of street coverage distribution in categories.



(a) New York.

(b) Los Angeles.

(c) London.

Figure 4.19: Amount of check-ins throughout the hours of the day in different cities.

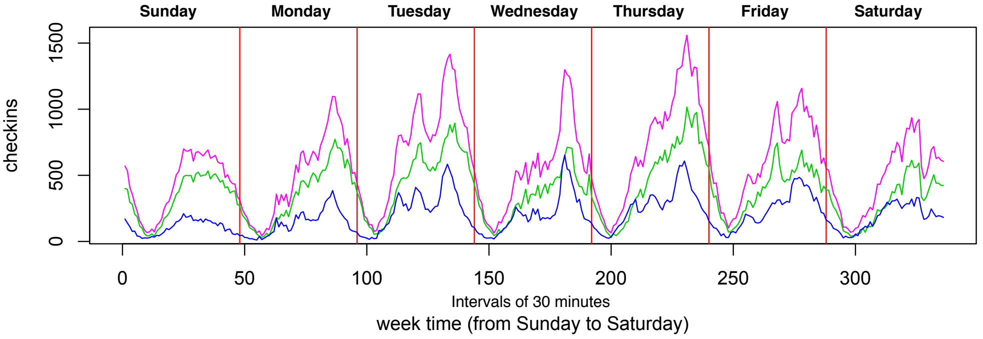
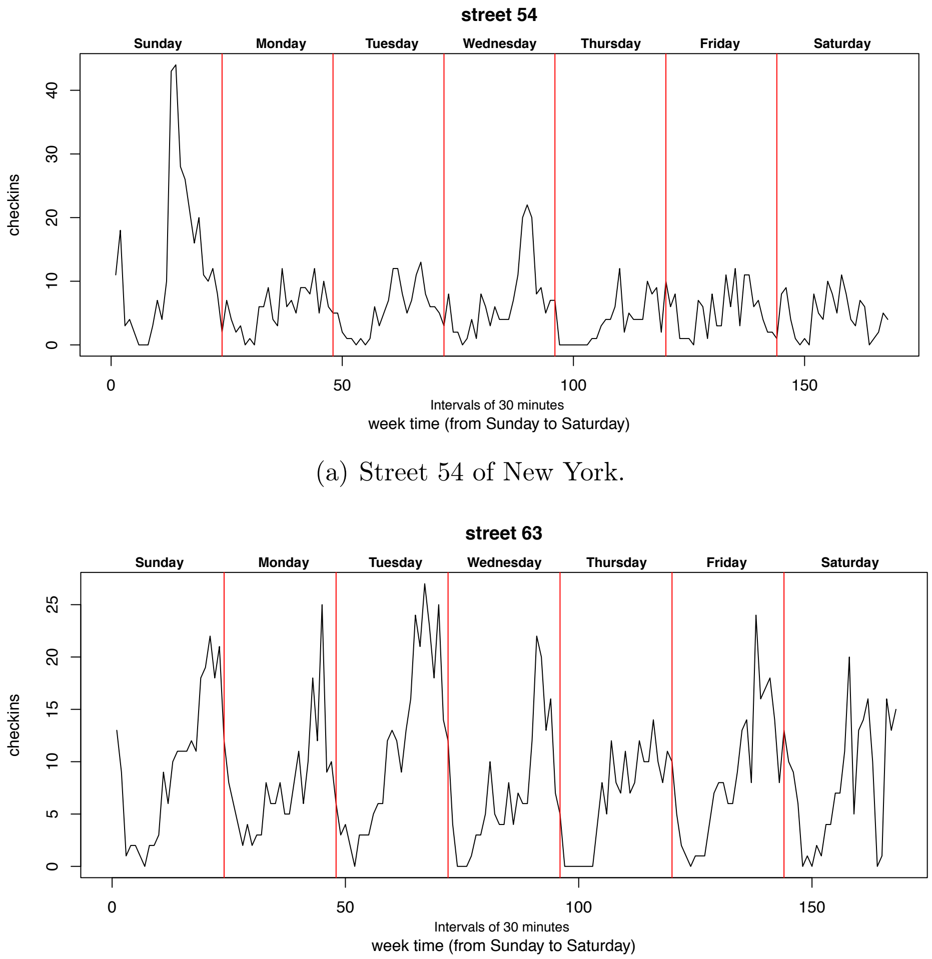


Figure 4.20: Time series of accumulated check-ins on streets per days of the week for New York.

check-ins from Foursquare than from Instagram. The amount of check-ins is higher on weekdays, specially in the evening. Tuesday and Thursday have more check-ins.

This is the average pattern for the city, but if we analyze in the street level, we can see that check-ins vary spatially and temporally. As an example, Figure 4.21 illustrates two time series of the amount of check-ins on a street segment in the half-hours of one



(a) Street 54 of New York.

(b) Street 63 of New York.

Figure 4.21: Time series of the amount of check-ins on a street segment for one specific day.

specific day. Notice that for street 54 we have considerably more check-ins on Sunday than on other days, which is different from street 63, in which we have approximately the same amount of check-ins per day. Also, we can observe that the pattern of check-ins varies considering the street segment, day and time. For this reason, in Section 4.3, we consider the time series in the street level and for different time intervals so as to apply check-ins data to the traffic jam forecasting model.

4.2 Phase Transitions

4.2.1 Definition

In this section we define and present an example of a phase transition.

Definition 5. Traffic Flow Phase: time frame when the traffic flow in a specific street does not vary, that is, has always the same Bing value.

Considering the definition of a traffic flow phase, we can define a phase transition:

Definition 6. Phase Transition (PhT): occur when the traffic flow changes abruptly from one phase to another phase, different from the previous.

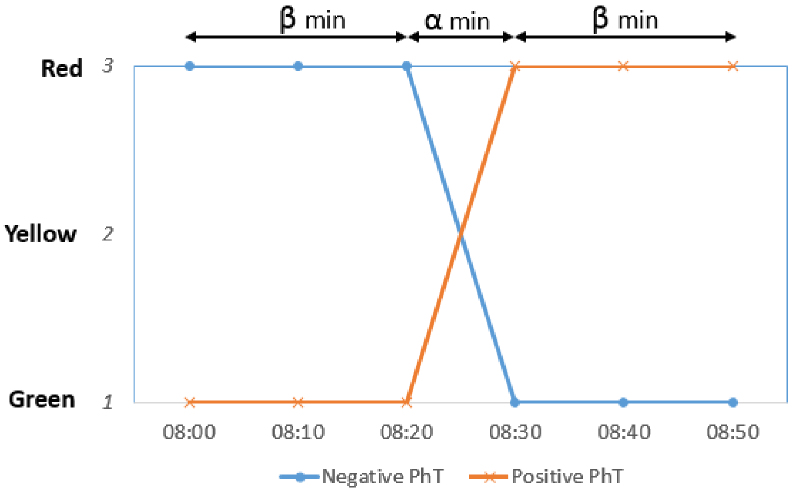


Figure 4.22: Illustration of the definition of a positive and a negative PhT.

That is, considering Figure 4.22, notice that we have two phases with β duration, and an abrupt change in the Bing value – one series from red to green and the other from green to red. Since this abrupt change (α) is shorter than the phase duration (β), this can be considered as a phase transition. In this example, a street has a traffic jam for β = 20min, and after α = 10min, the traffic flow changes to a free phase for β = 20min. Similarly, the opposite can occur: a street has a free traffic flow for β = 20min, and after α = 10min, the traffic flow is jammed for β = 20min.

These two situations indicate abrupt changes in the traffic flow. In one case, the traffic is getting better, and in the other, the traffic is getting worst. Therefore, we have the definitions of positive and negative phase transitions.

Definition 7. Positive Phase Transition (PhT+): occur when the phase transition is from free flow to traffic jam (green-to-red transition), indicating that the traffic flow is getting worst ("positive to jam").

Definition 8. Negative Phase Transition (PhT-): occur when the phase transition is from traffic jam to free flow (red-to-green transition), indicating that the traffic flow is getting better ("negative to jam").

Nevertheless, identifying these moments of PhT is an arduous task as its prediction depends on the amount of available occurrences, which can vary. PhT can be classified as regular or irregular. Regular PhT can happen in " moments" socially known by the population such as rush hours, while the irregular PhT occurs in adverse times, as in the occurrence of traffic accidents, congestion waves or unexplained congestions. Therefore, algorithm 4 presents how to identify a PhT. With a sliding window, we can identify a PhT based on the values of α and β.

Algorithm 4 Algorithm to identify a PhT

```
Require: Dataset from our traffic flow acquisition methodology, based on Bing Maps
    information, time limit $t_{limit}$, $\alpha$, and $\beta$
Ensure: Periods of time that occured a PhT.
while current time $t < t_{limit}$ do
    Initiate a sliding windows of $\alpha + 2\beta$ minutes
    if traffic flow matches with a PhT then
        Register the PhT
    end if
end while
```

4.2.2 Evaluations

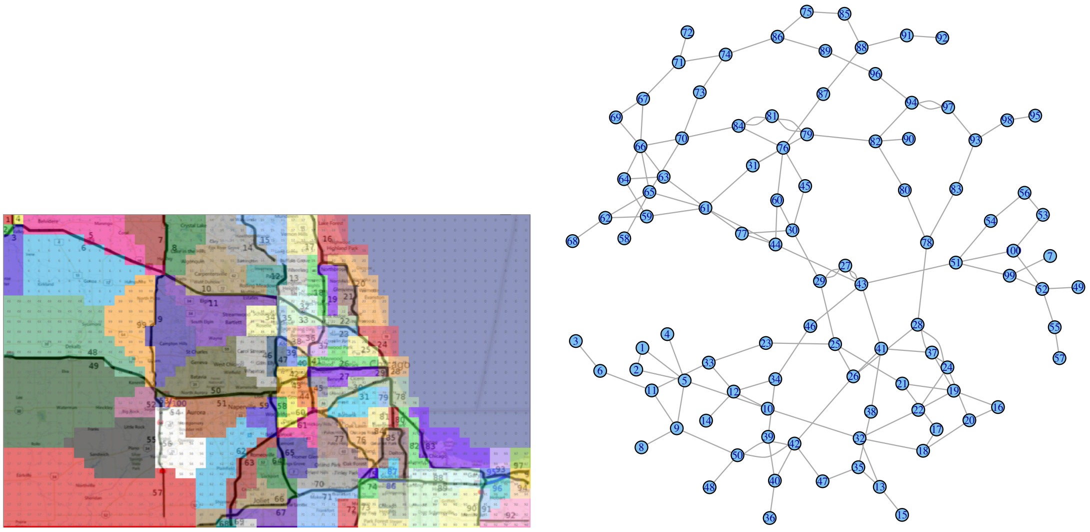
Here we have made two case studies involving the concept of phase transitions. First, we have used a dataset from Chicago. Second, we have used a dataset from Manhattan, in New York.

4.2.2.1 Chicago Case Study

Chicago was our first study case. The methodology of Flow Acquisition was applied. We collected data from April 10th 2013 to April 24th 2013. Data from Bing maps were collected every 7 minutes. Next, the database has the following information: (I) date; (II) hour; (III) street number; (IV) number of green pixels; (V) number of yellow pixels; (VI) number of red pixels; and (VII) number of no category pixels.

Chicago map was divided into geo-code sectors, as Figure 4.23 illustrates. Each street has its influence area, considering the direction of the street. It is important to notice each street number. Also, we mapped Chicago as a graph (see Figure 4.23(b)): the vertexes are roads and the edges are the direction that can be taken by a vehicle to continue its path.

In order to present the most important streets, we have calculated the betweenness centrality, which indicates the number of shortest paths from all vertexes to all



(a) Street sectors over Chicago map.

(b) Graph representation

Figure 4.23: Street sectors and the graph of Chicago.

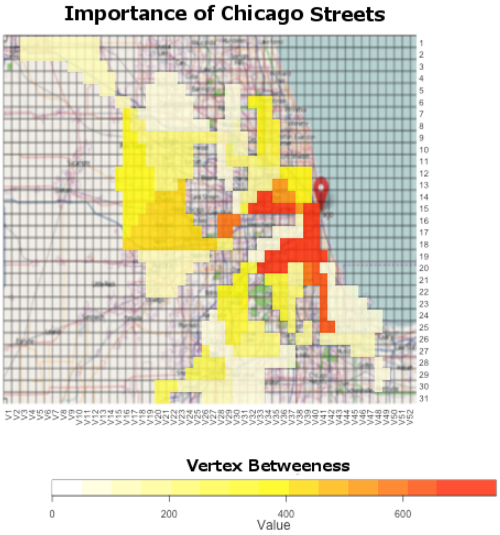


Figure 4.24: Street importance according to the vertex betweenness.

others that pass that vertex. Figure 4.24 summarizes the most important roads, which are the downtown streets (red squares in the heat map). The top is street 79 (759), followed by street 28 (731), street 78 (719), street 31 (713) and street 26 (702).

Another graph metric that has been evaluated was the edge betweenness, i.e. the number of shortest paths between pairs of nodes that run along the edge. This metric evaluates how much relevance this road is considering both infrastructure and traffic jams. Here we named this as convergence importance, which is calculated as the edge betweenness of the weighted graph, considering as weight the Bing value.

Table 4.1: Important roads and convergences according with the city infrastructure.

Road	Importance	Edge	Convergence Importance	
79	759	31 → 79	775	100.00%
28	731	78 → 28	754	97.29%
78	719	80 → 78	633	81.68%
31	713	79 → 82	628	81.03%
26	702	82 → 80	615	79.35%
82	611	26 → 42	520.5	67.16%
80	597	50 → 9	427	55.10%
42	582	25 → 26	383	49.42%
25	531	76 → 31	380	49.03%
50	454	28 → 26	372	48.00%

Table 4.1 presents the top-10 roads importance and the convergence importance (edge betweenness). One can notice that the top-5 roads involve the top-5 vertex betweenness that is vertexes 79, 28, 78, 31, and 26. A traffic jam in such roads has more impact on the network availability.

Next analysis consists in the identification of phase transitions. It is presented in different periods of the day (dawn, morning, lunch, afternoon, and night), indicated by Table 4.2 with their corresponding times. Notice that the x-axis is presented in the time frame ID. For instance, if the time frame is 5 min (0:05AM), than 1 is the first interval of 5 min, 2 is the second interval of 5 min (0:10AM), and so on.

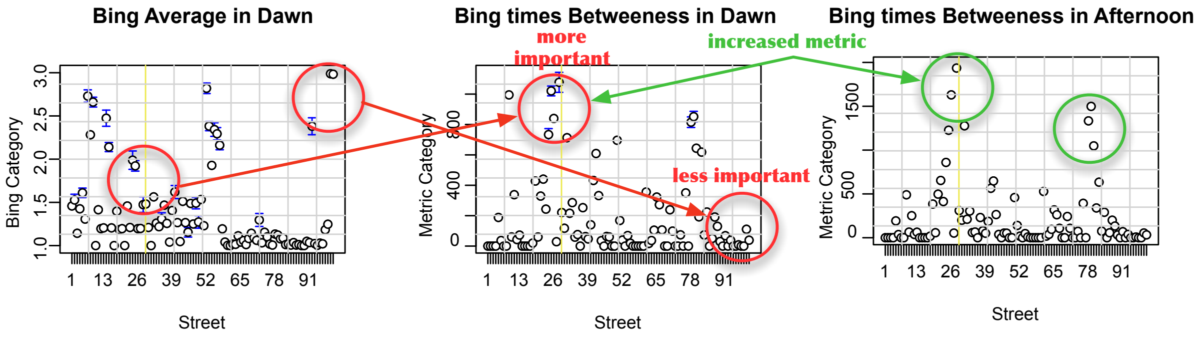


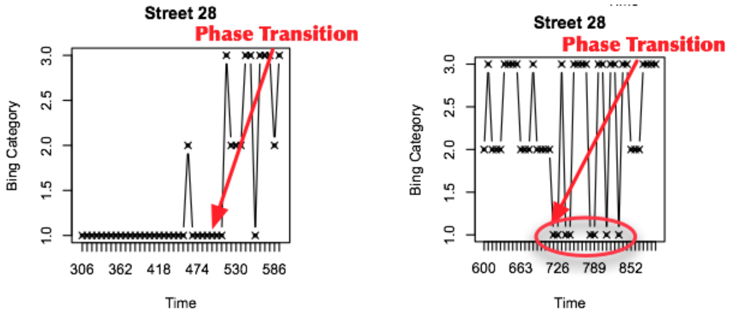
Figure 4.25: Average traffic congestion and such value updated by the edge betweenness.

Figure 4.25 presents the scatterplot of the average traffic congestion (metric 1) of weekdays in the left graphic while the two last graphics illustrate the multiplication of the average traffic congestion and the edge betweenness (metric 2) for each street at dawn and in the afternoon. Notice that with metric 1, we can not differ clearly the most important roads with a higher value, which is the advantage of metric 2. We

Table 4.2: Parameters of the period of the day.

	Description	Time
1	Dawn	0:00am–5:00am
2	Morning	5:00am–10:00am
3	Lunch	10:00am–15:00am
4	Afternoon	15:00am–20:00am
5	Night	20:00am–0:00am

can see that some streets have more trafic flow, but this comparison do not consider infrastructure characteristics, which can influence in the traffic flow impact in other roads. In other words, it is not fair to compare one avenue with 4 lanes and a small street with 1 lane. Traffic congestion in those streets will have different impacts in the propagation of the traffic congestion. Also, we can see that the period of the day influence in the traffic congestion, which will influence in which street is more important. For instance, in the afternoon, streets with higher congestion will be more important than at dawn (meaning that their traffic flow will have a higher impact in the traffic congestion of other streets during the afternoon). Therefore, distinct behavior patterns stand out depending on the period of the day. The comparison of such metrics is discussed in Section 4.2.2.3.



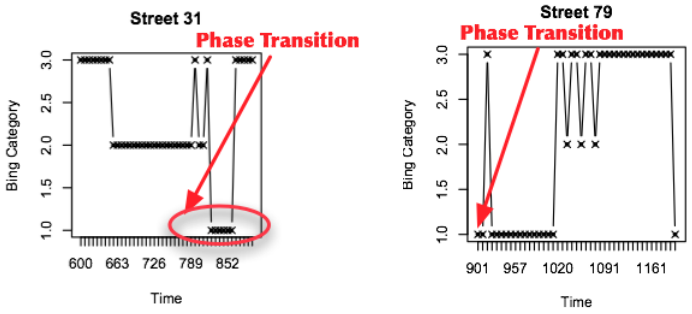
(a) Morning

(b) Lunch

Figure 4.26: Flow intensity analysis for street 28.

Besides, phase transitions are difficult to identify in a daytime analysis, but not in each period of the day. Figure 4.26 presents the flow intensity in April 19th 2013 in the morning and during lunch. In Figure 4.26(a), we have used a larger value of β to identify the phase of traffic flow, when we compare with Figure 4.26(b), in which we have used a lower value of β. One can notice that are several hours of the day that if we leave 7 minutes earlier, it will make a difference.

Figure 4.27 presents the flow intensity for streets 31 and 79, which is the most important road. About street 31, we can identify 852 minutes (14:12 hours) as a phase transition due the street flow intensity remains highly congested (level 3).



(a) Lunch (b) Afternoon

Figure 4.27: Flow intensity analysis for streets 31 and 79.

By the daytime analysis, we can identify traffic patterns during hours of the day, as well as the phase transitions. Figure 4.28 presents the flow intensity result for the top-4 important streets in April 19th, 2013. Notice that the street 79 has more traffic jam at night, as well as the street 28. Notwithstanding, the street 28 has more flow fluctuation during lunch, and in the afternoon and in the night it is highly congested for hours.

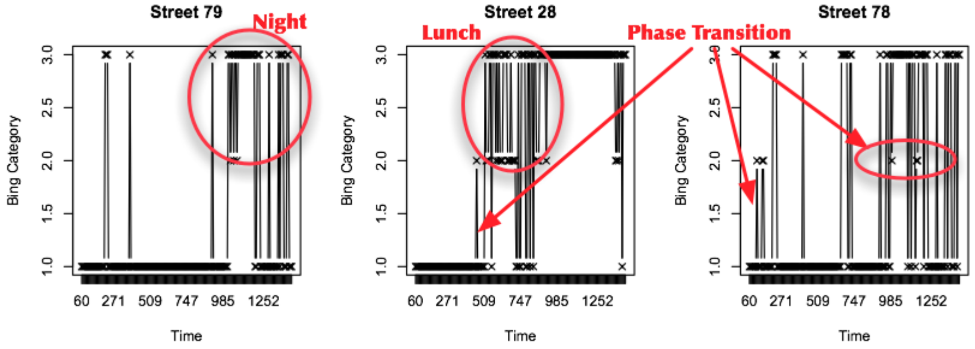


Figure 4.28: Day time analysis of the top-4 streets' flow intensity.

Finally, four clusters have been made, according to the average flow intensity so as to preserve the street main characteristics: (i) streets with values between 1 and 1.5; (ii) streets with values between 1.5 and 2; (iii) streets with values between 2 and 2.5; and (iv) streets with values higher than 2.5. Such division has been made with data of weekdays and of weekends, creating two distinct clustering of behavior.

So as to define the street categories, Figure 4.29 presents the probabilities of flow intensity for each. One can notice that category 1 has a higher probability of being green during almost all the day, except at dawn that presents a higher probability of being yellow. The same happens for category two, highlighting the red higher probability at

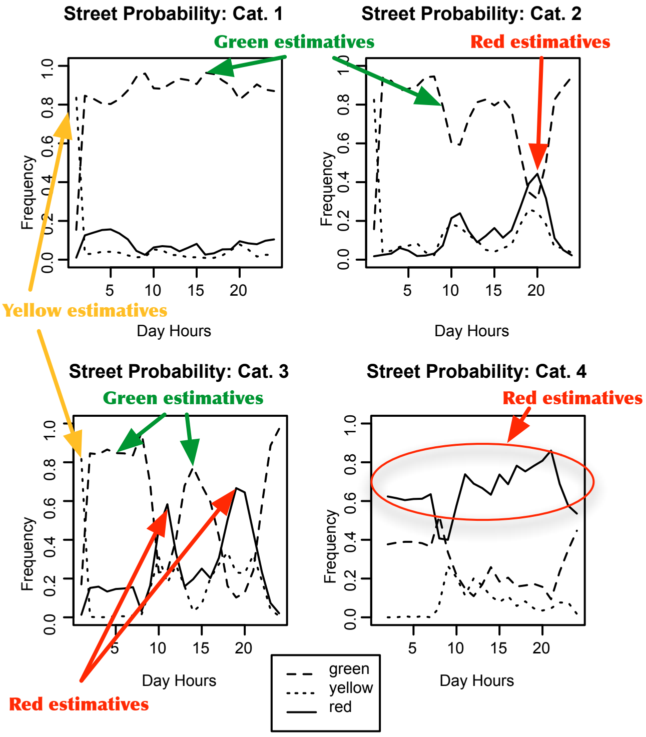


Figure 4.29: Probabilities of flow intensity per street category.

8pm. Streets of category 3 presents a variation of green and red probabilities, being more susceptible of being red at 11am and 20pm. Finally, category 4 presents a red probability during most of the day, except for 9am that has a higher probability of being green.

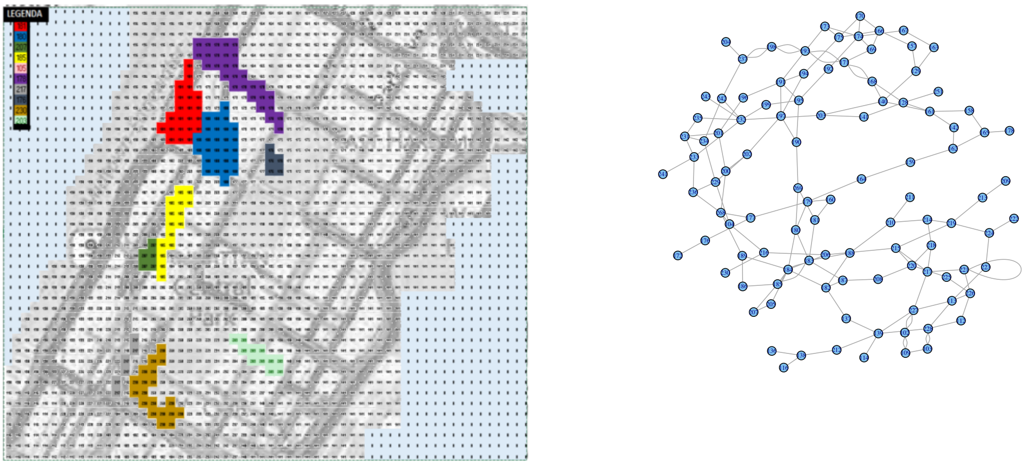
Table 4.3 shows the classification of streets 79, 78, 28, and 31. One can notice that almost every period of the day the street category is 1, being sometimes 2. Categories 3 and 4 are rare, standing out street 28 that change its category during the periods of the day. Even in the weekend, one can notice that street 28 has a distinct behavior in comparison with streets 79, 78, and 31.

Table 4.3: Classification of streets in periods of the day.

	weekdays				Weekends			
Streets	79	28	78	31	79	28	78	31
Dawn	1	1	1	1	1	1	1	1
Morning	1	2	1	1	1	1	1	1
Lunch	1	3	1	2	1	2	1	1
Afternoon	2	4	2	2	1	4	1	1
Night	1	2	1	1	1	2	1	1

4.2.2.2 Manhattan Case Study

Another case study was New York, Manhattan area. In our dataset, data was collected from June 10 of 2013 to August 22 of 2013. Data were collected every 1 min. Next, the database has the following information: (I) date; (II) hour; (III) street number; (IV) number of green pixels; (V) number of yellow pixels; (VI) number of red pixels; and (VII) number of no category pixels.



(a) Street sectors over New York map.

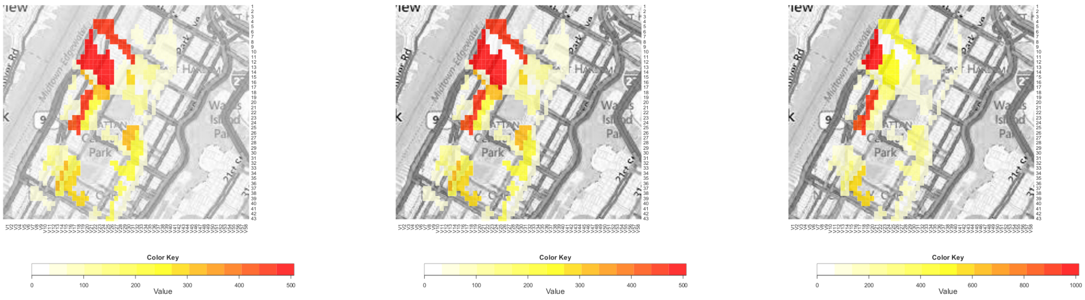
(b) Graph representation

Figure 4.30: Street sectors and the graph of New York.

The same process that was made with Chicago was conducted with Manhattan data. Figure 4.30 illustrates New York geo-code sectors and its graph (see Figure 4.30(b)). We mapped a graph in which the vertexes are roads and the edges are the direction that can be taken by a vehicle.

In order to present the most important streets, the betweenness centrality has been calculated. Figure 4.31 summarizes the most important roads, which are the downtown streets (red squares in the heat map). The top is street 181 (522), followed by street 180 (497), street 207 (445), street 185 (388) and street 105 (369). Figure 4.31b and c presents the flow importance on June, 11th at 08:00 and at 22:00, which is the multiplication of the edge betweenness by the Bing value, which is the edge weight. One can notice that important roads change according to the traffic flow besides the traffic infrastructure.

Figure 4.32 shows the traffic flow and the flow importance for each street during weekdays. One can notice that roads with higher id are more important. The flow



(a) Vertex betweenness (characteristic of the infrastructure).

(b) Example of flow importance on June, 11th at 08:00.

(c) Example of flow importance on June, 11th at 22:00.

Figure 4.31: Street importance according to the vertex betweenness.

importance discriminates the roads based on their infrastructure nature and their flow. More details are presented in Section 4.2.2.3.

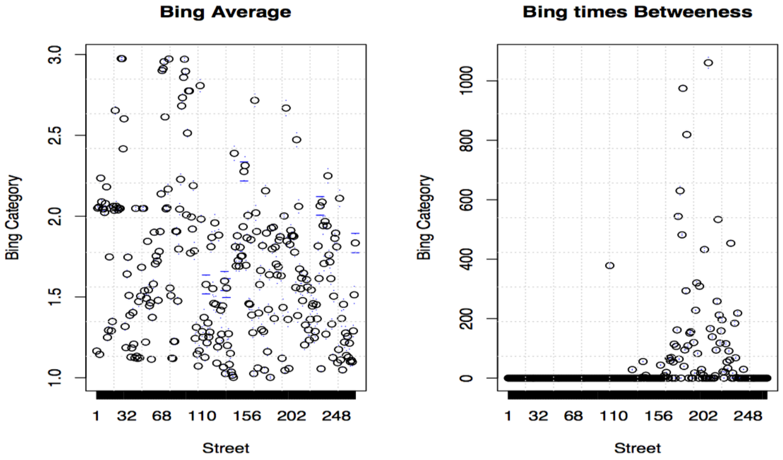


Figure 4.32: Comparison between the betweenness and the flow importance for each street during weekdays.

In summary, Table 4.4 presents the top-10 roads importance and the convergence importance (edge betweenness). One can notice that the 5 most important roads involve the top-5 vertex betweenness that is vertexes 181, 180, 185, 178, and 207. A traffic jam in such roads has more impact on the network availability.

For the streets category, Figure 4.33 presents the probability of the road be congested or not. One can notice that category 1 is more likely to be free (green), while category 3 is more likely to be yellow and category 4 tends to be congested (red). Category 2 has a probability of presenting phase transitions.

Finally, with our methodology to identify phase transitions, we can develop ap-

Table 4.4: The most important roads according to flow and infrastructure.

Road	Edge	Convergence	Importance
181	181 → 180	522	100.00%
180	180 → 178	497	95.21%
207	207 → 185	445	85.25%
185	185 → 181	388	74.36%
105	105 → 217	369	70.69%
178	178 → 269	349	66.86%
217	217 → 207	348	66.67%
176	176 → 195	319	61.11%
230	230 → 105	297	56.90%
203	203 → 230	247	47.32%

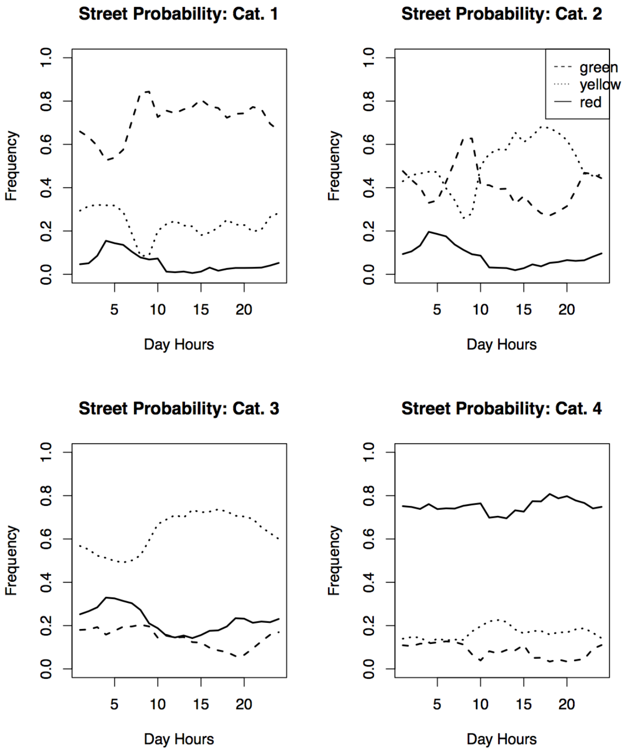
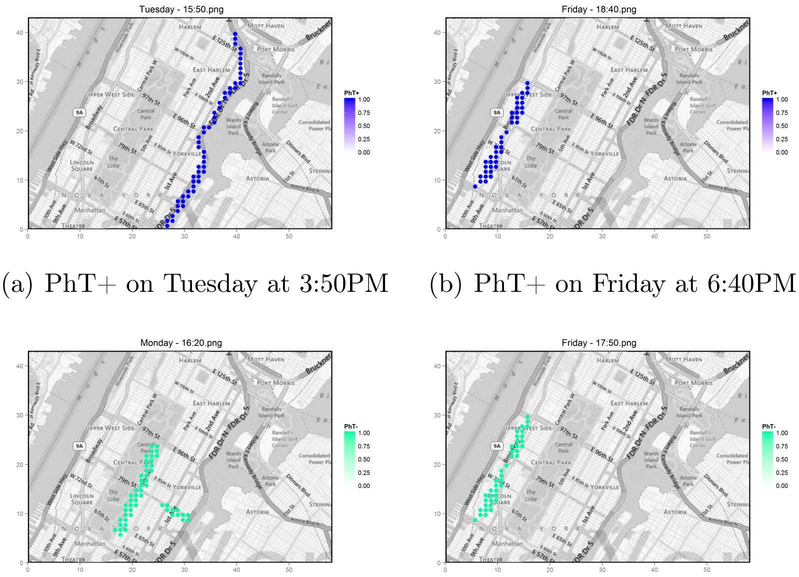


Figure 4.33: Street probability based on the streets' four clusters.

plications that shows spatially and temporally when and where these shocks in the system occur. Figure 4.34 illustrates this idea, presenting two PhT+ and two PhT- in different times and weekdays. We can see that these transitions occur on Broadway avenue and the 5th street, besides the highway.



(a) PhT+ on Tuesday at 3:50PM (b) PhT+ on Friday at 6:40PM

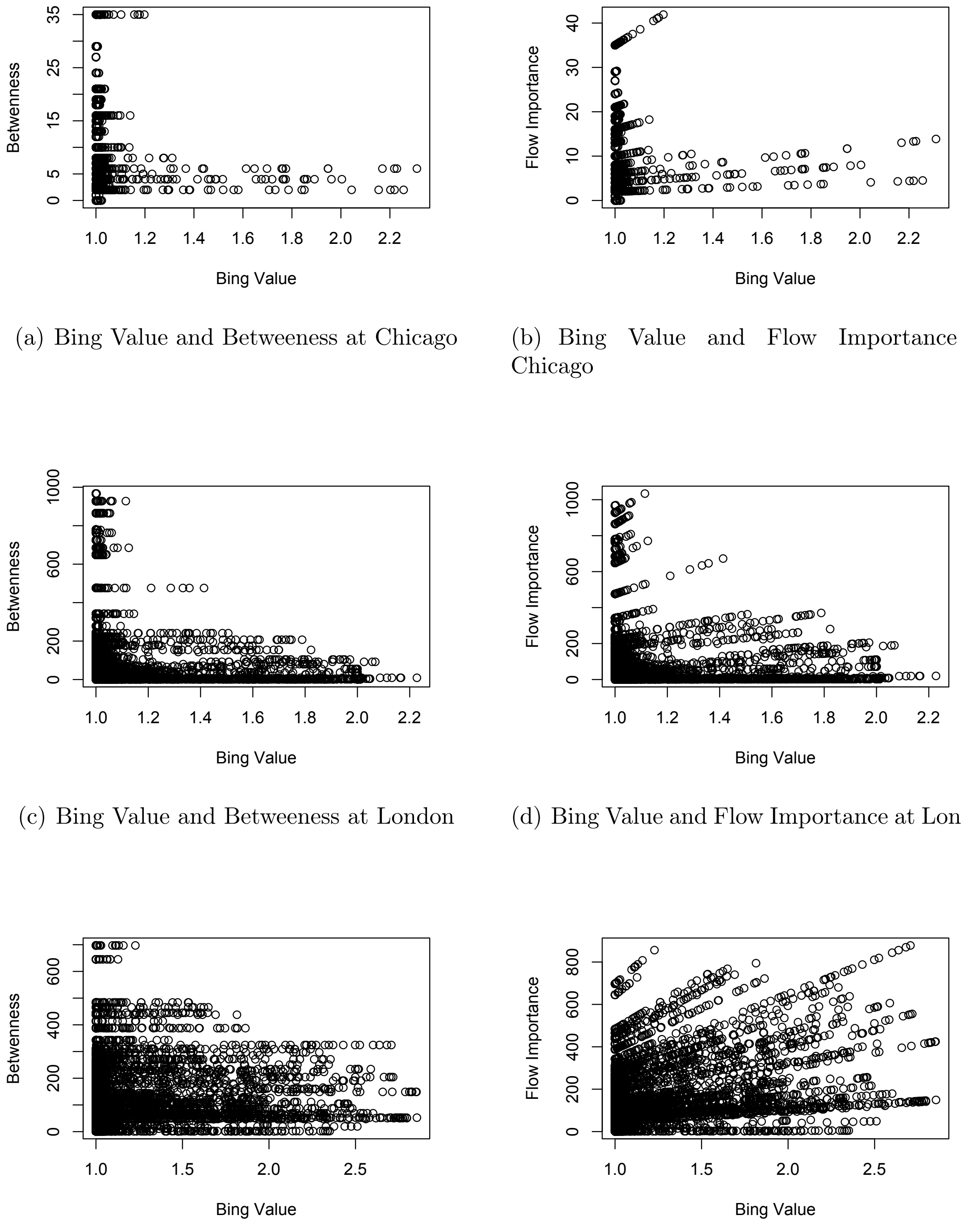
(c) PhT- on Monday at 4:20PM (d) PhT- on Friday at 5:50PM

Figure 4.34: Spatial representation of phase transitions in Manhattan.

4.2.2.3 Flow Importance in Several Cities

Here we evaluate the correlation between the average Bing Value of each street in comparison with its betweeness and its flow importance for several cities. Figure 4.35 presents the results of this comparison. We can observe in Figures 4.35(a), 4.35(c) and 4.35(e) that there are different categories of streets, some with lower betwenness and others with higher. Although, we can not differ such streets with the same range of betwenness.

In order to consider the traffic flow and the infrastructure measured by the betwenness, we have proposed the flow importance metric. The scatterplot between the average Bing Value and the flow importance is illustrated in Figure 4.35(b), 4.35(d) and 4.35(f). In these graphs, we can observe that for the same category of streets, streets with higher traffic flow have higher flow importance. In other words, we can affirm that there are streets that even with traffic jam, considering their infrastructure (its neighbors), do not impact much for a greater traffic congestion in the city. Streets with higher flow importance, by the other hand, they impact more in the traffic flow of the city than those with lower flow importance.



(a) Bing Value and Betweeness at Chicago

(b) Bing Value and Flow Importance at Chicago

(c) Bing Value and Betweeness at London

(d) Bing Value and Flow Importance at London

(e) Bing Value and Betweeness at Los Angeles

(f) Bing Value and Flow Importance at Los Angeles

Figure 4.35: Scatterplots comparing the correlation between the Bing value, betweenness and flow importance.

4.3 Data Correlation: Check-ins X Traffic Jam

In [Tostes et al., 2014], we have analyzed how do social sensors, derived from Twitter, Foursquare, and Instagram, are correlated with traffic jam. Recall that, in our work, every data shared in them is called a check-in. We want to investigate if we can use data from social sensors (check-ins) to better understand the traffic condition, that is, the traffic flow from Bing Maps. The objective is to verify if check-ins can be used as a hint of traffic conditions changes or current situation.

As people move around in cities, they share their location, pictures and videos throughout their routine through the check-in. Between one check-in and the next one users have to move in a certain way, for example by bus or by car. Thus, since check-ins are correlated with inhabitants routines Silva et al. [2014], they might implicitly have embedded information about the traffic conditions. Consider the following scenario: a person makes a check-in at home, gets his/her car and drives to work, gets stuck in a traffic jam, and arrives at work and makes another check-in. Imagine how many people on Foursquare follow the same routine. Finally, when we have more check-ins, we can affirm that there are more people moving there, and this is the relation of causality: the more people, the more traffic.

Therefore, can we affirm that the information of check-in can reflect the behavior of the traffic? Do people make more check-ins when traffic is congested? Do check-ins can be a sign of heavy traffic? In other words, do the information of check-ins are correlated with heavy traffic? In summary, can we use the check-ins as a sensor to indicate and to predict traffic jams? Answer those questions is a fundamental step in order to use check-ins as a complementary source of information to improve intelligent traffic systems.

To investigate our hypothesis that check-ins and traffic jams are correlated, we analyze two datasets from the same period of time at Manhattan, New York City: (1) check-ins from Twitter, Foursquare and Instagram; (2) traffic flow from Bing Maps. We chose New York City, Manhattan region, because it is a popular region, having a high number of people performing check-ins, as it was verified in [Silva et al., 2012]. Besides that, according to the NYMTC report [Council, 2013], it has the third lowest daily vehicle miles traveled, as it was mentioned. Besides, the U.S. cities present a good quality of data in Bing Maps traffic flow.

4.3.1 Definition

In order to evaluate the influence of check-ins in traffic jam, we have developed the following methodology:

1. As the time interval between check-ins is high, we aggregate the check-ins into a 3-hour period. Thus, for the 24 hours of a day, we have 8 periods of 3 hours. However, if we aggregate the traffic flow in the same time interval we can miss valuable information, as the transit varies widely. Traffic flow in Bing Maps varies from fast traffic (represented by green color or the integer 1), moderate traffic (represented by yellow color or the integer 2), and slow traffic (represented by red color or the integer 3). As we mentioned, these are the definition of the Bing Value (see definition 1). Knowing that, we categorized the aggregated traffic flow (provided by Bing Maps), named τ, for a particular street as it follows:
   - $\tau = 1$, when the average traffic flow of that specific street segment is less than 1 (recall that we also represent traffic conditions by an integer);
   - $\tau = 2$, when the average traffic flow of that specific street segment is between 1 and 1.5;
   - $\tau = 3$, when the average traffic flow of that specific street segment if higher than 1.5.
2. For each street segment[4], we calculate the mean value and standard deviation of check-ins. These information will be used to map the check-ins category into 1, 2, or 3 (the same as traffic flow categories).
3. For each time interval (1–8), we perform the following steps:
   - Capture the traffic flow category (1, 2 or 3);
   - Calculate the total number of check-ins in which its location is in the street segment;
   - Based on the total number of check-ins every street segment, the mean and standard deviation calculated on step 2, apply the following steps to categorize check-ins:
     - If the number of check-ins in the street segment is between the confidence interval (based on the mean and one third of the standard deviation), then the check-in category is 2;

[4] Street segment is the part of the street between consecutive intersections, considering a specific direction.

- If the number of check-ins in the street segment is higher than the sum of the mean with one third of the standard deviation, then the check-in category is 3;
- Otherwise, the category is 1.

4. To analyze if check-ins are a good signal to indicate traffic flow condition (free or congested), we aggregated the data according to a time interval of δ minutes. We separate the analysis in weekdays and weekends. For each dataset, two distributions were plotted, one with the frequency of check-ins during 24 hours, and another with the frequency of congested traffic flow (average traffic flow category for all streets). With these distributions, we want to demonstrate that they have similar shapes, i.e., even if a certain threshold shifts them, acting then as a sort of signal about traffic condition changes. This could help to improve the prediction of, for example, problems such as jams.

5. After this, it makes sense that there are streets with more check-ins over the time than others. Maybe the correlation would change temporally and spatially. In some periods of the day, some streets can be more correlated with the traffic jam than in other times. So in this step, we analyze the correlation of check-ins and traffic conditions in a lower granularity, in the street level, using the following strategy:

   - Cluster streets according to the number of check-ins and the frequency of intense traffic conditions. For this we have used K Means algorithm with four centers. Our hypothesis is that four clusters will be formed: (i) less check-ins and less traffic jam; (ii) more check-ins and less traffic jam; (iii) less check-ins and more traffic jam; and (iv) more check-ins and more traffic jam, which is the category that we are interested on. So, we count the amount of check-ins and the frequency of intense traffic jam (τ 2 and 3) per weekday and per street. We are doing this considering the seasonality of both time series.

     In order to apply K Means algorithm, some precautions are necessary. When the variables are in different units, they should be standardize to minimize the effect of scale differences. For that we use the following equation:

$$f(x) = log((x - \mu_x)/\sigma_x + 1)$$

where x is either the amount of check-ins and the frequency of intense traffic conditions in a time interval, μ_x is the average and σ_x is the standard deviation. We add $+1$ at the end, because there might be cases in which the amount of check-ins or the frequency of jam is 0 (the log(0) issue).

Besides clustering, with this analysis, we can answer questions such as: *which weekday has more check-ins? which weekday has more traffic? when and where this happens?*.

For proof of concept, in this step we should create the distribution graph as it was don't in step 4, but with the street level. This is important to analyze the possibility of using the check-ins data as another source in the traffic jam forecast models.

- After that, we are gonna choose, for each street in the higher category (more check-ins and jam), the most representative days, that is, days with more check-ins and more traffic variation that matches with their time series seasonality.
- Finally, we analyze the correlation for each street during each time interval of the chosen days. It is important to consider the seasonality, weekdays and weekends. Our focus is on weekdays since in this days the traffic jam is more intense.

Our hypothesis is that these two functions are shifted δ minutes. For that we propose the equation

$$D(\delta) = \sum_{t} \left(\left| y_t - y^*_{(i-\delta)} \right| \right) \tag{4.1}$$

in which y_t is the value of traffic flow in time t and y^*_k the value of check-ins in time k. To discover the value of δ, we calculated the discrepancy value through the sum of the difference between the y value (traffic flow) in time t and the other y^* value (check-ins) in time $t - \delta$. Then, it is possible to calculate the above function for $\delta = 1, 2, 3, \cdots$ in order to find the δ value that minimizes the discrepancy. This equation represents the error between the two input distributions, which is also a random variable.

4.3.2 Results

This section presents the results about our investigation to discover if check-ins are correlated with traffic jams. In other words, if check-ins are good signal of traffic conditions changes and matters. We made four analyses: (i) spatial analysis; (ii) temporal analysis; (iii) offset analysis; and (iv) correlation. Here we present the results of the spatial analysis for New York, Manhattan.

Spatial Analysis. We show the traffic flow and check-ins categories during the eight periods of time per day (3 hours of interval) considered in this study, using the entire dataset.

Figures 4.36(a)–4.36(c), 4.36(d)–4.36(f) and 4.36(g)–4.36(i) show the check-ins, the traffic flow and the check-ins categories in different periods of time, respectively, when we have different traffic flows. For a better visualization, videos featuring each result during all periods of time considered have been made. They can be found online[5].

Based on Figure 4.36(a)–4.36(c), we can see that the number of check-ins increases gradually throughout the day. At dawn we have less check-ins than during the morning, when we have less check-ins than at night. One can also notice that check-ins are more concentrated at downtown. Note that we can see the shape of Manhattan, showing a distribution of our data.

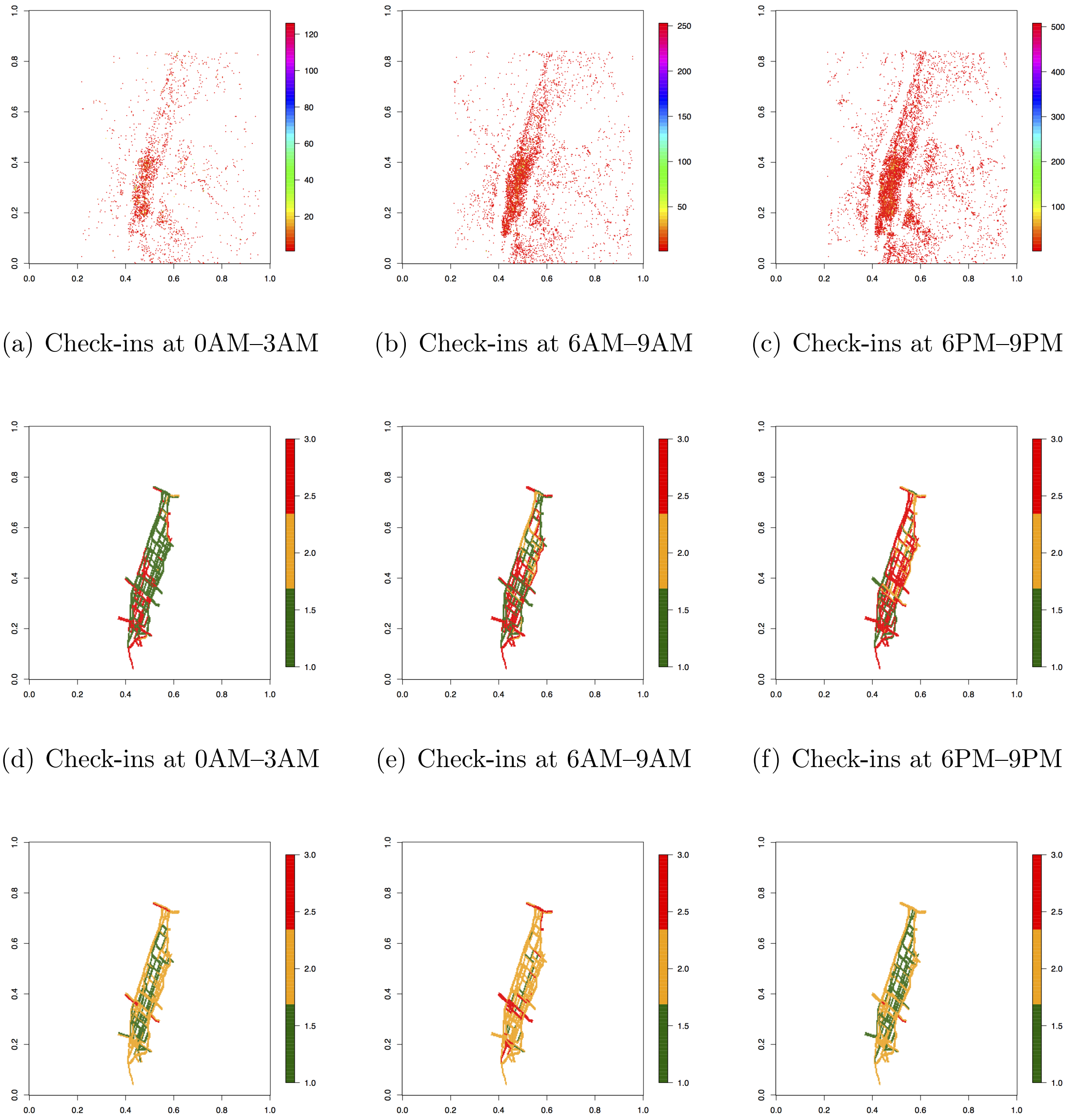
The categories of check-ins in the same periods of time presented in Figure 4.36(a)–4.36(c) can be seen in Figure 4.36(d)–4.36(f). We can see that when we have less check-ins, we have more green streets. When the number of check-ins increases, the intensities of the streets (color) change from green to yellow and then to red. Regarding to the average check-ins in the street, the illustration enable the identification of where (street) there are more check-ins, beyond the usual, and where there are not. We can see that the red lines are more concentrated in downtown (see Figure 4.36(e)), but the period of 6PM–9PM (see Figure 4.36(f)) is when people tend to make more check-ins, above the average, in the entire region (not just in downtown).

On the other hand, Figure 4.36(g)–4.36(i) presents the traffic flow from the same periods. One can notice that the yellow traffic is more common at morning (see Figure 4.36(h)) and less common in other periods. This behavior is expected because it is the time when people go to work.

When we compare all graphics, we can see that when the traffic flow is higher (red lines), check-ins are lower (green lines). In the first look, it seems like the check-ins and the traffic flow are inverse correlated. But this is not what happens. The explanation is that the traffic flow behavior will only be seem in the check-ins category after δ minutes, as it will be shown in next topics of evaluation (temporal and offset analysis).

Despite that, a better analysis can be performed, because a period of 3 hours is too large to represent the dynamics of traffic. So we need to investigate shorter

[5] `http://annatostes.azurewebsites.net/check-ins-and-traffic-flow/`



(a) Check-ins at 0AM–3AM (b) Check-ins at 6AM–9AM (c) Check-ins at 6PM–9PM

(d) Check-ins at 0AM–3AM (e) Check-ins at 6AM–9AM (f) Check-ins at 6PM–9PM

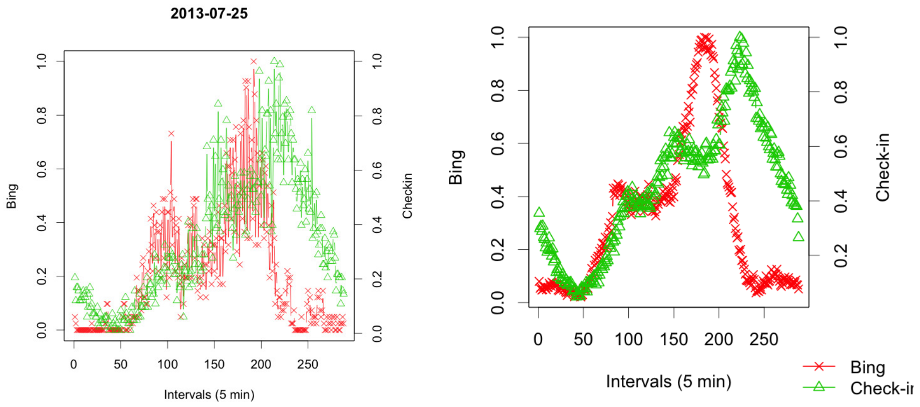
(g) Traffic flow at 0AM–3AM (h) Traffic flow at 6AM–9AM (i) Traffic flow at 6PM–9PM

Figure 4.36: Results of check-ins and traffic flow in different periods of time.

intervals of time, such as 5 min, 10 min, and so on. This analysis is presented in the next section.

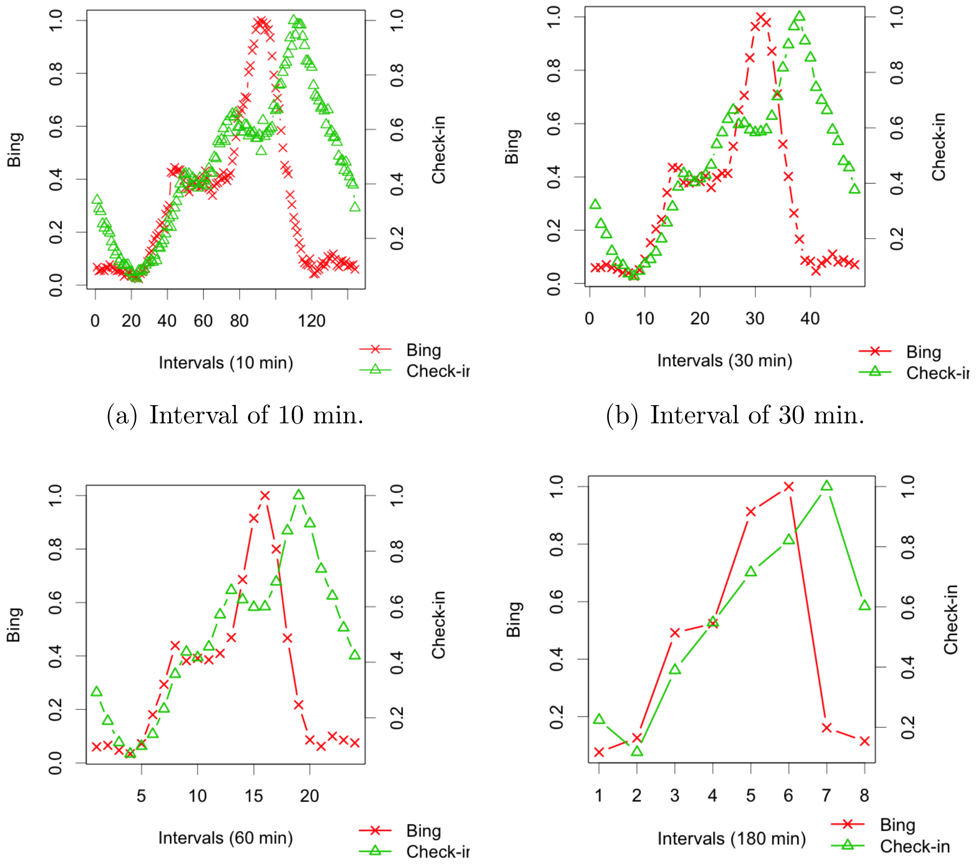
Temporal Analysis. We present a temporal analysis of check-ins and traffic flow, describing how the check-ins and the *congested* traffic flow vary during the day.

Figure 4.37 presents two graphs where we can compare the real data from an specific day (07/25/2013 – see Figure 4.37(a)), and the typical distribution for weekdays (see Figure 4.37(b)).



(a) 07/25/2013. (b) Typical weekdays.

Figure 4.37: Frequency of traffic flow in intervals of 5 minutes.



(a) Interval of 10 min. (b) Interval of 30 min.

(c) Interval of 1 hour. (d) Interval of 3 hours.

Figure 4.38: Frequency of traffic flow in different time intervals and weekdays.

Comparing the two first graphs, we can see that the traffic flows and check-ins distributions seems to be almost the same, but shifted. And the typical distribution represents the specific day. During weekends the behavior changes, presenting only one peak in the late afternoon for both check-ins and congested traffic flow. This finding is very surprising and shows that the social sensing might reflect real traffic conditions.

A natural question that emerges is the impact in the results when changing the time interval of our analysis. To evaluate the impact of choosing another time interval, we calculated the distribution for intervals of 5, 10 and 30 min, 1 hour and 3 hours, depicted in Figures 4.37(b) and 4.38. As we can see, the difference between weekdays and weekends are clearer in the traffic flow. However choosing one specific time interval does not impact in the distribution, but only in the amount of data. For this reason we chose a time interval that provides the bigger amount of data, which in our case is the time interval of 5 min.

Complementarily, in order to apply the check-ins data so as to improve the accuracy of a traffic forecasting model, we need to analyze both distributions in a lower level, the street level, according to step 5 of our methodology. Figure 4.39 shows the time series of check-ins and intense traffic flow during weekdays. Each dashed-line indicates midnight.

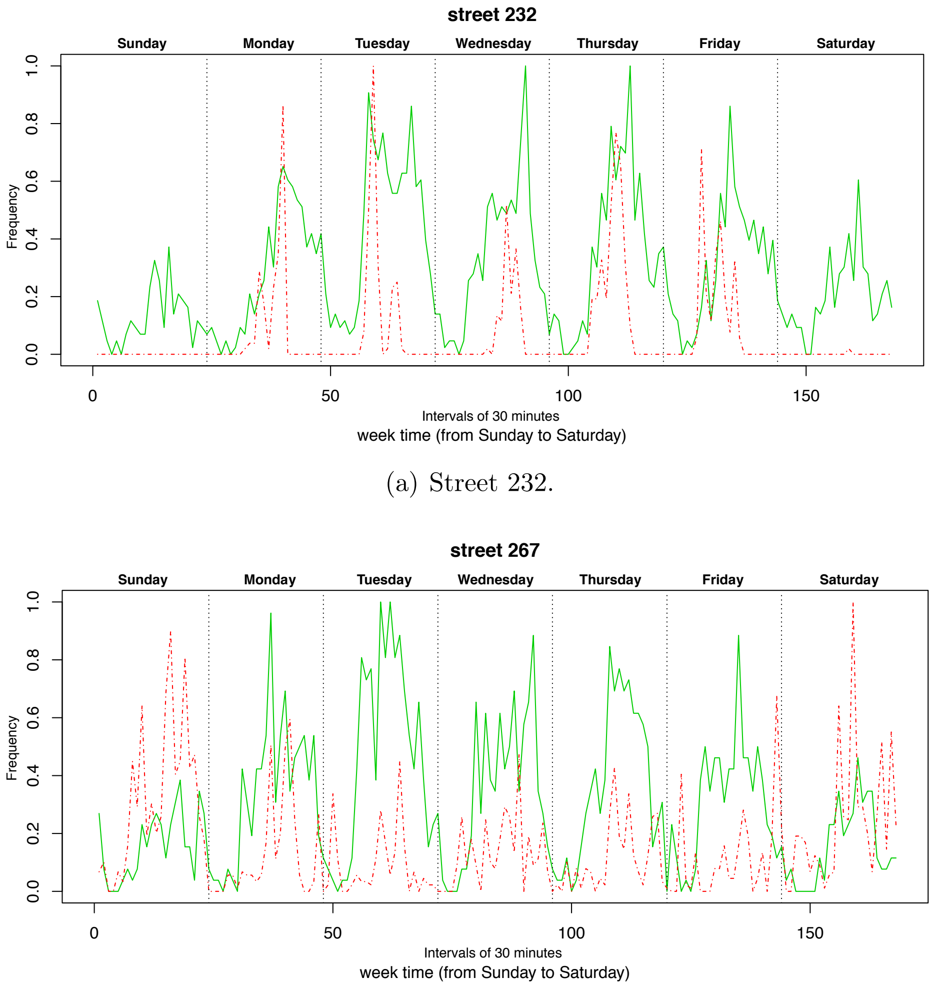
Thus, we chose the time interval of 5 min to go on. In all graphics the distributions are shifted. But how to discovery this offset of minutes? This is investigated in next section.

Offset Analysis. Our hypothesis is that these two functions are shifted δ minutes. For that we propose the Equation 4.2, in which y_t is the value of traffic flow in time t and $y{*}_k$ the value of check-ins in time k. To discover the value of δ, we calculated the discrepancy value through the sum of the difference between the y value (traffic flow) in time t and the other y* value (check-ins) in time $t - \delta$, as shows Equation 4.2:

$$D(\delta) = \sum_t \left(\left| y_t - y{*}_{(i-\delta)} \right| \right) \tag{4.2}$$

It is possible to calculate the above function for $\delta = 1, 2, 3, \cdots$ in order to find the δ value that minimizes the discrepancy. That equation represents the error between the two input distributions, which is also a random variable.

In order to formally discover the value of the shift δ between the two distributions (check-ins and traffic flow), we calculated the distribution given by the Equation 6.1. It represents the discrepancy distribution, illustrated in Figure 4.40. The red line indicates the value of δ (time) that minimizes the discrepancy error. As we can see, for Manhattan, the δ should be equal to 36 minutes.



(a) Street 232.

(b) Street 267.

Figure 4.39: Time series of check-ins and traffic jam on streets

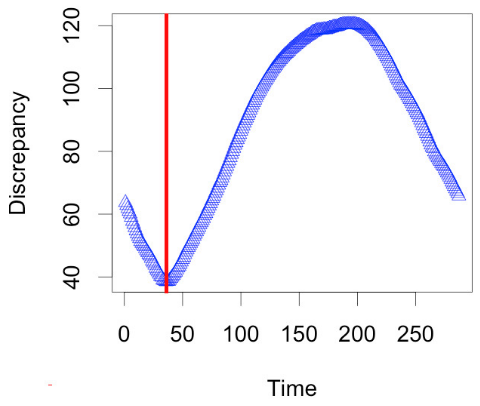


Figure 4.40: Discrepancy distribution.

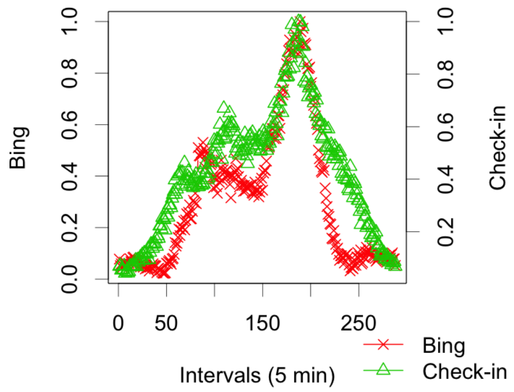


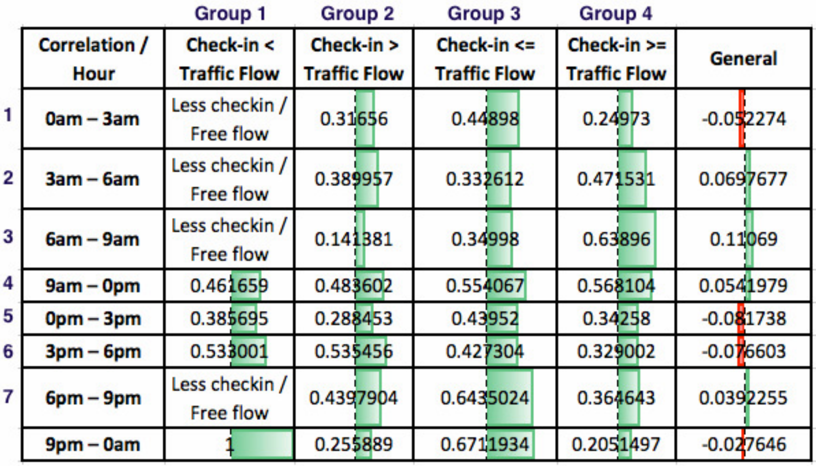
Figure 4.41: Finding the δ that minimizes the discrepancy.

Then, we shifted the check-ins distribution in 36 intervals and Figure 4.41 presents both distributions but shifted. With this result, we can see that the check-ins distribution is equal do the congested traffic flow distribution. This indicates that check-ins can be used to forecast intense traffic flow.

Correlation. Here the main idea is to understand if check-ins and traffic jam variables are correlated. Steps 1–3 from our methodology have been applied with intervals of 3 hours due to the high inter-sharing times of check-ins. A positive correlation, between check-ins and traffic flow, means that the more check-ins the worse traffic condition.

Table 4.5 presents the results of correlation for the five groups through different periods of the day. Each column represents a group (1–4) from left to right. As explained before, Group 5 was omitted because the correlation is always one, when we have data. We can see that the highest correlation is in Group 3 ("Check-in <= Traffic Flow" → {0.44, 0.33, 0.34, 0.55, 0.43, 0.42, 0.64, 0.67}), where the category of check-ins is lower than or equal to the traffic flow. In times when the traffic flow is more intense, we can see a higher correlation (after 6pm – 0.64 and 0.67).

Table 4.5: Correlation between check-ins and traffic flow.



		Group 1	Group 2	Group 3	Group 4	
	Correlation / Hour	Check-in < Traffic Flow	Check-in > Traffic Flow	Check-in <= Traffic Flow	Check-in >= Traffic Flow	General
1	0am – 3am	Less checkin / Free flow	0.31656	0.44898	0.24973	-0.052274
2	3am – 6am	Less checkin / Free flow	0.389957	0.332612	0.471531	0.0697677
3	6am – 9am	Less checkin / Free flow	0.141381	0.34998	0.63896	0.11069
4	9am – 0pm	0.461659	0.483602	0.554067	0.568104	0.0541979
5	0pm – 3pm	0.385695	0.288453	0.43952	0.34258	-0.081738
6	3pm – 6pm	0.533001	0.535456	0.427304	0.329002	-0.076603
7	6pm – 9pm	Less checkin / Free flow	0.4397904	0.6435024	0.364643	0.0392255
	9pm – 0am	1	0.255889	0.6711934	0.2051497	-0.027646

Another interesting result is the comparison between the Groups 3 and 4 during 3am–6am and 6pm–9pm. We can see that the correlations are inverses. It suggests that in the morning, people makes more check-ins before going to work when the traffic is free, and then they move in the city and the traffic flow gets more intense (potentially congested). This can be seen by the greater correlation of Group 4 (0.47) compared to the correlation of Group 3 (0.33). At night, the opposite occurs, people tend to make more check-ins after getting stuck in traffic jams. This is an expected behavior because people do note have many incentives to perform check-ins while driving in intense traffic conditions. As we can see, the correlation of Group 3 (0.64) is higher than the correlation of Group 4 (0.36), enforcing our conjecture.

As we can see, those correlations are still low, suggesting that some regions might be more correlated than others. This indicates the usefulness of check-ins as a hint to predict traffic conditions.

4.4 Our Analytical Model ALLuPIs

As it was mentioned, there are several routing protocols proposed and evaluated for VANETs [Li and Wang, 2007] related to whether their performance can satisfy throughput and delay requirements of safety and emergency applications. Both traditional and geographical ad hoc routing protocols cannot be used due to either route instability that causes packet loss and high overhead of the forward process. To solve such matter, most of the routing protocols use road information to choose the crossroad to forward packets, but they do not use information of the crossroad behavior. In this context, we believe that crossroads can be used to improve the performance of these protocols.

The assessment of protocols depends on the chosen simulation scenario model. Simulation models of real-world scenarios are proposed to assess several protocols in VANETs. There are several mobility models, which differ from their realistic mobility description at both macroscopic and microscopic levels. There are models where vehicles move at a random direction and high speed inside a limited area of simulation scenario [Li and Wang, 2007]. Other models include the drivers' decision of direction [Chen et al., 2001b] and city maps to produce real vehicular mobility patterns. There are also patterns that resemble a real behavior such as of Zurich [Naumov et al., 2006] and the United States [Saha and Johnson, 2004]. Although evaluating its efficacy, such simulations are only capable to review a limited subset of all possible behaviors, leading to unpredictable behavior due to its incompleteness analysis. This work's context is to provide a complementary tool for the evaluation of the protocols for VANETs.

4.4.1 Definition

This section discusses the impact of traffic contention at crossroads from a real mobility model. The main goals are the following: (i) to estimate the crossroad traffic throughput, i.e., the number of vehicles that cross an crossroad at an specific time; (ii) the impact of a higher vehicles load in the average response time; and (iii) the time that a crossroad has vehicles traffic. To achieve such objectives, this work proposes an analytical model for analyzing the performance of traffic at crossroads (intersection). It is called ALLuPIs – Analytical modeL to evaLUate the Perfomance of IntersectionS. Based on queuing theory, the Mean Value Analysis (MVA) algorithm [Menasce et al., 2004] has been applied to estimate the desired metrics for a map of intersections, considering real data of vehicular mobility. The analytical model represents the behavior of the system performance. Its main advantage is being faster than simulations [Menasce et al., 2004]. Ideally the model needs to be simple and precise to capture the main aspects of system characteristics. Therefore, the simulation must achieve the stationary state and it needs to be executed various times.

Several questions were answered: (i) *in average, what is the optimal point of operation at an intersection (throughput) with respect to the number of vehicles that cross that intersection?* (ii) *what is the impact of a higher number of vehicles crossing an intersection in relation to the average response time?* (iii) *how does the intersection utilization changes with the increase in the number of vehicles?* By using the proposed analytical model, we evaluate a traffic behavior, estimating the impact of a higher load in the throughput, average response time and utilization in a vehicular network. Results show that the increase of concurrency at an intersection implies in an increase of time spent by a vehicle to cross an intersection. Thus, we can predict a traffic jam and specify the level of contention (low, medium, high).

Figure 4.42 presents the methodology followed to design our analytical model. The followings steps were followed: (i) simulation; (ii) model definition; (iii) validation. In this section, we describe steps 1 and 2. The next section presents step 3.

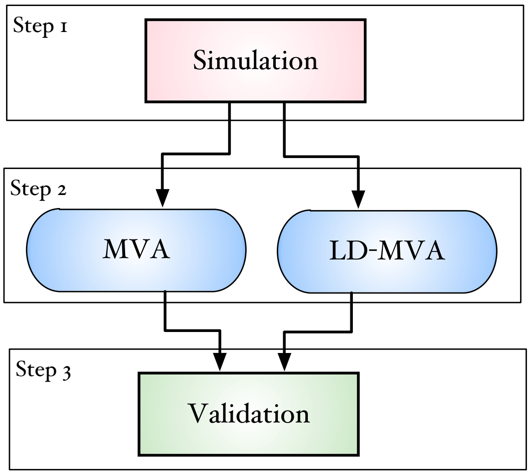
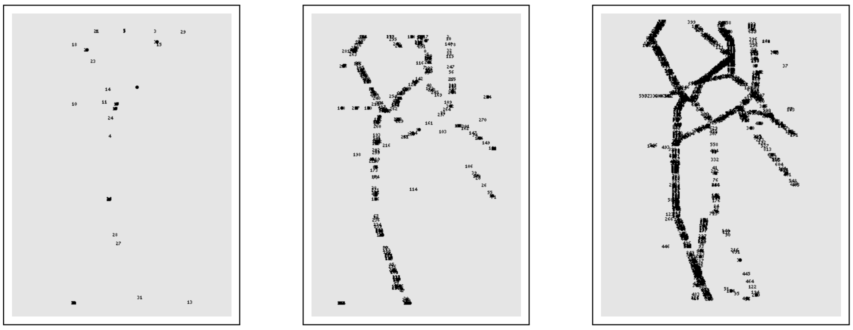


Figure 4.42: Methodology to design the analytical model.

1. Simulation description: the first step to design the analytical model was to reproduce the simulation of [Naumov et al., 2006]. This simulation has two objectives: (i) to produce inputs to the analytical model; and (ii) to validate it. Due the limitations of event-oriented network simulators to process over 10.000 nodes, Naumov et al. [Naumov et al., 2006] divided Zurich in sub-regions (lanes). With 30-40 vehicles/km, we have 30-40 vehicles passing through every 1 km of road. Each sub-region has three levels of motor activity: (i) low: less than 15 vehicles/km, (ii) average: 30-40 vehicles/km, and (iii) high (rush hour): more than 50 vehicles/km. Each simulation time (300 s) corresponds to 30 min.

The following sub-regions have been modeled: *enge oberstrass*, *unterstrass*, *zentrum bellevue* (Zurich's center) and highways (*aarau oftingen*, *hurgen jona*, *effi winti*). Each simulation generates as result a trace with the mobility of users and a Network ANimator (NAM) file. The NAM file has been used in order to view NS-2 simulations. Figure 4.43 illustrates the simulation results of the three density levels of vehicles in the *enge oberstrass* subregion in NAM.



(a) Low. (b) Medium. (c) High.

Figure 4.43: NAM output: vehicular simulated traffic in different traffic density levels.

The focus of this work is in the *enge oberstrass* region for the high density traffic. We conducted simulations with 520 vehicles during 300 s in a scenario of 5 km × 7.5 km. These are the same values used in [Naumov et al., 2006]. The communication channel is wireless and the radio propagation model is FreeSpace, with omnidirectional antennas and droptail priority queue (FIFO). The Media Access Control (MAC) protocol is IEEE 802.11p (WAVE).

NS-2 simulation generates a trace file summarizing the vehicles' mobility. Each line represents an event and each column represents the event. With this trace, we calculate each service center demand, the response time of an intersection, the number of vehicles in the queue, the number of intersection visits by one vehicle, and the number of vehicles in traffic during a simulation period.

2. Model definition: the step 2 of our methodology involves the proposal of the analytical model to meet the requirements for performance analysis. The inputs are the demand of each road intersection, which indicates the total average time that a vehicle spends to cross it. The system workload consists in the number of vehicles at time. So as to validate the analytical model, real workload has been used based on [Naumov et al., 2006]. Simulations were conducted in NS-2. When a vehicle enters the simulated scenario, it immediately begins to travel through the roads. The vehicle exits the system when it leaves the simulated area or its trip is finished. Since the number of vehicles in the system is unlimited, we have an open workload.

Queue
Server
Input
Output
Queue
Server
Input
Output

(a) Load independent device.

(b) Load dependent device.

Figure 4.44: The simplest queue system.

An intersection set represents the queue network. Each intersection is a service center. The simplest queue systems [Menasce et al., 2004] have one service center (a queue and a server), as it is illustrated by Figure 4.44. A service center can be load independent (LI) or dependent (LD), according to the following characteristics: (i) time between arrivals of vehicles crossing, and (ii) average demand service, which indicates the minimum time (not considering the queue) that the vehicle takes to cross the intersection. LD is a service center whose service time depends on the number of vehicles waiting to cross an intersection. The more vehicles in the queue, the longer the service time. In this work, vehicles arrive in the queue waiting to cross an intersection (server). The response time is the total time a vehicle spends to cross an intersection. It depends on the number of vehicles waiting in the queue and the absence of traffic jam, which is measured by the number of vehicles in the system. This dependency implies in the modeling of a LD center [Menasce et al., 2004].

The model abstracts several output possibilities in an intersection because of the tunneling of the road characteristics. The load of vehicles at an intersection implies in the time spent by a vehicle to cross it. If there are many cars at an intersection, the vehicle takes longer to continue its route. Table 4.6 presents the variables notation of the model.

Figure 4.45 presents the simulated scenario with 21 centers (black dots). Besides

Table 4.6: Notation of analytical model variables.

#	Variable Description
N	Number of competing vehicles.
N^*	Optimal number of competing vehicles (saturation threshold of the region).
λ	Arrival rate of vehicles.
D_k	Demand on center k.
$\mu_k(j)$	Throughput of j vehicles in center k.
μ	Throughput in region.
$\alpha_k(j)$	Slowdown with j vehicles in center k.
R_k	Response time in center k.
R	Region response time.
Q_k	Queue size of center k.
Q	Region queue size.
U_k	Utilization of center k.
$p_k(j\|n)$	Proportion of time the center k have j vehicles when the population size is n.

each center $C_1 \ldots C_{21}$, we put it respective coordinate (x, y) in the simulated area. Each highlighted point represents a vehicle in the system (speed ranges from 10 to 35 km/h). The arrival rate of vehicles in the system (λ) is calculated by the following equation:

$$\lambda = \frac{\gamma}{\zeta} = \frac{520}{300} \approx 1.7\,\text{vehicles/s} \tag{4.3}$$

in which γ is the number of arrival vehicles and ζ is the simulation time. For Zurich, the result is 1.7 vehicles/s.

From the simulation we calculate each center demand. We measure the number of vehicles crossing an intersection (center) to estimate the occupancy time of each vehicle served by this center, in average. We obtain the system throughput when the intersection has $N = 1, 2, 3 \cdots \gamma$ vehicles (the respective symbols of $\mu_k(1), \mu_k(2), \mu_k(3), \ldots, \mu_k(\gamma)$), in which γ represents the number of simultaneous vehicles in the system. The demand of a center k is calculated by dividing 1 by $\mu_k(1)$, as follows:

$$D_k = \frac{1}{\mu_k(1)}.$$

This is important since those variables are inputs to the system performance evaluation.

Algorithm 5 MVA Algorithm for LI Center

Require: m as the number of service centers and the demand of each center (D_k)
Ensure: Q, Q_k, R_k, U_k, X

```
procedure MVA-LI ($m$)
    $Q_k = 0; n = 1;$
    repeat
        for k in 1:$m$ do
            $R_k \leftarrow D_k(1 + Q_k)$
        end for
        R $\leftarrow \sum R_k$
        $X \leftarrow \frac{n}{R}$
        for k in 1:$m$ do
            $Q_k \leftarrow XR_k; \quad U_k \leftarrow XD_k$
        end for
        Q $\leftarrow \sum Q_k$
        $n \leftarrow n + 1$
    until $n \neq \gamma$
end procedure
```

Algorithm 6 MVA Algorithm for LD Center

Require: m as the number of service centers and the demand of each center (D_k)
Ensure: Q_k, R_k, U_k, X

```
procedure MVA-LD ($m$)
    $Q_k = 0; n = 1; p_k(0|0) = 1$
    repeat
        for k 1:$m$ do
            $R_k \leftarrow D_k \sum_{j=1}^{n} \frac{j}{\alpha(j)} p_k(j-1|n-1)$
        end for
        R $\leftarrow \sum R_k$
        $X \leftarrow \frac{n}{R}$
        for k in 1:$m$ do
            for j in 1:$n$ do
                $p_k(j|n) \leftarrow \frac{D_k X}{\alpha_k(j)} p_k(j-1|n-1)$
            end for
            $p_k(0|n) \leftarrow 1 - \sum_{j=1}^{n} p_k(j|n)$
            $U_k \leftarrow 1 - p_k(0|n)$
            $Q_k \leftarrow \sum_{j=1}^{n} j p_k(j|n)$
        end for
        $n \leftarrow n + 1$
    until $n \neq \gamma$
end procedure
```

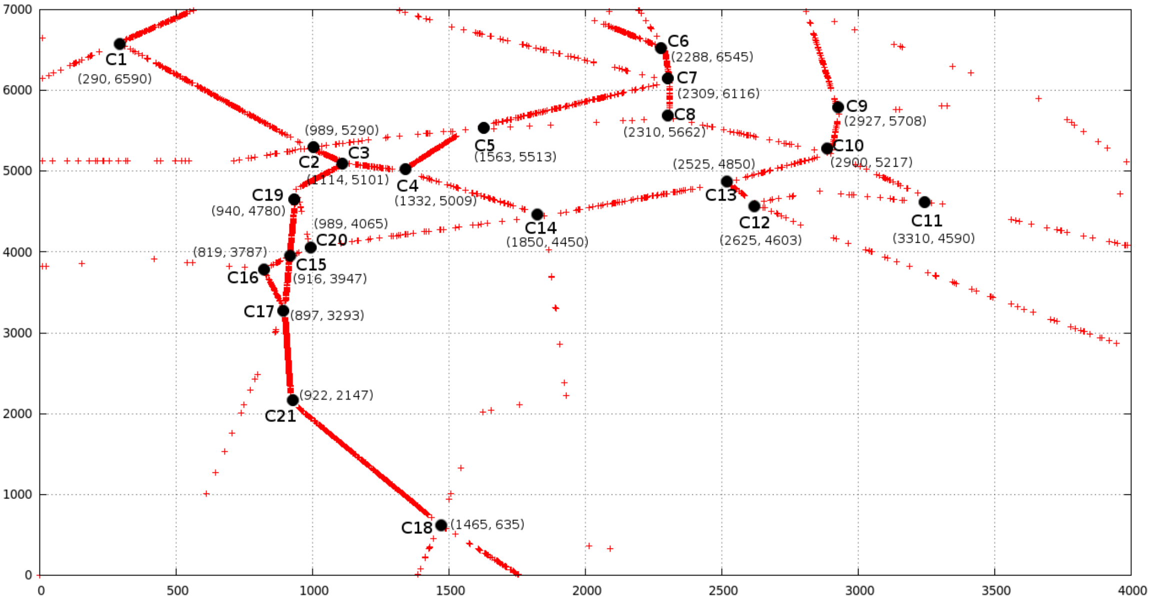


Figure 4.45: Simulation scenario with load dependent centers.

Next, we obtain two results. The first is the performance evaluation of a region. In the second, we identify the intersections responsible for the contention. To evaluate (1) we have used a Mean Value Analysis (MVA) model [Menasce et al., 2004], presented by Algorithm 5, that represents the contention at an intersection as the waiting time. Hence, we use an LI center, analyzing each region performance in terms of response time, throughput and slowdown. The goal is to generate the throughput when the center has $N = 1, 2, 3, \ldots, \gamma$ vehicles. In the second result, we consider LD centers and use the MVA algorithm for LD centers [Menasce et al., 2004]. It is important to represent the traffic jam since this work aims to analyze which intersection leads to contention.

Algorithm 6 presents the MVA Algorithm for LD center. The algorithm requires the slowdown and the queue size distribution, i.e., the fraction $p(j|n)$ of time in which each possible size exists in the center. Slowdown is the throughput in each center divided by the system throughput, as follows:

$$\alpha(j) = \frac{\mu(j)}{\mu(1)}.$$

We calculate the system response time (R), the center response time (R_k), system throughput (X), the utilization of each intersection (U_k), and the size of system queue (Q) that includes the vehicle in service.

4.4.2 Validation

Before we have validated our proposed model, we have obtained the values of demand, response time and utilization for each center (crossroad) based on the simulation described in step 1.

Table 4.7 summarizes the simulation results. As we can see, different centers obtained different demands. For example, centers C_{11} (8.30 s) and C_{17} (0.11 s). C_{11} presents lower utilization than C_{17}, but C_{11} has a higher response time. This justifies the demand variations. The service centers with higher utilization are 9 and 13, with 99.67% of utilization. All services are saturated, except center 11 with 85% of utilization. Using Little's Law, we calculate the utilization in the MVA algorithm (demand times throughput). C_7 is the center with the highest average response time (27.39 s). We measure the response time by dividing W_k, which is the cumulative vehicle time in the center, by C_k, which is the number of vehicles crossing an intersection k.

Table 4.7: Simulation Results

Center	Demand	Response Time (R_k)	Utilization (U_k)
C_1	10.16 s	24.54 s	92.67%
C_2	5.52 s	18.07 s	98.34%
C_3	7.08 s	11.39 s	99.00%
C_4	8.63 s	15.01 s	99.01%
C_5	8.83 s	23.86 s	97.01%
C_6	9.38 s	17.41 s	95.00%
C_7	11.76 s	27.39 s	95.34%
C_8	9.39 s	14.57 s	98.67%
C_9	16.90 s	24.12 s	99.67%
C_{10}	11.82 s	25.91 s	92.67%
C_{11}	8.30 s	20.92 s	85.00%
C_{12}	9.65 s	13.34 s	97.67%
C_{13}	12.71 s	19.43 s	99.67%
C_{14}	11.95 s	25.30 s	99.67%
C_{15}	16.03 s	20.75 s	96.00%
C_{16}	10.39 s	16.25 s	97.01%
C_{17}	0.11 s	14.61 s	98.34%
C_{18}	12.76 s	20.61 s	94.67%
C_{19}	7.52 s	15.38 s	95.67%
C_{20}	17.26 s	23.04 s	98.67%
C_{21}	10.05 s	16.93 s	99.34%

Response Time (R)	123.49]s
Throughput (X)	1.7 vehicles/s
Queue size (Q)	214 vehicles

To estimate each center response time, we need each center throughput. This section confirms that each intersection can be modeled as an LD center. We run the LD-MVA algorithm with the same slowdown values ($\alpha_k(j)$) for each center from the inputs ($\mu_k(i)$).

We validate the results using the MVA and LD-MVA algorithms, to verify if the LD center can model road intersections. We made two tests: (i) the evaluation, through the MVA algorithm, of the intersection performances (response time, throughput and slowdown); and (ii) the analysis of which intersections are responsible for contention, using the LD-MVA algorithm. Based on these results and the simulation data, described previously, we can calculate the error of the model. If the error is lower than 30%, than the model is validated and the intersection can be modeled as a LI or LD center.

Table 4.8a presents the validation of each center utilization by the MVA algorithm. Errors up to 30% are reasonable for MVA [Menasce et al., 2004]. In this work, the errors for response time, throughput and queue system size are less than 5%, and all others are less than 10%, validating the proposed model. The simulation results also validate this table.

The second part of this work (LD-MVA algorithm) validates each center response time. Table 4.8b presents these results. The error rate of response time obtained by the model in comparison to the simulation was less than 30%, which indicates a good approximation. This validation indicates that an intersection can be modeled as a LD center.

To analyze the impact of the load increase in the center response time and utilization, we calculate the system saturation threshold, i.e., the area beneath the function after the optimum system level. In this work, the threshold is 12 (vehicles). The optimum system level depends on the system demand and the maximum center demand. Based on the simulation results, Center 20 has the maximum demand (17.26 s). The sum of center demands is 216.18 s. We calculate the optimum system level through Eq. 4.4. The response time is 34.83 s.

$$N^{*} = \frac{D}{D_{max}} = \frac{216.18}{17.26} \approx 12 \; simultaneous \; vehicles \tag{4.4}$$

Figure 4.46(a) illustrates the impact of load increase in the response time for Center 10. The number of vehicles crossing the intersection 10 was increased over time. The higher the number of vehicles in the center, the greater the center response time in an exponential scale.

On the other hand, Figure 4.46(b) presents the impact of increasing the number of

Table 4.8: Analytical model results and validation

a. Validation of system performance results (test 1).

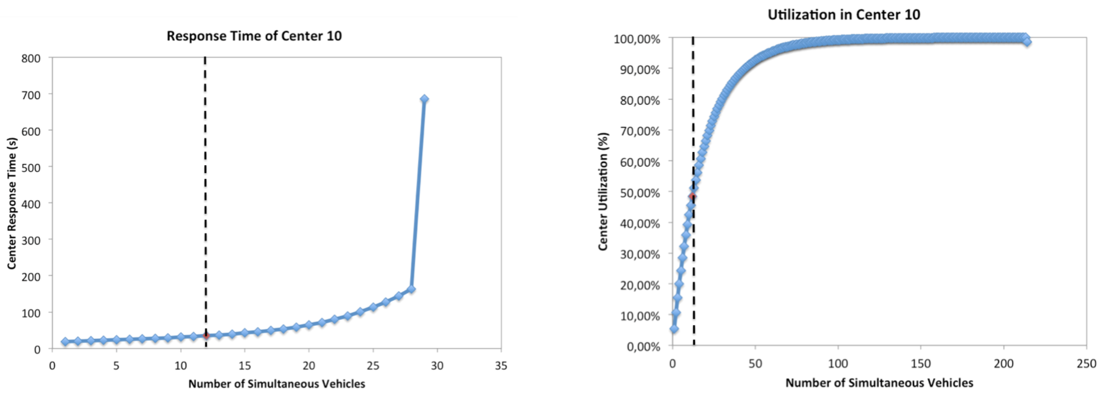
Variable	Simulation	MVA	Error
U_1	92.67%	99.99%	7.90
U_2	98.34%	99.98%	1.67
U_3	99.01%	100.00%	1.00
U_4	99.01%	99.99%	0.99
U_5	97.01%	99.98%	3.07
U_6	95.01%	99.93%	5.18
U_7	95.34%	100.00%	4.89
U_8	98.67%	98.08%	0.60
U_9	99.67%	94.04%	5.65
U_{10}	92.67%	98.48%	6.27
U_{11}	85.01%	87.05%	2.40
U_{12}	98.76%	99.98%	1.23
U_{13}	99.67%	99.44%	0.23
U_{14}	96.00%	99.38%	3.52
U_{15}	93.73%	97.93%	4.48
U_{16}	98.34%	99.80%	1.49
U_{17}	94.67%	99.97%	5.59
U_{18}	95.67%	100.00%	4.52
U_{19}	98.67%	100.00%	1.34
U_{20}	97.01%	99.47%	2.54
U_{21}	99.34%	99.99%	0.65
R	125.92	122.81	2.47%
X	1.70	1.74	2.53%
Q	214.00	214.00	0.00%

b. Validation of center response time (test 2).

Response time	R_k		Error
	Simulation	MVA	
R_1	24.54	20.99	14.46
R_2	18.07	13.04	27.84
R_3	11.39	12.54	10.08
R_4	15.01	12.55	16.35
R_5	23.86	17.49	26.72
R_6	17.41	18.00	3.37
R_7	27.39	19.35	29.34
R_8	14.57	16.79	15.24
R_9	24.12	28.51	18.20
R_{10}	25.91	29.23	12.81
R_{11}	20.92	25.86	23.62
R_{12}	13.34	15.02	12.66
R_{13}	19.43	21.09	8.54
R_{14}	25.30	26.70	5.50
R_{15}	20.75	26.27	26.61
R_{16}	16.25	18.78	15.56
R_{17}	14.61	14.65	0.29
R_{18}	20.61	23.57	14.36
R_{19}	15.38	11.44	25.66
R_{20}	23.04	26.87	16.64
R_{21}	16.93	19.22	13.50

vehicles running simultaneously through Center 10. It has a significant grow, achieving the system saturation with 12 vehicles. The dotted line indicates the optimum center level. The tangent in a function point from the optimum point has a lower inclination angle, which indicates a slowdown and, thus, validating the ideal number of concurrent vehicles in an intersection. The multiprogramming level of the service center is 12 vehicles, i.e., the intersection must have 12 vehicles (operating and standby) to have the best cost/benefit in response time versus center utilization. When one vehicle is in service, the others are on hold. Thereby, one vehicle has a significant impact on increasing the center response time and utilization, since the center is already saturated.

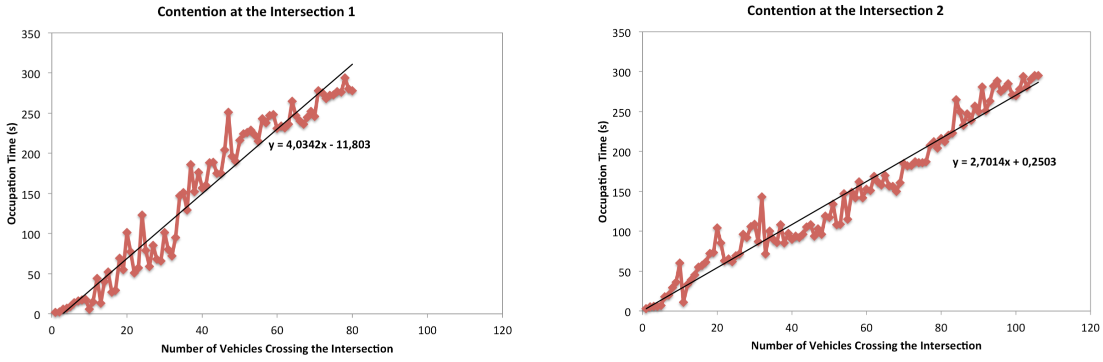
We evaluated an intersection contention evolution through the increased load. This analysis considers the center response time. Figure 4.47 presents the contention



(a) Response time.

(b) Utilization.

Figure 4.46: Performance results of center 10.



(a) Center 1.

(b) Center 2.

Figure 4.47: Contention in a service center in accordance with the load level.

of Centers 1 and 2. The time spent by a vehicle (in average) to cross an intersection depends on the intersection concurrency. This function increases according to the angle of the linear function that describes the intersection contention. The larger the angle, the faster increase the center service time. For instance, assume 40 vehicles are crossing Center 1 with an average response time of 150 s, and the same amount of vehicles cross the intersection 2 during approximately 110 s. This shows a reduction of 26.67%. We can use this information to predict a road contention, and start a contingency plan.

4.4.3 Application of ALLuPIs

One of the purposes of vehicular networks is to tackle the traffic jam matter. In this context, researchers simulate real urban scenarios to evaluate vehicular network protocols [Naumov et al., 2006; Saha and Johnson, 2004]. However, routing protocols consider only road information, but not crossroad performance.

In this work, we proposed a model to evaluate crossroad intersections, considering the vehicles' mobility and the road characteristics, i.e., the model includes only vehicles data. The model is a queue network to represent the contention at each road intersection. We believe that this model can be applied to general traces with vehicle movement. The goal of this model is to analyze the impact of road intersection as possible traffic contention points through a real mobility model. Simulation of the scenario described in [Naumov et al., 2006] has been made. We believe that this model can be applied to general traces with vehicle movement.

Notice that we can use this information to predict a road contention and start a contingency plan. Protocols can initiate a congestion control method while quantifying the contention level. Other studies can use the information of crossroad performance to apply it in a routing or dissemination protocols in the search for the next hop.

4.5 Chapter Remarks

This chapter has investigated how the traffic flow, traffic incidents evolve spatially and temporally, how are they correlated, and whether social data (check-ins) can be used to signalize congested traffic flow. In other words, *can we use check-ins to better predict traffic jam?*

According with our results: *check-ins may be useful to better predict traffic jam.* We found that data from social sensors and traffic conditions, provided by Bing Maps, are surprisingly very correlated. The social data distribution is equal to the traffic condition distribution, shifted by an offset that can be easily calculated. That is, the time difference between the measurements is δ minutes, which indicates that social data signalize the traffic flow δ intervals latter. For Manhattan, δ is equal to 36 intervals. We believe that the 36 intervals threshold occurs because there is a time between the congestion and the time that the person arrives at the check-in point. The threshold can vary from place to place, but as shown in the paper, with our methodology this value can be easily calculated.

This information can be extremely valuable in several scenarios. For instance to build more efficient services that take advantage of such information, as in traffic prediction models or to build new VANET protocols to control and to minimize the traffic problems (e.g. jams).

Another approach that we have used in the characterization module is the proposal of ALLuPIs, in order to evaluate crossroad intersections, considering the mobility of vehicles and the road characteristics. This means that the model includes only vehicles data. ALLuPIs is a queue network to represent the contention at each road intersection. We believe that this model can be applied to general traces with vehicle movement. The goal of this model is to analyze the impact of road intersection as possible traffic contention points through a real mobility model. Simulation of the scenario described in [Naumov et al., 2006] has been made. We believe that this model can be applied to general traces with vehicle movement.

Our results have shown the impact of vehicle load in the throughput, response time and utilization. They indicate that there is an optimal number of vehicles competing for intersections to achieve a better cost/benefit relation in center response time versus utilization. The ideal number of vehicles crossing an intersection is the multiprogramming system level, which has a direct impact on the response time (time spent by a vehicle in a traffic jam). In this work, this multiprogramming level was 12 (vehicles). It means that if there are less than 12 vehicles, the cost/benefit relation in response time versus utilization is positive, and the system is available, otherwise it occurs a system saturation. Moreover, the multiprogramming level and the center response time affect the vehicular network performance. The traffic performance changes according to the number of vehicles that want to cross an intersection. The advantage is that the analytical model provides faster results than simulations.

Chapter 5

Prediction and Inferences

In a smart city, physical and virtual (or social) sensors can be used spread in the city to collect context data in order to provide more information about the city behavior in real-time. This was demonstrated in Chapters 3 and 4 through the data collection and characterization of traffic flow, incidents and check-ins. Since distinct cities have different characteristics and needs, the usage of such characterized data can help prediction models to better calibrate to different cities.

Figure 5.1 illustrates several data spread in the city, most of them discussed in the previous chapter. We can see check-ins (social sensor), traffic incidents and traffic flow (virtual sensors), and street detectors (physical sensor). Notice that they are all collected in a specific time interval, that can be different for each kind of sensor. We also know that dealing with such different time intervals is a challenge that, when overcome, facilitates the characterization of information. That is, how such data evolve temporally and spatially. Moreover, now we can understand the correlation between them in order to discover which data might influence or depend on others. For instance, we saw that check-ins distribution is similar to the intense traffic flow distribution, but with a given offset. Also, the occurrence of traffic incidents explains intense traffic flows.

Figure 5.2 summarizes this chapter goals. We propose the following prediction models: (1) MoVDic – Mobility Vehicular preDiction model [Tostes et al., 2015a]; (2) STRIP – Short-term TRaffIc jam Prediction [Tostes et al., 2013, 2015b]. While MoVDic uses only street detectors information from the above modules, STRIP uses both traffic flow and check-ins. Recall that the answers from both models can (and are) stored in inferences databases, presented in the following module (4. Inferences). Finally, about the evaluation of our models, we discuss:

- Which data are required, at least, to predict traffic mobility and traffic jam in the city?
- Which data can contribute to a better prediction?
- How much can they contribute in the accuracy of the data forecasted?

This chapter is organized in four sections. Section 5.1 defines and validates MoVDic, while Section 5.2 defines and validates STRIP. A brief discussion of our results is presented in Section 5.3. Finally, Section 5.4 concludes this chapter.

5.1 MoVDic

Here we present the proposed model MoVDic (Mobility Vehicular preDiction model), which predicts the mobility of vehicles in a city using a Bayesian network. The Bayesian network predicts the likelihood of a vehicle to follow a certain direction based on its current location and trajectory. The key idea is to predict the future locations of vehicles. To accomplish that, the following questions must be answered: (i) *if the vehicle is on the Y street, then what is the probability of driving to the adjacent streets of Y?*; (ii) *if the vehicle is on the Y street, then what is the probability of driving to a specific direction?*; (iii) *what is the probability of a vehicle being on the street Y given*

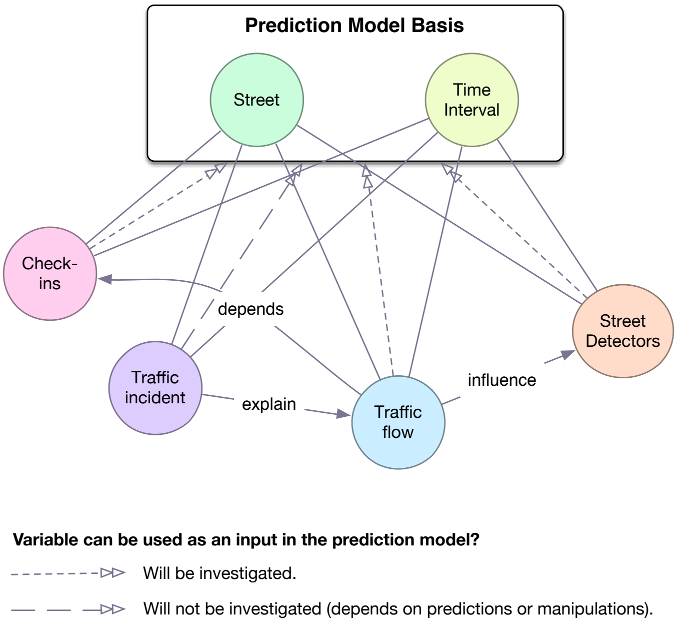


Figure 5.1: Which (and how) the data collected and characterized can be used in prediction models?

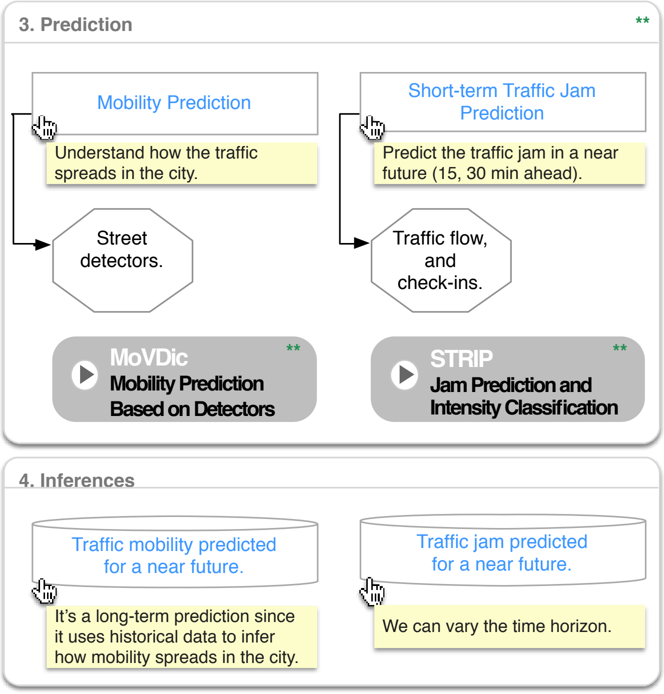

3. Prediction
**
Mobility Prediction
Understand how the traffic spreads in the city.
Street detectors.
Short-term Traffic Jam Prediction
Predict the traffic jam in a near future (15, 30 min ahead).
Traffic flow, and check-ins.
MoVDic
Mobility Prediction Based on Detectors
**
STRIP
Jam Prediction and Intensity Classification
**
4. Inferences
Traffic mobility predicted for a near future.
It's a long-term prediction since it uses historical data to infer how mobility spreads in the city.
Traffic jam predicted for a near future.
We can vary the time horizon.

Figure 5.2: Description of the Prediction and Inferences chapter.

that it is on the X street driving in a specific direction?; (iv) *how can we use our model to forecast future paths in order to reduce the emission of CO_2 and fuel consumed?*

However, the focus of this work is to answer the question (ii) by a Bayesian model learned through a Markov chain Monte Carlo algorithm. Notice that the only known variables are the data of street detectors, therefore we do not know the destinations of vehicles route. First, we assume a priori a uniform distribution for the turning choices, i.e., a vehicle has the same probability for following in any direction before the algorithm sees any empirical data. This uniform distribution corresponds to a random walk on the route graph. This uninformative distribution is updated by the Bayesian model through the MCMC algorithm. Despite knowing that this is not true in reality, this discrete prior distribution will be improved by MCMC. Moreover, the model uses only the data of street detectors, i.e., the number of vehicles driving on the street during a time interval, dispensing with the need to collect data on the actual turning choices at each intersection. The main idea is that the simple frequency of vehicles in each edge of the graph, and the implied many tight constraints it ensues, the entire flow can be predicted by a learning algorithm.

For instance, consider the scenario illustrated in Figure 5.3. Given the position and direction in which the vehicle is driving, the goal is to find out the odds on it following possible future directions. In the case of the vehicle A (blue), it can go in all

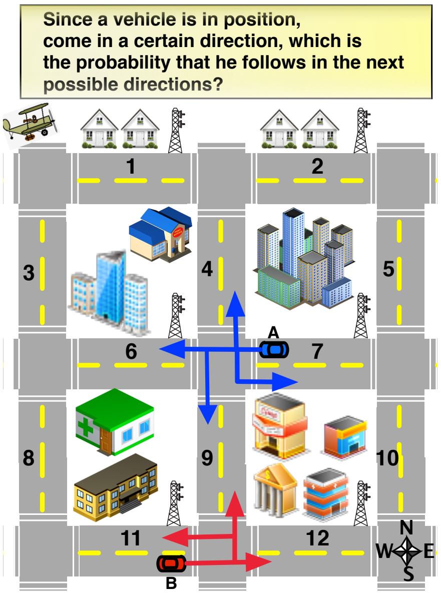

Since a vehicle is in position, come in a certain direction, which is the probability that he follows in the next possible directions?
1
2
3
4
5
6
7
A
8
9
10
11
12
B
N
W
E
S

Figure 5.3: Illustration of a question answered by MoVDic.

four directions. But the vehicle B (red) has a 0% probability of heading south (right conversion). However, other than this logical exclusion, we can need to predict from the remaining possibilities of conversion appropriately. Therefore, the issue faced by this work can be summarized by answering the following question: *for the vehicles on the street segments 7 and 11, when they arrive at an intersection, what is the probability of their future directions, that is, the likelihood of going to any of the adjacent street segments? How the mobility of vehicles spreads in the city?*

After updating the uniform prior distribution with the edge frequencies by mean of the Bayesian model, we obtain the posterior distribution, which describes the likely directions where there is greater flow of vehicles. In possess of such information, it is possible to improve routing algorithms and data dissemination protocols. As discussed in [Schougaard, 2007], protocols can be based on the behavior of the vehicular network itself. The context information (temperature and humidity sensors on roads) can be used in prediction models to save energy, for example. Furthermore, the forecast allows the discovery of the best route for vehicles, the reservation of resources (vehicles traveling on a road), the delivery of messages in real time with a lower loss of packets.

In order to evaluate our proposal, we validate our model with the large-scale mobility dataset of TAPAS Cologne [Uppoor and Fiore, 2012]. Two sub-scenarios are considered: (i) a small scenario with 125 streets; (ii) and a bigger scenario with 2216 streets, to represent the central area. Our evaluations were made to answer two main questions: (i) “How well the model fits when we want to know which will be the road with more vehicles in the near future?”; and (ii) “How well the model fits when we want to know which road will have more vehicles in the near future, when we are in a crossroad?” These two questions can be useful to dissemination protocols, i.e. when we need to decide where to send the packet when we are in a crossroad. Results have shown that the mean error is lower than 0.002, overcoming other approaches. Also, regarding the road with more turnings, our model performs well when choosing a street, achieving the mean accuracy of:

- **For the small scenario:** considering only crossroads with more than α vehicles ($\alpha = 21$), the accuracy for crossroad with two exit-streets is 88%, for three exit-streets is 83%, and for four exit-streets is 100%, and five exit-streets is 58%.

- **For the bigger scenario:** considering only crossroads with more than α vehicles ($\alpha = 21$), the accuracy for crossroad with two exit-streets is 95%, for three exit-streets is 94%, and for four exit-streets is 91%.

5.1.1 Definition

Our model MoVDic relies only on information of street detectors. It uses a Markov Chain Monte Carlo (MCMC) algorithm to learn a model providing the probability for vehicles movements when they arrive at an intersection.

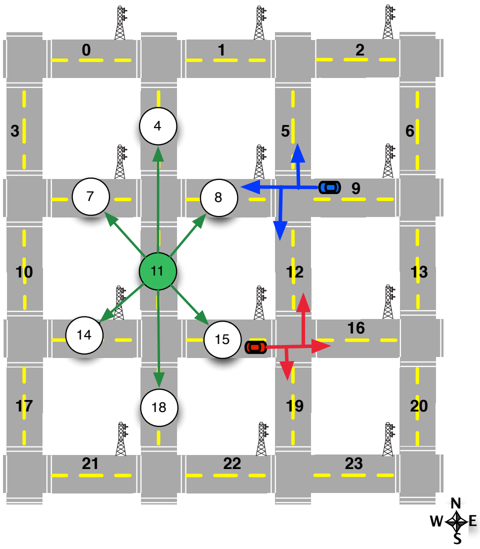


Figure 5.4: An example of graphical model over the traffic scenario.

Consider the scenario illustrated in Figure 5.4, which presents a 4 × 4 grid with 24 street segments, which corresponds to 48 lanes when we consider the both opposite moving directions. Each directed street segment is a vertex. We design an edge if we can go from one vertex to another crossing an intersection. Each street segment has a detector that counts the number of vehicles that drive on the road in a specific time interval. Based on the above scenario, our model calculates the probabilities of turning choices at the intersections of a street segment, based on the observed data from detectors in a street segment, considering the observed data from detectors.

Definition 9. θ_k**:** the probability of seeing a vehicle in the street segment k, which is defined as the one-way part of the street between two consecutive crossroads.

Definition 10. $\underset{\sim}{\theta}$**:** the vector of θ_k for all street segments in the region.

Definition 11. $P(\theta_k)$**:** considering that the vehicle is on the j-th segment, we assume that the probability of moving to the k-th segment is given by the equation:

$$P(\theta_k) = \frac{\theta_k}{\sum_{r \in N(k)} \theta_r} \tag{5.1}$$

where $r \in N(k)$ means that r is in $N(k)$, which is the set of neighbors of k, and this sum is over all neighbors of j, including k.

Based on the definitions 9, 10 and 11, we assume that the prior distribution regarding the movement of vehicles when they arrive at an intersection is a Dirichlet distribution for variables $\underset{\sim}{\theta}$ =$(\theta_1, \theta_2, \theta_3, \theta_4, \cdots, \theta_k)$ and parameters $\underset{\sim}{a} = (a_1, a_2, \cdots, a_k)$, as it shows Equation 5.2.

$$\underset{\sim}{\theta} = (\theta_1, \theta_2, \cdots, \theta_k) \sim Dir(k) \tag{5.2}$$

The Dirichlet distribution is defined in Equation 5.3:

$$\frac{1}{Z(\underset{\sim}{a}) \prod_{i=1}^{k} \theta_i^{a_i - 1}} \tag{5.3}$$

being $Z(\underset{\sim}{a})$ normalization constant. Notice that we do not need to know the normalization constant since it will be removed due to the Markovian property. Also, the sum of all $\underset{\sim}{\theta}$ must be 1. Therefore, our goal is to discover the probability of $\underset{\sim}{\theta} = (\theta_1, \theta_2, \cdots, \theta_k)$.

Consider $\underset{\sim}{X} = (X_0, X_1, X_2, \cdots, X_t)$ as the path taken by a vehicle. Hence, X_0 means that the vehicle is at a given segment at time $t = 0$, and X_1 means that this vehicle is at an adjacent segment at time $t = 1$, and henceforth. The likelihood function is the conditional probability

$$\mathbb{P}(X_0, X_1, X_2, \cdots, X_t | \theta_1, \theta_2, \cdots, \theta_k).$$

The posterior distribution can be calculated as the prior distribution times the likelihood function.

To impose sparsity and decrease complexity in estimation, we assume a Markovian property for the probabilities. That is, θ_j depends only on the last position of the vehicle, and not on the entire trajectory taken until then, and this is valid for each variable of $\underset{\sim}{\theta}$. Therefore, the historical paths can be discarded. We only need to know the last position of the vehicle to predict its future position, and this information is described in the data of street detector. The two pieces of information that the system must know about the trajectories are: (i) the number of times the stretch j was traveled by any vehicle (N_j) and (ii) the number vehicles that began their route at stretch j (M_j). Thereafter, according to the Bayes theorem, we can simplify the conditional probability distribution of θ_j, for each variable of $\underset{\sim}{\theta}$, given by Equation 5.4 less than a constant.

$$\mathbb{P}(\theta_j | \cdots) \quad \propto \quad \theta_j^{a_j+N_j+M_j-1} \cdot \theta_k^{a_k+N_k+M_k-1} \cdot \alpha_j^{N_j} \tag{5.4}$$

The symbol "$\cdots$" after the conditioning symbol stands for all the other θ_j values but θ_k, which are all the neighbors of θ_j. This means that the proposal of θ_j depends only on its last value (proposed in the last iteration) and the values of its neighbors, represented by θ_k. To propose the last θ_k, we can do the normalization, because the sum of all variables of $\underset{\sim}{\theta}$ is 1. This conditional distribution is what the MCMC algorithm requires to obtain the joint posterior distribution.

Algorithm 7 presents our prediction model. The values proposed to $\underset{\sim}{\theta}$ is stored in the matrix Θ, defined in line 4 and 5. The iterations of the MCMC can be seen at line 11. The distribution of θ_j is calculated from Equation 5.4. To draft a new value of θ_j for each street segment, we use the Metropolis-Hasting MCMC algorithm (lines 12, 13, 16–22). It proposes a new value for each θ_j (line 13), except for the last, which is obtained by the linear restriction (the sum of $\underset{\sim}{\theta}$ is 1). The real data used by the MCMC are the values N_j and M_j, according to Equation 5.4.

To implement the MCMC, there were two main challenges:

1. How to deal with *underflow*?

Algorithm 7 MoVDic Algorithm

```
Require: $nsim$: number of iterations in MCMC;
$seg$: number of segments in the street; MCMC inputs
Ensure: Distribution of $\underset{\sim}{\theta}$, which is in the last line of the matrix $\Theta$
procedure PREDICTION ($nsim$, $seg$, MCMC inputs)
                                              ▷ Initial guess: uniform distribution
    $\Theta \leftarrow$ numeric($nsim$*$seg$)
    $\Theta \leftarrow$ matrix($\Theta$, nrow = $nsim$, ncol=$seg$)
    $ratio \leftarrow 1/seg$
    for i in 1:$seg$ do
        $\Theta[1, i] \leftarrow$ ratio
    end for
                                              ▷ Iterations of MCMC
    for i in 2:nsim do
        for k in 1:$seg$ do
            Calculate the $\phi_k$ as $\log(\theta_k)$
        end for
        for j in 1:($seg$-1) do
            Store the $\theta_j$ ($\Theta[i, j]$) from the previous iteration ($\Theta[i - 1, j]$)
            $\Theta[i, j]$ = Propose a $\theta_j*$ that is not zero
            Calculate the $\phi_j*$ of such $\theta_j*$
            Calculate the probability of $\theta_j$ according to eq. 5.4 (in function of $\phi_j$)
            $\alpha* \leftarrow P(\phi_j*) - P(\phi_j)$
            $U* \leftarrow$ log(runif(1))
            if $U* < min(0, \alpha*)$ then
                Accept the candidate
            else
                Reject the candidate
            end if
        end for
                                              ▷ Update the last $\theta_k$
        $\Theta$[i, $seg$] $\leftarrow$ 1 - $\sum$($\Theta$[i,1:($seg$-1)])
        Update the probability of this iteration as the previous
    end for
    return $\Theta[nsim]$
end procedure
```

2. How to propose a new value according to Equation 5.4?

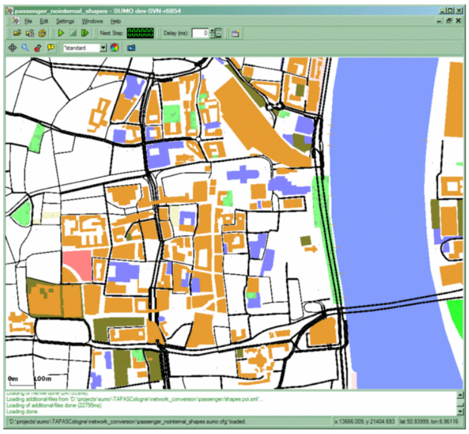
To deal with (1), we use the logarithmic transformation of $\phi_j = \log(\theta_j)$, and then we modify the Equation 5.4. For (2), the challenge was to propose a new valid value for θ_j. This new value is proposed according to a uniform distribution between zero and a maximum value, which is calculated as the excess to complete all the probabilities. That is, $\theta_j^* = \{1 - \sum_{i=1}^{N} \theta_i + \theta_j\}$. This proposed value can be accepted or rejected according to the Metropolis-Hastings test. During such process, a number of the last values of θ_j must be stored to empirically evaluate the posterior distribution.

Therefore, MoVDic provides the following information based on only street detectors: (i) the probability of seeing a vehicle on the street segment k (θ_k – based on Equation 5.4); and (ii) the probability of turning choices in a crossroad (based on Equation 5.1). The evaluation of achieving such information is in the next section.

5.1.2 Validation

5.1.2.1 Simulation Description

To validate our model, we use two sub-scenarios from the large-scale realistic mobility dataset of TAPAS Cologne. Such dataset spans two hours of vehicles' movements over a 400km^2 area of the city of Cologne, Germany [Uppoor and Fiore, 2012]. Such dataset is realistic from both macroscopic and microscopic viewpoints [Uppoor and Fiore, 2011]. Moreover, it contains information such as different types of roads, traffic light programs and road signalization, which is used as input for the Simulation of Urban MObility (SUMO) [Behrisch et al., 2011]. Notice that, this turns our analysis much more accurate than using a regular Manhattan grid, as the one illustrated in Figure 5.5.



```
Loading of net-file done (247353ms).
Loading additional-files from 'D:\projects\sumo\TAPASCologne\network_conversion\passenger/shapes.poi.xml'...
Loading of additional-files done (22795ms).
Loading done.
'D:\projects\sumo\TAPASCologne\network_conversion\passenger_nointernal_shapes.sumo.cfg' loaded.   x:13666.009, y:21404.693   lat:50.93999, lon:6.96116
```

Figure 5.5: Part of the TAPAS Cologne Scenario [Uppoor et al., 2013].

To assess the behavior of the model under different road traffic conditions, we predicted the mobility of vehicles at different times of the day. Initially, we assume the first 1800 seconds as the warm-up period, and then, at every 30 seconds interval, our model generates an output, which is produced offline (that is, not during the simulation). Note that the choice of 30 minutes is empirical, and this does not affect our results, it is just to count the amount of cars on the road. That is, our model

Table 5.1: Vehicle density for different times of the day for the real city scenario.

Time of the day (a.m.)	Density (vehicles/km^2)
06:30	61
06:45	82
07:00	92
07:15	102
07:30	108

predicts the positions of vehicles at time steps 1830 s, 1860 s, 1890 s, $\cdots$, 5400 s, for a total of 120 intervals. Notice that, as the time of the day increases, so does the road traffic, as shown in Table 5.1.

Our model uses a graph in which the street segments are the vertexes. Notice that, this is the opposite of the road topology graph, in which the junctions are the vertexes and the roads are the edges on the graph. Therefore, we created the equivalent line graph for the road topology graph. Due to scalability issues, we have used subgraphs of the original graph. Here, we present the model validation for two scenarios: (i) with 125 streets, named S125; (ii) with 2216 streets, named S2216. The process to create these scenarios was the following: (i) choose the street with the highest betweenness centrality; (ii) initiate a depth-first search and add the streets to the sub-scenario until the maximum number of streets is achieved.

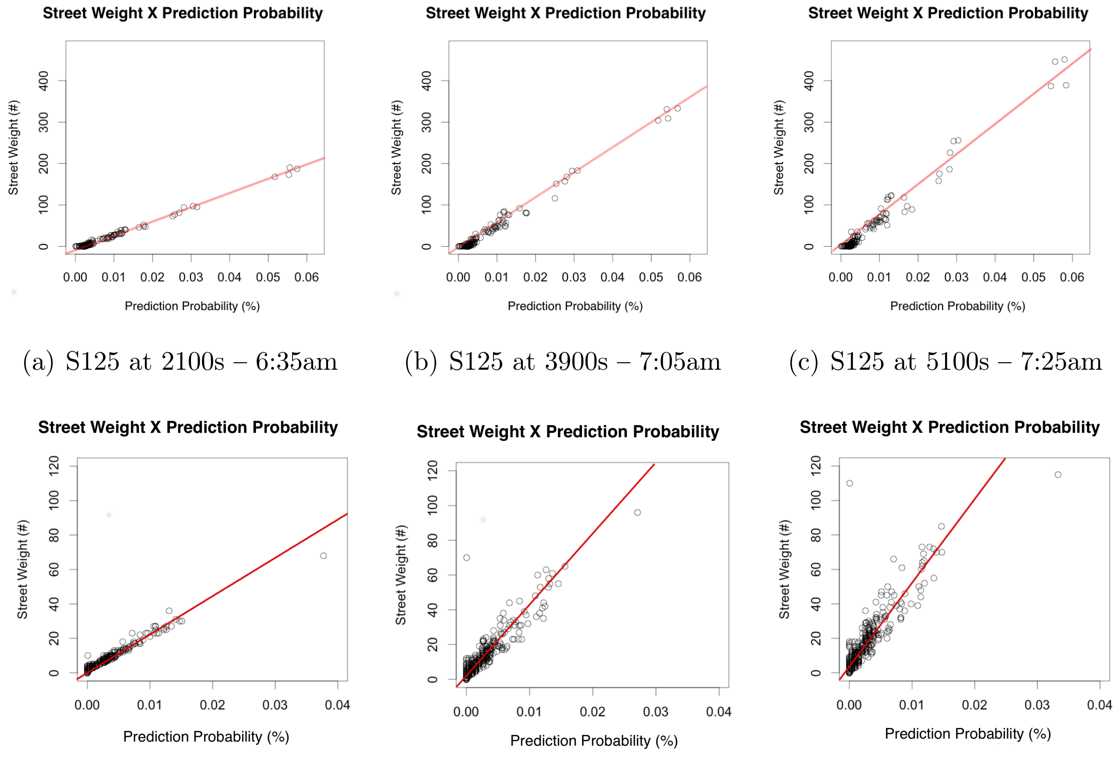
5.1.2.2 Evaluations

Here we present the evaluations of MoVDic.

Evolution of $\underset{\sim}{\theta}$ against the demand of vehicles: Common sense indicates that the number of vehicles driving through a street segment is proportional to the probability of seeing a vehicle in that street. To support that, we plotted Figures 5.6 and 5.7, which demonstrate the correlation between the prediction probability and the values from street detectors (weight). The red dots indicate a higher amount of vehicles that the one predicted.

One can notice the difference between the smaller and bigger scenarios through Figure 5.7. In the sub-scenario S2216, due to the amount of data, we can clearly see the correlation between the variables weight and prediction during a time variation. As the time increases and there are more vehicles on the roads (weight), the greater the probability of seeing a vehicle (prediction).

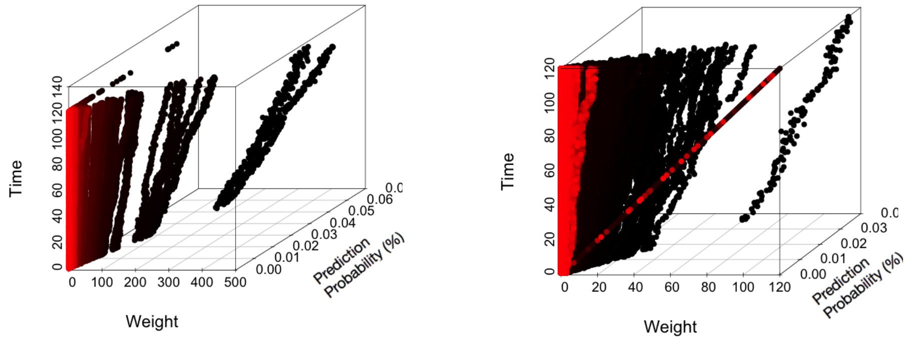
Besides static images, we have also developed videos containing graph images for all time intervals in order to allow a temporal and spatial analysis about the mobil-



(a) S125 at 2100s – 6:35am (b) S125 at 3900s – 7:05am (c) S125 at 5100s – 7:25am

(d) S2216 at 2100s – 6:35am (e) S2216 at 3900s – 7:05am (f) S2216 at 5100s – 7:25am

Figure 5.6: Correlation between the street detectors and the prediction during different simulation times.

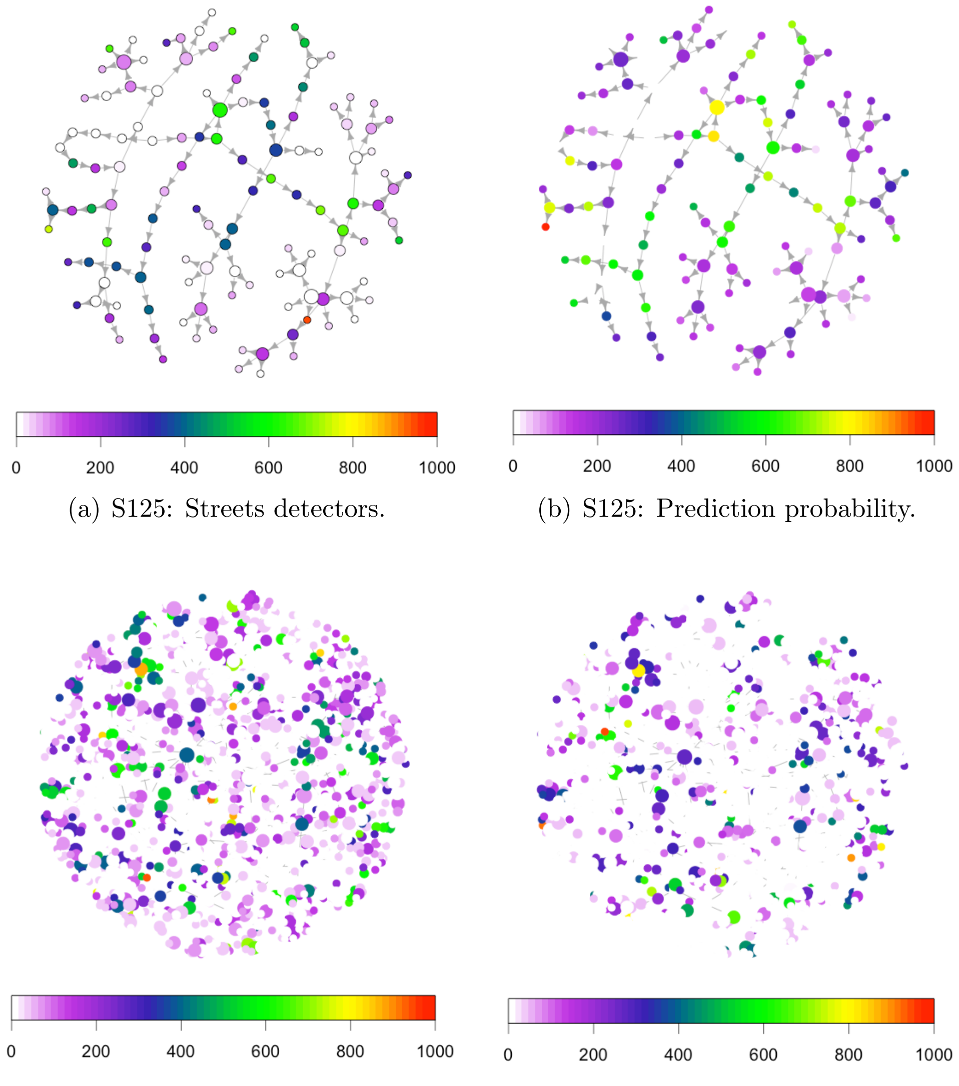


(a) S125 (b) S2216

Figure 5.7: Scatterplot of time X weight X prediction.

ity of vehicles. They are available online (`http://annatostes.azurewebsites.net/mobility-prediction-model/`). The videos show the traffic flow increasing in specific vertexes and the prediction model output predicting this evolution.

Figures 5.8(a), 5.8(b), 5.8(c) and 5.8(d) present the weight (detector) and prediction graphs at time $t = 3900$s, respectively, for the sub-scenarios S125 and S2216. The



(a) S125: Streets detectors.

(b) S125: Prediction probability.

(c) S2216: Streets detectors.

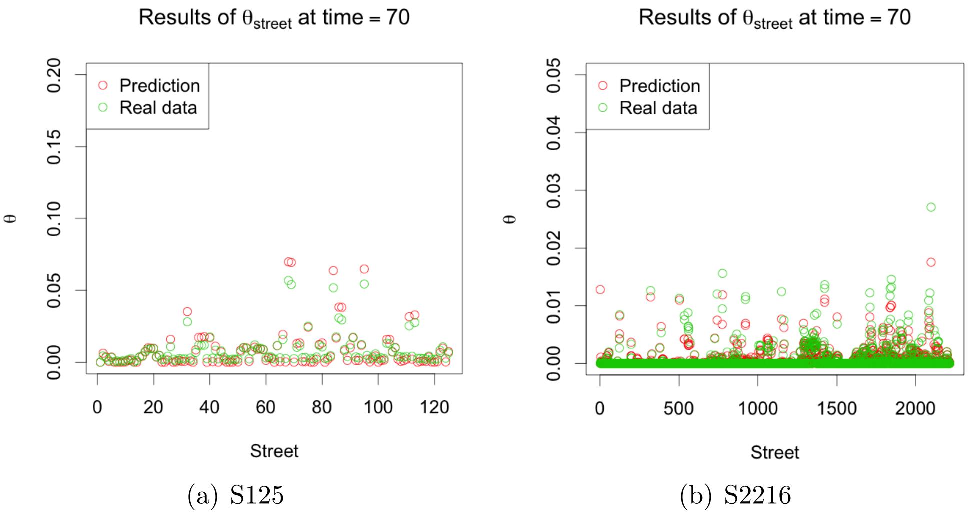
(d) S2216: Prediction probability.

Figure 5.8: Graph illustrations at 3900s of simulation (real time: 7:05am).

colors in vertexes indicate the detectors values, and their sizes indicate their degree. One can notice that the vertexes with colors dark blue or higher (color bar) have a higher probability in our prediction model (green or higher).

Probability of $\underset{\sim}{\theta}$: The main results are summarized in Figure 5.9, which presents the probability of seeing a vehicle in each street segment (θ). One can notice that the predicted probabilities follow the real probabilities, but there are mistakes. The question is: how much distant from the real probabilities they are? This is answered in this section.

To answer this question, first we have used the metric known as Prediction Error



(a) S125 (b) S2216

Figure 5.9: Prediction and real probabilities of the variables at 3900s (7:05am).

(ϵ), represented by

$$\epsilon(\theta) = max(|\theta_k^p - \theta_k^r|)$$

that measures the difference between the real value of θ_k^r and the predicted value of θ_k^p. If the crossroad has a minimum number of vehicles (α) and the prediction error ϵ is less than 0.1, than the model is accurate. Otherwise, it is not.

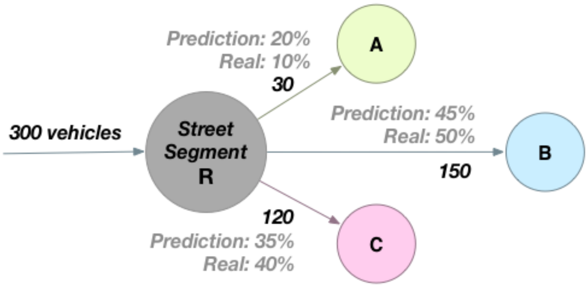


Figure 5.10: Example of future possible paths.

First, we classified the crossroads based on the number of exit-streets that they present (2, 3, 4 or 5). For instance, consider the example illustrated by Figure 5.10 that indicates possible paths for the vehicles in street segment R with 300 vehicles, that is, to take route A, B or C (3 exit-streets). For each street segment, we calculate the probability of vehicles going to the adjacent streets, having for instance $\alpha_A, \alpha_B, \alpha_C$ for the real data (observed – that is the hypothetical values of 10%, 50%, and 40%) and $\beta_A, \beta_B, \beta_C$ for our probability model (that is 20%, 45% and 35%), that is calculated through the Dirichlet distribution (see Equation 5.1). For this example we have

$$\alpha_A = \frac{\theta_A}{(\theta_A + \theta_B + \theta_C)},$$

in which the sum of α_i for each i representing an street segment adjacent of R is 100%.

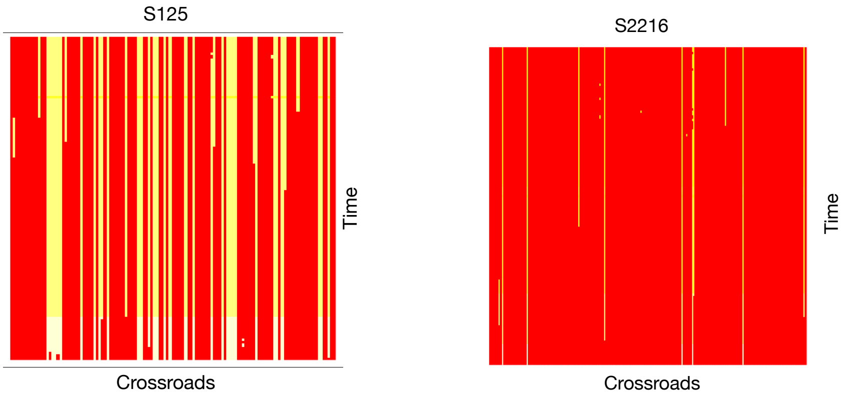
Second, we varied the minimum number of vehicles, represented by α. This variation was made based on the density of vehicles in the scenario, described in Table 5.1 in Section 5.1.2.1 (61, 82, 92, 102, 108).

Third, we answered the following questions.

Q1. How many crossroads have the proper conditions to MoVDic makes its prediction? And based on this amount of valid crossroads, how many of them are accurate?

Here we have calculated the percentage of crossroads where the mean prediction error is lower than 0.10 with a minimum of α vehicles. Figure 5.11 shows a heatmap of crossroads where MoVDic's performance is accurate over the time. In this graph, for each crossroad in every second of simulation, we indicate whether the crossroad has (yellow) or has not (red) proper conditions for MoVDic prediction. Since scenario S125 (500 vehicles) has more vehicles than S2216 (130 vehicles), than it has more crossroads with the proper conditions to be predicted than in S2216. We can notice that most crossroads with proper conditions tend to keep this way during all simulation, depending on the amount of vehicles.

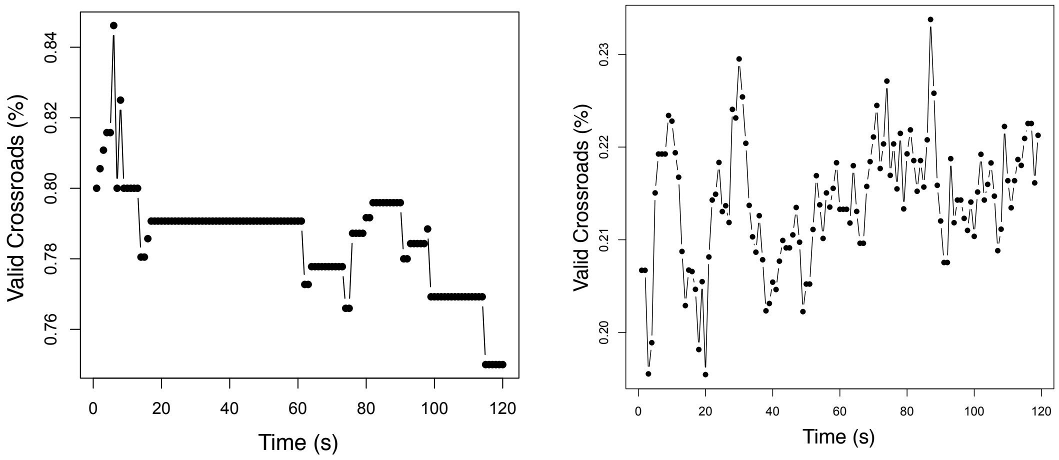
To complement, recall to Figure 5.12. It shows the percentage of valid crossroads, that is, the frequency of crossroads that have proper conditions for MoVDic prediction.



(a) S125

(b) S2216

Figure 5.11: Indicative of crossroads where MoVDic is accurate over the time.

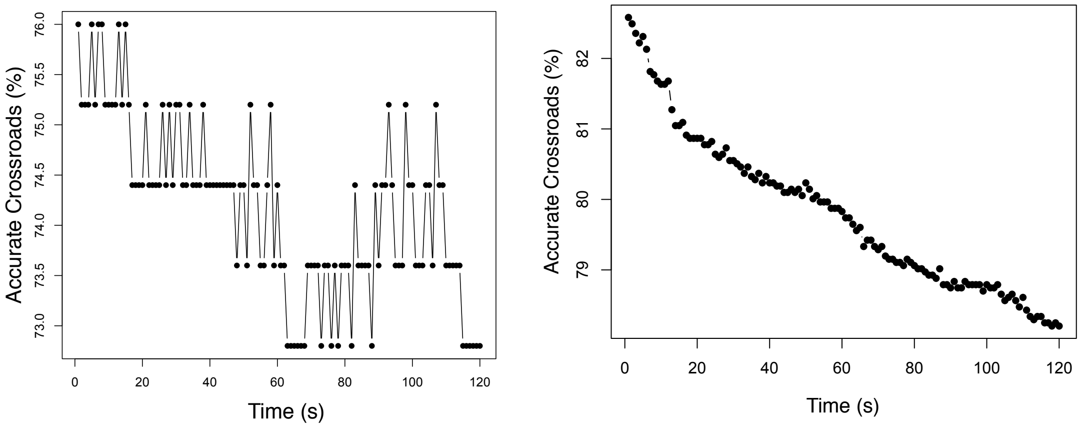


(a) S125 (b) S2216

Figure 5.12: Percentage of valid crossroads, with proper conditions to predict.

We can see that the amount of valid crossroads is higher in the smaller scenario (S125) than in the bigger scenario (S2216), ranging from 84% to 75% of crossroads over the simulation time. Therefore, we can conclude that scenario S125 has more example cases to evaluate MoVDic than S2216.

Q2. How does the frequency of accurate crossroads varies over the time, considering that the demand of vehicles also increases?



(a) S125 (b) S2216

Figure 5.13: Percentage of crossroads where MoVDic is accurate over the time.

To make this evaluation, we have calculated the mean value of ϵ over the time only for the valid crossroads. If the value of ϵ was lower than 0.10, than MoVDic fitted well and we considered this as a accurate crossroad. Therefore, this metric indicates the percentage of accurate crossroads for MoVDic. Figure 5.13 illustrates the results. We can see that more MoVDic performs well for more than 70% of crossroads. For the scenario S2216, we can see that the frequency reduces from 83% down to 78%. This occurs because of the lower amount of vehicles in the scenario. Also, we can observe that the variation of the prediction accuracy in the entire simulation is lower, being easier to predict.

Q3. How well the model adjust to predict the street with more turnings, according with Equation 5.1?

To answer this question, we have considered the hypothesis that a vehicle will turn in the street with the highest probability to turn. In the example of Figure 5.10, we consider that a vehicle will turn to B because it presents the highest probability (real of 50% and prediction of 45%), in comparison with the other streets A and C. If this is correct, than we consider this a hit.

Table 5.2: Frequency of hits in the prediction of the street with more turnings, considering a crossroad with 2 up to 5 exit-streets.

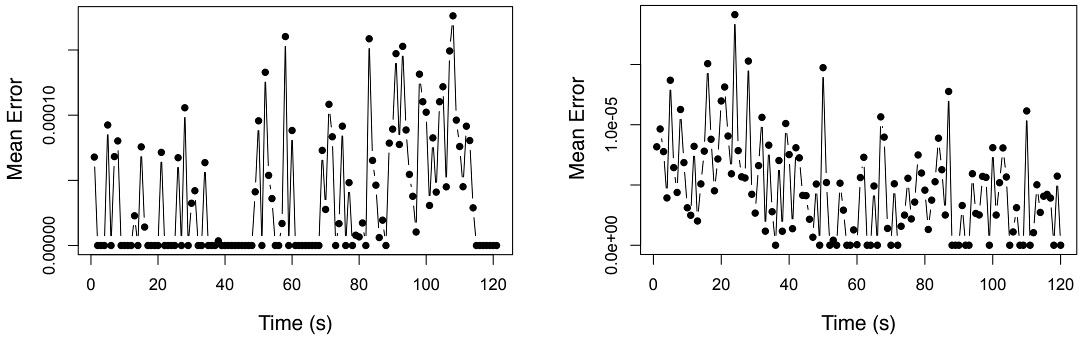
	S125				S2216			
α	2 streets	3 streets	4 streets	5 streets	2 streets	3 streets	4 streets	5 streets
1	0.89	0.87	0.80	0.58	0.95	0.88	0.90	0.89
11	0.88	0.85	0.96	0.58	0.90	0.88	0.91	0.92
21	0.88	0.83	1.00	0.58	0.95	0.94	0.91	–
31	0.91	0.83	–	0.54	0.99	0.99	0.92	–
41	0.93	0.82	–	0.53	1.00	–	–	–
51	0.93	0.83	–	0.51	1.00	–	–	–
61	0.96	0.85	–	0.47	1.00	–	–	–
82	1.00	0.90	–	0.41	–	–	–	–
92	1.00	0.92	–	0.38	–	–	–	–
102	1.00	0.96	–	0.29	–	–	–	–
108	1.00	0.96	–	0.27	–	–	–	–
120	1.00	0.99	–	0.20	–	–	–	–
150	1.00	1.00	–	–	–	–	–	–
200	–	–	–	–	–	–	–	–

Based on this, table 5.2 summarizes the frequency of hits in the prediction of the street with more turning in a crossroad, also varying the minimum number of vehicles (α). Notice that we have clustered the crossroads with similar features, that is, the number of exit streets (2, 3, 4 or 5).

Notice that the performance of MoVDic for the smaller scenario (S125) is higher than 0.8 (80%) for crossroads with 2, 3 and 4 exit-streets, but it is not good for 5 exit-streets. Also, when we increase the value of α, the performance of MoVDic also improves. Regarding the amount of vehicles in the scenario with more exit-streets, MoVDic does not perform well, since it requires more features than only detectors. Finally, in the bigger scenario, with less number of vehicles (120 vehicles agains 500 vehicles in the smaller scenario), MoVDic also performs well (more than 0.88 – 88% of crossroads). Notice that the sample size for both scenarios is different: S125 has more valid crossroads than S2216.

Q4. What is the mean value of ϵ for each category (number of exit-streets)? How does it varies over the time?

Here we have calculated the prediction error (ϵ). Figure 5.14 presents the evolution of the mean value of ϵ over the time. One can notice that the mean error is lower than 0.002 during the entire simulation, being lower for the bigger scenario than the small scenario.



(a) S125

(b) S2216

Figure 5.14: Mean error for valid crossroads.

5.2 STRIP

In this section, we describe the proposed prediction model STRIP – Short-term TRaffIc jam Prediction model [Tostes et al., 2015b], based on logistic regression. In statistics, logistic regression is a type of regression analysis used for predicting the outcome of a categorical dependent variable based on one or more predictor variables. Its main characteristic is that the probabilities describing the possible outcomes of a single trial are modeled, as a function of the predictor variables, using a logistic function. Logistic regression measures the relationship between a categorical dependent variable and one or more independent variables. The probability function follows the logistic function:

$$P(t) = \frac{1}{1 + e^{-t}}$$

5.2.1 Definition

STRIP is composed of two logistic regressions (LRs), one to predict whether the flow is free or congested, and another one to predict the congestion intensity. The prediction is based on several variables such as weekday, hour of the day, historical traffic conditions (crawled from Bing Maps using the methodology described in [Tostes et al., 2013]) and social sensors data (crawled from Foursquare and Instagram as described in [Ribeiro et al., 2014]). The traffic flow can be: (1) green, representing a free flow; (2) yellow, representing a low congested flow; or (3) red, representing a high-congested flow.

Figure 5.15 depicts the architecture of STRIP. We have several basic inputs (weekday, hour, street ID, historical traffic and check-ins) to predict the level of traffic congestion (low or high). STRIP consists of one Logistic Regression (LR) per street, predicting the traffic conditions locally (a distributed version). For vehicular networks, this model can contribute to a lower overload in the network, a lower exchange of information between roadside units and vehicles, and an increase in the performance of the prediction itself.

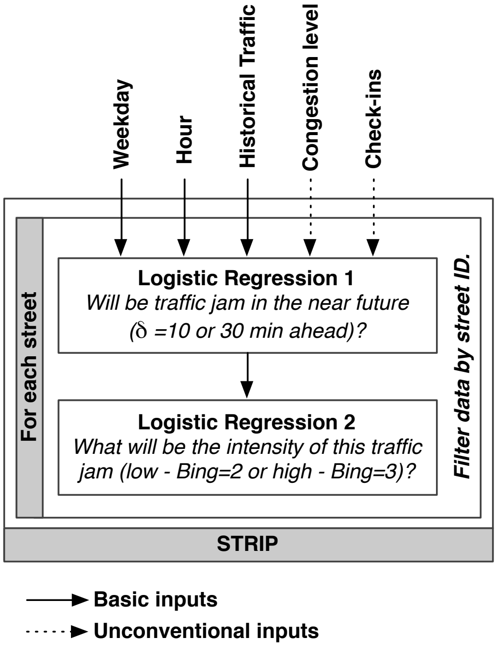


Figure 5.15: Architecture of STRIP

LR is a type of regression analysis used for predicting the outcome of a categorical dependent variable based on one or more predictor variables. Its main characteristic is that the probabilities describing the possible outcomes of a single trial are modeled as a function of the predictor variables, using a logistic function. LR measures the relationship between a categorical dependent variable and variables. The probability function follows the logistic function (Equation 5.5).

$$P(t) = \frac{1}{1 + e^{-t}}. \tag{5.5}$$

We have used LR because it refers to the problem in which the dependent variable is binary, which is our case. In addition, different from the linear regression, LR considers the conditional distribution as a Binomial distribution, and not a Gaussian distribution. In the case of a traffic jam, we are indicating where there is a probability of having intense traffic condition or not (Binomial). Also, since LR predicts the probability of the instance being positive, the estimated probabilities are restricted to $[0, 1]$ through the logistic distribution function.

In this context, STRIP differs from other approaches in terms of the used information sources in the prediction. While most solutions use only road data (e.g., [Kong et al., 2013; Kurihara, 2013; Li et al., 2013]) and GPS traces (e.g., [Niu et al., 2014]), some approaches use additional distinct inputs such as weather (e.g., [Li et al., 2013; Microsoft Research, 2013b]), events (e.g., [Ghosh et al., 2009; Microsoft Research, 2013b]), and traffic incidents (e.g., [Inrix, 2006; Microsoft Research, 2013b]). What is quite interesting is that none of the prediction models we found in the literature has used social sensors data, and that is exactly what we did. Another contribution of this work is that we can evaluate our model in real time and apply a street-level granularity to assess how much each input variable contributes to the predicted output, allowing us to evaluate if we will consider this input or not.

Algorithm 8 shows the STRIP algorithm, being executed for each street in the city every 15 min. First, the algorithm randomly selects 70% of the data for the training set (Lines 3–11). Second, we perform the LR for the requested street (Lines 12–19). Third, we perform the second LR, to classify the intensity of the traffic jam (Lines 20–35). Four, we perform the validation and the Chi-squared test to evaluate if all inputs have contributed to the prediction output or not (Lines 38–41). This allows us to indicate, in real time, if an input is significant to the traffic forecasting or not.

Each independent variable is categorical or numeric. The green flow (Bing $= 1$) intensity is considered as class 0, while class 1 is yellow (Bing $= 2$), and red (Bing $=$ 3) represents the congested flow. By removing the green data, the second LR classifies

Algorithm 8 STRIP

```
Require: Database data of inputs and street id
Ensure: Logistic regressions of STRIP
                                        ▷ Run this algorithm for each street in the city.
procedure LogisticRegression (data, id)
    data1 ← randomly chooses 70% weekday samples from data (training dataset)
                                        ▷ The categorical dependent variable
    y ← vector of bing flow intensity (1, 2 or 3) of data1
    x ← matrix of the six independent variables of data1
                                        ▷ Consider bing = 1 as class 0 and others as class 1
    y ← y−1                             ▷ in order to consider bing as 0, 1 and 2
    y[y==2] ← 1                         ▷ considering bing=2 as 1
    class0 ← y == 0
    class1 ← !class0
                                        ▷ Run the logistic regression 1 – classifier 1, with all inputs
    flrm1 ← glm(y ∼ (factor(x[,1])
            + factor(x[,2])
            + as.numeric(x[,3])
            + as.numeric(x[,4])
            + as.numeric(x[,5])
            + (...)
            ), family=binomial("logit"))
                                        ▷ Prepare data for classifier 2
    y ← vector of bing flow intensity (only 2 or 3)
    x ← matrix of the six independent variables data
                                        ▷ Consider the bing value 2 as class 0 and the bing value 3 as class 1
    y ← y−2
    class0 ← y == 0
    class1 ← !class0
                                        ▷ Run the logistic regression 2
    flrm2 ← glm(y ∼ (factor(x[,1])
            + factor(x[,2])
            + as.numeric(x[,3])
            + as.numeric(x[,4])
            + as.numeric(x[,5])
            + as.numeric(x[,6])
            + (...)
            ), family=binomial("logit"))
    data2 ← the remaining 30% weekday samples from data (validation dataset)
    Evaluate the model accuracy for street id
    Perform the Chi-squared test and produce the p-value for all inputs.
    if p−value >= 0.05 then
        Input not significant; suggest to remove the input.
    end ifreturn flrm1 and flrm2
end procedure
```

the flow intensity as yellow (class 0) and as red (class 1), measuring the intensity of the traffic jam. The STRIP classifiers of were written using the `glm` library in R–Project.

The basic inputs are: (i) weekday, (ii) time, (iii) street number, (iv) historical traffic flow (i.e., traffic condition obtained from Bing for a specific time interval in the past at the same day of the week). As new unconventional inputs, we are using data from social sensors (check-ins) performed in PSNs. As discussed in [Ribeiro et al., 2014], part of this thesis, data from social sensors and traffic conditions are correlated. The distribution of the number of observed social data is equal to the traffic condition distribution, shifted by an offset easily calculated. This information can be extremely valuable to build more efficient traffic condition predictors, but the problem is to know in real time when and where (street level) this data really contributes to the traffic prediction.

The main advantage of the STRIP architecture is its flexibility for defining inputs. We can add new inputs to the LR and, during the training, evaluate how much each variable contributes to the predicted output. For this specific evaluation, we used the Chi-squared test. If the probability value (p–value[1]) is lower than 0.05, the more significant the input is.

Finally, it is important to compare our initial prediction model [Tostes et al., 2013] with STRIP, proposed in this work. In [Tostes et al., 2013], we predicted the traffic jam in the next instant of time (next minute), which is less useful. The inputs for our previous model were date, time and historical traffic flow. The model was based on two logistic regressions for the entire city, which is not scalable. In this work, we propose a scalable distributed model, STRIP, that predicts the traffic jam in a near future with a higher time horizon (15 min, 30 min, 1 hour). Another difference is that we use a new input that considers data from PSNs as a social sensor (virtual sensor). Besides, since STRIP presents a street granularity, that is scalable, we can now measure in real time how significant this input is to the model accuracy. This is possible due to the Chi-squared test, based on the probability value, as explained above.

5.2.2 Validation

In this section, we evaluate STRIP considering the dataset of traffic flow described in Section 3.1.2, in Chapter 3.

Q1. If we increase the amount of historical data while increasing the time horizon (δ), how this will affect the prediction accuracy in terms of the quality metrics (accuracy, recall, false positive rates, specificity, precision)?

The hypothesis of Q1 is that there will be a specific threshold of time frame that contributes to a better prediction. Moreover, this amount will not be statistically significant to an increase in the prediction accuracy.

In order to evaluate STRIP in terms of the amount of historical data necessary to achieve a better accuracy, we selected New York (Manhattan), which also appears

[1] It shows that the observed result is highly unlikely under a null hypothesis.

in our previous work [Tostes et al., 2013]. We collected a different and longer time interval for this dataset: from June 24, 2013 to August 22, 2013, with a lower time interval for the Bing maps acquisition (every 1 min instead of 7 min). We have selected the same amount of data for each category (free and congested).

Table 5.3 summarizes STRIP's accuracy against the amount of historical data. The best accuracy is achieved when we consider the last 15 min of traffic flow as input. Initially, we hypothesized that the predictor is not as sensitive to the amount of data when we give more than 5 min information. This was confirmed by the data of Table 5.3: 81% of accuracy with 5 min in comparison with 81% of accuracy with 15 min. Since we can ignore the significant digits in the accuracy, we can observe that the increase of historical data does not improve the accuracy of the prediction model, and does not reduce the rate of false positive as well. This means that knowing the last 5 min of traffic flow, it is enough for the good performance of STRIP.

Table 5.3: Accuracy of STRIP while increasing the amount of historical data.

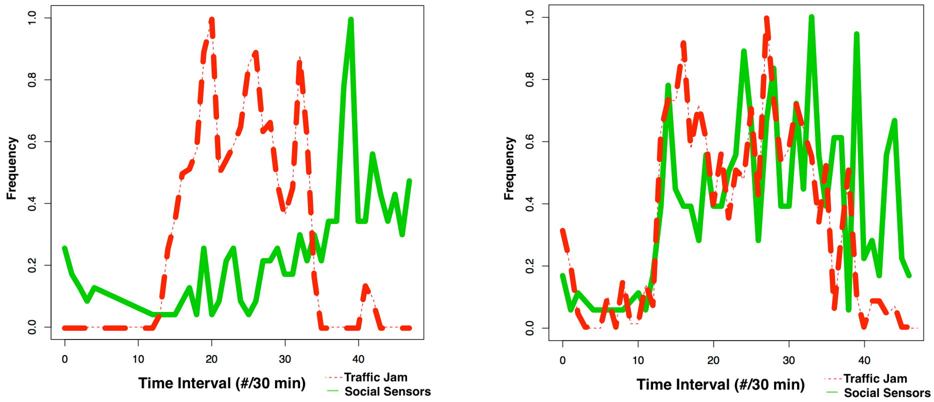
Minutes	Accuracy	Recall	False Positive Rate	Specificity	Precision
1	75.80%	84.99%	44.65%	55.35%	80.89%
5	81.03%	88.57%	35.72%	64.27%	84.64%
10	80.94%	88.25%	34.31%	64.69%	84.75%
15	81.06%	88.24%	34.89%	65.11%	84.91%

Q2. What is the temporal behavior of intense traffic conditions and the amount of check-ins, considering a street granularity?

The hypothesis of Q2 is that the higher the number of check-ins on the streets, the higher the probability of occurring a traffic jam. For comparison purposes, we have used the same NYC dataset, covering the period from June 24, 2013 to August 22, 2013, to predict future traffic jam in the next 30 min. We compare STRIP with our previous model (accuracy of 80%) [Tostes et al., 2013]. Additionally, we demonstrate why and how data from social sensors can be used for a better prediction. We used two social sensor datasets: (SS1) measures the amount of check-ins, referred here as τ, that occurs on or near the street; (SS2) measures the difference between two consecutive check-ins near the street, i.e., between the SS1 data τ_t (t is the current time) and $\tau_{t-\delta}$.

Figure 5.16 shows the likelihood of the SS 1 (green) and of the traffic jam (red) for two different streets (IDs 35 and 266) for NYC. We present the results for these streets, because they are popular in terms of number of check-ins and traffic jam. Recall that we consider check-ins as data from social sensors, as we mentioned above.

We can observe that during the night, when there is less data from social sensors, it is less likely to have a traffic jam. After 35 intervals of 30 min (= 5:30pm), there are



(a) Accuracy of 85.74% (street 35)

(b) Accuracy of 90.48% (street 266)

Figure 5.16: Likelihood of SS data (green) and traffic jam (red).

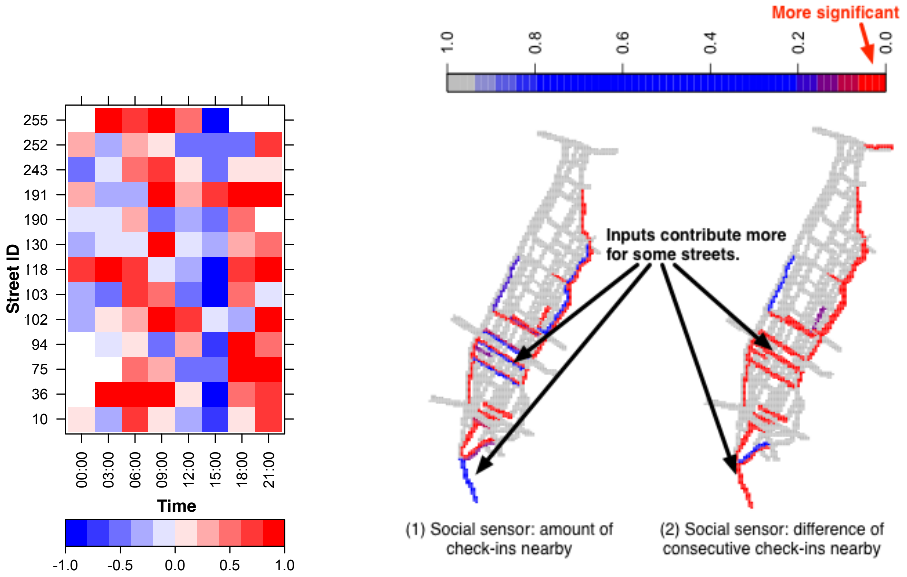
more data from social sensors and less traffic jam. Differently, in street 266, we have similar distributions for data from social sensors and jams. Thus, we have two patterns indicating whether the social sensor data will contribute (or not) to the prediction model. We can observe that for streets like 266, the accuracy increased from 80% up to 90%, showing that social sensors can improve the performance of prediction models for traffic jam.

Q3. Can we identify in real time when and where the check-ins (social sensor) can be significant to the prediction model?

The hypothesis of Q3 is that for some streets, which have more check-ins and more variation of traffic conditions, this will be correct. The social sensor will possibly contribute, but varying spatially and temporally. The question addressed here is: *how correlated is social sensor data and traffic jam and how significant they are to the prediction?* To answer that, we correlated the check-ins occurring on streets with the average traffic jam. For this, we have used the NYC dataset. Figure 5.17 presents how correlated (Figure 5.17(a)) and significant (Figure 5.17(b)) they are.

Figure 5.17(a) shows that correlations vary in terms of time and space. Blue and red colors indicate correlation -1 and 1, respectively, while white cells indicate no correlation at all. Recall that data from social sensors, in some streets, are more correlated with jams in some periods. In certain periods and locations, they are not correlated (e.g., Street 75 at 0:00am and 3:00am). With this, we have a way of knowing when and where it is appropriate to use social sensor data.

For how significant social sensors are, we can look back in Figure 5.17(b), the probability values (p–value) for streets, i.e., if there exists a significant impact of the



(a) Correlation

(b) Significant

Figure 5.17: Analyzing the importance of social data with traffic conditions

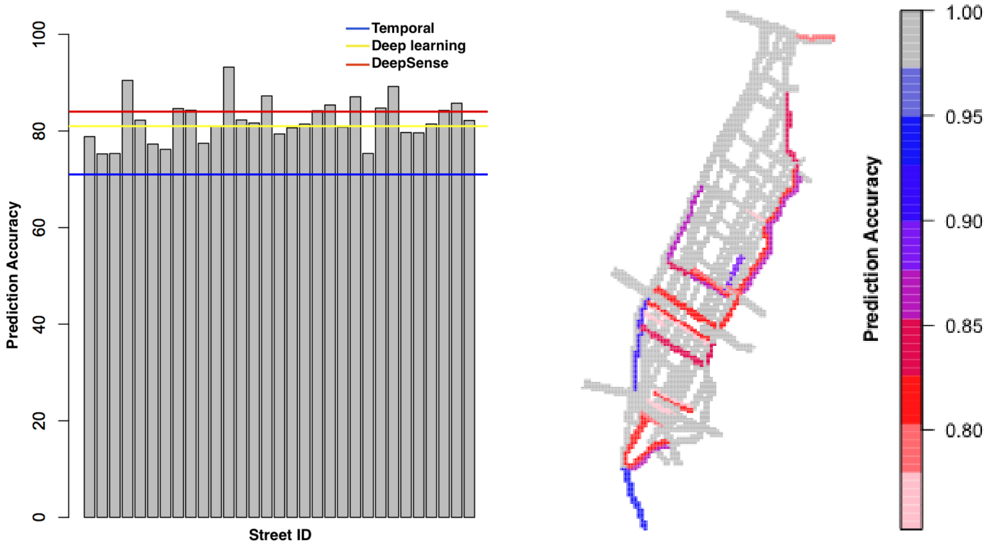
value used to test the correlation in the prediction output. Red color indicates more significant (p–value < 0.05) and blue no significant. Notice that each social sensor contributes differently spatially. For the highlighted streets, we can see that SS2 is more meaningful than SS1. Overall, SS1 contributes for 90.32% streets while SS2 contributes for 58.07% streets.

Q4. What is the performance of STRIP, considering additional social sensors as inputs, in comparison with baseline models? Is STRIP better in terms of higher accuracy, recall, specificity, and precision, and lower false positive rates?

The hypothesis of Q4 is that the STRIP accuracy will be better than the prediction model of [Kurihara, 2013; Niu et al., 2014; Tostes et al., 2013]. Here we discuss the STRIP performance in terms of accuracy, recall, false positive rates, specificity and precision. We have used 5 min as a historical data period. First, we have used the same dataset of last section (NYC dataset).

A) Comparison with baseline models. Figure 5.18 represents results for popular streets in our datasets, regarding the number of social sensor data and the occurrences of traffic jam. For comparison, we plotted the results of [Niu et al., 2014] as baselines for our prediction that are approximately: (i) 70% of accuracy in the temporal approach,

(ii) 81% of accuracy in deep learning and (iii) 84% in DeepSense. As we mentioned, the related work [Kurihara, 2013] has an accuracy of 86%, in average.



(a) Precision accuracy with social sensors (b) Spatial analysis for popular streets

Figure 5.18: Accuracy of the traffic jam forecasting

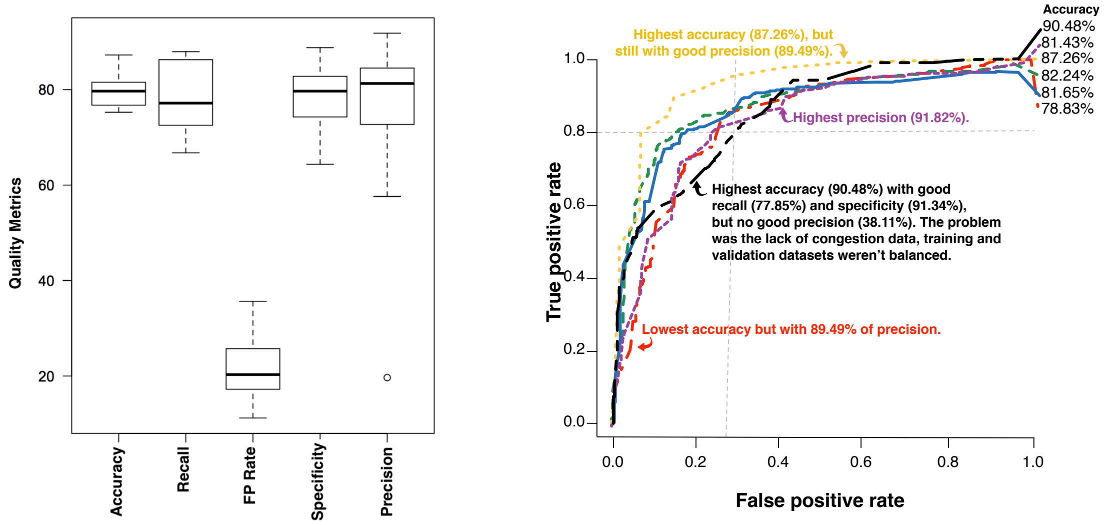
In Figure 5.18(a), we can observe that the STRIP accuracy is above the baseline for eight streets, achieving up to 90.49%, which is 7.71% higher than DeepSense, in average. In comparison with deep learning, our results are 11.7% better for 19 streets and 27.44% better than temporal for all streets. A spatial representation of the prediction accuracy of the most significant streets is depicted in Figure 5.18(b).

Furthermore, in comparison with our model proposed in [Tostes et al., 2013], we improved the results for some streets: from 80%, in average, up to 93.21% for some streets, which corresponds of an improvement of approximately 16% in accuracy. In comparison with the results of Table 5.3, STRIP is, in average: 3.89% better in accuracy, 5.25% in recall, 82.72% in FP (false positive) rate, 42.87% in specificity and 8.48% in precision, even considering a prediction for a higher time horizon (30 min).

B) Evaluation of quality metrics. Figure 5.19(a) shows the box plots for quality metrics. The thicker line indicates the average. Individual outliers are plotted as individual points (observe in Accuracy, FP rate and Specificity). Recall that the average accuracy is 82.21% (confidence interval of 80.57%–83.85%). Although having the accuracy up to 93.21% with 91.82% of specificity and precision, if we ensure a balanced dataset for training[2], the quality metrics are, overall, better (87%) and higher than

[2] Same amount of samples for each class in the prediction output, in average.

baseline models (84%) in average.



(a) Boxplot of the quality metrics in popular streets (with more social data and more traffic jam).

(b) Comparing the FP rate (sensitivity) to the false positive rate (complementarity of specificity).

Figure 5.19: Using the correlation to decide when and where to use social data.

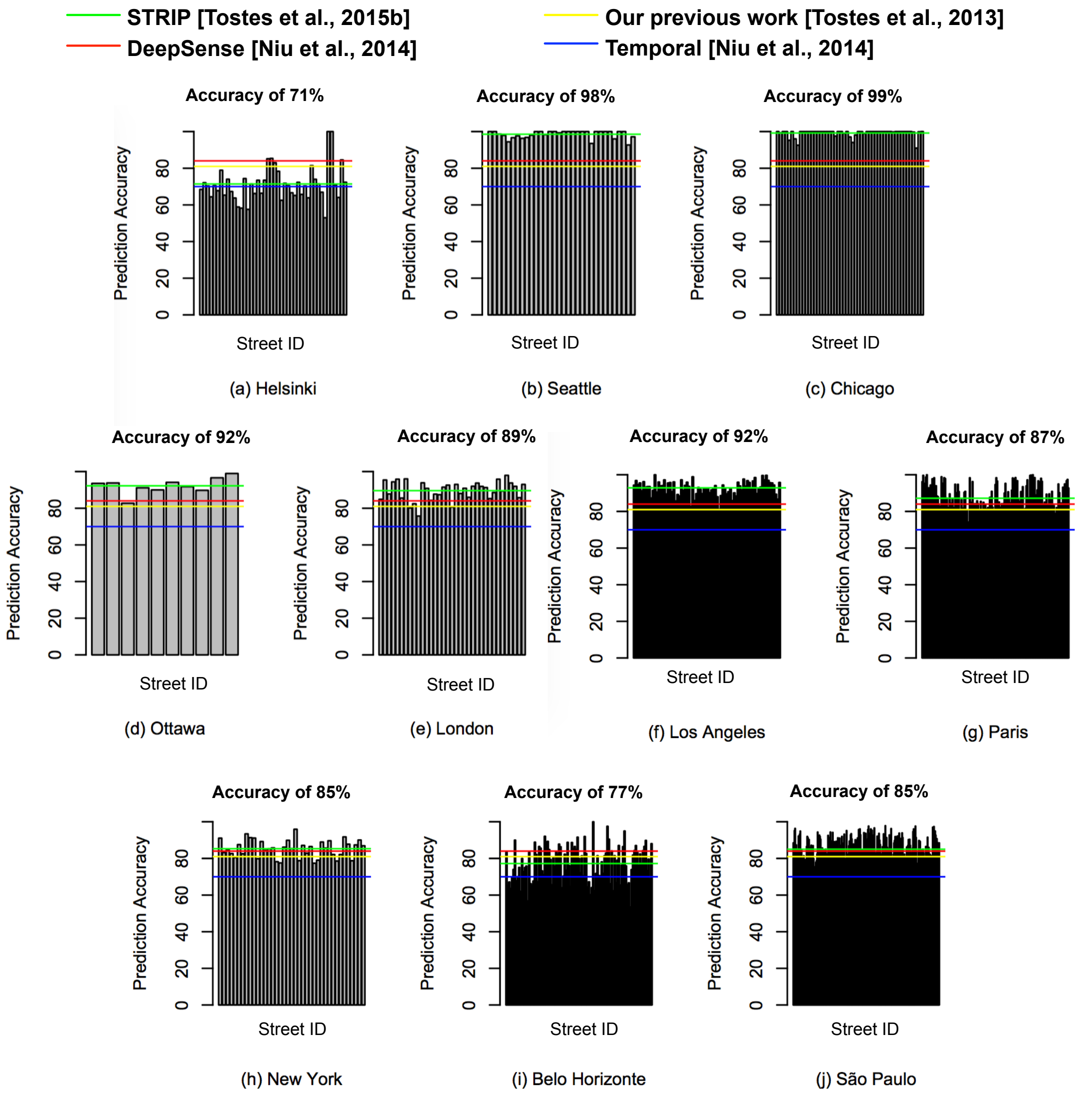
In addition, Figure 5.19(b) compares the sensitivity of the model with the FP rate (complement of specificity). Each line represents the result of STRIP V2 LR for one street. We can observe that the highest accuracy (87.26%) had still a good precision (up to 89.49%). The same was noticed to the highest precision (91.82%), also presenting a good accuracy (81.43%). Although we had streets with a lower accuracy (78.83%), the precision of prediction for such streets was high (89.49%), being still above our baseline (84%). For the accuracy of 90.48%, we had a good recall (77.85%) and specificity (91.34%), but no good precision (38.11%) due to the lack of congestion data in datasets (unbalanced outputs). Therefore, besides being better than baseline models, STRIP can improve its accuracy to NYC when considering PSN data.

Q5. What is the performance of the prediction model for different cities?

The hypothesis of Q5 is that different cities have different traffic patterns, culminating to different traffic prediction accuracies. Here we have used datasets from different cities using the same methodology as we described in Section 3.1.2. From these datasets, we considered the same as the NYC dataset: only weekdays in our training datasets, since weekdays and weekends have different traffic patterns.

Figure 5.20 shows the STRIP accuracy for the streets that have more traffic flow variation. We can observe the average STRIP accuracy as a green line, temporal and

DeepSense from [Niu et al., 2014] as blue and red lines, respectively, and our previous model [Tostes et al., 2013] as yellow.



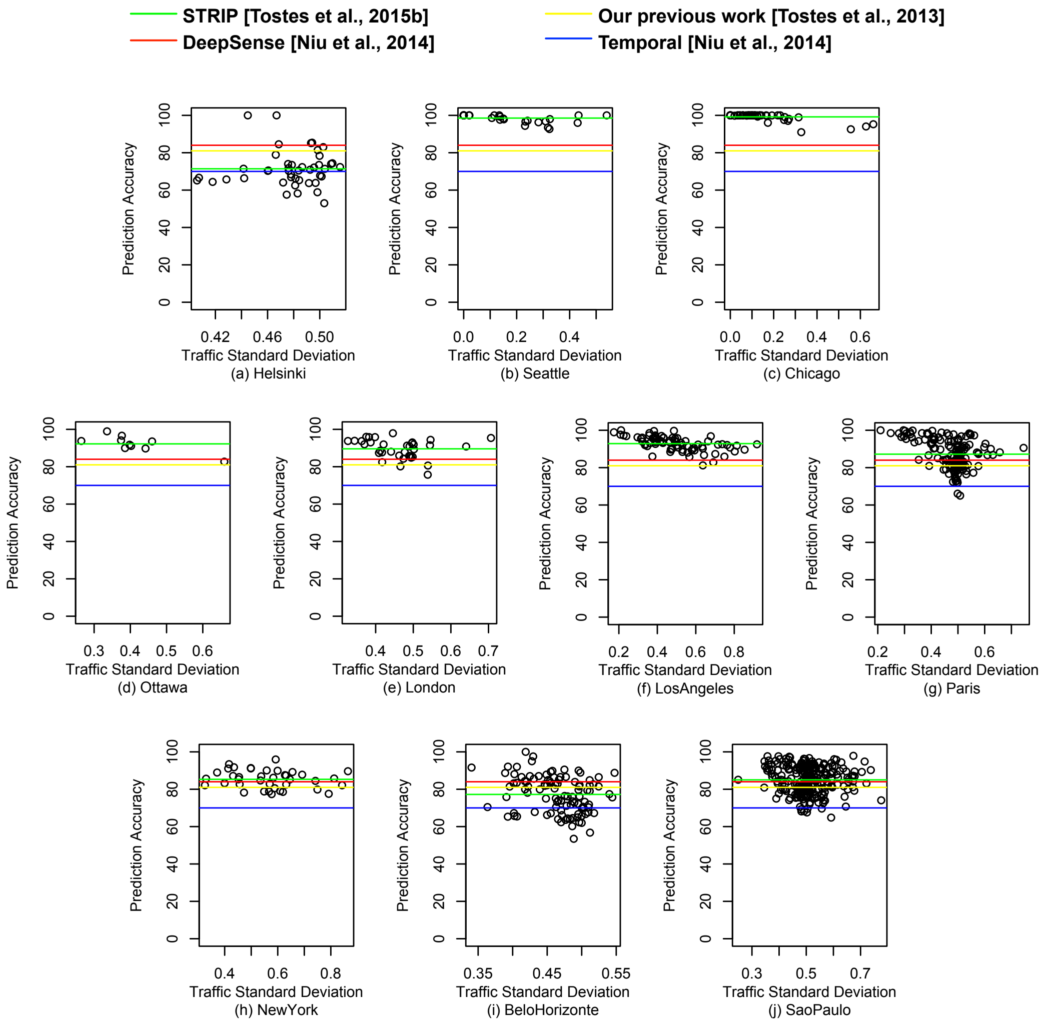
(a) Helsinki (b) Seattle (c) Chicago

(d) Ottawa (e) London (f) Los Angeles (g) Paris

(h) New York (i) Belo Horizonte (j) São Paulo

Figure 5.20: Prediction accuracy of STRIP for different cities, in comparison with the overall performance of baseline models.

For all cities, our results were better than the baseline models. For São Paulo and Paris, which are the most congested cities in our dataset, the accuracy was up to 85% and 87%, respectively. For New York, the accuracy was 85%. For London, the accuracy was 89%. For Los Angeles, the accuracy was 92%. For median cities, like Ottawa and Seattle, the accuracy was also very good, up to 98%, representing our best results. Therefore, STRIP has a good prediction accuracy (above 80%), in average, for both median and large cities. Therefore, this indicates that the size of the traffic



(a) Helsinki

(b) Seattle

(c) Chicago

(d) Ottawa

(e) London

(f) LosAngeles

(g) Paris

(h) NewYork

(i) BeloHorizonte

(j) SaoPaulo

Figure 5.21: Evolution of the STRIP accuracy in comparison with the standard deviation of the traffic conditions in the training dataset, considering different cities.

fleet does not affect STRIP performance. The important aspect is to keep the training dataset representative for each city.

In order to explain such results, Figure 5.21 shows a graph comparing the prediction accuracy with the standard deviation of traffic flow considering only the training dataset. We can observe that the prediction models with a lower accuracy, Helsinki and Belo Horizonte, present a standard variation between 0.35–0.55. The other datasets present a larger standard deviation interval (0.2–0.8). This indicates that the dataset is probably more representative than others are.

5.3 Discussion

As we have mentioned, there are several context data that can be crawled and characterized in a smart city, such as traffic flow, incidents, check-ins, and physical sensors (street detectors). In this work, we focused in traffic flow and check-ins, since they have already demonstrated a correlation between them.

Notice that future works may investigate the usage of other variables such as traffic incidents. In addition, it is important to recall that we have not used traffic incidents to predict the traffic jam, since its raw form can only explain why jams occurred, but in order to predict, we would have to also predict whether we would have a traffic incident or not. This complex topic is out of this thesis outline.

Therefore, we have proposed and evaluated the prediction models MoVDic and STRIP. But how can they be applied or be combined in vehicular networks? What can be possible applications for them? This is discussed in this section.

Figure 5.22 illustrates an example of how to use the proposed models, MoVDic, STRIP, and ALLuPIs, mentioned in the previous chapter. In this scenario, vehicles 5 (red – street s2) and 6 (blue – street s5) want to turn left, in street s4 that has a traffic jam. But when the vehicles 5 and 6 communicate with the infrastructure, our models enter in action. In this context, we can have the following applications, based mainly in the service of route suggestion:

- ALLuPIs, discussed in Chapter 4, knows that the next intersection has a better performance than others, since it can measure the response time and the utilization of passing through a crossroad (turning or moving forward).
- MoVDic knows that more vehicles turn left in this crossroad because of its prediction (by definition). Therefore, based with these two information, the infrastructure can send a new path to vehicle 5 in order to avoid the traffic jam.
- STRIP, in turn, could previously tell vehicles that street s4 will be congested in the next time horizon, changing the decision of the last route for vehicle 5.

Recall that all these processes can occur in real time, given a more precise information to the traffic jam minimization procedure. Also, for sure the performance of such application will depend on how routes will be suggested. Therefore, an important task is to investigate how can we better suggest routes while minimizing the traffic jam, i.e. better spreading the traffic flow in the city, and also to evaluate these proposals and their impact in the travel time, fuel consumed and CO_2 emissions. This is shown in the next chapter.

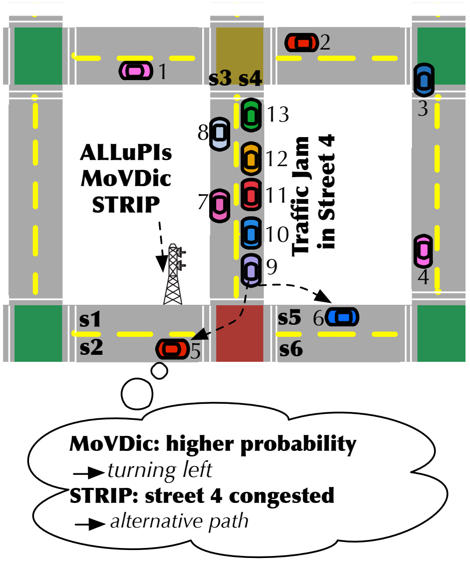


Figure 5.22: Example of how to use the proposed models.

5.4 Chapter Remarks

This chapter presented two prediction models: (i) MoVDic, a mobility vehicular prediction model; and (ii) STRIP, a short-term traffic jam forecasting model.

First, MoVDic has been proposed to predict how the traffic flow spreads in the scenario. The driving idea of this work was to propose a model to predict the mobility of vehicles in a city through Bayesian networks. The Bayesian network predicts the likelihood of a vehicle to follow a certain direction based on its current location and trajectory. The key idea is to predict the future locations of vehicles. That is, consider a vehicle that follows some path. When it arrives at an intersection, it can turn left, turn right or move forward. *How can we forecast the probability of vehicles that will turn left, turn right or move forward when they arrive at an intersection? How to recover such mobility in a highly dynamic environment as VANETs in real-time?* MoVDic calculates the probabilities of seeing a vehicle in a street segment, considering the observed data from detectors. The Metropolis-Hastings algorithm has been used considering the Dirichlet as the prior distribution.

Two sub-scenarios have been used to validate our model, a smaller one with 125 streets and a bigger one with 2216 streets. Results have shown that the values observed from detectors and the mobility of vehicles are correlated. The higher the number of vehicles on the street at an instant of time, the higher will be the probability of seeing a vehicle in that street. We have also evaluated the distance between the real mobility of vehicles and the obtained from the prediction model through distinct metrics: (i) hits to predict the street with more turnings; (ii) mean prediction error considering several crossroad categories. We have also evaluated how many crossroads are valid and accurate to the model, that is, having a minimum of vehicles in the crossroad and predicting correctly to street with more turning choices. Comparing both scenarios, results have demonstrated that MoVDic is accurate for up to 74% in S125 and 80% in S2216, in average, for up to 85% of the crossroads in S125 and 24% in S2216. This indicates that our model is quite accurate for valid crossroads. The mean error was lower than 0.002 for both scenarios, being lower to S2216 since it presented less vehicles than S125 (130 vehicles against 500 vehicles).

We have also vary the minimum number of vehicles in the crossroads, based on the density of vehicles in the scenario. Recall that MoVDic performs well for crossroads with 2, 3 and 4 exit-streets, predicting correctly the crossroads with more turning choices more than 82% of crossroads. It has also achieved up to 100% in some cases, when we have a higher number of vehicles (150 vehicles). Since the amount of valid crossroads with 2 and 3 exit-streets is higher than with 4 and 5 exit-streets, then the result for 2 and 3 exit-streets is more representative than for 4 and 5 exit-streets.

Second, we have proposed a short-term traffic jam prediction, named STRIP. Its has a distributed architecture, with one logistic regression per street. We forecasted traffic jam for a near future (30 min). We gave a proof of concept that check-ins can be used as social sensors to improve the accuracy of prediction models. Then, the correlation between check-ins and traffic jam has been following a methodology that allows us to indicate when and where such sensors are significant to STRIP.

According with our results, the social sensors are very significant for 90.32% of streets with SS1 and 58.07% streets with SS2. As proved, social sensors can improve the prediction quality (from 80% up to 90%, in general). We have also evaluated the performance of STRIP in comparison with baseline approaches, including our previous model [Tostes et al., 2013]. Results have shown that the accuracy of STRIP in predicting the next 30 min-traffic jam increased up to 90% for some streets, being above the baseline (84% for [Niu et al., 2014] and 86% for [Kurihara, 2013]).

Therefore, we have explored the proposal and the evaluation of two prediction models: MoVDic and STRIP. Notice that we can use both models to predict a road contention and to start a contingency plan. Protocols can initiate a congestion algorithm while quantifying the contention level. Other studies can use the information of crossroad performance to apply it in a routing protocol while searching for the next hop. Also, we can use MoVDic to design a new mobility model for any city. In dissemination and routing protocols, we can use MoVDic to inform future densities instead of using the real-time densities. We can also investigate new approaches to use the forecasted paths. About STRIP, future works can investigate its performance in other cities and with new sensors (physical, virtual or social).

Chapter 6

Services

In this chapter, we discuss one specific service approach to prevent and to control traffic jam: the load balance of traffic flow. Recall that, besides the load balance of flow, we have also made collaborations in other services, such as protocols for congestion detection and control [Araújo et al., 2014; Araújo et al., 2014; de Brito et al., 2015, 2014b], and data dissemination [de Castro et al., 2013; Soares et al., 2014b].

Related work have different focus in the context of services to deal with traffic jam. Some focus on ways in which VANET could be used to support algorithms which reduce emissions and fuel consumption (e.g. [Doolan and Muntean, 2013] and [Füßler et al., 2005]). Others propose novel communication protocols in terms of how to identify, minimize and disseminate the information of traffic congestion, after it happens (e.g. [Araujo et al., 2014; Soares et al., 2014a]). There are also proposals of traffic jam forecasting (e.g. [Jain and Sethi, 2012; Kurihara, 2013; Manasseh and Sengupta, 2013]) and of novel navigation algorithms, deciding how to determine quicker and more fuel-efficient routes for vehicles (e.g. [Chen et al., 2011; Sommer et al., 2010]). Instead, we study ways to suggest routes while mitigating the traffic congestion.

Here we assign routes to vehicles according with the graph modeling that minimizes the traffic congestion (a mitigation process), considering also the roads capacity and flow. Suppose that at least one crossroad has a static node representing the RSU. When a vehicle commutes from a source to a destination, it requests a path to the nearby RSU. Recall that we consider both source and destination as crossroads. As a solution, we use the modeling of mode (iii) with Graph Theory, as the union of two graphs: the graph of mode (i), defined as $G_1 = (V_1, E_1)$ (vehicles); and the graph of mode (ii), defined as $G_2 = (V_2, E_2)$ (RSU). The edges of this graph G $= G_1 \cup G_2$ are $E_t = E_1 \cup E_2$, which enables the communication between vehicles, infrastructures and between the vehicle with the infrastructure. Therefore, the RSU determines the best route between all possible routes, which are the k–shortest-paths algorithm in G_2.

The traffic congestion was modeled as the Multicommodity Flow Problem (MFP). We use a non-directional graph $G = (V, E)$ and a set of source (s_i) and destination (t_i) pairs defined as $\{(s_i, t_i) : 1 \leq i \leq k\}$ [Raghavan and Tompson, 1987]. The set of vertexes V consists in the definition of several possible sources. For each source s_i, there is a path to its respective destination t_i through the set of edges E of the graph G. Each edge $e \in E$ has a capacity $c(e)$, which is the superior bound for the total flow capacity in e.

We map the problem of congestion minimization as the following graph problem: assign disjoint paths from multiple sources to multiple destinations. From the set $\wp = P_i : 1 \leq i \leq k$ such that P_i is the path from s_i to t_i in G, the MFP problem is how to find a route for each pair of source/destination in $\wp$ that minimizes the number of paths that use same edges. Since this problem is NP–hard [Even et al., 1975], we use heuristic to find solutions closer to the optimal in a polynomial time [Karp, 1972].

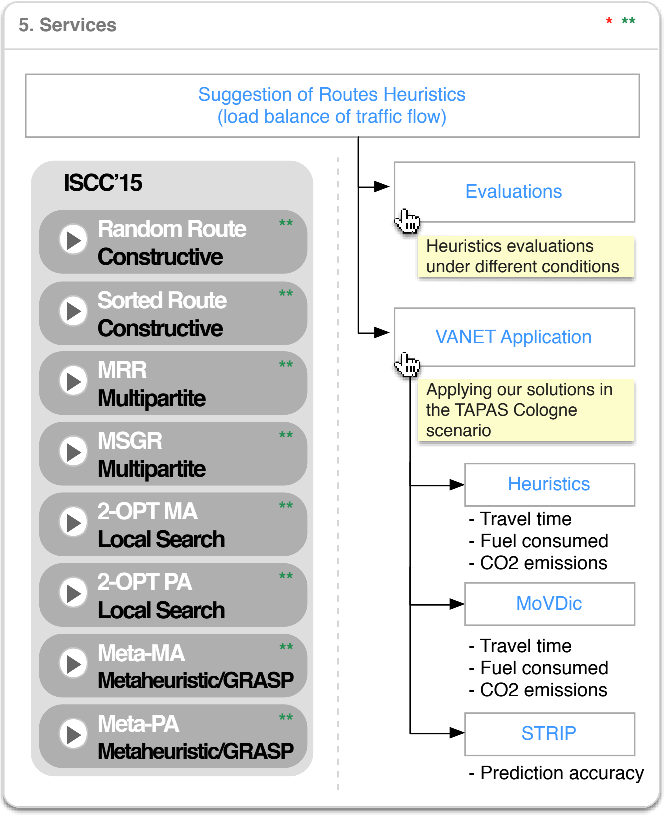

5. Services
* **
Suggestion of Routes Heuristics
(load balance of traffic flow)
ISCC'15
Random Route Constructive **
Sorted Route Constructive **
MRR Multipartite **
MSGR Multipartite **
2-OPT MA Local Search **
2-OPT PA Local Search **
Meta-MA Metaheuristic/GRASP **
Meta-PA Metaheuristic/GRASP **
Evaluations
Heuristics evaluations under different conditions
VANET Application
Applying our solutions in the TAPAS Cologne scenario
Heuristics
- Travel time
- Fuel consumed
- CO2 emissions
MoVDic
- Travel time
- Fuel consumed
- CO2 emissions
STRIP
- Prediction accuracy

Figure 6.1: Description of the Services chapter.

Figure 6.1 summarizes this chapter goals. First, we propose the following heuristics, with several levels of complexity [Tostes et al., 2015]: (i) two constructive heuristics; (ii) two multipartite heuristics; (iii) two local search heuristics; and (iv) two metaheuristics. Second, we have evaluated all heuristics in terms of time and cost, varying the scenario size and requested routes. Third, we have evaluated a realistic scenario based on a trace from vehicular networks in terms of travel time, CO_2 emissions and fuel consumed. Three strategies have been used: (i) with our proposed heuristics; (ii) with MoVDic prediction; and (iii) with STRIP prediction. Results have shown that, with our proposed heuristics, we can have a decrease of up to 56% in travel time, and up to 18% in both CO_2 emissions and fuel consumed, in average, against 33% in travel time and up to 11% in CO_2 emissions and fuel consumed when using MoVDic prediction. Also, the accuracy of STRIP was more than 95%, in average.

The rest of this chapter is organized in five sections, as it follows. Section 6.1 explain the heuristics proposed. Section 6.2 describes the evaluations that have been made in our heuristics. Section 6.3 presents the developed applications based on our heuristics and both of our prediction models, MoVDic and STRIP. In Section 6.4, we summarize our results with a brief discussion. Finally, this chapter remarks is presented in Section 6.5.

6.1 Suggestion of Routes Heuristics

In this section, we explain the heuristics proposed as solution to MFP in vehicular networks.

6.1.1 Constructive Heuristics

Here we propose two constructive heuristics: Random Routes and Sorted Routes. These heuristics assign a route for every vehicle, considering three parameters: (i) a road graph; (ii) a capacity matrix; and (iii) a list of source/destination pairs. They differ in the selection of which vehicle will be firstly assigned with a route. Random Routes selects randomly the vehicle that will be assigned with a route, while in Sorted Routes we assign a route to the farthest vehicle first.

The first proposed heuristic is called Random Routes, described by Algorithm 9. In Line 3, we randomly remove vehicles from the list of routes, until the list becomes empty. At each iteration, in Lines 4 and 5, we calculate and assign a new route to the selected vehicle, by using the shortest–path algorithm of [Dijkstra, 1959]. For each assigned route, in Line 6, we update the weights of edges according with the $\phi(x)$ function (see Equation 6.1), proposed in [Buriol et al., 2003] based on the weight/capacity relation (l_a/c_a) in the capacity matrix.

Algorithm 9 Random Routes

```
Require: Graph $G = (V, E)$, list $R$ of $n$ pairs of source/destination (required routes).
Ensure: Array of suggested routes to vehicles.
Create an array $R^{ret}$ of $n$ lists of vertexes to store the routes suggested.
for each vehicle $r \in R$ randomly chosen do
    Remove the vehicle $r$ from the list $R$
    Assign to the vehicle $r$ the route $e$, calculated by the Dijkstra algorithm.
    Add the route $e$ to the array $R^{ret}$ in the index of $r$.
    Update the weights of the edges of the route $e$ according with the $\phi$ function.
end for
return $R^{ret}$
```

The second heuristic is called Sorted Routes, described in Algorithm 10. First, in Lines 3-6, we assign a route for each vehicle through the algorithm of [Dijkstra, 1959]. Second, in Line 7, we sort the list of routes in a decreasing order according with the size of the routes. Third, in Lines 8-13, for each vehicle in the sorted list, we assign a new route calculated by [Dijkstra, 1959] and we update the weights of the edges by using the $\phi(x)$ function.

$$\phi(x) = \begin{cases} 1, & \text{se } 0 \leq l_a/c_a < 1/3 \\ 3, & \text{se } 1/3 \leq l_a/c_a < 2/3 \\ 10, & \text{se } 2/3 \leq l_a/c_a < 9/10 \\ 70, & \text{se } 9/10 \leq l_a/c_a < 1 \\ 500, & \text{se } 1 \leq l_a/c_a < 11/10 \\ 5000, & \text{se } 11/10 \leq l_a/c_a < \infty \end{cases} \tag{6.1}$$

Algorithm 10 Sorted Routes

```
Require: Graph $G = (V, E)$, list $R$ of $n$ pairs of source/destination (required routes).
Ensure: Array of suggested routes to vehicles.
 1: Create an array $R^{ret}$ of $n$ lists of vertexes to store the routes suggested.
 2: Create an array $D$ with the size of the $n$ requested routes.
 3: for each vehicle $r \in R$ do
 4:     Assign a route $aux$ to the vehicle $r$, calculated by the Dijkstra algorithm.
 5:     Store the size of the route $aux$ in the array $D[r]$.
 6: end for
 7: Sort the list $R$ in descending order according with the size of the routes in the array $D$.
 8: for each vehicle $r \in R$ do
 9:     Remove the vehicle $r$ from the list $R$
10:     Assign the route $e$ to the vehicle $r$, calculated by the Dijkstra algorithm.
11:     Add the route $e$ to the array $R^{ret}$ in the index of $r$.
12:     Update the weights of the edges of the route $e$ according with the $\phi$ function.
13: end for
14: return $R^{ret}$
```

6.1.2 Multipartite Heuristics

One technique that improves the quality of the constructive heuristic solutions is the multipartite approach. The idea is to perform several times a constructive heuristic with a perturbation in the input data. Accordingly, at each execution, we have a different solution with different costs. What is important is to randomize the solution, producing several options among which we choose the best. This section describes two multipartite heuristics, based on constructive heuristics.

First, we propose the multipartite heuristic known as Multipartite Random Routes (MRR), presented in Algorithm 11. At each execution of Random Routes in Line 3, we find a new solution based on the random choice of the vehicle that will be assigned with a route, using an uniform distribution. This means that all vehicles have equal probability of being selected, regardless of the distance to the destination. In Line 4, we keep the best solution amount the solutions proposed.

Algorithm 11 Multipartite Random Routes

```
Require: Graph $G = (V, E)$, list $R$ of $n$ pairs of source/destination (required routes), the
    number of iterations $t$.
Ensure: Array of suggested routes to vehicles.
1: Create an array $R^{ret}$ of $n$ lists of vertexes to store the routes suggested.
2: for $i = 1 \rightarrow t$ do
3:     $R^{ret}$ = RandomRoutes($G, R$)
4:     Update $R^{ret}$ if this is the best solution that has been found.
5: end for
6: return $R^{ret}$
```

Algorithm 12 Multipartite Gaussian Sorted Routes

```
Require: Graph $G = (V, E)$, list $R$ of $n$ pairs of source/destination (required routes), the
    number of iterations $t$.
Ensure: Array of suggested routes to vehicles.
1: Create an array $R^{ret}$ of $n$ lists of vertexes to store the routes suggested.
2: for $i = 1 \rightarrow t$ do
3:     $R^{ret}$ = GaussianSortedRoutes($G, R$)
4:     Update $R^{ret}$ if this is the best solution that has been found.
5: end for
6: return $R^{ret}$
```

We also propose the multipartite heuristic known as Multipartite Gaussian Sorted Routes (MGSR), presented in Algorithm 12. The idea is to choose first the most requested routes. In this case, we privilege the routes with average size instead of short and long routes. The advantage is the flexibility, because we can change the probability distribution if there are more long or short routes. MGSR is based on a

Algorithm 13 Gaussian Sorted Routes

```
Require: Graph $G = (V, E)$, list $R$ of $n$ pairs of source/destination (required routes).
Ensure: Array of suggested routes to vehicles.
 1: Create an array $R^{ret}$ of $n$ lists of vertexes to store the routes suggested.
 2: Create an array $D$ with the size of the $n$ requested routes.
 3: Create two doubles: $\mu$ and $\sigma$.
 4: for each vehicle $r \in R$ do
 5:     Assign the route $aux$ to the vehicle $r$, calculated by the Dijkstra algorithm.
 6:     Store the size $k$ of the route $aux$ in the array $D[r]$.
 7:     Calculate $\mu = \mu + k$
 8: end for
 9: Calculate $\mu = \mu/n$
10: Sort the list $R$ in descending order according with the size of routes in the array $D$.
11: Calculate $\sigma$ by Equation 6.1.
12: for $R$ is not empty do
13:     Randomly choose a vehicle $r$ through the Gaussian distribution $(\mu, \sigma)$ and remove this
        vehicle from the list $R$.
14:     Assign the route $e$ to the vehicle $r$, calculated by the Dijkstra algorithm.
15:     Add the route $e$ to the array $R^{ret}$ in the index of $r$.
16:     Update the weights of the edges of the route $e$ according with the $\phi$ function.
17: end for
18: return $R^{ret}$
```

version of Sorted Routes, the Gaussian Sorted Routes described in Algorithm 13. We have inserted a perturbation in the choice of the vehicle made by Sorted Routes. We highlight the differences in Lines 3, 7, 9, 11 and 13. The selection of a vehicle occur based on a Gaussian distribution (normal) with mean μ and a standard deviation σ that are calculated based on the distance from sources to destinations.

6.1.3 Local Search Heuristics

Despite multipartite heuristics have better results than constructive heuristics, they still lack the ability to improve the result. Additionally, a good solution cannot be guarantee due to the randomness of the results. The idea of local search heuristics is to improve the solution of a constructive heuristic or multipartite, to achieve optimal local (or global) solution.

In this section, we propose two local search heuristics: (i) best improvement (2–OPT–BI), described in Algorithm 14; and (ii) first improvement (2–OPT–FI), presented in Algorithm 15. The difference between them is in Line 4. While 2–OPT–BI moves to the best solution among the neighbors, 2–OPT–FI moves the solution that gives the first improvement.

The local search algorithm starts through a multipartite heuristic, also known

Algorithm 14 2–OPT–BI

```
Require: Graph $G = (V, E)$, list $R$ of $n$ pairs of source/destination (required routes), the
    number of iterations $t$.
Ensure: Array of suggested routes to vehicles.
Create an array $R^{ret}$ of $n$ lists of vertexes to store the routes suggested.
while there is improvement in the solution do
    $R^{ret}$ =MultipartiteRandomRoutes $(G, R)$
    for each one of the $k$ neighbors
 do
        Randomly choose an edge.
        Save the weight of this chosen edge.
        Set the weight of this chosen edge to infinite.
        Randomly choose a route, saving its source $s$ and its destination $t$
        Find the shortest path from $s \rightarrow t$ using the Dijkstra algorithm.
        Set this new path as the vehicle route
        Calculate the cost of this new solution $S^*$
        Set the weight of the chosen edge to its original weight
        if cost($S^*$) $\leq$ cost($R^{ret}$) then
            $R^{ret} = S^*$
        end if
    end for
end while
return $R^{ret}$
```

Algorithm 15 2–OPT–FI

```
Require: Graph $G = (V, E)$, list $R$ of $n$ pairs of source/destination (required routes), the
    number of iterations $t$.
Ensure: Array of suggested routes to vehicles.
Create an array $R^{ret}$ of $n$ lists of vertexes to store the routes suggested.
while there is improvement in the solution do
    $R^{ret}$ =MultipartiteRandomRoutes $(G, R)$
    while there is no neighbor with a better solution than cost($R^{ret}$) do
        Randomly choose an edge.
        Save the weight of this chosen edge.
        Set the weight of this chosen edge to infinite.
        Randomly choose a route, saving its source $s$ and its destination $t$
        Find the shortest path from $s \rightarrow t$ using the Dijkstra algorithm.
        Set this new path as the vehicle route
        Calculate the cost of this new solution $S^*$
        Set the weight of the chosen edge to its original weight
        if cost($S^*$) $\leq$ cost($R^{ret}$) then
            $R^{ret} = S^*$
        end if
    end while
end while
return $R^{ret}$
```

as randomized constructive heuristic (Line 3). The neighbor is defined as the set of solutions that can be achieved by removing and edge (from the solution) by another route, calculated through the shortest path algorithm of [Dijkstra, 1959]. For each iteration the algorithm tries to change one edge from a route by other, evaluating this exchange in terms of the quality of the solution (cost, which, in this case, we measure by the number of repeated edges in different routes). This happens in Lines 6–10 in Algorithm 14 and in Lines 5–9 in Algorithm 15. The heuristic searches for a better solution (Lines 11–15 in Algorithm 14 and Lines 10–14 in Algorithm 15), based on an initial solution. The algorithm is executed while having a better solution (Line 2), analyzing all the neighbors.

6.1.4 Metaheuristics

Here we propose two metaheuristics of reactive Greedy Randomized Adaptive Search Procedure (GRASP) [Feo and Resende, 1995], each with two phases: (i) create a viable solution; (ii) apply a local search. The idea is to improve an initial solution by using the local search to achieve an optimal solution (local or global). Since we calculate the initial solution by a multipartite, we generate several local optimums. At the end, we return the best solution among all the solutions found. In order to control the diversification of solutions and the intensification of improved solution, we use the parameter of randomness α, which is self-adjusting according with the previously found solutions.

Algorithm 16 Metaheuristic of Reactive GRASP

```
Require: Graph $G = (V, E)$, list $R$ of $n$ pairs of source/destination (required routes), the
    number of iterations $t$.
Ensure: Array of suggested routes to vehicles.
 1: Create an array $R^{ret}$ of $n$ lists of vertexes to store the routes suggested..
 2: for each $i$ of MAX(Iterations) do
 3:     $R^{ret}$ = MultipartiteRandomRoutes$(G, R)$
 4:     Run local search approach in $R^{ret}$
 5:     Update $\alpha$ every $k$ iterations.
 6:     if $custo_i \leq$ lowest cost $+\alpha *$ ( highest cost $-$ lowest cost ) then
 7:         Save the solution and its cost
 8:     end if
 9: end for
10: return Best solution found in the procedure
```

Algorithm 16 describes the metaheuristics Meta–BI and Meta–FI. The difference is in the utilization of the two local search algorithms proposed in previously, respectively 2–OPT–BI and 2–OPT–FI. Line 5 shows the automatic update in the parameter of randomness (α), which is initialized with value of 1, and is decreasing by 0.1 every k iterations. When α reaches zero, we increase until it reaches 1, also every k iterations.

6.2 Evaluations

We have defined a metric of cost for the heuristics. In this work, the cost means the number of edges in more than one route. That is, for each route assigned, if two routes have the same edge, than we increase in one unit in the cost of the solution (per every edge). A heuristic has cost zero when there is only routes with disjoints of edges. Then, we evaluate the heuristics in terms of time and cost. For the time, we have made an asymptotic evaluation. For the heuristic cost, we have varied the size of requested routes and the size of the scenario. Finally, to have an overview of all pairs of sources/destination, before starting a route, vehicles request routes to the nearest RSU, which responds with a set of routes, calculated by the heuristics proposed. At each time interval, the RSU has a set of requests and initiates the proposed heuristic.

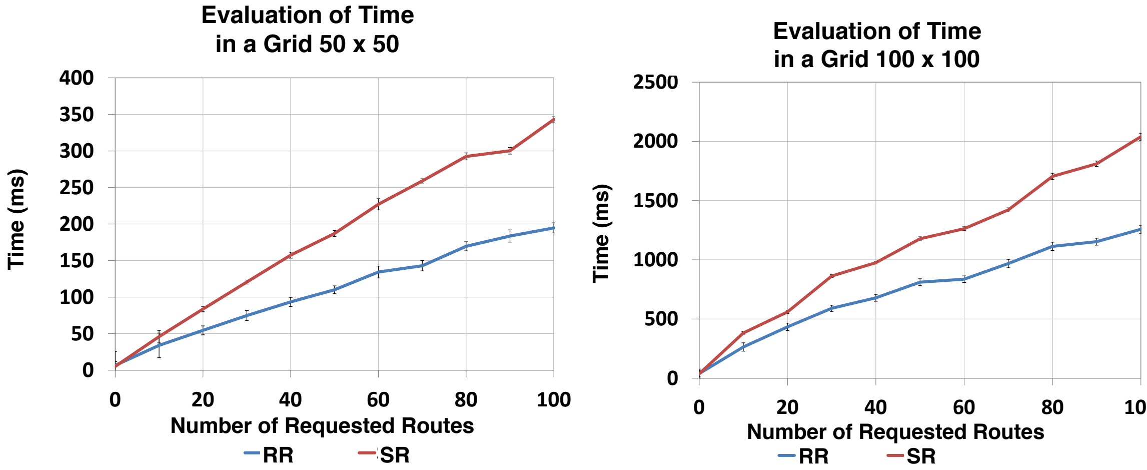
6.2.1 Asymptotic Evaluation

According with the number of requested routes, we made an asymptotic evaluation of the constructive heuristics proposed (Random Routes – RR and Sorted Routes – SR). Consider a graph $G = (V, E)$, with v vertexes (crossroads) and e edges (streets), representing the road network where we request k routes. Based on the algorithm description, one can notice that RR has complexity of $O(k \times (e + v \log v + e)) = O(k(e + v \log v)) = O(ke + kv \log v)$. For each k routes, the shortest path algorithm is performed ($O(e + v \log v)$) and the weights of all edges are updated ($O(e)$). On the other hand, SR generates k shortest paths and sort them by the size of the route ($O((v-1)\log(v-1))$[1]). We generate the final routes while we update the weights of edges through the ϕ function, at each iteration ($O(ke + kv \log v + v \log v)$ – see Equation 6.2). Therefore, RR has an asymptotic upper bound lower than SR.

$$\begin{aligned} O(k \times (e + v \log v) + (v-1)\log(v-1) + \\ k \times (e + v \log v) + e) = \\ O(ke + kv \log v + v \log v + ke + kv \log v) + ke) = \\ O(ke + kv \log v + v \log v) \end{aligned} \tag{6.2}$$

[1] In the worst case, each route will go through $v - 1$ vertexes.

In addition, we made an evaluation of time (in milliseconds)[2], considering the following: (i) time to create the graph; (ii) time to create the array of routes; (iii) calculation of routes by the heuristic that is being evaluated; and (iv) retrieval of the heuristic cost. These computations were executed for both RR and SR. We have used the grid scenarios of 50 × 50 and 100 × 100 vertexes. Figure 6.2 illustrates the results. It can be seen that the time spent by RR is lower than the time spent by SR as far as the number of routes increases. This was expected due to the asymptotic upper bound.



(a) 50 × 50

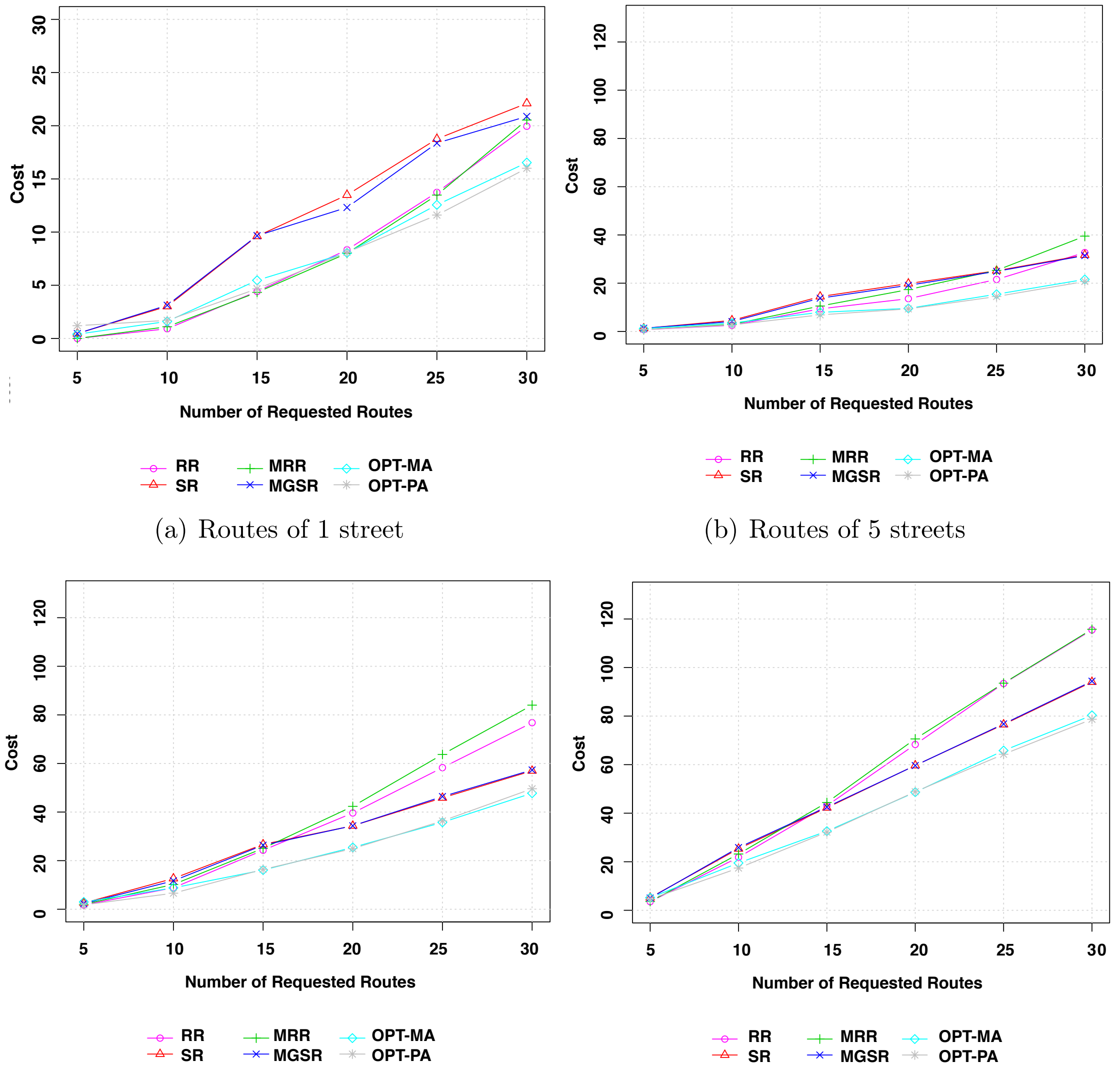
(b) 100 × 100

Figure 6.2: Time evaluation results for Random Routes and Sorted Routes.

6.2.2 Evaluation by the Size of Requested Routes

Here we varied the size of the requested routes to evaluate our heuristics. We have two categories of assessment: (i) requests of short routes; and (ii) requests of long routes. We consider a route as short if it has two or less edges, and as long if a commuter can travel from one corner of the grid to another. For this analysis, we have used a grid of 4 × 5. Figure 6.3 presents the results. We can observe that the higher is the size of the requested route, the higher is the cost of the heuristic since it is harder to balance the load of vehicles. Independent of the scenario, the best approach were OPT–PA and OPT–MA. When there are more requested routes, SR is better than RR, while RR is better when we request a small number of routes (5 up to 15). Notice that 15 requested routes is the inflection point, that is, over this point, an additional requisition provokes more impact on the cost than what it was.

[2] The machine we use was completely dedicated to experiments, with the following configuration: Mac OS X version 10.6.8, with Intel Core 2 Duo of 2.4GHz and 4GB of RAM DDR3.



(a) Routes of 1 street

(b) Routes of 5 streets

(c) Routes of 10 streets

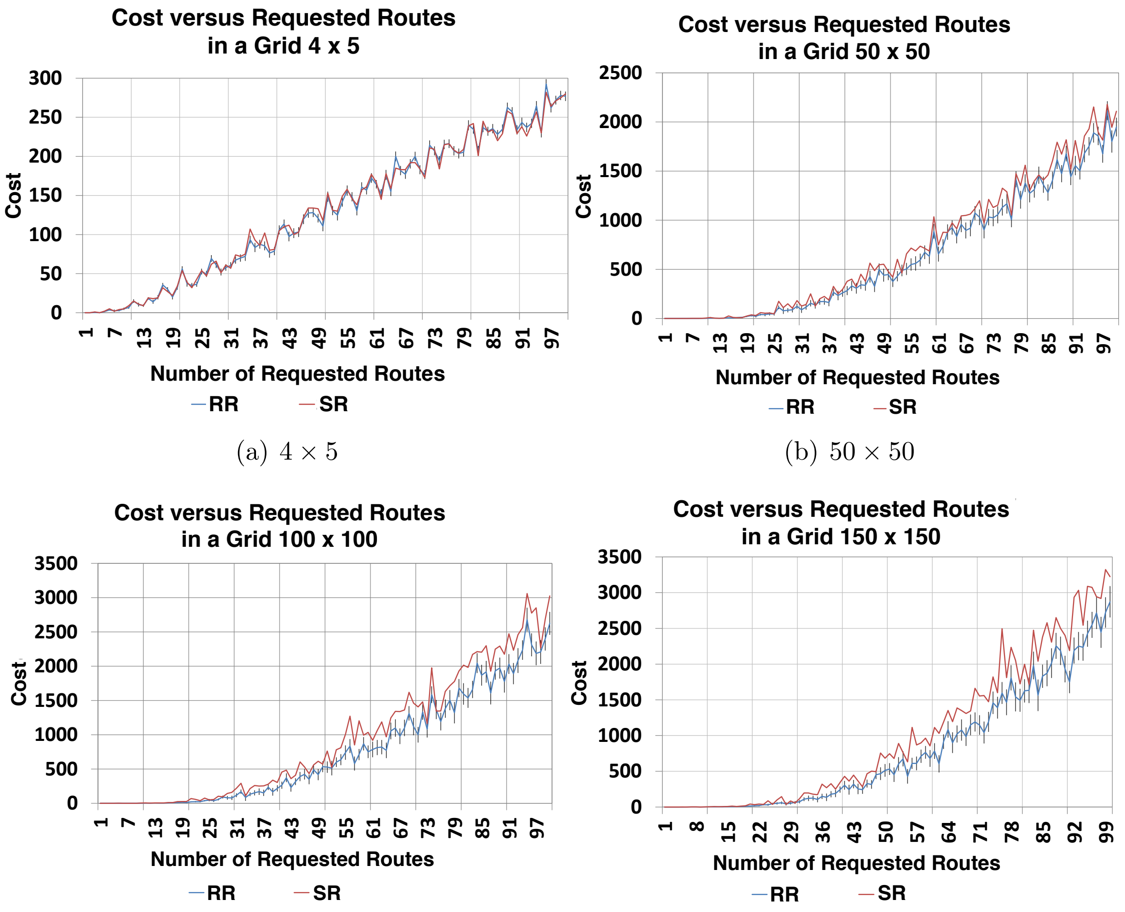
(d) Routes of 20 streets

Figure 6.3: Cost of heuristics in terms of the size of requested routes.

6.2.3 Evaluation by the Request of Random Routes

To evaluate heuristics sensitivity to the number of routes requested we have generated randomly n sources and n destinations in scenarios of increasing size. The number of routes varied from 1 to 100 routes in a grid 4×5, 50×50, 100×100 and 150×150. For Random Routes, we evaluate the average of 100 performances.

Figure 6.4 has the results for the constructive heuristics. Notice that both have similar behavior for the small scenario (grid of 4×5 – Figure 6.4(a)), but, for bigger scenarios (Figures 6.4(b), 6.4(c) and 6.4(d)), Random Routes has a better performance than Sorted Routes in terms of cost. This can be explained by the edges congestion caused by Sorted Routes at each iteration (each new route), which consequently increases the solution cost. When the selection of vehicles is random, we assign distinct routes that reduces the cost, in average.



(a) 4 × 5

(b) 50 × 50

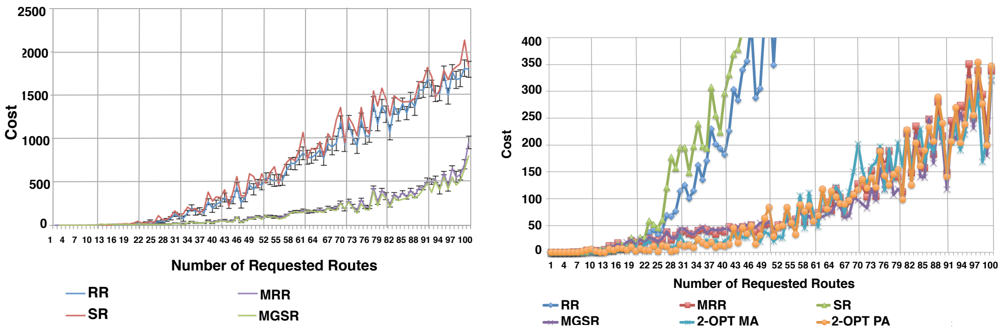
(c) 100 × 100

(d) 150 × 150

Figure 6.4: Sensibility to the size of routes for constructive heuristics.

Figure 6.5 illustrates a comparison between the constructive, multipartite and local search heuristics. Through Figure 6.5(a), we can observe that, as expected, for the grid of 50 × 50, the multipartite approaches have better solutions, closer to the optimal, than the constructive heuristics. The higher is the number of routes requested, the higher is the cost of the solution. Besides, recall that there is almost no distinction in the performance of the multipartite heuristics, i.e., they are equivalents. For grids bigger than 50 × 50, the results were the same, and we omit them due to space issues.

In addition, Figure 6.5(b) shows the results of the local search heuristics in a bigger grid, of 100 × 100. Similarly, local search solutions have similar behavior to the multipartite heuristics, presenting small peaks of improvements for some number of routes requested. This is a indicative that we did not choose the neighbors properly. For instance, when we requested 100 routes, the heuristic 2–OPT–BI had a higher cost



(a) Multipartite results in the 50×50 grid. (b) Local search results in the 100×100 grid.

Figure 6.5: Sensibility to the size of routes for multipartite and local search.

than the heuristic 2–OPT–FI, which is an evidence of multiple optimal solutions in the space of MFP solutions.

6.2.4 Evaluation by the Request of Predefined-size Routes

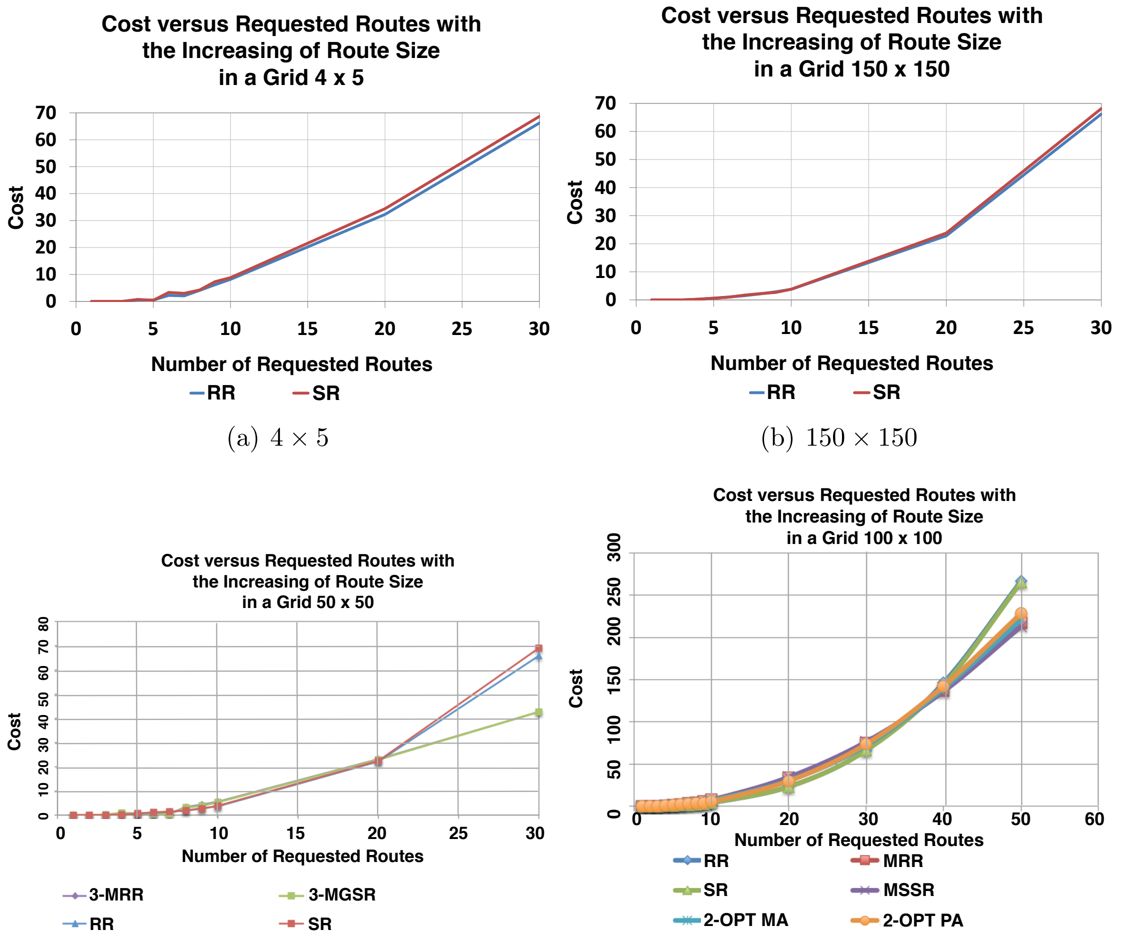
The heuristics proposed regarding to request of routes with sizes predefined increases gradually until it achieves the maximum size of a route in the graph (30 routes).

As Figure 6.6 shows, there is no clear difference in cost between heuristics when we increase the number of routes in the scenario. RR has lower cost than SR. For both, the cost increases as the number and the size of routes is incremented.

According with the graph of Figure 6.6(c), both multipartite heuristics have the same performance in terms of cost. Although, they are only better than the constructive heuristics when we request more than 20 long routes (size 20). Therefore, this means that there is no different when we request short routes. When we observe Figure 6.6(d), which indicates that most heuristics had the same cost, we can see that Sorted Routes has slightly higher cost after requesting up to 40 routes.

Since some heuristics had similar behavior, we have also evaluated the convergence time to a solution closer to the optimal. In the grid of 150×150, we request 100 routes from different sources to different destinations, randomly selected. Through the graphs of Figure 6.7, we can observe that MRR has a better performance than MGSR, since MRR converges faster to the optimal solution ($Z^* = 108$) while MGSR took substantially more time to outstrip the cost of 117.

As for the local search heuristics, we present the cost to achieve an optimal solution (evolution of the heuristic's cost) in Figure 6.7(b). There is a significant improvement comparing 2–OPT–BI and 2–OPT–FI. In 1,000 iterations, we can observe that 2–OPT–BI achieves a better solution faster than 2–OPT–FI. This happens because



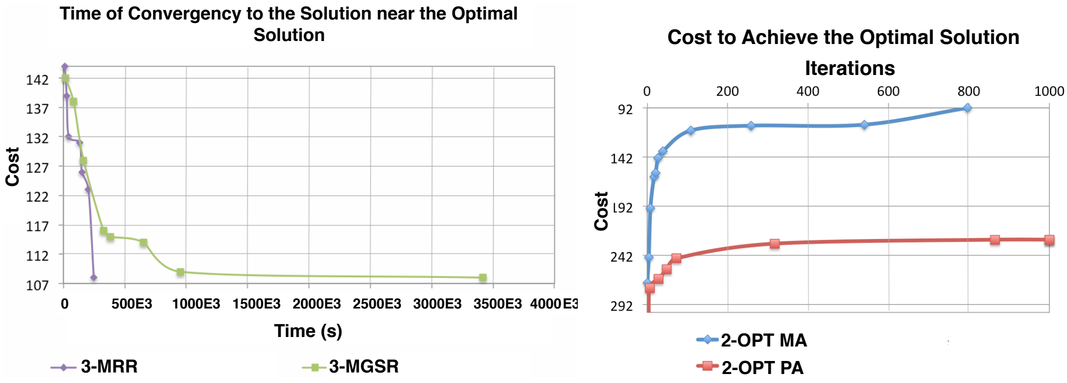
(a) 4 × 5

(b) 150 × 150

(c) 50 × 50

(d) 100 × 100

Figure 6.6: Results when requesting routes with sizes predefined.



(a) 150 × 150

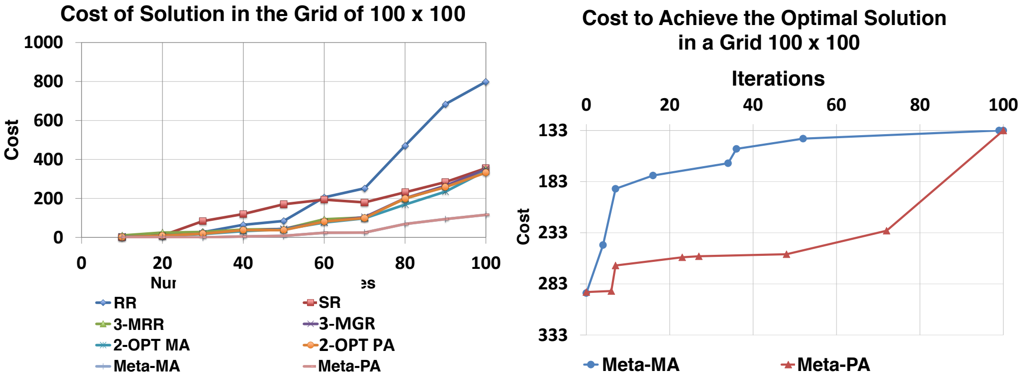
(b) 100 × 100

Figure 6.7: Efficiency of multipartite and local search heuristics.

2–OPT–FI can make detours while searching for a local optimal solution, since it is satisfied with the first improvement. Rather, 2–OPT–BI always ensures the choice of the lower cost solution between neighboring solutions. Therefore, in this scenario, the best is 2–OPT–BI.

6.2.5 Evaluation of Metaheuristics

We had two goals in order to evaluate the metaheuristics: (i) to know the cost (and the solution) of the best metaheuristic; and (ii) to analyze the time spent to achieve this solution. For that, we have requested from 10 to 100 routes, randomly. We have chosen sources and destinations crossroads randomly with a uniform distribution. We have made 100 iterations with both metaheuristics. The parameter of randomness (α) was self-adjusted every 20 iterations. First, α starts with the value of 1 for diversification, and then we reduce this value until it reaches zero, for intensification. After this, we have increased the value until 1 and repeat this cycle all over again. In diversification, the metaheuristic prioritizes viable solutions.



(a) Cost of the solution.

(b) Evolution in heuristics' cost.

Figure 6.8: Results for the metaheuristics in a grid of 100 × 100.

Figure 6.8 presents the results for Meta–BI and Meta–FI. In Figure 6.8(a), we can observe the performance of the metaheuristics in comparison with the other heuristics. We can observe that both had similar performance (same cost), and that RR was the worst heuristic. Until 60 routes requested, both found the optimal cost. To distinguish which one was the best, Figure 6.8(b) presents the evolution of the heuristic cost (until achieving the optimal solution), i.e. the robustness of heuristics[3] when they achieve the solutions with same cost. Meta–BI is better than the Meta–FI, because Meta–BI tends to the optimal cost faster than Meta–FI.

[3] Robustness indicates how fast a metaheuristic decreases its solution heading towards the optimal.

6.3 VANET Applications

In this section, we have evaluated our proposed solutions in a VANET scenario. The chosen scenario is the large-scale realistic mobility dataset of TAPAS Cologne. Such dataset spans two hours of vehicles' movements over a 400km^2 area of the city of Cologne, Germany [Uppoor and Fiore, 2012], being realistic from both macroscopic and microscopic viewpoints [Uppoor and Fiore, 2011]. Moreover, it contains information such as different types of roads, traffic light programs and road signalization, which is used as input for the Simulation of Urban MObility (SUMO) [Behrisch et al., 2011]. Notice that, this turns our analysis much more accurate and complete than using a regular Manhattan grid, since streets have different characteristics and this scenario is more complex than grid.

We have made three analyses. First, in terms of our proposed heuristics, discussed and already evaluated in this chapter. Second, as an application of our proposed model MoVDic. Third, to evaluate the performance of our proposed model STRIP.

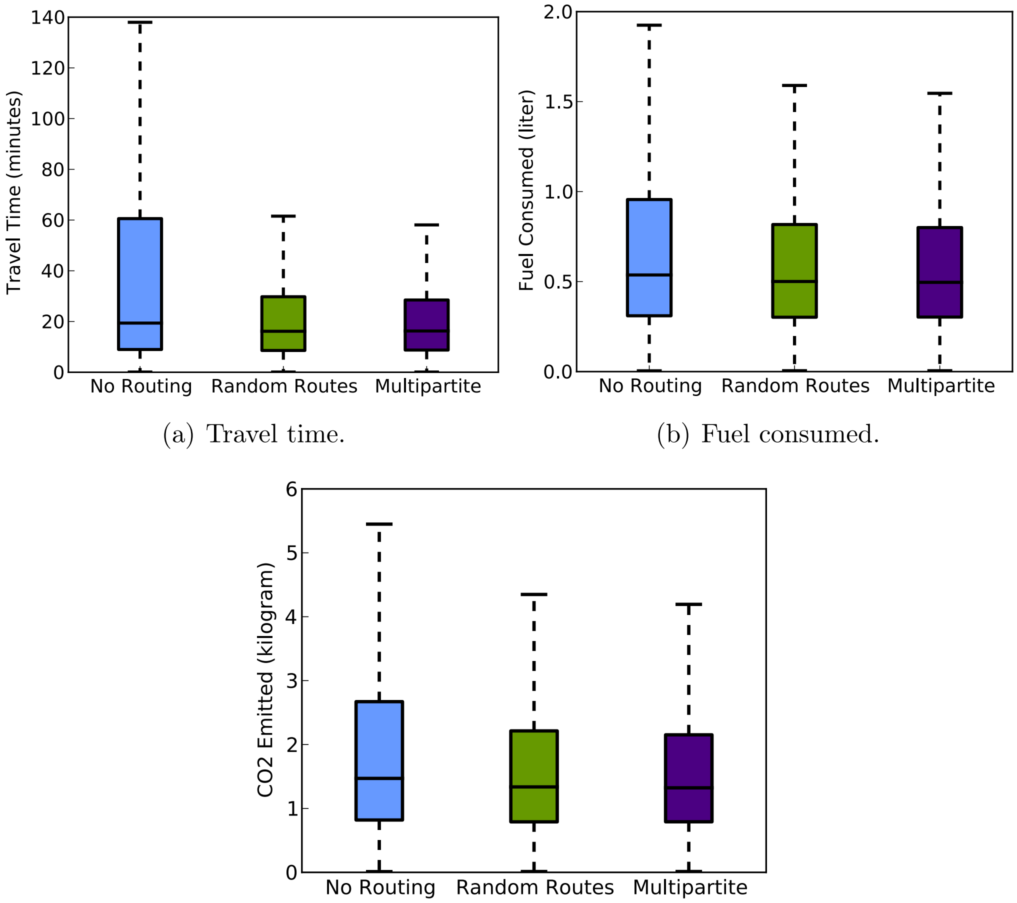
6.3.1 Our Proposed Heuristics

In order to evaluate the performance of our heuristics with real data, we have used the following metrics: travel time, fuel consumed and CO$_2$ emissions. Our goal is to assess the behavior of our heuristics under different road traffic conditions. For that, we made the load balance of flow in the network while suggesting routes for vehicles at different times of the day, i.e. at every 100 seconds interval, we suggested a different route according with the proposed heuristic (RR or MRR) for each vehicle.

Results are demonstrated in Table 6.1. We compared these two scenarios with the original scenario (in Table 6.1 "None"), in which there is no route suggestion (we use the original routes for each vehicle). For MRR, we have used 5 iterations that produce a good cost–benefit. When we increase the number of iterations in MRR, its performance gets better but it takes longer time to make the suggestion. According with the results shown in Table 6.1, as the time goes, the traffic density increases (at 6am is 61 vehicles/km while at 7:30 is 108 vehicles/km) and the travel time, fuel consumed and CO$_2$ emissions decrease.

Table 6.1: RR and MRR results in TAPAS Cologne.

Method	Average Travel Time		Average CO$_2$ Emissions		Fuel Consumption	
None	47.8990	0.3422	2.0322	0.009	0.8102	0.004
RR	23.5762	0.1282	1.7046	0.007	0.6796	0.0029
MRR	20.7990	0.0867	1.6623	0.0070	0.6627	0.0028



(a) Travel time.

(b) Fuel consumed.

(c) CO_2 emitted.

Figure 6.9: The main results for the load balancing. The boxes reach from the 25% to 75% quartile while the whiskers extend to 1.5× InterQuartile Range. The bold line within the box marks the median.

Figure 6.9 summarizes the main results for the load balance with our proposed heuristics. The travel time reduced significantly when we started suggesting the routes (50% for RR and 56% for MRR, in average). We can see a drop in the average travel time when we use MRR instead of RR. The same occurs for the average CO_2 emissions and fuel consumption. We can see that the average CO_2 emissions was reduced in 16% for RR and 18% for MRR, in average. About the fuel consumption, in average, RR decreased in 16% while MRR decreased in 18% compared with no rerouting.

6.3.2 MoVDic Predictions

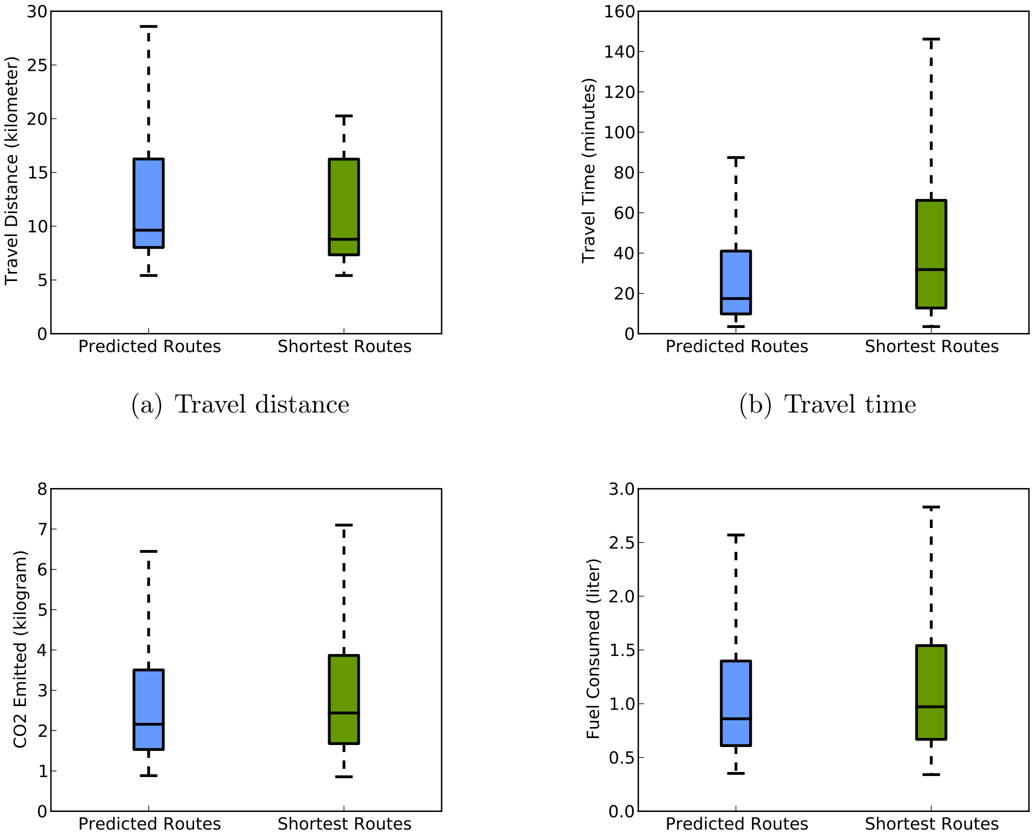
In this section, we show how the prediction from our proposed model MoVDic may be used to improve the overall road traffic condition at a large city. For that, we simulate a route recommendation application for the city of Cologne, as mentioned. The idea is to suggest the best routes to vehicles based on predicted traffic conditions, and then, assesses possible improvements regarding efficiency and environment metrics.

Initially, we randomly select 50 trips, i.e., origin-destination pairs for Cologne. Then, starting at time 1800 s of the dataset (06:30 a.m.), every 30 s, we define the routes for these trips using two strategies. In the first one (Predicted Routes), we use the traffic prediction for the next 30 s to select the best route. For instance, if the routes being generated are for vehicles departing at time 1800 s, then we use the model output for time 1830 s to choose the best routes. The second strategy (Shortest Routes) consists in choosing the routes using distance information, i.e., it always selects the shortest route.

This trip selection and route generation process continues until time 5370 s, for a total of 120 time steps. Therefore, a total of 6000 vehicles are added to the original mobility dataset. This amount represents only 2% of the total number of vehicles when we consider the resulting dataset. The resulting dataset is then given as input to SUMO in order to execute the routes and collect both efficiency and environmental statistics. It is worth noticing that these statistics are collected only for the 6000 vehicles added to the original dataset.

Figure 6.10 shows the main results for both the routes generated using MoVDic and the shortest routes. In particular, Figure 6.10(a) shows the results for the traveled distance. As expected, routes generated by our model are longer. However, on average, they are only 5% longer than the shortest routes. Figure 6.10(b) shows the travel times for vehicles to complete the routes. As can be observed, vehicles using routes generated by our proposed model takes much less time to complete their journey. Indeed, on average, the travel time is 33% lower when compared to the shortest routes. Despite the small increase on the average distance, the decrease in the travel time is substantial. Figure 6.10(c) shows the amount of CO_2 emitted per vehicle. As can be observed, vehicles that follow the routes suggested by the model emit less CO_2 when compared to vehicles that take the shortest routes. In fact, on average, they emit about 11% less CO_2. Moreover, by considering only this small amount of vehicles (2% of the dataset), a total reduction in the amount of CO_2 emitted was about 1,900 kilograms, which is quite impressive. Finally, Figure 6.10(d) shows the fuel consumption for vehicles. Once again, the vehicles that take the routes suggest by the model consume less fuel. On average, they consume 11% less and the total saving was about 760 liters.

In summary, these results show that even when manipulating the routes of a small amount of vehicles, the gains can be profound.



(a) Travel distance

(b) Travel time

(c) CO_2 emitted

(d) Fuel consumed

Figure 6.10: The main results for the route recommendation application. The boxes reach from the 25% to 75% quartile while the whiskers extend to 1.5× IRQ (interquartile range). The bold line within the box marks the median

6.3.3 STRIP Prediction

In this section, we show how accurate is STRIP prediction considering a VANET scenario such as the TAPAS Cologne. First, we have produced a trace with the travel time, density, and occupancy in each street at every 1 second of simulation. There was up to 69,000 edges. We have executed the STRIP prediction model considering the last 5 seconds of data from the different combinations of those variables:

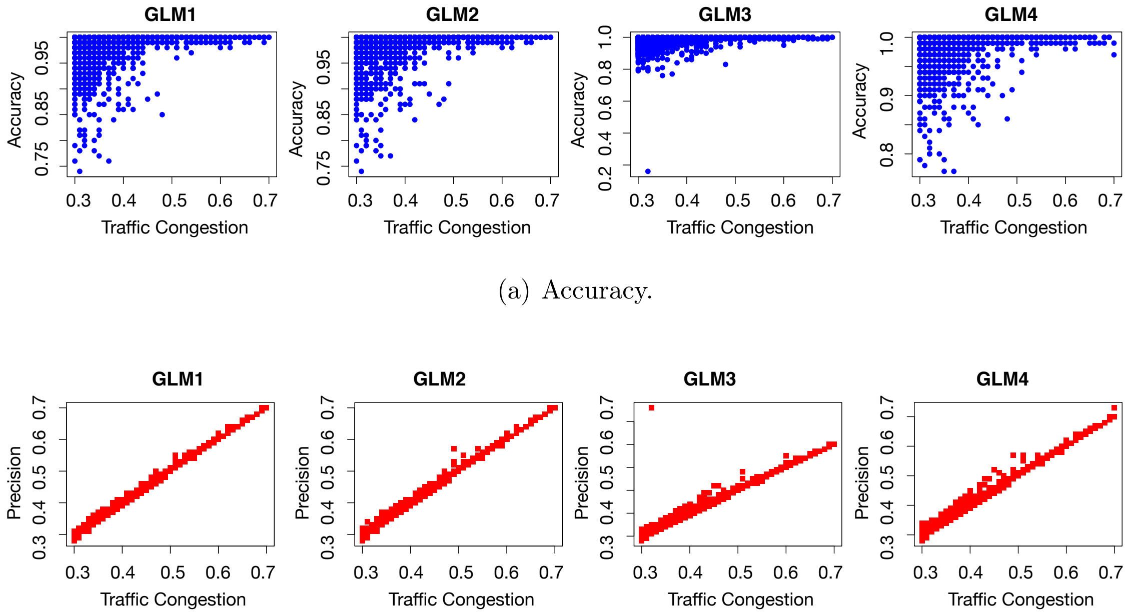
GLM1. Density only;

GLM2. Density and occupancy;

GLM3. Density and travel time;

GLM4. Density, occupancy and travel time.

For each logistic regression (GLM1–GLM4), we have considered 70% for training and 30% for the validation dataset. The metrics evaluated were accuracy and precision. Figure 6.11 presents the results of STRIP for each GLM already mentioned, considering the average of traffic congestion in the street. Figure 6.11(a) indicates the accuracy of STRIP while Figure 6.11(b) indicates its precision, in accordance with the traffic congestion in the street, that is, when the traffic density is higher than the average density in that street.



(a) Accuracy.

(b) Precision.

Figure 6.11: Results of STRIP performance with the different GLMs in comparison with the traffic congestion.

We can observe that, overall, STRIP accuracy is higher than 75% for most GLMs, except for GLM3 that had instances with an accuracy of 20%. For GLM4, the accuracies were higher than 80%. Although not having much difference, notice that GLM4 is better than the others. Also, regarding STRIP precision, notice that it is linear proportional to the traffic congestion, achieving up to 70% of precision when the traffic congestion is 70%.

6.4 Discussion

Considering constructive heuristics, Random Routes (RR) is better than the Sorted Routes (SR) when there are more route required in the scenario. For scenarios with less route requirements, RR is better. SR performs better with more vehicles. Although, when improving the constructive heuristics, the utilization of RR is better in terms of time and complexity.

For the multipartite heuristics, MRR and MGSR have better performance than the constructive heuristics, reaching solutions with a lower cost. The complexity depends on the number of iterations in the multipartite approach, which consequently determines the solution quality. MRR converges faster and closer to the optimal solution than MGSR.

The local search heuristics have no significant improvement in the cost of the heuristic when compared with the multipartite heuristics. Despite that, the local search 2–OPT–BI consumes less time to achieve a solution with the same quality as the other approaches. In 1,000 iterations, 2–OPT–BI achieved a better solution, with a lower cost, than 2–OPT–FI.

Regarding the metaheuristics, both Meta–BI and Meta–FI achieved the optimal solution, when 60 routes were required in the scenario of 100×100. For scenarios higher than that, both had solutions with the same optimal cost. About their robustness, Meta-BI was better than Meta–FI, since we achieve the optimal solution faster. Thus, Meta–BI was the best heuristic of this work, considering 100 iterations.

Besides, when we have used services for suggestion of routes based on our proposals in the TAPAS Cologne scenario. First, we have evaluated the application of MoVDic prediction in the suggestion of routes. Second, we evaluated the suggestion of routes based on load balancing with RR or MRR.

In the first evaluation, we showed how the prediction model could be used as a building block for a route suggestion application. In summary, the routes recommended by the application led to a 5% increase on the average distance traveled, when compared to the shortest routes. However, it incurred a 33% decrease on the travel time and 11% decrease on the total amount of CO_2 emitted and fuel consumed, which is promising.

In the second evaluation, we observed a decrease in the travel time, CO_2 emitted and fuel consumed. In average, RR reduced about 50% in the travel time and 16% in both CO_2 emitted and fuel consumed, while MRR reduced about 56% in the travel time and 18% for both CO_2 emitted and fuel consumed. Notice that these percentages are even better than the previous approach with MoVDic only. Although MRR is better, it also takes longer to suggest the routes. Despite the improvement in MRR, we could use RR as a faster and efficient approach. If we really need all improvements, than we suggest MRR.

In the third evaluation, we observed that STRIP is very accurate and precise in all cases. Its accuracy is higher than 80% for all streets, and its precision is linear proportional to the traffic congestion, achieving up to 70% of precision when the traffic congestion is 70%.

Finally, we can compare our solution with Waze [Google, 2009]. Waze is an application made by Google to suggest routes based on the current traffic flow of vehicles, which is inferred according to data reported by users. Basically, it uses only the current traffic conditions to suggest a less-congested-faster route. When we have phase transitions, this less-congested-faster route might be not less congested in a near future, nor even the faster route. To overcome this problem, our application can suggest routes based on the predicted traffic flow, which behave better when we have phase transitions. This evaluation is being conducted as future work.

6.5 Chapter Remarks

In this chapter, we have studied the problem of traffic congestion minimization in vehicular networks, also known as MFP. It consists in minimizing the vehicles that moves on the same roads, based on a graph of roads and a list of routes of vehicles. We proposed and evaluated several heuristics: (i) two constructive heuristics (RR and SR), (ii) two multipartite heuristics (MRR and MGSR), (iii) two heuristics of local search (2–OPT–BI and 2–OPT–FI), and (iv) two metaheuristics of reactive GRASP (Meta–BI, and Meta–FI).

Accordingly to our evaluations, the best heuristics were Meta–BI and MRR, respectively. The advantage of the proposed heuristics is to avoid future congestion while keeping the load balance of vehicles in the roads. This contributes to reduce the travel time of commuters (up to 56%, in average), the emissions of CO_2 (up to 18%, in average), and the fuel consumption (up to 18%, in average), being bigger than the simple usage of MoVDic in the suggestion of route mechanism. Finally, STRIP performed very well, achieving accuracies higher than 80% for all streets with a precision of up to 70% when the traffic congestion is 70%.

The disadvantage of our proposed heuristics can be the cost to find the solution. Since MFP is a NP–hard problem, the challenge is to find the balance between good solutions in a viable time for its computation. Some heuristics such as Meta–BI demand many resources due to their complexity. The quality of the solution will be proportional to its execution time. Therefore, this work contributes with various heuristics in different levels of cost in a viable execution time.

Chapter 7

Final Remarks

Vehicular congestion is getting worse every year. New solutions to minimize traffic jam are necessary. People believe that traffic congestion occurs because the demand exceeded roads capacity, and only with additional infrastructure the issue will be solved. However, as [Chen et al., 2001a] discuss, the major cause of traffic jams is inefficient operation of highways during periods of high demand. The increase in the vehicle fleet is not the main cause, although it contributes to a worse traffic. They say that congestion can reduce highway efficiency by 20–50%. These 20% of efficiency loss mean more loss of time, more stress, more pollution, more CO_2 emissions, and less money. As [Chen et al., 2001a] mentioned: *"The best way to combat congestion is through increases in operational efficiency."* Besides developing smart mechanisms, understanding the traffic behavior is essential for such a complex matter.

In this thesis, we propose to tackle traffic jam by exploiting the traffic behavior through probabilities, traffic incidents, phase transitions, analytical model and prediction models. In particular, we have analyzed the traffic flow from Chicago and from New York, Manhattan region. We have also analyzed the traffic incidents from Manhattan in comparison with USA. Then, we have identified the phase transitions, which has provided a clearly understand of the traffic behavior. Furthermore, we proposed an analytical model, whereby metrics for evaluating the performance of intersections, and two prediction models, in which we can respectively predict, with a granularity of road segment, *how the mobility of vehicles spreads in the city* and *what traffic flow shall be in the next instant of time.* We have also correlated the intense traffic flow with participatory sensing systems, such as Foursquare and Instagram.

Throughout this thesis, we proposed and validated models. In this chapter, Section 7.1 summarize our conclusions and contributions drawn from the previous chapters, while in Section 7.2 we lay out future work.

7.1 Conclusions

In this section, we summarize the main conclusions drawn from the thorough and comprehensive evaluation of our models, including the main contributions of this thesis. In particular, these conclusions validate the statement of this thesis, as presented in Section 1.2.

- **The characterization of real-time traffic flow and incidents from map services, and social data:** In Sections 4.1.1, 4.1.2 and 4.1.3, we have analyzed the properties of traffic flow, traffic incidents, and check-ins, such as, for instance, how they evolve spatially and temporally. In addition, in Section 4.2, we have proposed and evaluated phase transitions with two case studies, one from Chicago and other from New York, where we have analyzed the traffic behavior. Notice that we have compared different traffic patterns in several cities, besides these ones. Results have demonstrated that the traffic usually has two peaks, one in the morning and another in the evening, remarking rush-hours.

- **The correlation between intense traffic flow and social sensing:** Although being part of the characterization module, this is an important contribution that it is worth it to be highlighted. In Section 4.3, we have analyzed social sensing sources, mainly from Twitter, Foursquare and Instagram, in order to correlate them with intense traffic flow. We have shown that the distributions of intense traffic flow and of social sensing are similar, but shifted from δ time intervals. For example, in New York (Manhattan), intense traffic flow and check-ins are shifted from $\delta = 36$ intervals. This indicates that check-ins can be used to better predict the traffic congestion, as it was proved in Section 5.2.2.

- **A novel analytical model to measure the performance of intersections:** In Section 4.4, we have proposed a model to evaluate crossroad intersections, considering the vehicles' mobility and the road characteristics. This model, named ALLuPIs, is a queue network to represent the contention at each road intersection. The goal of this model is to analyze the impact of road intersection as possible traffic contention points. The impact of vehicle load in the throughput, response time and utilization has been analyzed in Section 4.4.2. They indicate that there is an optimal number of vehicles competing for intersections to achieve a better cost/benefit relation in center response time versus utilization. For the scenario that we have evaluated, this ideal number is 12 vehicles. Notice that we can use this information to predict a road contention and start a contingency plan. Moreover, we believe that this model can be applied to general traces with vehicle movement.

- **A novel prediction model on the mobility of vehicles:** In Section 5.1, we have proposed a model to predict the mobility of vehicles (MoVDic) in a city through Bayesian network, in which the model learned through a Markov chain Monte Carlo algorithm. The model uses only the data of street detectors, i.e., the number of vehicles driving on the street during a time interval, dispensing with the need to collect data on the actual turning choices at each intersection. The main idea is that the simple frequency of vehicles in each graph edge, and the implied many tight constraints it ensues, the entire flow can be predicted through a learning algorithm.

  In Section 5.1.2, we validated our model when we have contrasted the values from the detectors and the mobility of vehicles with the outputs from MoVDic. Its validation has been conducted using the large-scale mobility dataset of TAPAS Cologne [Uppoor and Fiore, 2012], overcoming other approaches in terms of mean error (less than 0.002). To predict the road with more turnings in a crossroad, comparing both scenarios, results have demonstrated that MoVDic is accurate for up to 74% in S125 and 80% in S2216, in average, for up to 85% of the crossroads in S125 and 24% in S2216. This indicates that our model is quite accurate for valid crossroads. The mean error was lower than 0.002 for both scenarios, being lower to S2216 since it presented less vehicles than S125 (120 vehicles against 500 vehicles). Also, recall that MoVDic performs well for crossroads with 2, 3 and 4 exit-streets, predicting correctly the crossroads with more turning choices more than 82% of crossroads. It has also achieved up to 100% in some cases, when we have a higher number of vehicles (150 vehicles).

- **A novel prediction model on future traffic flow:** In Section 5.2, we have proposed a short-term prediction model (STRIP), through logistic regression, to discovery future flow intensities for a target street. We have also discussed about the impact of the amount of previous data that are necessary to better predict the traffic flow.

  In Section 5.2.2, we have evaluated STRIP in accordance with Manhattan data, using as input variables the traffic flow from the current minute, from the last α minutes, besides the time and the street identification. We have tested different values of α in order to discover the best value with the best cost/benefit relation. Results have shown that more than five minutes of information to the model does not make much difference in the performance of STRIP. Also, STRIP improves the accuracy of the state-of-the-art studies, especially when using social sensors data as an input (from 80% up to 90%). Its accuracy to predict the traffic jam for the next 30 min increased up to 90% for some streets, being above the baseline (84% for [Niu et al., 2014] and 86% for [Kurihara, 2013]).

- **The suggestion of routes service with the load balance of flows:** In Chapter 6, we have proposed a service to suggest new routes to vehicles while making the load balance of vehicles flow. We have studied the problem of traffic congestion minimization in vehicular networks, also known as the Multicommodity Flow Problem. It consists in minimizing the vehicles that moves on the same roads, based on a graph of roads and a list of routes of vehicles. We proposed and evaluated several heuristics: (i) two constructive heuristics (RR and SR), (ii) two multipartite heuristics (MRR and MGSR), (iii) two heuristics of local search (2–OPT–BI and 2–OPT–FI), and (iv) two metaheuristics of reactive GRASP (Meta–BI, and Meta–FI).

  In Section 6.2, we have evaluated these heuristics in terms of time and cost, varying the scenario size and requested routes. Besides, we have applied these heuristics, MoVDic and STRIP in a suggestion of routes service that was evaluated in the realistic scenario of TAPAS Cologne. When considering the service based on our proposed heuristics RR and MRR, results have shown a decrease of up to 56% in travel time, and up to 18% in both CO_2 emissions and fuel consumed, in average. By using MoVDic prediction, results led to 33% decrease on the travel time, and 11% decrease on the fuel consumed and CO_2 emissions. Finally, to predict future traffic jam in the TAPAS Cologne scenario, STRIP achieved accuracies higher than 80% with up to 70% of precision when the traffic congestion was 70% (linear proportional).

Finally, Table 7.1 provides a comparison between this thesis contributions on Intelligent Transportation Systems (ITS) and the state-of-the-art. Recall that there are some works about participatory systems, others about congestion detection and traffic level estimation. There are also researches on dissemination protocols for VANETs and prediction models related to traffic jam and mobility. In addition, there are studies about traffic rerouting techniques in order to avoid or to minimize jams. Although, this thesis is the only research that provides contributions in all layers of the ITS.

Table 7.1: The state-of-the-art on ITS with this thesis contributions.

	Congestion Detection	Traffic Level Estimation	Cooperative Validation	Dissemination	Feasibility of inputs	Participatory systems	City-aware	Context-aware	Characterization-based	Prediction-based	Real-time Short-term Prediction	Rerouting	Prediction-based rerouting
[Silva et al., 2013b]						✓	✓						
[Silva et al., 2014]						✓	✓						
[Manasseh and Sengupta, 2013]					✓					✓			
[Okutani and Stephanedes, 1984]					✓			✓		✓			
[Inrix, 2006]	✓												
[Google, 2008]	✓												
[Microsoft Research, 2013b]	✓						✓	✓		✓			
[Google, 2009]	✓					✓							
[Min et al., 2009]					✓			✓	✓	✓	✓		
[Chan et al., 2012]					✓					✓			
[Kong et al., 2013]					✓			✓	✓	✓			
[Li et al., 2013]										✓			
[Kurihara, 2013]										✓	✓		
[Niu et al., 2014]										✓			
[Quercia et al., 2014]						✓	✓					✓	
[Liang and Wakahara, 2014]	✓				✓					✓		✓	
[de Souza et al., 2014]	✓	✓	✓	✓								✓	
[Brennand et al., 2015]	✓			✓								✓	
[Meneguette et al., 2015]	✓	✓		✓								✓	
[Pan et al., 2012]												✓	
This thesis	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

7.2 Future Work

Here follows a list of possible future research directions based on this thesis content:

- Perform a comprehensive study of the traffic conditions based on complex networks metrics. We can study how can we apply metrics from complex networks into our statement, providing new metrics with traffic flow information to measure the level of congestion.

- Apply our models in other applications and protocols using VANETs.

- To investigate the reason for the correlations of check-ins and traffic jam (e.g. due to restaurants, bars?).

- To optimize MoVDic's performance in order to achieve a real-time prediction (e.g. create a map to update only partial data of MoVDic's prediction).
- To compare STRIP's performance with other models, specially with a Waze-like strategy.
- To use the contention of a crossroad calculated by ALLuPIs as a variable in STRIP and to evaluate its performance.
- To analyze the entropy of check-ins according with their types and their effects.
- To evaluate the suggestion of routes service in VANETs when only a percentage of commuters follow suggested routes.
- To evaluate our solutions in VANET regarding the network performance, such as delay, packet loss and throughput.
- In dissemination and routing protocols, we can use our models to inform future densities instead of using the real-time densities.
- Investigate new approaches to use the forecasted paths.
- Propose new mechanisms and protocols that can take advantage of the prediction output, such as suggestion of routes, congestion control and dissemination protocols for vehicular networks.
- Evaluate other techniques, such as fuzzy logic, to classify the level of traffic congestion in the prediction models.
- Analyze and predict traffic incidents.
- Suggest routes based on predicted traffic flow and with a certain level of criminality in the city.
- Generalize the ALLuPIs model and apply the unmarked and marked intersection theory [Chevallier and Leclercq, 2007; Rouphail et al., 2000].

These are only examples of future work that could be made from this thesis. Certainly, there are several other possibilities that can also be proposed.

Bibliography

Apacible, J., Sarin, R., and Liao, L. (2005). Prediction, expectation, and surprise: Methods, designs, and study of a deployed traffic forecasting service. *Twenty-First Conference on Uncertainty in Artificial Intelligence, UAI-2005.*

Araújo, G., Duarte-Figueiredo, F., Tostes, A. I. J., and Loureiro, A. A. F. (2014). Um protocolo de identificação e minimização de congestionamentos de tráfego para redes veiculares. In *SBRC 2014*.

Araújo, G., Queiroz, M., de Lima Procópio Duarte-Figueiredo, F., Tostes, A. I. J., and A.F. Loureiro, A. (2014). CARTIM: a proposal toward identification and minimization of vehicular traffic congestion for VANET. In *19th IEEE Symposium on Computers and Communications (IEEE ISCC 2014)*, Madeira, Portugal.

Araujo, G. B., Queiroz, M. M., de L. P. Duarte-Figueiredo, F., Ribeiro, A. I. J. T., and Loureiro, A. A. F. (2014). CARTIM: A proposal toward identification and minimization of vehicular traffic congestion for VANET. In *IEEE Symposium on Computers and Communications, ISCC 2014, Funchal, Madeira, Portugal, June 23-26, 2014*, pages 1--6.

Atechian, T. and Brunie, L. (2008). Dg-castor for query packets dissemination in vanet. In *Mobile Ad Hoc and Sensor Systems, 2008. MASS 2008. 5th IEEE International Conference on*, pages 547 –552.

Awerbuch, B. and Khandekar, R. (2007). Greedy distributed optimization of multi-commodity flows. In *Proceedings of the twenty-sixth annual ACM symposium on Principles of distributed computing*, PODC '07, pages 274--283, New York, NY, USA. ACM.

Ban, X. J. and Gruteser, M. (2012). Towards fine-grained urban traffic knowledge extraction using mobile sensing. In *Proceedings of the ACM SIGKDD International*

Workshop on Urban Computing, UrbComp '12, pages 111--117, New York, NY, USA. ACM.

Banzi, A. S., Pozo, A. T. R., and Duarte Jr., E. P. (2011). Bio-inspired event dissemination in dynamic and decentralized networks. In *GECCO (Companion)*, pages 223–224.

Bauza, R., Gozalvez, J., and Sanchez-Soriano, J. (2010). Road traffic congestion detection through cooperative vehicle-to-vehicle communications. In *Local Computer Networks (LCN), 2010 IEEE 35th Conference on*, pages 606–612. ISSN 0742-1303.

Behrisch, M., Bieker, L., Erdmann, J., and Krajzewicz, D. (2011). Sumo - simulation of urban mobility: An overview. In *SIMUL 2011, The Third International Conference on Advances in System Simulation*, pages 63–68, Barcelona, Spain.

Binglei, X., Zheng, H., and Hongwei, M. (2008). Fuzzy-logic-based traffic incident detection algorithm for freeway. In *Machine Learning and Cybernetics, 2008 Int'l Conference on*, volume 3, pages 1254–1259.

Bollen, J., Mao, H., and Zeng, X. (2011). Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1--8.

Boukerche, A., Turgut, B., Aydin, N., Ahmad, M. Z., Bölöni, L., and Turgut, D. (2011). Routing protocols in ad hoc networks: A survey. *Computer Networks*, 55(13):3032–3080.

Brennand, C. A. R. L., de Souza, A. M., Maia, G., Boukerche, A., Ramos, H., A.F. Loureiro, A., and Villas, L. A. (2015). An intelligent transportation system for detection and control of congested roads in urban centers. In *20th IEEE Symposium on Computers and Communications (ISCC2015)*, Larnaca, Cyprus.

Briesemeister, L., Schafers, L., Hommel, G., and Ag, D. (2000). Disseminating messages among highly mobile hosts based on inter-vehicle communication. In *IEEE Intelligent Vehicles Symposium*, pages 522--527.

Buriol, L., França, P., Resende, M., and Ribeiro, C. (2003). Otimizando o roteamento do tráfego na internet. *Anais do XXXV Simpósio Brasileiro de Pesquisa Operacional*, pages 1722--1732.

Capone, A. and Martignon, F. (2007). A Multi-Commodity flow model for optimal routing in wireless MESH networks. *Journal of Networks*, 2(3).

Chan, K. Y., Khadem, S., Dillon, T. S., Palade, V., Singh, J., and Chang, E. (2012). Selection of significant on-road sensor data for short-term traffic flow forecasting using the taguchi method. *Industrial Informatics, IEEE Transactions on*, 8(2):255–266. ISSN 1551-3203.

Chang, I.-C., Wang, Y.-F., and Chou, C.-F. (2011). Efficient vanet unicast routing using historical and real-time traffic information. In *Proceedings of the 2011 IEEE 17th International Conference on Parallel and Distributed Systems*, ICPADS '11, pages 458--464, Washington, DC, USA. IEEE Computer Society.

Chen, C., Jia, Z., and Varaiya, P. (2001a). Causes and cures of highway congestion. *Control Systems, IEEE*, 21(6):26–32. ISSN 1066-033X.

Chen, L., Sharma, P., and Tseng, Y. (2011). Eco-sign: a load-based traffic light control system for environmental protection with vehicular communications. In *Proceedings of the ACM SIGCOMM 2011 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, Toronto, ON, Canada, August 15-19, 2011*, pages 438--439.

Chen, L., Wei, S., and Shi, L. (2009). Research on location prediction of vehicular networks. *Computer Science and Information Technology, International Conference on*, 0:558–561.

Chen, Z. D., Kung, H., and Vlah, D. (2001b). Ad hoc relay wireless networks over moving vehicles on highways. In *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, MobiHoc '01, pages 247--250, New York, NY, USA. ACM.

Chevallier, E. and Leclercq, L. (2007). A macroscopic theory for unsignalized intersections. *Transportation Research Part B: Methodological*, 41(10):1139 – 1150. ISSN 0191-2615.

Chou, L.-D., Yang, J.-Y., Hsieh, Y.-C., Chang, D.-C., and Tung, C.-F. (2011). Intersection-based routing protocol for vanets. *Wirel. Pers. Commun.*, 60(1):105--124. ISSN 0929-6212.

Chunmei, Z., Xiaoli, X., and changpeng, Y. (2010). The research of method of short-term traffic flow forecast based on ga-bp neural network and chaos theory. In *Information Science and Engineering (ICISE), 2010 2nd International Conference on*, pages 1617–1620.

Civilis, A. (2006). Prediction of crossroad passing using artificial neural networks. In *Databases and Information Systems, 2006 7th International Baltic Conference on*, pages 229–234.

Council, N. N. Y. M. T. (2013). Congestion management process report – 2013 status report. Technical report MSU-CSE-00-2, New York Metropolitan Transportation Council, New York.

Davis, G. and Nihan, N. (1991). Nonparametric regression and short-term freeway traffic forecasting. *Journal of Transportation Engineering*, 117(2):178–188.

de Brito, M. R., Silva, B., Tostes, A. I. J., Duarte-Figueiredo, F., and Loureiro, A. A. F. (2015). Transtree: Detecção de congestionamento utilizando redes veiculares. In *SBRC 2015 - WGRS*.

de Brito, M. R., Tostes, A. I. J., Duarte-Figueiredo, F., and Loureiro, A. A. F. (2014a). Simulação e análise de congestionamento em redes veiculares. In *CTIC 2014*.

de Brito, M. R., Tostes, A. I. J., Duarte-Figueiredo, F., and Loureiro, A. A. F. (2014b). Simulação e análise de métodos de detecção de congestionamento de veículos em vanet. In *SBRC 2014 - WGRS*.

de Castro, M. S., Tostes, A. I., Duarte-Figueiredo, F., and Loureiro, A. A. F. (2013). Disseminação de mensagens de acidente em redes veiculares. In *SEMISH 2013*.

de Souza, A. M., Boukerche, A., Maia, G., Meneguette, R. I., Loureiro, A. A., and Villas, L. A. (2014). Decreasing greenhouse emissions through an intelligent traffic information system based on inter-vehicle communication. In *Proceedings of the 12th ACM International Symposium on Mobility Management and Wireless Access*, MobiWac '14, pages 91--98, New York, NY, USA. ACM.

de Souza, A. M., Yokoyama, R. S., Botega, L. C., Meneguette, R. I., and Villas, L. A. (2015). Scorpion: A solution using cooperative rerouting to prevent congestion and improve traffic condition. In *IEEE CIT 2015*.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269--271. ISSN 0029-599X.

Ding, B., Chen, Z., Wang, Y., and Yu, H. (2011). An improved aodv routing protocol for vanets. In *Wireless Communications and Signal Processing (WCSP), 2011 International Conference on*, pages 1 –5.

Ding, Y. and Xiao, L. (2010). Sadv: Static-node-assisted adaptive data dissemination in vehicular networks. *Vehicular Technology, IEEE Transactions on*, 59(5):2445–2455. ISSN 0018-9545.

Domingos Da Cunha, F., Boukerche, A., Villas, L., Viana, A. C., and Loureiro, A. A. F. (2014). Data Communication in VANETs: A Survey, Challenges and Applications. Research Report RR-8498, INRIA Saclay.

Doolan, R. and Muntean, G.-M. (2013). Vanet-enabled eco-friendly road characteristics-aware routing for vehicular traffic. In *Vehicular Technology Conference (VTC Spring), 2013 IEEE 77th*, pages 1–5. ISSN 1550-2252.

Duan, G., Liu, P., Chen, P., Jiang, Q., and Li, N. (2011). Short-term traffic flow prediction based on rough set and support vector machine. In *Fuzzy Systems and Knowledge Discovery*.

Endarnoto, S., Pradipta, S., Nugroho, A., and Purnama, J. (2011). Traffic condition information extraction amp; visualization from social media twitter for android mobile application. In *Electrical Engineering and Informatics (ICEEI), 2011 International Conference on*, pages 1–4. ISSN 2155-6822.

Even, S., Itai, A., and Shamir, A. (1975). On the complexity of time table and multi-commodity flow problems. In *Proceedings of the 16th Annual Symposium on Foundations of Computer Science*, SFCS '75, pages 184--193, Washington, DC, USA. IEEE Computer Society.

Fahmy, M. and Ranasinghe, D. N. (2008). Discovering automobile congestion and volume using vanet's. In *ITS Telecommunications, 2008. ITST 2008. 8th International Conference on*, pages 367–372.

Fasolo, E., Zanella, A., and Zorzi, M. (2006). An effective broadcast scheme for alert message propagation in vehicular ad hoc networks. In *Communications, 2006. ICC '06. IEEE International Conference on*, volume 9, pages 3960--3965.

Feo, T. A. and Resende, M. G. (1995). Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6:109--133.

Fortz, B. and Thorup, M. (2000). Internet traffic engineering by optimizing ospf weights. In *INFOCOM 2000. 19th Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 519--528.

Foursquare (2014). *About Foursquare.* Foursquare. Available at https://foursquare.com/about.

Frias-Martinez, V., Soto, V., Hohwald, H., and Frias-Martinez, E. (2012). Characterizing urban landscapes using geolocated tweets. In *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom)*, pages 239--248. IEEE.

Fukumoto, J., Sirokane, N., Ishikawa, Y., Wada, T., Ohtsuki, K., and Okada, H. (2007). Analytic method for real-time traffic problems by using contents oriented communications in vanet. In *Telecommunications, 2007. ITST '07. 7th International Conference on ITS*, pages 1 –6.

Fulkerson, L. and Ford, D. (1962). Flows in networks.

Füßler, H., Hartenstein, H., Mauve, M., Effelsberg, W., and Widmer, J. (2004). Contention-based forwarding for street scenarios. In *1st International Workshop in Intelligent Transportation (WIT 2004).*

Füßler, H., Moreno, M. T., Transier, M., Festag, A., and Hartenstein, H. (2005). Thoughts on a Protocol Architecture for Vehicular Ad-hoc Networks. In *Proceeding of the 2nd International Workshop on Intelligent Transportation*, pages 41--45.

Gabrel, V., Knippel, A., and Minoux, M. (2003). A comparison of heuristics for the discrete cost multicommodity network optimization problem. *Journal of Heuristics*, 9(5):429--445. ISSN 1381–1231.

Ghosh, B., Basu, B., and O'Mahony, M. (2009). Multivariate short-term traffic flow forecasting using time-series analysis. *Intelligent Transportation Systems, IEEE Transactions on*, 10(2):246–254. ISSN 1524-9050.

Gomide, J., Veloso, A., Jr., W. M., Almeida, V., Benevenuto, F., Ferraz, F., and Teixeira, M. (2011). Dengue surveillance based on a computational model of spatio-temporal locality of twitter. In *ACM Web Science Conference (WebSci).*

Google (2008). Available at http://googlesystem.blogspot.com.br/2008/04/google-maps-predicts-traffic-conditions.html.

Google (2009). Waze. Available at http://www.waze.com/.

Hartenstein, H. and Laberteaux, K. P. (2008). A tutorial survey on vehicular ad hoc networks. *IEEE Communications Magazine.*

Hayashi, T. and Yamada, K. (2009). Predicting unusual right-turn driving behavior at intersection. In *Intelligent Vehicles Symposium, 2009 IEEE*, pages 869–874. ISSN 1931-0587.

Horvitz, E., Apacible, J., Sarin, R., and Liao, L. (2005). Prediction, expectation, and surprise: Methods, designs, and study of a deployed traffic forecasting service. *CoRR*, abs/1207.1352.

Horvitz, E. and Mitchell, T. (2010). From data to knowledge to action: A global enabler for the 21st century. *Data Analytic Series, Computing Community Consortium, Computing Research Association (CRA).*

Hu, S.-R. and Wang, C.-Y. (2006). Decision of route diversion point on freeway corridors under a global route guidance system framework. In *Systems, Man and Cybernetics, 2006. SMC '06. IEEE International Conference on*, volume 3, pages 2523–2528.

Hu, T. (1963). Multicommodity network flows. *Oper. Research*, 11:344--360.

Huzita, E., de Souza, T., and Kabuki, Y. (2012). A system to capture and generation of traffic information from posted messages on social networks. In *Collaborative Systems (SBSC), 2012 Brazilian Symposium on*, pages 174–180.

Inrix (2006). Inrix. Available at http://www.inrix.com.

Instagram (2014). *Instagram Today: 200 Million Strong.* Instagram. Available at http://blog.instagram.com/post/80721172292/200m.

Intellione (2006). Intellione. Available at http://www.intellione.com.

Jain, P. and Sethi, M. (2012). Fuzzy based real time traffic signal controller to optimize congestion delays. In *Proc. of ACCT'12*, pages 204–207.

Jerbi, M., Senouci, S. M., Rasheed, T., and Ghamri-Doudane, Y. (2009). Towards efficient geographic routing in urban vehicular networks. *Vehicular Technology, IEEE Transactions on*, 58(9):5048–5059. ISSN 0018-9545.

Jun, M. and Ying, M. (2008). Research of traffic flow forecasting based on neural network. In *Intelligent Information Technology Application, 2008. IITA '08. Second International Symposium on*, volume 2, pages 104–108.

Kaaniche, H. and Kamoun, F. (2010a). Mobility prediction in wireless ad hoc networks using neural networks. *CoRR*, abs/1004.4610.

Kaaniche, H. and Kamoun, F. (2010b). Mobility prediction in wireless ad hoc networks using neural networks. *CoRR*, abs/1004.4610.

Kamarianakis, Y. and Prastakos, P. (2003). “forecasting traffic flow conditions in an urban network: Comparison of multivariate and univariate approaches. *Proc. 82nd Annu. Meeting TRB.*

Karagiannis, G., Altintas, O., Ekici, E., Heijenk, G., Jarupan, B., Lin, K., and Weil, T. (2011). Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions. *Communications Surveys Tutorials, IEEE*, 13(4):584–616. ISSN 1553-877X.

Karp, B. and Kung, H. T. (2000). Gpsr: greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 243--254, New York, NY, USA. ACM.

Karp, R. (1972). Reducibility among combinatorial problems. In Miller, R. and Thatcher, J., editors, *Complexity of Computer Computations*, pages 85--103. Plenum Press.

Kihl, M., Sichitiu, M., and Joshi, H. (2008). Design and evaluation of two geocast protocols for vehicular ad-hoc networks. *Journal of Internet Engineering.*

Kong, Q.-J., Xu, Y., Lin, S., Wen, D., Zhu, F., and Liu, Y. (2013). UTN-model-based traffic flow prediction for parallel-transportation management systems. *IEEE Trans. on ITS*, 14(3):1541–1547. ISSN 1524-9050.

Korkmaz, G., Ekici, E., Özgüner, F., and Özgüner, U. (2004a). Urban multi-hop broadcast protocol for inter-vehicle communication systems. In *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, VANET ’04, pages 76--85. ACM.

Korkmaz, G., Ekici, E., Özgüner, F., and Özgüner, U. (2004b). Urban multi-hop broadcast protocol for inter-vehicle communication systems. In *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, VANET ’04, pages 76--85, New York, NY, USA. ACM.

Krumm, J. (2008). A markov model for driver turn prediction. In *SAE Technical Paper 2008-01-0195.*

Krumm, J. (2010). Where will they turn: Predicting turn proportions at intersections. *Personal Ubiquitous Comput.*, 14(7):591--599. ISSN 1617-4909.

Kurihara, S. (2013). Traffic-congestion forecasting algorithm based on pheromone communication model. *Ant Colony Optimization - Techniques and Applications.*

Lee, J.-W., Lo, C.-C., Tang, S.-P., Horng, M.-F., and Kuo, Y.-H. (2011). A hybrid traffic geographic routing with cooperative traffic information collection scheme in vanet. In *Advanced Communication Technology (ICACT), 2011 13th International Conference on*, pages 1496–1501. ISSN 1738-9445.

Lee, R. and Sumiya, K. (2010). Measuring geographical regularities of crowd behaviors for twitter-based geo-social event detection. In *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks*, pages 1--10. ACM.

Li, F. and Wang, Y. (2007). Routing in vehicular ad hoc networks: A survey. *Vehicular Technology Magazine, IEEE*, 2(2):12 –22. ISSN 1556-6072.

Li, L., Xia, H., Li, L., and Wang, Q. (2013). Traffic prediction based on svm training sample divided by time. *TELKOMNIKA Indonesian Journal of Electrical Engineering*, 11(12):7446–7452.

Liang, Z. and Wakahara, Y. (2014). A route guidance system with personalized rerouting for reducing traveling time of vehicles in urban areas. In *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, pages 1541–1548.

Liu, H., Gao, Y., Lu, L., Liu, S., Qu, H., and Ni, L. (2011). Visual analysis of route diversity. In *Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on*, pages 171–180.

Liu, Y. and Ozguner, U. (2003). A quantitative study on traffic network throughput. In *Intelligent Transportation Systems, 2003. Proceedings. 2003 IEEE*, volume 1, pages 108–113 vol.1.

Lu, J. and Cao, L. (2003). Congestion evaluation from traffic flow information based on fuzzy logic. In *Intelligent Transportation Systems, 2003. Proceedings. 2003 IEEE*, volume 1, pages 50–53 vol.1.

Manasseh, C. and Sengupta, R. (2013). Predicting driver destination using machine learning techniques. In *Intelligent Transportation Systems - (ITSC), 2013 16th International IEEE Conference on*, pages 142–147.

Marfia, G. and Roccetti, M. (2011). Vehicular congestion detection and short-term forecasting: A new model with results. *Vehicular Technology, IEEE Transactions on*, 60(7):2936–2948. ISSN 0018-9545.

Menasce, D. A., Almeida, V. A. F., and Dowdy, L. W. (2004). *Performance by Design: Computer Capacity Planning by Example.* Prentice Hall.

Meneguette, R., Filho, G. P. R., Bittencourt, L. F., Ueyama, J., Krishnamachari, B., and Villas, L. A. (2015). Enhancing intelligence in intervehicle communications to detect and reduce congestion in urban centers. In *20th IEEE Symposium on Computers and Communications (ISCC2015)*, Larnaca, Cyprus.

Microsoft Research (2013a). Developing business intelligence and data visualization applications with web maps. [Online; accessed May 31, 2013].

Microsoft Research (2013b). Predictive analysis for traffic. [Online; accessed May 31, 2013].

Min, X., Hu, J., Chen, Q., Zhang, T., and Zhang, Y. (2009). Short-term traffic flow forecasting of urban network based on dynamic starima model. In *Intelligent Transportation Systems, 2009. ITSC '09. 12th International IEEE Conference on*, pages 1–6.

Mourão, F. P. (2009). Aplicação de métodos heurísticos aos problemas de fluxo multiproduto mono-objetivo e multiobjetivo. Master's thesis, Mestrado em Modelagem Matemática e Computacional, Centro Federal de Educação Tecnológica de Minas Gerais, Belo Horizonte.

Naumov, V., Baumann, R., and Gross, T. (2006). An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces. In *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '06, pages 108--119, New York, NY, USA. ACM.

Niu, X., Zhu, Y., and Zhang, X. (2014). Deepsense: A novel learning mechanism for traffic prediction with taxi gps traces. *Globecom 2014 - Symposium on Selected Areas in Communications: GC14 SAC Internet of Things*, pages 2745---2750.

Nzouonta, J., Rajgure, N., Wang, G., and Borcea, C. (2008). Vanet routing on city roads using real-time vehicular traffic information. *IEEE Transactions on Vehicular Technology.*

Nzouonta, J., Rajgure, N., Wang, G., and Borcea, C. (2009). Vanet routing on city roads using real-time vehicular traffic information.

Okutani, I. and Stephanedes, Y. J. (1984). Dynamic prediction of traffic volume through kalman filtering theory. *Transportation Research Part B: Methodological*, 18(1):1–11.

Pan, J., Khan, M., Popa, I., Zeitouni, K., and Borcea, C. (2012). Proactive vehicle re-routing strategies for congestion avoidance. In *Distributed Computing in Sensor Systems (DCOSS), 2012 IEEE 8th International Conference on*, pages 265–272.

Parkinson, B. and Gilbert, S. (1983). Navstar: Global positioning system – ten years later. *Proceedings of the IEEE*, 71(10):1177–1186. ISSN 0018-9219.

Perkins, C. and Royer, E. (1999a). Ad-hoc on-demand distance vector routing. In *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on*, pages 90--100.

Perkins, C. E. and Royer, E. M. (1999b). Ad hoc on-demand distance vector routing. In *IEEE Workshop on Mobile Computing Systems and Applications*, pages 90--100.

Quercia, D., Schifanella, R., and Aiello, L. M. (2014). The shortest path to happiness: recommending beautiful, quiet, and happy routes in the city. In *25th ACM Conference on Hypertext and Social Media, HT '14, Santiago, Chile, September 1-4, 2014*, pages 116--125.

Raghavan, P. and Tompson, C. D. (1987). Randomized rounding: a technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7(4):365--374. ISSN 0209-9683.

Ranjan, P. and Ahirwar, K. K. (2011). Comparative study of vanet and manet routing protocols. In *Proceedings of the International Conference on Advanced Computing and Communication Technologies (ACCT 2011)*.

Ribeiro, A. I. J. T., Silva, T. H., Duarte-Figueiredo, F., and Loureiro, A. A. (2014). Studying traffic conditions by analyzing foursquare and instagram data. In *Proc. of the 11th ACM Symp. on Performance Evaluation of Wireless Ad Hoc, Sensor, &#38; Ubiquitous Networks*, PE-WASUN '14, pages 17--24, New York, NY, USA. ACM.

Roess, R. and Prassas, E. (2014). *The Highway Capacity Manual: A Conceptual and Research History: Volume 1: Uninterrupted Flow*. Springer Tracts on Transportation and Traffic. Springer. ISBN 9783319057866.

Rouphail, N., Tarko, A., and Li, J. (2000). Traffic flow at signalized intersections. In H, L., editor, *Civil Engineering*, pages 9.1--9.28. Revised Monograph of Traffic Flow Theory, update and expansion of the Transportation Research Board (TRB) special report 165, "Traffic Flow Theory", published in 1975.

Saha, A. K. and Johnson, D. B. (2004). Modeling mobility for vehicular ad-hoc networks. In *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, VANET '04, pages 91--92, New York, NY, USA. ACM.

Sakaki, T., Okazaki, M., and Matsuo, Y. (2010). Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 851--860, New York, NY, USA. ACM.

Salnikov, V., Lambiotte, R., Noulas, A., and Mascolo, C. (2015). OpenStreetCab: Exploiting Taxi Mobility Patterns in New York City to Reduce Commuter Costs. Technical report arXiv:1503.03021. Comments: in NetMob 2015.

Schougaard, K. (2007). *Vehicular Mobility Prediction by Bayesian Networks*. DAIMI PB. Computer Science Department, Aarhus Univ.

Seada, K. and Helmy, A. (2006). Efficient and robust geocasting protocols for sensor networks. *Comput. Commun.*, 29(2):151--161.

Sera, M. (2007). Traffic jam prediction device and method. US Patent App. 11/476,384.

Shen, D., Chen, G., Cruz, J., and Blasch, E. (2008). A game theoretic data fusion aided path planning approach for cooperative uav isr. In *Aerospace Conference, 2008 IEEE*, pages 1–9. ISSN 1095-323X.

Shen, H., Zhu, Y., Liu, T., and Jin, L. (2009). Particle swarm optimization in solving vehicle routing problem. In *Intelligent Computation Technology and Automation, 2009. ICICTA '09. Second International Conference on*, volume 1, pages 287–291.

Silva, C. A. (2007). Uma abordagem de problemas de fluxo multiproduto via métodos heurísticos. Master's thesis, Mestrado em Modelagem Matemática e Computacional, Centro Federal de Educação Tecnológica de Minas Gerais, Belo Horizonte.

Silva, F., Silva, T., Ruiz, L., and Loureiro, A. (2013a). Conprova: A smart context provisioning middleware for vanet applications. In *Vehicular Technology Conference (VTC Spring), 2013 IEEE 77th*, pages 1–5. ISSN 1550-2252.

Silva, T., Vaz De Melo, P., Almeida, J., and Loureiro, A. (2014). Large-scale study of city dynamics and urban social behavior using participatory sensing. *IEEE Wireless Comm.*, 21(1):42–51.

Silva, T. H., da Cunha, F. D., Tostes, A. I. J., Neto, J. B. B., de S Celes, C. S. F., Mota, V. F. S., Ferreira, A. P. G., de Melo, P. O. S. V., Almeida, J. M., and Loureiro, A. A. F. (2015a). Users in the urban sensing process: Challenges and research opportunities (accepted). In *Pervasive Computing: Next Generation Platforms for Intelligent Data Collection*.

Silva, T. H., Vaz de Melo, P. O. S., Almeida, J. M., Borges Neto, J., Tostes, A. I. J., Celes, C. S. F. S., Mota, S., V. F., Cunha, F. D., Ferreira, A. P. G., Machado, K. L. S., and Loureiro, A. A. F. (2015b). Redes de sensoriamento participativo: Desafios e oportunidades. In *Minicursos / XXXIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 266--315, Porto Alegre. Sociedade Brasileira de Computação.

Silva, T. H., Vaz de Melo, P. O. S., Almeida, J. M., and Loureiro, A. A. F. (2012). Visualizing the Invisible Image of Cities. In *Proc. IEEE International Conference on Cyber, Physical and Social Computing*, pages 382--389, Besancon, France.

Silva, T. H., Vaz de Melo, P. O. S., Viana, A., Almeida, J. M., Salles, J., and Loureiro, A. A. F. (2013b). Traffic Condition is more than Colored Lines on a Map: Characterization of Waze Alerts. In *Proc. of the Int. Conference on Social Informatics (SocInfo'13)*, pages 309--318, Kyoto, Japan.

Skordylis, A. and Trigoni, N. (2008). Delay-bounded routing in vehicular ad-hoc networks. In *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '08, pages 341--350, New York, NY, USA. ACM.

Soares, R. B., Ribeiro, A. I. J. T., Nakamura, E. F., and Loureiro, A. A. F. (2014a). An adaptive data dissemination protocol with dynamic next hop selection for vehicular networks. In *IEEE Symposium on Computers and Communications, ISCC 2014, Funchal, Madeira, Portugal, June 23-26, 2014*, pages 1--7.

Soares, R. B., Tostes, A. I. J., Nakamura, E. F., and A.F. Loureiro, A. (2014b). An adaptive data dissemination protocol with dynamic next hop selection for vehicular networks. In *19th IEEE Symposium on Computers and Communications (IEEE ISCC 2014)*, Madeira, Portugal.

Sommer, C., Krul, R., German, R., and Dressler, F. (2010). Emissions vs. travel time: Simulative evaluation of the environmental impact of its. In *Vehicular Technology Conference (VTC 2010-Spring), 2010 IEEE 71st*, pages 1–5. ISSN 1550-2252.

Son, D., Helmy, A., and Krishnamachari, B. (2004). The effect of mobility-induced location errors on geographic routing in mobile ad hoc sensor networks: analysis and improvement using mobility prediction. *Mobile Computing, IEEE Trans. on*, 3(3):233 – 245. ISSN 1536-1233.

Son, T. T., Minh, H. L., Sexton, G., Aslam, N., and Ghassemlooy, Z. (2013). Bayesian model for mobility prediction to support routing in mobile ad-hoc networks. In *Personal Indoor and Mobile Radio Communications (PIMRC), 2013 IEEE 24th International Symposium on*, pages 3186–3190. ISSN 2166-9570.

Souza, A. M. and Villas, L. A. (2015). A new solution based on inter-vehicle communication to reduce traffic jam in highway environment. *Latin America Transactions, IEEE (Revista IEEE America Latina)*, 13(3):721–726. ISSN 1548-0992.

Stutz, C. and Runkler, T. (2002). Classification and prediction of road traffic using application-specific fuzzy clustering. *Fuzzy Systems, IEEE Transactions on*, 10(3):297–308. ISSN 1063-6706.

Su, W., Lee, S.-J., and Gerla, M. (2000). Mobility prediction in wireless networks. In *MILCOM 2000. 21st Century Military Communications Conference Proceedings*, volume 1, pages 491–495 vol.1.

Su, W., Lee, S.-J., and Gerla, M. (2001). Mobility prediction and routing in ad hoc wireless networks. *Int. J. Netw. Manag.*, 11(1):3--30. ISSN 1099-1190.

Sun, S., Yu, G., and Zhang, C. (2004). Short-term traffic flow forecasting using sampling markov chain method with incomplete data. In *Intelligent Vehicles Symposium, 2004 IEEE*, pages 437–441.

Tan, M.-C., Wong, S., Xu, J.-M., Guan, Z.-R., and Zhang, P. (2009). An aggregation approach to short-term traffic flow prediction. *Intelligent Transportation Systems, IEEE Transactions on*, 10(1):60–69. ISSN 1524-9050.

Thomas, T., Weijermars, W., and van Berkum, E. (2010). Predictions of urban volumes in single time series. *Intelligent Transportation Systems, IEEE Transactions on*, 11(1):71–80. ISSN 1524-9050.

Toole, J. L., Ulm, M., González, M. C., and Bauer, D. (2012). Inferring land use from mobile phone activity. In *Proceedings of the ACM SIGKDD International Workshop on Urban Computing*, UrbComp '12, pages 1--8, New York, NY, USA. ACM.

Tostes, A. I., Duarte-Figueiredo, F., Almeida, J., and Loureiro, A. A. F. (2012). Modelo analítico de contenção de tráfego em vanet usando dados reais de mobilidade. In *CSBC 2012 - WPerformance*, Curitiba-PR.

Tostes, A. I., Maia, G., Duarte-Figueiredo, F., and A.F. Loureiro, A. (2015). Suggestion of routes for vehicles in vehicular networks using the multicommodity flow model. In *20th IEEE Symposium on Computers and Communications (ISCC2015)*, Larnaca, Cyprus.

Tostes, A. I. J., de L. P. Duarte-Figueiredo, F., Assunção, R., Salles, J., and Loureiro, A. A. F. (2013). From data to knowledge: City-wide traffic flows analysis and prediction using bing maps. In *Proc. of ACM SIGKDD UrbComp'13*, pages 12:1--12:8, Chicago, Illinois.

Tostes, A. I. J., Maia, G., ao, R. A., Duarte-Figueiredo, F., and Loureiro, A. A. (2015a). MoVDic: predicting the mobility of vehicles through information of street detectors [submitted]. In *Ad Hoc Network Journal*.

Tostes, A. I. J., Silva, T. H., ao, R. A., Figueiredo, F. D., and Loureiro, A. A. (2015b). Strip: A short-term traffic jam prediction model based on online maps and social sensors (submitted). In *IEEE 82nd Vehicular Technology Conference (VTC2015-Fall)*.

Tostes, A. I. J., Silva, T. H., Duarte-Figueiredo, F., and A.F. Loureiro, A. (2014). Studying traffic conditions by analyzing foursquare and instagram data. In *PeWaSUN'14)*.

Tribune, C. (2011). Chicago no. 1 in road congestion. [Online; accessed May 31, 2013].

Uma Nagaraj, N. N. (2011). Study of statistical models for route prediction algorithms in vanet. *Journal of Information Engineering and Applications*, 1(4):28--33. ISSN 2224–5758.

Uppoor, S. and Fiore, M. (2011). Large-scale urban vehicular mobility for networking research. In *IEEE Vehicular Networking Conference (VNC '11)*, pages 62--69.

Uppoor, S. and Fiore, M. (2012). Insights on metropolitan-scale vehicular mobility from a networking perspective. In *ACM International Workshop on Hot Topics in Planet-Scale Measurement (HotPlanet '12)*, pages 39--44.

Uppoor, S., Trullols-Cruces, O., Fiore, M., and Barcelo-Ordinas, J. M. (2013). Generation and analysis of a large-scale urban vehicular mobility dataset. TAPAS (TAPAS Cologne Scenario) [Online]. `http://sourceforge.net/apps/mediawiki/sumo/index.php?title=Data/Scenarios/TAPASCologne/`.

Villas, L., Boukerche, A., Araujo, R., Loureiro, A., and Ueyama, J. (2013). Network partition-aware geographical data dissemination. In *Communications (ICC), 2013 IEEE International Conference on*, pages 1439–1443. ISSN 1550-3607.

Villas, L. A., Boukerche, A., Maia, G., Pazzi, R. W., and Loureiro, A. A. (2014). Drive: An efficient and robust data dissemination protocol for highway and urban vehicular ad hoc networks. *Computer Networks*, 75, Part A:381 – 394. ISSN 1389-1286.

Wang, S., Djahel, S., and McManis, J. (2014). A multi-agent based vehicles re-routing system for unexpected traffic congestion avoidance. In *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, pages 2541–2548.

Wang, Y., Zhu, X., Li, L., and Wu, B. (2013). Reasons and countermeasures of traffic congestion under urban land redevelopment. *Procedia - Social and Behavioral Sciences*, 96(0):2164 – 2172. ISSN 1877-0428. Intelligent and Integrated Sustainable Multimodal Transportation Systems Proceedings from the 13th {COTA} International Conference of Transportation Professionals (CICTP2013).

Williams, B. and Hoel, L. (2003). Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results. *Journal of Transportation Engineering*, 129(6):664–672.

Wisitpongphan, N., Jitsakul, W., and Jieamumporn, D. (2012). Travel time prediction using multi-layer feed forward artificial neural network. In *Proc. of CICSyN'12*, pages 326–330.

Wu, H., Fujimoto, R., Guensler, R., and Hunter, M. (2004). Mddv: a mobility-centric data dissemination algorithm for vehicular networks. In *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, VANET '04, pages 47--56, New York, NY, USA. ACM.

Xue, G., Li, Z., Zhu, H., and Liu, Y. (2009). Traffic-known urban vehicular route prediction based on partial mobility patterns. In *Parallel and Distributed Systems (ICPADS), 2009 15th International Conference on*, pages 369--375. ISSN 1521–9097.

Ye, S., He, Y., Hu, J., and Zhang, Z. (2008). Short-term traffic flow forecasting based on mars. In *Fuzzy Systems and Knowledge Discovery, 2008. FSKD '08. Fifth International Conference on*, volume 5, pages 669–675.

Y.Kamarianakis and P.Prastacos (2005). "space-time modelling of traffic flow. *Comput. Geosci.*, 31:119--133.

Yoon, J., Noble, B., and Liu, M. (2007). Surface street traffic estimation. In *Proceedings of the 5th International Conference on Mobile Systems, Applications and Services*, MobiSys '07, pages 220--232, New York, NY, USA. ACM.

Younes, M. and Boukerche, A. (2013). Efficient traffic congestion detection protocol for next generation vanets. *Communications (ICC), 2013 IEEE International Conference on*, pages 3764–3768. ISSN 1550-3607.

Yu, G., Hu, J., Zhang, C., Zhuang, L., and Song, J. (2003). Short-term traffic flow forecasting based on markov chain model. In *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*, pages 208–212.

Yuecong, S., Wei, H., and Guotang, B. (2007). Combined prediction research of city traffic flow based on genetic algorithm. In *Electronic Measurement and Instruments, 2007. ICEMI '07. 8th International Conference on*, pages 3–862–3–865.

Zhao, J. and Cao, G. (2008). Vadd: Vehicle-assisted data delivery in vehicular ad hoc networks. *Vehicular Technology, IEEE Transactions on*, 57(3):1910 –1922.

Zhao, J., Member, S., Zhang, Y., Member, S., Cao, G., and Member, S. (2007). Data pouring and buffering on the road: A new data dissemination paradigm for vehicular ad hoc networks. *IEEE Transactions on Vehicular Technology.*