Architectural Aspects of Sensing Applications for Vehicular Networks: from Data Collection to Data Model Management

Mohammad Nozari Zarmehri

Supervisor: Prof. Carlos Manuel Milheiro de Oliveira Pinto Soares

Doctoral Program in Telecommunications

May, 2016

© Mohammad Nozari Zarmehri: May, 2016

Faculdade de Engenharia da Universidade do Porto

Architectural Aspects of Sensing Applications for Vehicular Networks: from Data Collection to Data Model Management

Mohammad Nozari Zarmehri

Dissertation submitted to Faculdade de Engenharia da Universidade do Porto
to obtain the degree of

Doctor of Philosophy in Electrical and Computer Engineering

President: Prof. Eugénio da Costa Oliveira
Referee: Prof. José Hernandez Orallo
Referee: Prof. Alberto Jorge Lebre Cardoso
Referee: Prof. Alexandre Júlio Teixeira dos Santos
Referee: Prof. Pavel Bernard Brazdil
Referee: Prof. Carlos Manuel Milheiro de Oliveira Pinto Soares

---

May, 2016

It is my deepest gratitude and warmest affection that I dedicate this thesis to my family especially Zhaleh and Amirreza for their constant support.

Abstract

Vehicular Ad-hoc Networks (VANETs) are part of the communications infrastructure for Intelligent Transportation Systems (ITS), which are becoming an important type of information systems. Additionally, vehicular networks offer various opportunities for gathering data about a city. Vehicles continually sense events from streets and process sensed data. Therefore, utilizing vehicular networks as an infrastructure for urban sensing is a cost-efficient way of deploying an urban monitoring system without actually deploying connected sensors. ITS services based on VANETs require that massive amounts of data are gathered and transferred to a location for storing and making it available for applications that will analyze it. To collect this data using VANETs, an efficient protocol is needed that uses minimum communication time and network resources while providing low delay and high delivery rates. Two important components of the infrastructure of VANET-based ITS applications are the data communication and management to provide the collected data to entities inside or outside the VANET in almost real-time. In this thesis, to cover all aspects required for data communications and data management, the following tasks are done:

1. Design and implement a new protocol for urban sensing and data collection,
2. Performance evaluation of the proposed protocol and evaluate different broadcast suppression techniques,
3. Calculate the sensing capacity and validate it with simulation,
4. Investigate the use of existing hierarchies in the data to improve the performance of the learning process,
5. Propose a metalearning framework for data hierarchy level and algorithm selection,
6. Evaluate the framework by applying it to different datasets.

The results of the investigation show that the designed protocol is able to collect the VANETs data efficiently (with high packet delivery and low delay). In addition, the results of the experiment on collected data show that the designed framework is able to suggest the best part of data that should be used for training a model for each entity to obtain the best performance without applying each algorithm on each part of the data separately.

Keywords: Data Collection. Machine learning. Data mining. Metalearning. Data Hierarchy. Vehicular Networks.

Resumo

VANETs fazem parte da infra-estrutura de comunicações para Sistemas Inteligentes de Transportes (ITS), que estão a tornar-se um importante tipo de sistemas de informação. Além disso, as redes veiculares oferecem várias oportunidades para a recolha de dados sobre uma cidade. Os veículos estão constantemente a detectar eventos da ruas e a processar os dados adquiridos. Assim, a utilização de redes veiculares como uma infra-estrutura para detecção urbana é uma forma eficiente em termos de custo de implantação de um sistema de monitorização urbano sem realmente implementar sensores conectados. Os serviços ITS baseados em VANETs exigem que grandes quantidades de dados sejam recolhidos e transferidos para um local de armazenamento e tornando-os disponíveis para aplicações que os irão analisar. Para recolher esses dados usando VANETs, é necessário um protocolo eficiente que use tempo mínimo de comunicação e recursos de rede, proporcionando low delay e high packet delivery. Dois componentes importantes da infra-estrutura de aplicações ITS baseadas em VANET são a comunicação e gestão de dados. Para fornecer os dados recolhidos para entidades dentro ou fora da VANET aproximadamente em tempo real. Nesta tese, de forma a cobrir todos os aspectos necessários para a comunicação e gestão de dados, foram realizadar as seguintes tarefas:

1. conceptualizar e implementar um novo protocolo para sensorização urbana e recolha de dados,

2. Avaliação do desempenho do protocolo proposto e avaliação de diferentes técnicas de broadcast suppression,

3. Calcular a capacidade de detecção e validá-la com a simulação,

4. investigar o uso de hierarquias existentes nos dados para melhorar o desempenho do processo de aprendizagem,

5. Propor uma framework metalearning para o nível de hierarquia de dados e seleção de algoritmos,

6. Avaliar a framework, aplicando-o em diferentes conjuntos de dados.

Os resultados da investigação mostram que o protocolo projetado é capaz de recolher os dados VANETs de forma eficiente (com high packet delivery e low delay). Além disso, os resultados de ensaios em dados recolhidos mostram que a estrutura concebida é capaz de sugerir a melhor parte de dados que devem ser usados para o formo de um modelo para cada entidade para obter

o melhor desempenho sem necessidade de aplicação de cada algoritmo em cada parte dos dados separadamente.

Keywords: Data Collection. Machine learning. Data mining. Metalearning. Data Hierarchy. Vehicular Networks.

Acknowledgments

Firstly, I would like to express my sincere gratitude to my advisor Prof. Carlos Manuel Milheiro de Oliveira Pinto Soares for the continuous support of my Ph.D. study and related research, for his patience, motivation, and immense knowledge. His guidance helped me to have an opportunity improving my abilities in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D. study.

Besides my advisor, I would like to thank Prof. Eugénio Oliveira and Prof. Henrique Salgado for their support and encouragement. Without their precious support, it would not be possible to conduct this research.

In addition, I would like to thank the rest of my Ph.D. thesis committee: Prof. Eugénio da Costa Oliveira, Prof. Alexandre Júlio Teixeira dos Santos, and Prof. Alberto Jorge Lebre Cardoso, Prof. José Hernandez Orallo, Prof. Pavel Bernard Brazdil for their hard questions which motivate me to widen my research from various perspectives.

I thank my fellow lab-mates for the stimulating discussions and for all the fun we have had in the last four years. Also, I thank my friends in the following institution: Institution of Telecommunication and INESC Porto. In particular, I am grateful to Prof. Ana Aguiar for enlightening me the first glance of research.

Finally, I am very grateful to the Ph.D. scholarship (SFRH/BD/71438/2010) from the Portuguese Foundation for Science and Technology (FCT), who funded my study.

Last but not least, I would like to thank my whole family especially my wife, Zhaleh, and my son, Amirreza, for supporting me spiritually throughout this Ph.D. project and my life in general.

Mohammad Nozari Zarmehri

Publications

M. Nozari Zarmehri, A. Aguiar. *Novel Data Gathering Protocol for Sensing Applications in Vehicular Networks*. Polaris Workshop. Porto, Portugal, Oct 27/28. 2011.

M. Nozari Zarmehri, A. Aguiar. *Urban Data Collector Protocol for Sensing Applications in Vehicular Networks*. MAP-tele Workshop. Guimarães, Portugal, May 02. 2012.

M. Nozari Zarmehri, A. Aguiar. *Data Gathering for Sensing Applications in Vehicular Networks*. IEEE Vehicular Networking Conference (VNC). Amsterdam, Netherlands, Nov 13-17. 2011.

M. Nozari Zarmehri, A. Aguiar. *Supporting Sensing Application in Vehicular Networks*. ACM MobiCom Workshop on Challenged Networks. Istanbul, Turkey, Aug 22. 2012.

M. Nozari Zarmehri, A. Aguiar. *Urban Data Collector Protocol: Performance Evaluation with Different Suppression Techniques*. International Journal of Computer Science Issues, Vol. 9, Issue 5, No 1, ISSN (Online): 1694-0814. September 2012.

M. Nozari Zarmehri, A. Aguiar. *Numerical Limits for Data Gathering in Wireless Networks*. Proc IEEE International Symp. On Personal, Indoor and Mobile Radio Communication - PIMRC. London, United Kingdom, Vol. 1, pp. 1 - 6, September, 2013.

M. Nozari Zarmehri, Carlos Soares. *Improving Data Mining Results by taking Advantage of the Data Warehouse Dimensions: A Case Study in Outlier Detection*. The Brazilian Conference on Intelligent Systems (BRACIS) and Encontro Nacional de Inteligência Artificial e Computacional (ENIAC). São Carlos, SP, Brazil, October 19-23, 2014.

M. Nozari Zarmehri, Carlos Soares. *Metalearning to Choose the Level of Analysis in Nested Data: A Case Study on Error Detection in Foreign Trade Statistics*. International Joint Conference on Neural Networks (IJCNN). Killarney, Ireland, July 12-17, 2015.

M. Nozari Zarmehri, Carlos Soares. *A Metalearning Framework for Model Management*. European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD). Porto, Portugal, September 7-11, 2015.

M. Nozari Zarmehri, Carlos Soares. *Using Metalearning to Predict the Trip Duration for Taxis*. Lecture Notes in Computer Science, Springer International Publishing, Volume 9385, Pages 205-216, ISBN 978-3-319-24464-8. 2015.

M. Nozari Zarmehri, Carlos Soares. *Collaborative Data Analysis in Hyperconnected Transportation Systems*. 17th IFIP Working Conference on VIRTUAL ENTERPRISES, Springer International Publishing. Porto, Portugal, October, 2016.

M. Nozari Zarmehri, Carlos Soares. *Using Metalearning for Prediction of Taxi Trip Duration Using Different Granularity Levels*. in preparation to submit to Neurocomputing Journal - Elsevier. 2016.

M. Nozari Zarmehri, Carlos Soares. *A General Metalearning Framework for Model and Data Granularity Selection*. in preparation to submit to Neurocomputing Journal - Elsevier. 2016.

M. Nozari Zarmehri, Carlos Soares. *Using Data Hierarchies to Support the Development of Personalized Data Mining Models: a Case Study in Error Detection in Foreign Trade Transactions*. under review, International Journal of Intelligent Data Analysis. April, 2016.

"Research is what I'm doing when I don't know what I'm doing."

Wernher von Braun

Contents

List of Figures

List of Tables

List of Abbreviations

3G	Third Generation
ACM	Association for Computing Machinery
AIFS	Arbitration Inter Frame Space
AODV	Ad hoc On-Demand Distance Vector
ATIS	Advanced Travelers Information Systems
BI	Business Intelligence
BM	Backbone Member
BO	Back Off
BPF	Back off-based Per-hop Forwarding
BRACIS	The Brazilian Conference on Intelligent Systems
C2C-CC	Car-to-Car Communication Consortium
CAS	Content-Addressed Storage
CBF	Contention Based Forwarding
CBMAC	Cluster-Based Media Access control
CH	Cluster Head
COIN	Clustering for Open IVC Networks
CRISP-DM	CRoss Industry Standard Process for Data Mining
CRM	Customer Relationship Management
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CTB	Clear to Broadcast
CTS	Clear to Send
CW	Contention Window
DB	Data Base
DBA-MAC	Dynamic Backbone-Assisted MAC
DCF	Distributed Coordination Function
DF	Distance Factor
DFS	Dynamic Frequency Selection
DIFS	Distributed Inter Frame Space
DSR	Dynamic Source Routing
DSRC	Dedicated Short-Range Communications
DM	Data Mining
DMwR	Data Mining with R

DRIVE-IN	Distributed Routing and Infotainment through Vehicular Inter-Networking
DT	Decision Tree
DV-CAST	Distributed Vehicular broadCAST protocol
DW	Data Warehouse
ECML-PKDD	European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases
ENIAC	Encontro Nacional de Inteligência Artificial e Computacional
ERP	Enterprise Resource Planning
ETSI	European Telecommunications Standards Institute
EU	EUropean countries
FCC	Federal Communications Commission
FEUP	Faculdade de Engenharia da Universidade do Porto
FN	False Negative
FP	False Positive
GHz	Gigahertz
GPS	Global Positioning System
GPSR	Greedy Perimeter Stateless Routing
GSR	Geographic Source Routing
HV-TRADE	History-enhanced V-TRADE
ID	IDentifier
IEEE	The Institute of Electrical and Electronics Engineers
IFIP	International Federation for Information Processing
IJCNN	International Joint Conference on Neural Networks
INE	Instituto Nacional de Estatìstica
IQR	Interquartile Range
ISM	The Industrial, Scientific and Medical radio bands
ISBN	International Standard Book Number
ISSN	International Standard Serial Number
ITS	Intelligent Transportation Systems
IVC	Inter Vehicle Communication
Kbps	Kilobits per second
LM	Linear regression
LOF	Local Outlier Factor
m	Meter
MAC	Media Access Control
MAN	Metropolitan Area Networks
MANETs	Mobile Ad hoc Networks
MAP-tele	Joint Doctoral Program in Telecommunications between Universidade do Minho, Universidade de Aveiro and Universidade do Porto (MAP)
MAS	Mobility-Assist Storage

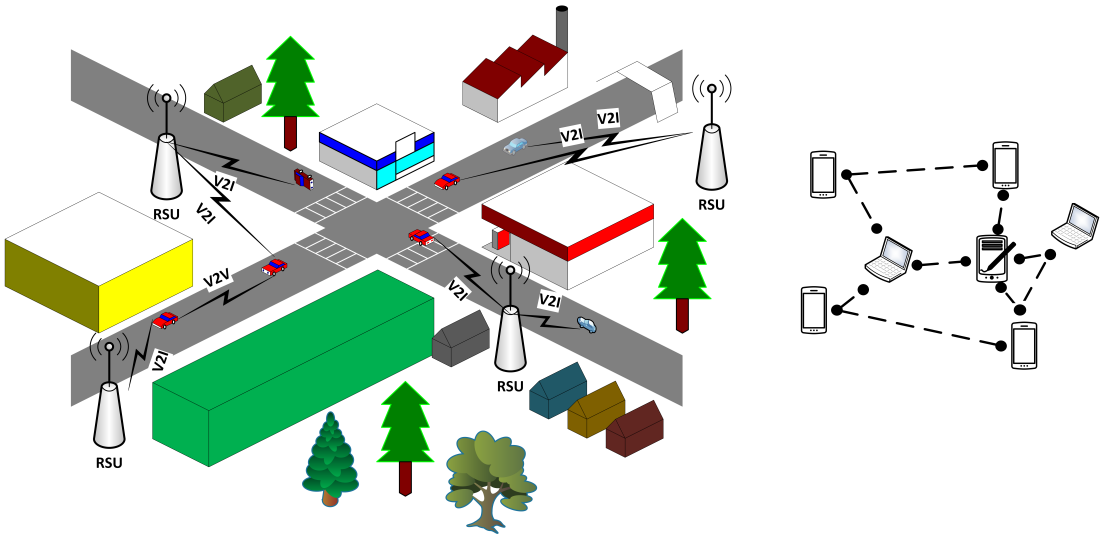
Mbps	Megabits per second
MDDV	Mobility-centric Data Dissemination algorithm for Vehicular networks
METAL	Meta-Learning assistant
MF	Mobility Factor
MHz	Megahertz
ML	Machine Learning
ML-RF	MataLearning-Random Forest
ML-DT	MataLearning-Decision Tree
MLT	Machine Learning Toolbox
mph	Miles Per Hour
MPR	Multi Point Relays
ms	Milliseconds
NRMSE	Normalized Root-Mean-Square-Error
NS3	Network Simulator 3
NV	Normal Vehicle
OBU	On-Board Unit
OFDM	Orthogonal Frequency-Division Multiplexing
PDR	Packet Delivery Ratio
PHY	Physical layer
PRAODV	Predicted AODV
PRAODVM	Predicted AODV with Maximum lifetime
QoS	Quality of Service
R2V	Roadside to Vehicle
RF	Random Forest
RMSE	Root-Mean-Square-Error
RSSI	Received Signal Strength Indication
RSSIF	Received Signal Strength Indication Factor
RSU	Road-Side Unit
RTB	Request to Broadcast
RTS	Request to Send
RZ	Risk Zone
SVM	Support Vector Machine
SVR	Support Vector Regression
TN	True Negative
TP	True Positive
TS	Time Slot
TTL	Time To Live
V2I	Vehicle-to-Infrastructure communication
V2V	Vehicle-to-Vehicle communication
VANETs	Vehicular Ad hoc Networks

VNC	Vehicular Networking Conference
WAVE	Wireless Access in Vehicular Environments
Wi-Fi	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Networks
WSN	Wireless Sensor Networks
UDC	Urban Data Collector
UMB	Urban Multi-hop Broadcast protocol
USA	United States of America
V-TRADE	Vector-based TRAcking DEtection
VANET	Vehicular Ad-hoc NETwork
ZOR	Zone Of Relevance
ZRP	Zone Routing Protocol

Chapter 1

Introduction

Vehicular ad-hoc networks (VANETs) are a class of Mobile Ad hoc Networks (MANETs) that has recently emerged. VANETs are naturally formed between moving vehicles provided with wireless interfaces, offering communication among nearby vehicles (vehicle-to-vehicle communication or V2V) as well as between vehicles and nearby fixed equipment (vehicle-to-infrastructure communication or V2I), usually described as roadside units (RSU).



(a) VANETs (b) MANETs

Figure 1.1: VANETs vs. MANETs

This kind of network differs from MANETs mainly in two aspects: high mobility of the nodes (cars) and restricted mobility along roads (Figure 1.1). As a consequence, VANETs have a highly volatile topology due to highly variable connectivity and communication range limited by the shadowing effect of buildings. In VANETs, the topology depends on the density and speed of cars, as well as on the environment. For example, in a highway scenario the speed of nodes is

higher and the node density lower than in an urban environment. However, in the former scenario, there are fewer obstacles and nodes do not change direction abruptly behind a building, as it can happen at a crossroad in a dense urban area.

VANETs are part of the communications infrastructure for Intelligent Transportation Systems (ITS), which are becoming an important type of information systems Chadwick et al. (1993). There are mainly three categories of ITS applications: public safety, traffic management, and infotainment. The overall goal of a public safety application is to reduce accidents, with many benefits beyond the obvious saving of lives. Examples of safety applications of ITS are: forward obstacle detection and avoidance Badal et al. (1994); turn accident warning and intersection collision warning Lee et al. (2009a). Using technology to improve the flow of traffic and reduction of congestion is the goal of traffic management. Smart traffic signs, rapid response to incidents, central traffic management, enhanced public transit systems and electronic toll collection are a few examples of traffic management applications Lee et al. (2006).

Another category of applications that could be provided in vehicles is infotainment. This kind of application provides additional information for drivers and entertainment to passengers like multimedia services, Internet connectivity and instant messaging. Also some other information provision services, like weather and parking services Lochert et al. (2008) belong in this category of VANET applications.

Additionally, vehicular networks offer various opportunities for gathering data about a city. Vehicles continually sense events from streets and process sensed data. Some examples using data gathered in real-time from the in-car sensors can be: Monitoring the fuel consumption from the in-car sensors in real-time can be used to identify areas of the city with high levels of pollution Liimatainen (2011); Zhou et al. (2013); Monitoring the brakes of cars to detect areas where drivers brake often without any apparent reason may be helpful in detecting dangerous conditions, like bad roads Murphy et al. (2006); Cheifetz et al. (2011); Detecting traffic jams and managing the traffic load in a different way Myr (2002); Improving traffic conditions Rudolf et al. (1997); Rescheduling traffic lights Zhou et al. (2010); Planing trip routes Yamamoto et al. (2002); Predicting trip duration of cars and public transports Balan et al. (2011); Mendes-Moreira et al. (2015); Eliminating traffic blind spots Keirstead (2004); Evaluating drivers behavior Noshadi et al. (2008a); Hasan et al. (1997); Measuring air quality and noise Lin and Yu (2008); Monitoring city environment Kamijo et al. (2000); Zhou et al. (2007). These examples show that using the in-car sensors as data sources and cars as data carriers, a macro vision of the city can be obtained.

Therefore, utilizing vehicular networks as an infrastructure for urban sensing is a cost-efficient way of deploying an urban monitoring system without actually deploying connected sensors Englund et al. (2015). Vehicles typically do not have energy constraints and can be equipped with powerful processing units, wireless transmitters, and sensing devices (GPS, detectors, video cameras, vibration sensors, acoustic detectors, car sensors, etc).

The gathered data can be processed and visualized live, enabling monitoring activities (Figure 1.2) and better decision making. However, this data can also be serve as the basis for predictive models that move the decision processes one step further. Given the availability of an increasing

amount of data, data mining approaches are being used to obtain models that are integrated into ITS applications (Figure 1.4) Thill (2000); Hauser and Scherer (2001); Chan and Marco (2004); Salim et al. (2007); Wang (2010); Qureshi and Abdullah (2013); He et al. (2014a,b).

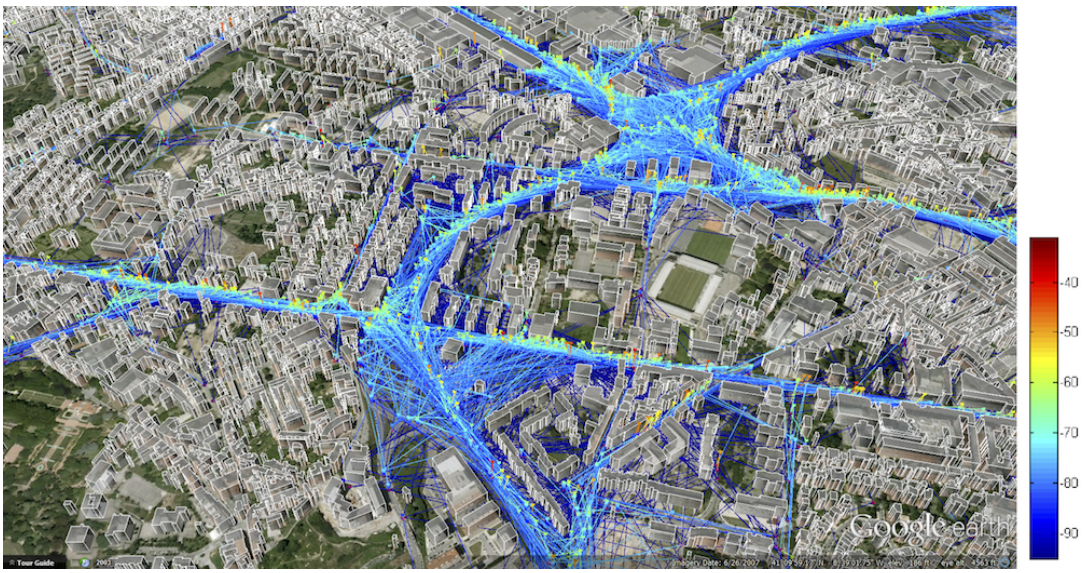


Figure 1.2: Snapshot showing a part of transmit-receive pairs in the city of Porto Boban et al. (2014)

1.1 Motivation

ITS services based on VANETs (Figure 1.3) require that massive amounts of data are gathered and transferred to a location for storing and making it available for applications that will analyze it (Figure 1.4). The existence of cars moving in the city can help collect this massive amount of data from the already implemented infrastructure without extra cost. To collect this data using VANETs, an efficient protocol is needed. An adequate data gathering protocol needs to use minimum communication time and network resources while providing low delay and high delivery rates.

Two important components of the infrastructure of VANET-based ITS applications are the data communication and management. Concerning data communication, the purpose is to provide the collected data to entities inside or outside the VANET in almost real-time. This corresponds to a system architecture where several or all nodes in the VANETs are data sources and the ultimate destination of the data lies outside the VANET in which it originated, whereby data can get there through one or more gateways.

On one hand, in a highly variable network (cars move in VANETs), unicast/multicast communication, in which a predefined route is selected for communication using continuously exchange neighbor information, is not feasible. On the other hand, in scenarios of high node density a

Figure 1.3: Intelligent Transportation Systems use cases and potential communication technologies, defined by ETSI ITS

broadcast communication impairs communication in VANETs. Therefore, a new solution for data gathering in VANETs is required.

Concerning data management for decision making, traditionally, data mining (DM) applications use a single model, created by applying algorithms on all data (or a carefully selected part of it). For example, a single model has been used to predict the trip duration in public transportations Mendes-Moreira et al. (2012). However, in a VANET, data is collected in different settings and the phenomena that is under analysis may also vary dramatically. Therefore, the best model may vary, which means that a different learning process must be used (i.e. different datasets, different parameter settings and/or different algorithms). This is consistent with what has already been shown in general for machine learning applications, namely that there is no commonly best algorithm for a broad domain of problems Wolpert and Macready (1997).

To have the best performance for each entity, different approaches can be used. To obtained the best performance, each entity can use its data or the data from its neighbors or even the data from whole network's entities for modeling. This decision can be made using different approaches. One of the traditional approaches is expert advice. They can suggest a specific setting for modeling including an algorithm and the part of data that should be used to have the best performance according to their experience. However, due to the variety of entities involved (i.e., different vehicles) and the variable nature of datasets, experts are unable to capture all the different possibilities

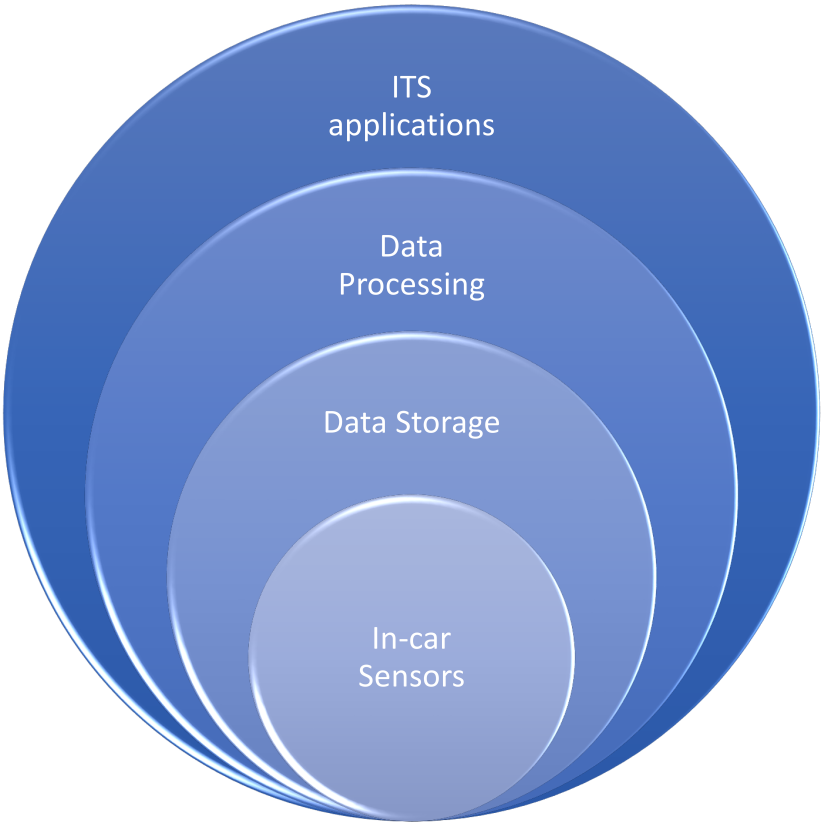


Figure 1.4: Data gathering hierarchy in vehicular networks

and the advice may end up being an inaccurate model. Another way of algorithm selection can be obtained by a trial-and-error approach. A set of algorithms is selected and applied to the datasets to obtain the best performance which needs a high computational cost. This approach may be helpful for small size datasets, but it is computationally costly for medium and large datasets.

An approach that has been used in several algorithm selection problems is metalearning Brazdil et al. (2003a); Soares et al. (2004, 1999); Torgo and Soares (2010). Metalearning models the relationship between the characteristics of the data with the performance of the algorithms. Given a new dataset, the (meta-)model is used to predict the algorithm that is expected to obtain the best performance.

1.2 Challenges

Data gathered from vehicles can be disseminated through the network in different ways depending on the application. For example, for a safety application, vehicle nodes broadcast a warning message and nodes that receive the message can use it and forward it by rebroadcasting. But for an infotainment application like games, the Internet and electronic toll collection, data needs to be sent unicast to a specific node or infrastructure.

For gathering data sensed by vehicles over VANETs in urban environment, the data from every node must be sent to one or more gateways (RSU or On-Board Units (OBU)) which are connected to a storage location to make the data available to applications outside the network. Due to high mobility of vehicles, being bounded by roads and dependent on density, vehicles cannot always communicate directly with each other or with a RSU. For getting a message to its final destination within the vehicle network, it must be forwarded by other nodes through several hops (multi-hops scenario). Since the wireless medium is a shared medium, it is necessary to keep the amount of messages in the network bounded. Therefore, a proper algorithm for data gathering is critical for efficient operation of a VANET-based urban sensor network.

All nodes in a VANET-based urban sensor network are simultaneously both data sources and potential relays for the messages from other nodes. It is yet unclear: what is the maximum amount of data that can be gathered from each node per unit time, which is the sensing capacity. Moreover, it depends on the data gathering protocol as well as on any mechanisms used for improving communication efficiency. In this respect, it must also be considered that a sensing application may tolerate a limited amount of losses, which can be traded off for efficiency mechanisms.

So a challenge is to provide a network protocol for efficiently (having high packet delivery and low delay) gathering data in an urban environment by vehicular nodes and making it available to outside applications.

In a city scenario, even by using an efficient algorithm, the amount of data gathered from different areas depends on the number of cars in that area. So it is probable to have a low amount of data or even no data for a certain area in the city. Therefore, this data may not be enough for decision making or creating a model.

Dealing with this challenge, the modeling can be made using the data from neighboring cars, the data in the nearest RSU, or even the data in the central point where the whole data is stored.

1.3 Research Objectives and Contributions

The most popular way to gather data from VANETs is broadcasting Li and Wang (2007). But broadcasting data in a dense network causes collisions, a problem also known as a broadcast storm problem, which causes severe impairment in the communications. The objective for data collection is to benchmark the performance of existing protocols in urban scenarios with different traffic densities and to propose a solution for the broadcast storm problem that addresses the requirements of massive urban sensing applications.

While all the VANET nodes regularly generate data for multi-hop communication over a shared medium, another objective is to address the problem of estimating the sensing capacity of a VANETs to determine the limits that a node should be considered for generating data.

The final goal of the project is to design a general data mining framework for data analysis, in which, by taking advantage of existing structure in the data, improve the performance of simple data mining and machine learning algorithms. Then, by extending the framework to use a met-

alearning, we try to reduce the computational costs and to improve the performance at the same time.

Therefore to meet the objectives of the project, the following contributions are done. The work is focused on improving data communications and management in VANET-based ITS applications.

To gather the data in servers that lie outside the VANETs in almost real-time, a very specific system – where there is no peer to peer communication, the data sources are distributed in the network (all network nodes are sources themselves), the ultimate destination of the data is a server outside the VANETs, that server is connected to the VANETs over one of several VANETs gateways (RSUs)– is required.

In this scenario, there are lots of data packet collisions. Especially, for the area far from the RSUs, the data packets should take a long multi-hop path to RSUs. So the probability of collisions increases. Therefore to design a protocol for this purpose, reducing the number of collisions should be taken into account for increasing the efficiency.

In this direction, the following contributions are done:

1. Design and implement a new protocol for urban sensing and data collection (Chapter 3).

2. Performance evaluation of the proposed protocol and evaluate different broadcast suppression techniques (Chapter 4).

3. Calculate the sensing capacity and validate it with simulation (Chapter 5).

Consequently, the collected data should be managed by choosing the right piece of data and the right algorithm to build a model for a particular unit in a specific area, taking into account the specificities of the VANET.

So an experimental framework was proposed and evaluated on different datasets, including a dataset unrelated to VANETs but with some common properties which made it adequate for this purpose.

Finally, the framework is applied to the data from a real world implementation (the DRIVE-IN project Cmuportugal.org (2014)) which was collected in an urban area from 440 taxis in the city of Porto.

The main contributions for this phase were:

4. Investigate the use of existing hierarchies in the data to improve the performance of the learning process (Chapter 6).

5. Propose a metalearning framework for data hierarchy level and algorithm selection (Chapter 7).

6. Evaluate the framework by applying it on different datasets (Chapters 8 and 9).

Chapter 2

State of the Art

In this chapter, we will summarize the state of the art related to the vehicular networks and machine learning and data mining. The standard and technologies which are used in VANETs are described in Section 2.1.

2.1 Technologies and Standards

The advent of Dedicated Short Range Communications (DSRC) makes the scenarios depicted in the previous chapter realistic in a near future. In October 1999, the United States Federal Communications Commission (FCC) allocated 75 MHz of the spectrum in the 5.9 GHz band to be used by Intelligent Transportation Systems (ITS) in the USA. DSRC supports high vehicle speeds, a transmission range of up to 1000 m, and default data rate of 6 Mbps (up to 27 Mbps) in that frequency band Commission (2015).

In Europe, in August 2008 the European Telecommunications Standards Institute (ETSI) allocated 30 MHz of spectrum in the 5.9 GHz band for ITS ETSI Headquarters (2008). In addition, the Car-to-Car Communication Consortium (C2C-CC) was created by car manufacturers with the main objective of increasing road traffic safety and efficiency by means of inter-vehicle communication Consortium (2008). The main goal of C2C-CC is to enable the different brands of cars to communicate with each other and with RSUs. IEEE has recently released the 802.11p standard for adding wireless access in vehicular communications iee (2013). The IEEE 1609 family of standards for wireless access in vehicular environments (WAVE) is a higher layer standard based on IEEE 802.11 iee (2013). WAVE is ensured the secure communication for V2V and V2I by introducing the standards and architectures. The IEEE 1609 family consists of four sub-protocols as follows:

1. IEEE 1609.1: it introduces the data and services within the WAVE architecture and also the format of messages and data storage that are required to communicate between different components of the WAVE architecture.

2. IEEE 1609.2: it defines the security issues to exchange messages.

3. IEEE 1609.3: it consists of all network and transport protocols required for supporting secure data exchange in WAVE.

4. IEEE 1609.4: it is an improvement of 802.11 Media Access Control (MAC) to support the WAVE functionalities.

2.2 VANET Standards

The DSRC spectrum is divided into 8 channels: one 5 MHz channel reserved for future use and 7 channels with 10 MHz bandwidth. Channels 174, 176 and channels 180, 182 can be aggregated to form 20 MHz channels, 175 and 181, respectively. Channel 178 is the control channel. Channels 172 and 184 are reserved for V2V advanced accident avoidance and high power public safety applications, respectively. The rest is available for both safety and non-safety channels. The channel allocation of DSRC in North America and Europe is shown in Figures 2.1 and 2.2, respectively.

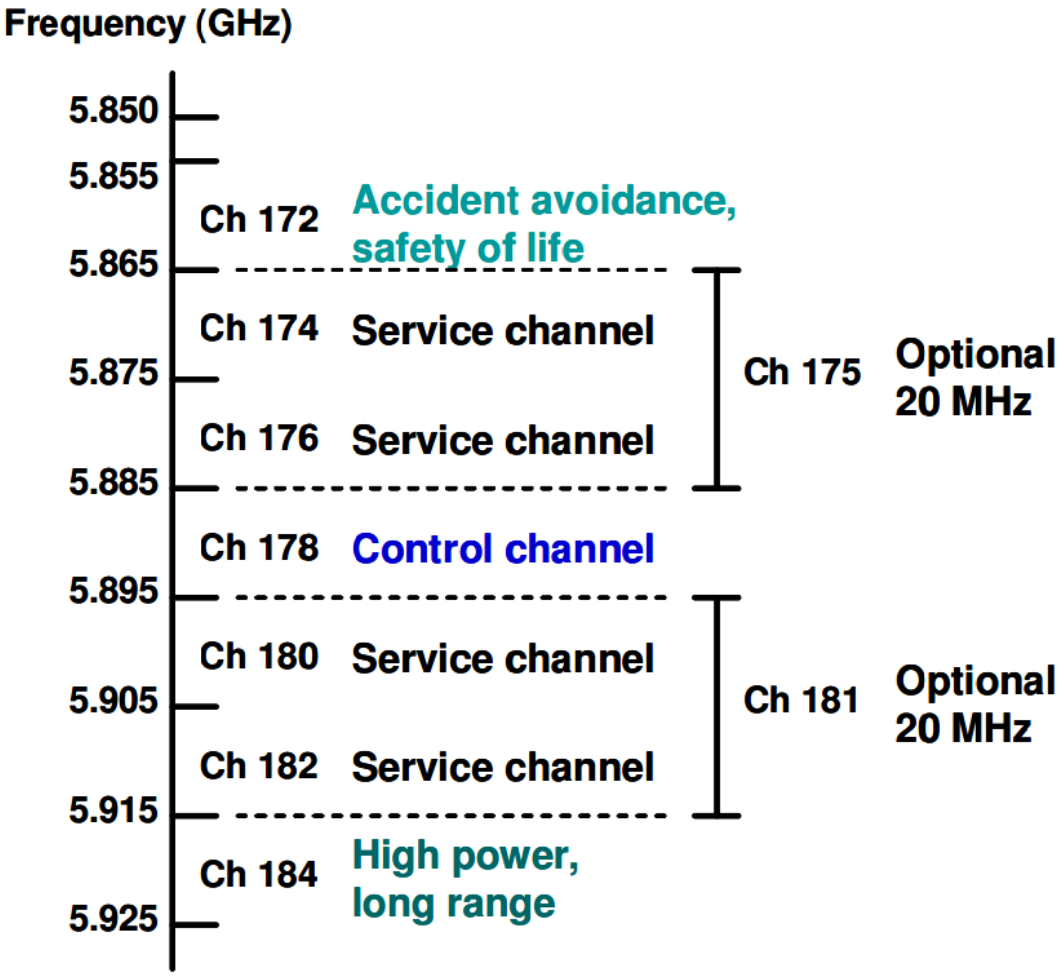


Figure 2.1: DSRC Channel allocation in north America Qian and Moayeri (2008)

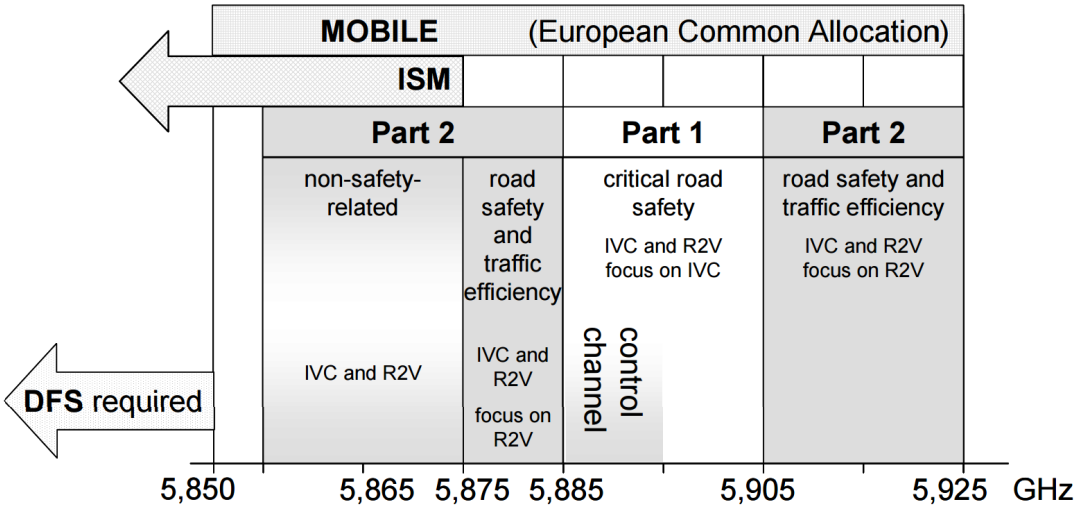


Figure 2.2: Frequency allocation in Europe Consortium (2008)

The 802.11p protocol uses Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) of 802.11 and the QoS supplement of 802.11e iee (2002). A node wishing to transmit should listen to the channel during the Arbitration Inter Frame Space (AIFS) (in 802.11e)/ Distributed Interframe Space (DIFS) (802.11a) and if the channel is busy during the listening period, then it performs backoff. The backoff procedure has two variables: Contention Window (CW) and Backoff count (BO). If the channel is busy, the node selects a random number between 0 and the CW ($[0, CW]$) for BO and only if the channel is free, it counts down the BO. If another station starts transmitting, the waiting nodes stop their counting down of BO and will do the same after the channel becomes free. Initially and after each successful transmission, CW is set to CW_{min} and after each unsuccessful attempt, it is set to $min\{2 \times CW + 1, CW_{min}\}$.

At the physical layer, IEEE 802.11p is essentially based on OFDM PHY defined for 802.11a iee (1999). But there are some differences between the two standards at the physical layer, as presented in Table 2.1.

Table 2.1: Differences between 802.11p and 802.11a

Parameter	802.11p	802.11a
Channel Bandwidth (MHz)	10	20
Data Rate (Mbps)	3 to 27	6 to 54
Slot Time (μs)	16	9
Air propagation time (μs)	< 4	$\ll 1$
CW (slots)	$[15, 1023]$	$[15, 1023]$
Latency	$< 50ms$	*Seconds*
Range (m)	< 1	< 100
Mobility (mph)	< 60	< 5

2.3 Data gathering

VANETs are an emerging technology which adds the capabilities of the new generation of wireless networks to vehicles. In-car users want to have connectivity to other networks like the internet. Enabling Intelligent Transportation Systems (ITS) for vehicle needs vehicle-to-vehicle and also vehicle-to-infrastructure communication. To collect data from VANETs, we need an efficient protocol. An excellent data gathering protocol needs to use minimum communication time and network resources while still providing low delay and high delivery rates. There are different approaches to making data from a node reach a gateway node: one is to have data delivery routes between each node and a gateway node, created and maintained by routing protocols; or data can be forwarded to a gateway node using hop-by-hop decisions, according to which one or more nodes forward packets until the gateway is reached. The latter class can be further divided according to the destination of the data, into data dissemination protocols, which deliver all data to all nodes, including the gateway nodes, configuring a network broadcast; and unicast/multicast/-geocast protocols, which deliver data to a subset of nodes only. We will present, in the following section, the state-of-the-art on these two types of protocols.

2.4 Routing Protocols

Routing protocols can be classified in different ways. In our scheme routing protocols have two categories, discussed in the following: Ad hoc Routing Protocols and Cluster-Based Routing Protocols.

2.4.1 Ad hoc Routing Protocols

Ad hoc On-Demand Distance Vector (AODV) Perkins and Royer (1999) and Dynamic Source Routing (DSR) Johnson et al. (2007a) are reactive protocols originally designed for MANETs. A number of studies have simulated and compared the performance of these protocols for VANETs Manish (2004); Wang et al. (2005a). One study shows that AODV does not have a good performance in the case of finding, maintaining and updating the routes in the VANET due to high mobility Wang et al. (2005a). Another study describes a highway scenario with few hops communication where the created route with AODV probably will break due to the high mobility of cars Manish (2004). To decrease this effect, the authors introduce the prediction-base AODV protocol: Predicted AODV (PRAODV) and Predicted AODV with Maximum lifetime (PRAODVM). These protocols use the speed and location information of nodes to predict the link lifetime. PRAODV creates an alternative route before the end of current link lifetime and PRAODVM, unlike AODV and PRAODV that use the shortest path, uses the path with longer lifetime through the multiple paths. But these methods depend on the accuracy of the prediction method.

2.4.2 Cluster-Based Routing Protocols

In this kind of routing protocol, a virtual network infrastructure is created by clustering the nodes in order to improve scalability. Each cluster has a cluster head which is responsible for intra- and inter-cluster communication. Nodes in a cluster can communicate directly, but communication between clusters is performed via cluster head. Many cluster-based routing protocols have been studied in MANETs Lin and Gerla (1997a); Jie Wu (1999). But due to different behavior of VANETs, these techniques are very unstable in VANETs and clusters created by these techniques are too short-lived. This issue has been addressed in the Clustering for Open IVC Networks (COIN) algorithm citeBlum2003. In this algorithm, the election of the cluster head is based on the dynamics of the vehicles instead of their IDs, as in traditional clustering methods. They show that COIN produces much more stable structures in VANETs by introducing lower overhead. COIN increases the average cluster lifetime by at least 192%.

Alternatively, a cluster-based approach to improving packet delivery ratio was proposed Gunter et al. (2007). It reduces the number of packets that are broadcast in the network. The authors have proposed a medium access scheme (CBMAC) for VANETs which is based on clustering of the vehicles. Their approach minimizes the hidden nodes problem by introducing clusters to provide better scalability. Also in this scheme, as the cluster head (CH) can assign bandwidth to the members of the cluster, fewer collisions occur, therefore increasing the reliability of the VANET. They form the cluster by using the following algorithm: each node can have four states: Undecided, Member, Gateway and Cluster Head. A node can enter the gateway state if it is a member of more than one cluster. Furthermore, all nodes have a neighbors table which is updated by exchanging hello messages. The authors evaluated CBMAC using simulation and compared their protocol with IEEE 802.11 MAC, showing that the loss rate is significantly lower. In addition, the results show that the protocol is able to form stable clusters for low and medium traffic densities.

2.4.3 Per-hop Forwarding

In this category, upon receiving a packet, each node analyzes it and decides to forward the packet by using broadcast (dissemination), unicast, multicast or geocast. Here nodes do not know about all paths from sender to destination and try to reduce the delay, packet travel time or increase the packet reachability. We categorize this into two different categories; first, all protocols that use broadcast for packet sending; second other protocols that use unicast, multicast or geocast for sending the packets. Depending on the knowledge of the node about its neighbors, we can divide these categories into two subcategories. When a node has information about the neighbors and decision for forwarding is based on that information, we put it in the “Sender-based” sub categories. When nodes have no information about their neighbors to decide for forwarding, we put it in “Receiver-based” sub categories. We review some of related work according to our classification.

2.4.4 Data Dissemination

VANETs can be used to gather and distribute sensing information for traffic management, safety and commercial application in urban environments. One of the challenges in these environments is how to retrieve data from the network. Vehicles gather information from sensors and send it across other vehicles to infrastructure nodes, which function as gateways.

Relaying has been used to disseminate data Wu et al. (2004a). The authors proposed the MDDV (Mobility-centric Data Dissemination algorithm for Vehicular networks) approach, designed to address the data dissemination problem in partitioned and highly mobile vehicular network. Since no end-to-end connectivity is assumed, the intermediate nodes store broadcast messages and forward them along a predefined trajectory geographically towards the destination. The message dissemination information, for instance, source id, source location, generation time, destination region, expiration time and forwarding trajectory is specified by the data source and is placed in the message header. This technique improves the packet delivery ratio by allowing multiple vehicles to forward the message.

Another approach uses geographical information for routing to improve packet delivery with less delay in urban area Lee et al. (2006). The authors propose two architecture models that consider the location and number of infrastructure nodes in the network. They also take into account the traffic density factor: the more vehicles, the more infrastructure nodes. Each infrastructure node is responsible for its area. They used a publish-subscribe system to forward the events through the network. Nodes interested in specific events must register themselves on infrastructure nodes. In the case of new event, it is first forwarded to the infrastructure node that is responsible for that area. The infrastructure node can then forward the event only to the interested nodes, based on their registered interests. The first architecture is Content-Addressed Storage (CAS) that takes advantage of infrastructure nodes by hashing the key of an event to a specific infrastructure node. The second architecture is Mobility-Assist Storage (MAS), which opportunistically disseminates events by "relaying" or sending events only to one's neighbors. CAS is appropriate for time-critical applications and MAS for delay tolerant applications.

Another approach for data dissemination in VANETs is Mobility centric Data Dissemination in VANETs (MDDV) Wu et al. (2004a). The authors use a combination of three types of forwarding: opportunistic, geographical and trajectory based forwarding. The vehicles know the road topology and their own location in the network. The source nodes specify the source ID, source location, generation time, destination region, expiration time and forwarding trajectory. The forwarding trajectory is the path between the source and the destination region. They use the geographical forwarding to send the packet from the source to destination. By geographically forwarding, the MDDV tries to move the messages closer to destination. The MDDV determines who has to forward the packets, when they have to forward them and when they can drop a packet.

Another protocol similar to CBMAC Gunter et al. (2007) is Dynamic Backbone-Assisted MAC (DBA-MAC) protocol Bononi and Di Felice (2007) that uses cross layer MAC and clustering for broadcasting the messages in VANETs. They assumed that alarm messages include: 1)

a direction of propagation, 2) a maximum time-to-live (TTL) and 3) a risk zone (RZ). Only nodes in this area forward the messages. The vehicles can have two states: normal vehicle (NV) or backbone member (BM) and each BM maintains the backbone record, namely information about vehicle ID, last hop and next hop. Creation of backbone has performed by broadcasting the BEACON messages. In addition, their approach allows the fast advertisement propagation of the alarm messages in the risk zone. Although all transmissions are broadcast, they use unicast along the backbone. The results of performance evaluation of their approach, when compared with IEEE 802.11 DCF, showed general advantages in performance, reliability and overhead reduction.

2.4.5 Unicast/Multicast/Geocast

2.4.5.1 Sender-Oriented

Using geographical position information in VANETs is more common and routing protocols that use this information have higher performance than topology based protocols like AODV and DSR Liu et al. (2004a); Füß ler et al. (2002a). One of the preliminary protocols in this category is the greedy routing protocol Bernsen and Manivannan (2008a) that always forwards the packet to the closest node to the destination. Another known protocol in the literature is GPSR (Greedy Perimeter Stateless Routing) Karp and Kung (2000a). This protocol consists of two different forwarding methods: greedy forwarding and perimeter forwarding. The packet consists of the destination location and forwarding nodes can select the next hop by greedy selection of the closest node to the destination. In this method, a beaconing algorithm is used for determining the neighbor position. In the case that the node does not have a neighbor closest to the destination but there are some nodes farther in geometric distance to destination, this protocol will use the perimeter forwarding which uses the right-hand rule to select the next hop.

As an alternative, the GPSR protocol has been used in a simulation of a highway scenario Füß ler et al. (2002a). It was shown that this geographical protocol achieves better performance when compared to the DSR protocol because of fewer obstacles in the highways. But in an urban environment, GPSR has some problems: first, greedy direct communication is not possible because of existing building and trees; secondly, packets need to travel a longer path with longer delay than with the DSR protocol; finally, packets are sometimes forwarded to wrong paths, causing longer delays. Lochert et al. Lochert et al. (2005a) proposed Geographic Source Routing (GSR), which uses the city digital map to get the destination position. By combining the geographical routing and knowledge of the city map, GSR has better average delivery rate, smaller total bandwidth consumption and similar latency of first delivered packet than DSR and AODV in urban areas.

Geocast routing Karp and Kung (2000a) is basically a location-based multicast routing. The objective of a geocast routing is to deliver the packet from a source node to all other nodes with a specified geographical region (Zone of Relevance, ZOR). A simple geocast scheme has been proposed to avoid packet collisions and reduce the number of re-broadcasts Briesemeister et al. (2000). When a node receives a packet, it does not rebroadcast it immediately but has to wait some waiting time to make a decision about whether to do it or not. The waiting time depends on

the distance of this node to the sender. Thus mainly nodes at the border of the reception area take part in forwarding the packet quickly. When this waiting time expires and if it does not receive the same message from another node then it will rebroadcast this message. By this way, a broadcast storm is avoided and the forwarding is optimized around the initiating vehicle.

One of the protocols that has been designed is DV-CAST Tonguz et al. (2010a). This protocol has three major components: neighbor detection, broadcast suppression and store-carry-forward mechanism. They use hello messages to estimate the network topology and GPS information to determine the direction of vehicles for broadcasting the data, reducing protocol overhead and complexity. The simulation results show that the DV-CAST performs well in heavy traffic during rush hours and very light traffic during certain hours of the day and also is robust against various extreme traffic conditions.

In a very recent work Wu et al. (2010a), the authors have proposed a fuzzy logic based multihope broadcast for VANETs. In the protocol, a selection of the relaying node is done by neighborhood evaluation. Upon receiving the hello message from its neighbors, the node evaluates them according to distance, mobility and signal strength. Then it uses a fuzzy function to convert the numerical value to some non-numerical value to combine three different metrics for decision making. After measuring the numerical value from the received packet, they convert them to three factors: Distance Factor (DF), with three possible values (Small, Medium and Large), Mobility Factor (MF), with three different values (Fast, Medium and Slow) and Received Signal Strength Indication Factor (RSSIF), with three values (Bad, Medium and Good). For example they use these values for conversion to the three factor: distance Large: 1, Medium: 0, Small: 0, mobility Slow: 0.75, Medium: 0.25, Fast: 0 and RSSI Good: 0.5, Medium: 0.5, Bad: 0. In this step they combine three non-numerical values to determine the final ranking. This ranking is used to select the best neighbor for relaying, using the Min-Max method. This method consists of two phases: first, it selects the minimum value of three factors to determine the neighbors rank; second, it evaluates more than one neighbors with the same rank (Perfect, Good, Acceptable, non-acceptable, bad and very bad), and then it uses the maximum rule to select the best neighbor to relay the packet.

Then by using fuzzy logic, they convert the non-numeric value to a numeric value and use those values in their process to select the relaying node. “The higher the value is, the better the neighbor node will be.” Their simulation results show that in the case of some metric such as: Number of broadcast per packet, Packet reception and End-to-end delay their protocol has better performance when compared with other protocols, such as (Flooding, Weighted p-persistence Wisitpongphan et al. (2007a), Multi Point Relays (MPR) Broadcast Qayyum et al. (2002) and Enhanced MPR Broadcast Wu et al. (2010b)). Their protocol reduces the number of broadcast messages by selecting only a subset of neighbors to forward the messages.

2.4.5.2 Receiver-Oriented

Broadcast is very common in VANETs and has been used for data dissemination through VANETs and for finding an efficient route to destination in unicast protocols. A very simple way is flooding.

Each node rebroadcasts a packet after receiving it. In this way, we will have multi-hop communication when the final destination is far from the source node. Flooding has a good performance for small number of nodes in the network, but when the number of nodes increases, the performance drops exponentially due to the increased probability of collision.

The performance of broadcast protocols depends on node density. We can have three types of node density: dense area, sparse area and normal area. Each of those has some problems and there are some solutions to cope with them. In dense areas like urban areas at rush hours, one of the problems is collision, which is also known as the broadcast storm problem. To deal with this problem is VANETs, broadcast suppression techniques are needed. There are three types of broadcast suppression techniques: Weighted p-Persistence, Slotted 1-Persistence and Slotted p-Persistence broadcasting Tonguz* et al. (2007); Tonguz et al. (2006a). In weighted p-persistence broadcasting, upon receiving the packet from node i, node j checks the packet ID and rebroadcast the packet with probability p_{ij} if it receives the packets for the first time, otherwise it discards it. And the probability of broadcasting is obtained in a simple way from the distance between nodes i and j (D_{ij}) and average communication range (R):

$$p_{ij} = \frac{D_{ij}}{R} \tag{2.1}$$

In slotted 1-persistence broadcasting, upon receiving the packet from node i, node j checks the packet ID and rebroadcasts it at assigned timeslot TS_{ij}, if it receives the packets for the first time and has not received any duplicates before the assigned timeslot. Otherwise it discards the packet. TS_{ij} can be calculated by the following expression:

$$TS_{ij} = S_{ij} \times t \tag{2.2}$$

where t is the estimated 1-hop delay and S_{ij} is the assigned slot number and can be calculated by:

$$S_{ij} = N_s \times (\frac{1 - D_{ij}}{R}), \quad D_{ij} \leq R, \quad D_{ij} > R \tag{2.3}$$

and N_s is the predetermined number of slots.

The last broadcast suppression technique is slotted p-persistence that a combination of the two previous approaches. Upon receiving the packet from node i, node j checks the packet ID and rebroadcast the packet with probability p_{ij} at the assigned timeslot TS_{ij}, if it receives the packets for the first time and has not received any duplicates before the assigned timeslot. Otherwise it discards the packet.

In sparse areas, when a node has the packet and wants to broadcast it, maybe there is no node to relay it too. So it needs to store the packet and when it finds another node in the communication range, it can forward the packet to that node. This technique is known as store-carry-forward mechanism Briesemeister and Hommel (2000); Wisitpongphan et al. (2007b).

In normal traffic density areas (i.e. not very dense and not very sparse), every node has different connectivity. It is possible to have many neighbors. We can consider this case as a dense

area and use one of the three broadcast suppression techniques. On the other hand, it is possible that some nodes have very few neighbors and so they can use the store-carry-forward technique, which is typically used in sparse areas.

Furthermore, there are some other approaches that discuss the broadcast issue in VANETs in general. For example, Durrest et al. Durresi et al. (2005a) introduced BROADCOMM for emergency applications which is based on geographical routing and has a virtual structure. A highway is divided into some virtual section that moves with vehicle. They suggest IEEE802.11 for the MAC layer and 350-400 m as the length of each section. The first level of the hierarchy is all the nodes within a section and the second level is represented by nodes which are usually located close to the geographical center of the section, and which are called cell reflectors. The live time of cell reflector is defined for a certain time interval and they will handle the emergency messages coming from members of the same section, or close members from neighbor sections. The source node broadcasts the emergency message within its section. Cell reflectors will multicast the message to other cell reflectors and so on. If a cell reflector becomes a bottleneck, another node will be selected to become the second cell reflector and the traffic will be shared between the two cell reflectors.

The Urban Multi-Hop Broadcast protocol (UMB) Korkmaz et al. (2004a) is designed to overcome broadcast storm, packet collisions, and hidden nodes problems in an urban area. In this protocol, the source node tries to select the furthest node in the broadcast direction (directional broadcast) and then will send the messages to that node for forwarding. The nodes do not have any prior information about their neighbors. The authors proposed to install repeaters in all intersections which will forward the packets in all road segments. They used Request to Broadcast (RTB)/Clear to Broadcast (CTB) instead of RTS/CTS. The UMB protocol utilizes the channel very efficiently since the forwarding duty is assigned to only one node in the broadcast direction.

Vector-based TRAcking DEtection (V-TRADE) and History-enhanced V-TRADE (HV-TRADE) Yamada et al. (2002) are GPS-based message broadcasting protocols. The goal is to improve bandwidth utilization. The basic idea is similar to the unicast routing protocol Zone Routing Protocol (ZRP) Haas et al. (2002). Each node has a message to rebroadcast, send a position request to other nodes and wait for a certain period of time for a reply from its neighbors. Based on position and movement information received from its neighbors, their methods classify the neighbors into five different forwarding groups: same_road_same_direction_ahead, same_road_same_direction_behind, same_road_opposite_direction_ahead, different_road and same_road_opposite_direction_behind. For each group, only a small subset of vehicles (called border vehicles) is selected to rebroadcast the message. They show significant improvement of bandwidth utilization with slightly loss of reachability, because of fewer vehicles to re-broadcast the messages. The V-TRADE has a little higher bandwidth utilization than HV-TRADE, but HV-TRADE has better reachability than V-TRADE.

2.5 Algorithm Selection

2.5.1 Meta-Learning

The algorithm selection problem was formally defined by Rice in 1976 Rice (1976). The main question was that which algorithm has the best performance for a specific problem? The meta-learning started to shape in the late eighties Brazdil et al. (2009).

Finding the relevant meta-features for predicting the performance of the base-level algorithms is discussed in different research works Aha (1992a); Michie et al. (1994); Gama and Brazdil (1995); Brazdil (1998); Keller et al. (2000); Brazdil et al. (2003b). Meta-features may have information regarding the error-rate of base-level algorithms which is called landmarkers Bensusan and Giraud-Carrier (2000); Pfahringer et al. (2000).

Several project with the relevant results to the meta-learning have been launched. The first formal project in this area was MLT project Kodratoff et al. (1992). The MLT project create a special system called *Consultant-2* which can help to select the best algorithm for a specific problem. The next two projects in this area were: Statlog Michie et al. (1995) and METAL Brazdil et al. (2003a). In these projects, the level of adaptation was the main difference between meta-learning and the traditional base-learning approaches.

It may also be important to select the best base-level algorithm not for the whole data set, but rather for subset of the data set Brodley (1995) of even for the individual example Todorovski and Džeroski (2003). Tuning the parameter of specific base-level algorithm is another task that meta-learning can be helpful Soares et al. (2004). They try to tune the width of the Gaussian kernel. Rijn et al. van Rijn, JanN. and Holmes, Geoffrey and Pfahringer, Bernhard and Vanschoren (2014) have investigated the use of meta-learning for algorithm selection on Data Streams. Calculated Meta-features on a small data window at the start of the data stream provide information about the best algorithm. Meta-learning uses this information for the algorithm prediction in the next data windows. And finally, a survey of meta-learning as reported by the machine-learning literature is provided in Vilalta and Drissi (2002).

One of the main objective of research in the community is to develop a meta-learning approach which is able to deal with the increasing number of models. It can produce advice dynamically on the algorithm selection problem.

Dealing with this problem – the selection of different levels for different products –, meta-learning approach is useful. It maps the characteristics of the data with the ideal level of granularity Brazdil et al. (2009). DM models can be learned at different level of granularity. Consequently, the data characteristics can be calculated and used to determine the best level for each product Soares et al. (1999); Torgo and Soares (2010).

In 1976, John Rice used the term algorithm selection Rice (1976) which correlates the data characteristics of a specific problem with the performance of the algorithms. Characterization of a classification problem and its effect on algorithm performance is investigated by Rendell and Cho Rendell and Cho (1990). They use the size and concentration of the classes as features. This idea was extended in 1992 by Aha Aha (1992b). Aha creates rules for learning, i.e.,

if a given dataset has specific characteristics (C_1, C_2, ..., C_n) then algorithm A_1 should be selected.

The number of examples, number of classes, number of prototypes per class, number of relevant and irrelevant attributes, and the distribution range of examples and prototypes were the selected features.

On the far side of the task of algorithm selection problem, there are many other problems that the same idea can be derived. For example to select the best parameter settings (Levels in our case), a meta-learning approach can be used. In 2005, Ali and Smith (2005) utilized meta-learning to find the best kernel to use within support vector machines (SVMs). By changing the kernel, the algorithm changes and in results there will be different performance for each setting. Also an interesting framework for optimizing algorithm parameter by using meta-learning is presented in Duch and Grudzinski (2001).

In addition, a meta-learning algorithm for supporting the selection of learning algorithms is presented in Brazdil et al. (2003c). It uses k-Nearest Neighbor algorithm to determine the datasets that are most similar to the under evaluation dataset. They use ranking rather than classification as a new contribution.

Similar approach to the one presented in this section can be used for selecting the right level of granularity in our problem. Our method is described in the next section.

2.5.2 Trip Duration

There has been several research on the trip duration prediction. Kwon et al. (2000) use the flow and occupancy data from single loop detectors and historical trip duration information for to forcasting trip duration on a freeway. Using real traffic data, they found out that simple prediction method can have a good estimation of trip duration for trip starting in near future (up to 20 minutes) while historical data can help better for the trip which starts in more than 20 minutes. The same approach is used in Chien and Kuchipudi (2003). Zhang and Rice (2003a) uses a linear model to predict the short-term freeway trip duration. In their model, the trip duration is varied as a smooth function of departure time. Their results show that for a small data set, the error varies from 5% to 10% while for the bigger data, the variation is from 8% to 13%.

Support Vector Regression is used for prediction of trip duration in Wu et al. (2004b). They utilize real highway traffic data for their experiment. They suggest a set of SVR parameter for the prediction which is able to outperform other baseline trip duration prediction model. Balan et al. (2011) is a real-time information system that provide the expected fare and trip duration for passengers. They use historical data consisting of approximately 250 million paid taxi trips for the experiment. But the use of meta-learning for the prediction of trip duration is still missing.

In our knowledge, there is no previous work which uses meta-learning for trip duration prediction. Considering the rapid changing of behavior of vehicular network, using a single algorithm for forecasting the travel time will end in unreliable prediction. In addition, using Trial and error to find out the algorithm which fits well to the specific data set (data set for a specific vehicle and in a specific time) would be time consuming and probably is not useful.

Part I

Data Collection

Chapter 3

Data Gathering for Sensing Applications in Vehicular Networks

3.1 Introduction

Vehicular ad-hoc networks (VANET) were motivated mainly by safety and traffic management applications, followed by infotainment applications that provide an additional commercial utilization of the new communication infra-structure. Alternatively, we propose to use a VANET as the infrastructure for an urban cyber-physical system, an approach that has not been extensively explored so far.

Vehicles equipped with a wide range of sensing devices and the ability to communicate with each other offer a unique opportunity for gathering real-time data about a city, like traffic conditions, environmental parameters, video and audio for surveillance Gerla and Kleinrock (2011), or physical condition of the drivers Rodrigues et al. (2010a). A good overview of existing work on using vehicles or VANET for sensing can be found in Gerla and Kleinrock (2011). Existing VANET solutions either apply on-demand querying for local dissemination within the VANET Lee et al. (2009b), sometimes keeping the data in the location it pertains to Dikaiakos et al. (2007a), or rely on delay-tolerant networking and open Wi-Fi access points for sending the data to the Internet backbone Hull et al. (2006a). However, the first are inefficient for real-time traffic or environmental monitoring due to the query overhead and the need to globally access the data, and the latter cannot guarantee up-to-date data. Knowing the updated state of the various relevant variables for a city is necessary for applications such as navigation using real-time traffic information for regular and for emergency vehicles, or personal mobility and environmental monitoring.

The purpose of sensing in the sense of a cyber-physical system is to provide the sensed data to entities outside the VANET in almost real-time. This corresponds to a system architecture where several or all nodes in the VANET are data sources and the ultimate destination of the data lies outside the VANET, whereby data can get there through one or more gateways. This chapter proposes and evaluates a broadcast-based protocol for data collection over VANET.

In scenarios of high node density broadcast storms impair communication in VANET. Several algorithms have been proposed to mitigate them mostly in scenarios of safety message dissemination in highways, with some techniques focusing on reducing the amount of forwarders at each hop using probabilistic forwarding and suppression Tonguz et al. (2007, 2006b); Wisitpongphan et al. (2007c), some relying on using exchanged neighbor information to explicitly limit the number of forwarders Sepulcre et al. (2011a).

We consider that it is inefficient to continuously exchange neighbor information in a high-density volatile network for several reasons. First, there is the overhead of periodically exchanging the neighbor list. Second, additional mechanisms must verify whether the chosen forwarder actually forwards the packet. Third, another major reason for not using an explicit choice of a single forwarder is that, in urban scenarios, this choice would require knowledge of the road topology and car density towards the destination to avoid routing packets to a dead-end. And it does not seem feasible to do routing on the road topology on a packet-by-packet basis.

Instead, we take the approach of adding the geographic location of current forwarder and the destination to each data packet and use this information at the receivers to rank them as potential forwarders in a distributed fashion on a packet by packet basis. The potential forwarders are differentiated using back-off timers and suppression is used to limit the number of forwarders, extending existing techniques to the urban sensing scenario. We evaluate the proposed protocol using the NS3 simulator for large-scale simulation and compare its performance with well-known broadcast-based protocols in an urban setting. We analyze networking metrics, like the packet delivery rate, the end-to-end delay and the number of hops in the path, as well as the number of replicas that reach the destination, i.e. the redundancy added by the protocol, and the overhead in terms of total amount of packets created in the network.

The rest of the chapter is organized as follows: Section 3.2 describes the novel protocol and its parameters. The simulation setup is described in Section 3.3. Section 3.4 shows the results of the performance evaluation of the protocol, and finally Section 3.5 concludes the chapter.

3.2 Broadcast-based Data Gathering Protocol

The proposed protocol aims at collecting large amounts of data from sensors installed in vehicles in an urban environment, configuring a cyber-physical system for an urban area. We envision that vehicles move within the urban environment and collect information like pollution or traffic conditions and that the data generated in each node is periodically sent to a back-office using the VANET as sensing infra-structure. In this scenario, we have many-to-one communication pattern from sources to the final destination and we assume that each node knows its own geographical location information and that of the final destination. The goal of the data gathering protocol is to collect this data with high packet delivery ratio (PDR), limited delay and low amount of overhead using a VANET. It is more critical in a scenario where all nodes are data sources than in other VANET scenarios to avoid congestion collapse by limiting the amount of packets forwarded in the network.

We propose Back off-based Per-hop Forwarding (BPF), a data gathering protocol that uses the location information to select the forwarding order among the nodes receiving the packet by mapping it into back-off time, so that nodes likely to be nearer to the final destination have shorter back-off times. BFP has the following properties: 1) it does not require nodes to exchange periodic messages with their neighbors communicating their locations for keeping low the management message overhead; 2) it uses geographic information about the current sender and the final destination node in the header of each data packet to route it in a hop-by-hop basis; 3) it takes advantage of redundant forwarding to increase packet delivery to a destination. The novelty of this protocol is the use of the final destination for per-hop forwarding in a unicast urban scenario. It takes advantage of the geographic location of the destination to direct the forwarding towards the destination, being more efficient than destination-agnostic protocols commonly used for safety message dissemination. Moreover, it takes advantage of redundancy to be more effective than protocols that specify one single forwarder, since specifying one single per-hop forwarder in an urban environment requires additional knowledge of the full street map towards the destination, or the chances are high that a packet is routed to a dead-end or along a very long route.

3.2.1 BPF Protocol Design

Figure 3.1 illustrates the scenario in considered to explain the calculation of the per-hop back-off time: node i is the previous hop for node j and nodes 1, 2 and j are potential forwarders.

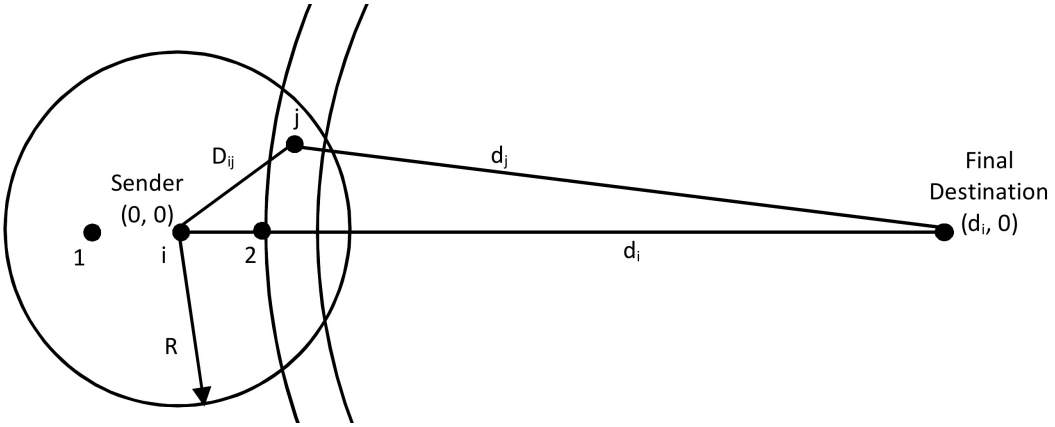


Figure 3.1: Node configuration used for calculations

The most straightforward choice for forwarding is the node geographically closest to the final destination Karp and Kung (2000b), but that information is not available when neighbor nodes do not exchange their locations with each other. So, the preferred forwarders are the nodes that represent the most progress from the previous sender, which are the nodes located closer to the end of the transmission range, which is taken by protocols like Contention Based Forwarding (CBF) or the 1-persistent broadcasting. We further reduce the number of forwarding nodes using the distance to the final destination in the component of the back-off calculation.

Usually, there are 2 constant values used to calculate the back-off time: D_{ij} and d_j. These values are the distance to the last hop and to the final destination, respectively. To analyze the effect of these two constant on the back-off time, we define two different components according to these values in this section: C_1 and C_2.

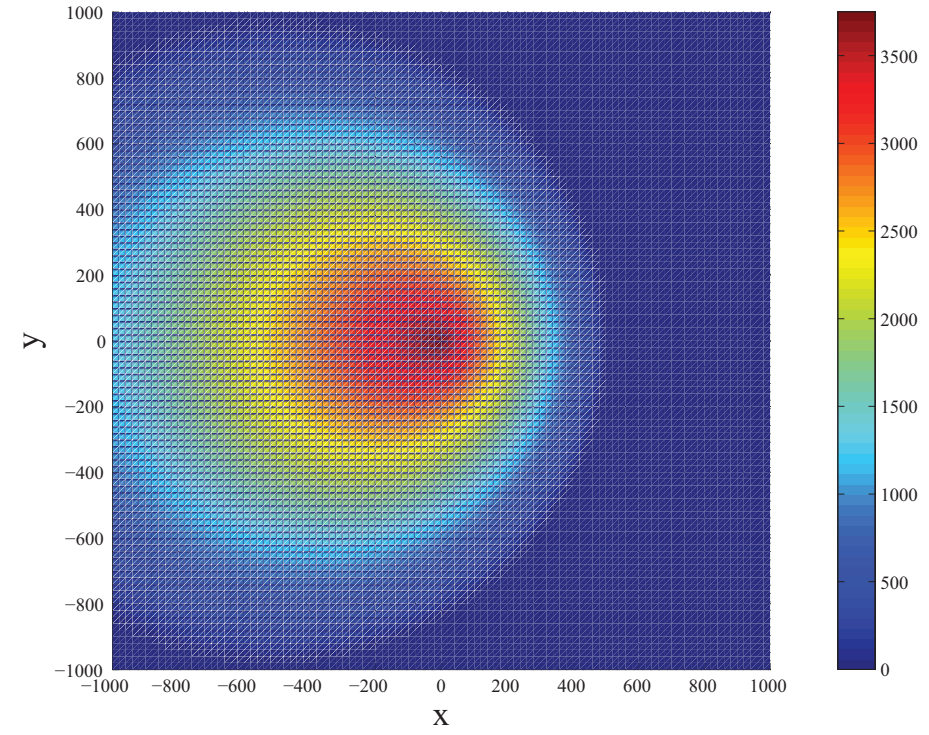
The first component is the distance between current node (j) and previous hop (i), D_{ij}, compared with the average communication range (R). By selecting nodes farther from the previous node to forward sooner than other nodes is a receiver-based greedy approach that makes packets travel the largest possible distance at each hop. We define the following component, which selects a node at the end of the communication range to have lower back-off time than other nodes:

$$C_1 = (1 - \frac{D_{ij}}{R}) \tag{3.1}$$

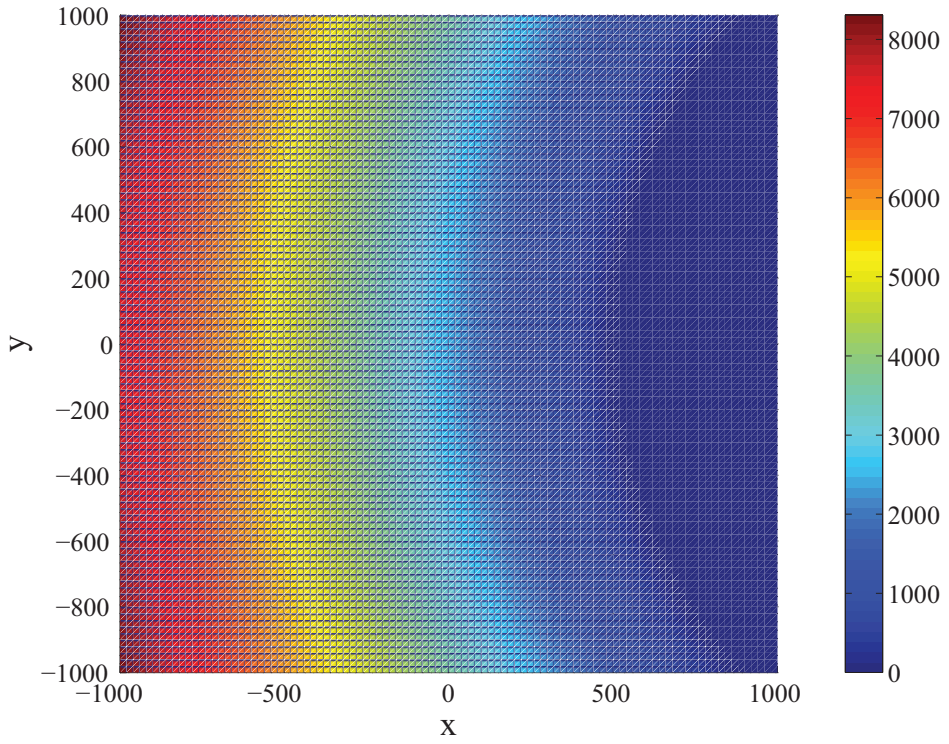
However, we wish to further concentrate the preferred forwarders in the direction of the final destination, since nodes at the end of the communication range in the opposite direction of the final destination can cause a useless increase in the number of replicas (in Figure 3.1 node 1 is in this situation). The second component in the calculation of the back-off time is the distance to the final destination, d_j, relative to the distance between the previous hop and the final destination, d_i:

$$C_2 = (1.0 + \frac{dj - di - R}{2R}) \tag{3.2}$$

Figure 3.2 shows the back-off time calculated as a combination of C_1 and C_2. In these plots, the previous node is located at $(0,0)$, the final destination is at $(2000,0)$ (horizontally to the right of the plot) and the communication range equals 500 m. The first plot depicts the back-off as an equally weighted sum of both components, while the second plot considers only C_2. The initial plot shows only little directionality towards the destination, because the C_1 component is dominant in the sum due to the fact that D_{ij} is much larger than $d_j - d_i - R$ except when the nodes are very close to the destination. Therefore, for the evaluation of the BPF, we only consider the calculation based on the C_2 component.



(a) C_1 and C_2



(b) C_2 only

Figure 3.2: Back off time in microseconds for different positions around a node located at $(0,0)$ according to C_1 and C_2. Location of the final destination $(2000,0)$ (horizontally to the right of the plot) and communication range is 500 m.

3.2.2 How to map the back-off value to time?

After calculating the back-off components, we need to map this value to the back-off time. Unlike other broadcast storm mitigation techniques Wisitpongphan et al. (2007c), we do not use the WAIT_TIME; we just use different back-off times to forward the packet and to distribute forwarding events at the time. The protocols in Wisitpongphan et al. (2007c) use a WAIT_TIME of 5000 μs to suppress as many duplicate packets as possible from previous forwarders. But we aim at forwarding the packet as fast as possible by the best-positioned nodes and cancel forwarding from nodes not so well positioned to reduce the amount of transmission in the network. So we give the shortest back-off time to the node with the most progress from the previous forwarder and that transmission will suppress forwarding on nodes with less progress to the destination.

In our protocol, the back-off value calculated from the components is between 0 and 1, and it is multiplied by 5000 μs which is the WAIT_TIME in the known broadcast suppression techniques Wisitpongphan et al. (2007c). So, the back-off time at any hop lies in the interval [0,5] ms. Note that this is the back-off time of the routing protocol and the MAC layer exponential back-off algorithm is run for every packet passed to the MAC layer.

3.2.3 Back off-based forwarding algorithm

The flow diagram of the per-hop forwarding algorithm executed in each node upon reception of a packet is shown in Figure 3.3. Each node upon receiving a packet, checks if it is the final destination node. If not, it checks if it received the packet before. If so, it cancels the forwarding event if the back-off time has not expired. In the case that it receives the packet for the first time, it calculates the back-off time and schedules the forwarding event on the back-off time and marks the packet as a received packet.

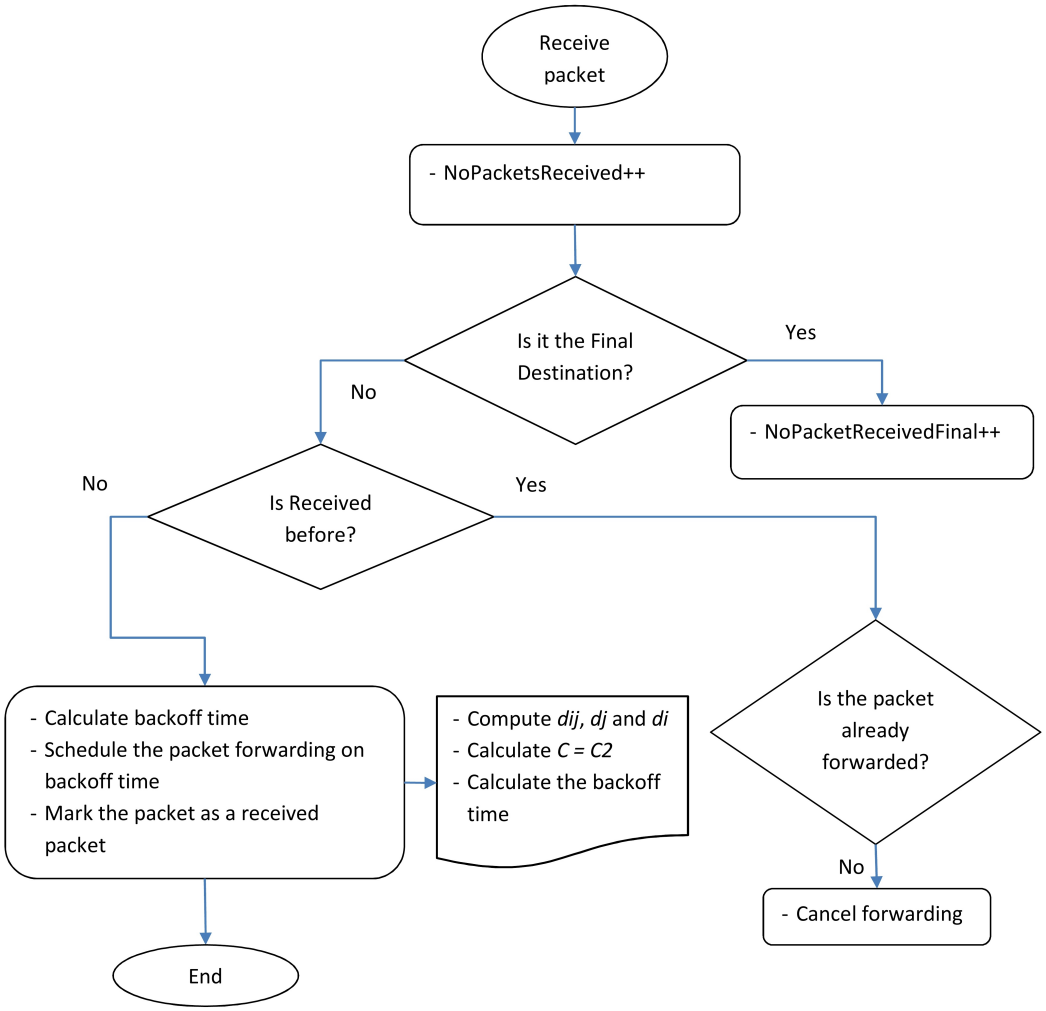


Figure 3.3: Flowchart of data gathering Protocol

3.3 Simulation

We used the Network Simulator 3 (ns-3) ns3 version 3.9. The topology used for movement of cars was Manhattan Grid with size 5×5 (from $(0m, 0m)$ to $(2500m, 2500m)$), so the simulated area was $2.5km \times 2.5km$ with $25km$ road length, and the final destination was located at $(1250m, 1250m)$ in the center of topology.

For the first results, and to keep feasible simulation durations, we simulate with a limited number of source nodes. We selected 8 source nodes located as far as possible in the topology. Node density is set to 2.4, 4.8, 7.2 and 9.6 nodes/km, totaling 61, 121, 181 and 241 nodes, respectively. The communication range has been set to 500 m, and each node had on average at least 2 nodes (low node density) and at most 20 nodes (high node density and at the intersections) within the communication range. Nodes move with an average speed of $14m/s$ and minimum speed of 3 m/s

without pause time. Each source node sends 512 Bytes packets with a rate of 5 packets/s (20$kbps$) and the simulation time is 200 seconds.

The underlying MAC protocol is set to 802.11p with PHY data rate equal to 6 Mbps and channel bandwidth is 10 MHz. The propagation loss model used in the simulation environment was Nakagami-m Propagation Loss Nak (July 13, 2011); Shigehiko (2003) with $m = 1.55$ which is the recommended value for urban environments Rubio et al. (2007a).

Each simulation configuration is done for 4 different protocols: BPF using only C_2, weighted p-persistence, slotted 1-persistence and slotted p-persistence with $p = 0.5$. For each combination of above parameters we ran 10 independent simulation runs and the results show the average and 95% confidence interval for each metric.

3.4 Performance Evaluation

For evaluating the performance of the BPF protocol we use the following metrics: packet delivery ratio (PDR%), the end-to-end delay between source nodes and the final destination, number of hops in the path, and number of replicas of a packet that reach the final destination.

3.4.1 Packet Delivery Ratio

Figure 3.4 shows the packet delivery ratio between sources and the final destination for the 4 mentioned protocols at four different traffic densities. BPF achieves higher end-2-end PDR% than any of the other 3 protocols. In areas of low node density, all protocols have the same PDR% performance because there are few nodes within the communication range to forward the packet and in many cases forwarded packets end in a dead-end. As the node density increases, BPF shows increasingly better behavior than other protocols.

The BPF protocol significantly improves the end-to-end PDR% in high node density by leveraging packet redundancy in the network. This effect overwhelms the additional collisions caused by the redundant forwarding.

As shown in Figure 3.4, the PDR% increases from 8% to 78% for the BPF protocol when the node density increases from 2.4 to 9.6 nodes/km, which is 85% more than the PDR% of the second best protocol in a well-connected network in high node density. Moreover, the PDR% increasing tendency is higher than that of the other 3 protocols, which seem to start saturating at the maximum node density simulated.

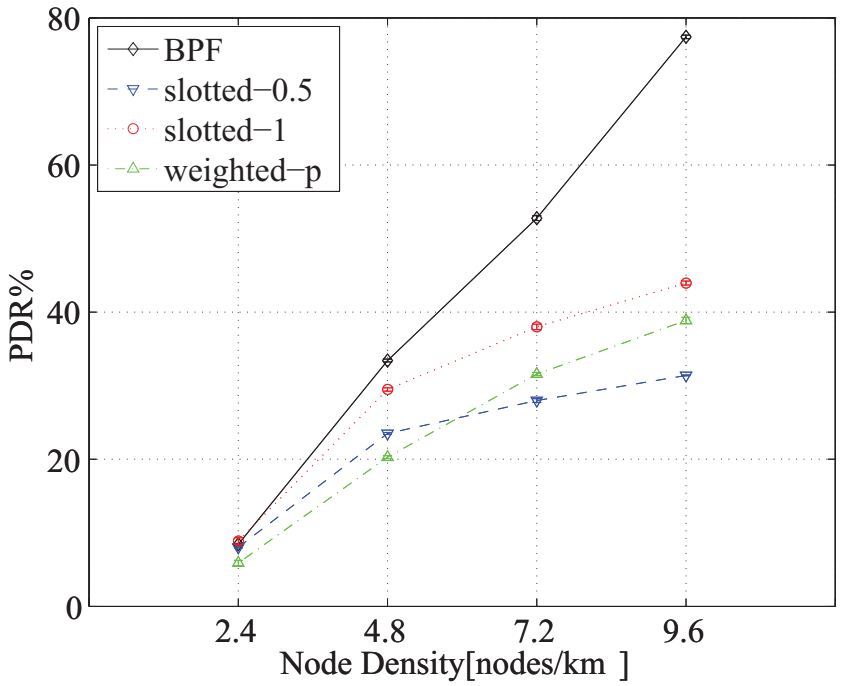


Figure 3.4: PDR% for 4 different protocols with 8 sources (20*kbps*)

3.4.2 End-to-End Delay

Figure 3.5 shows the end-to-end delay between sources and the final destination, which is an important metric since we aim at providing up-to-date data about a city in a timely manner. BPF also has a lower end-to-end delay when compared with the 3 other protocols, mainly because it does not have a WAIT_TIME of 5 ms on each hop, as do the other protocols.

When node density increases, the number of collisions increases because there are more nodes in the communication range of any node, and the probability of having a back-off time near 0 increases because of the higher number of nodes at the end of communication range.

On the other hand, the broadcast mitigation protocols can deal better with this because they use the WAIT_TIME before forwarding, sender nodes receive more duplicate packets from neighbor nodes before forwarding and choose the nearest node to itself for its calculation, so the probability of collision and the delay decrease. This slump is more significant for the protocols that use the probability for forwarding (slotted-0.5 and weighted-p). The protocols will not forward the packets with probability $1 - p$, reducing the probability of collisions, but enough other nodes forward the packets and the delay will decrease.

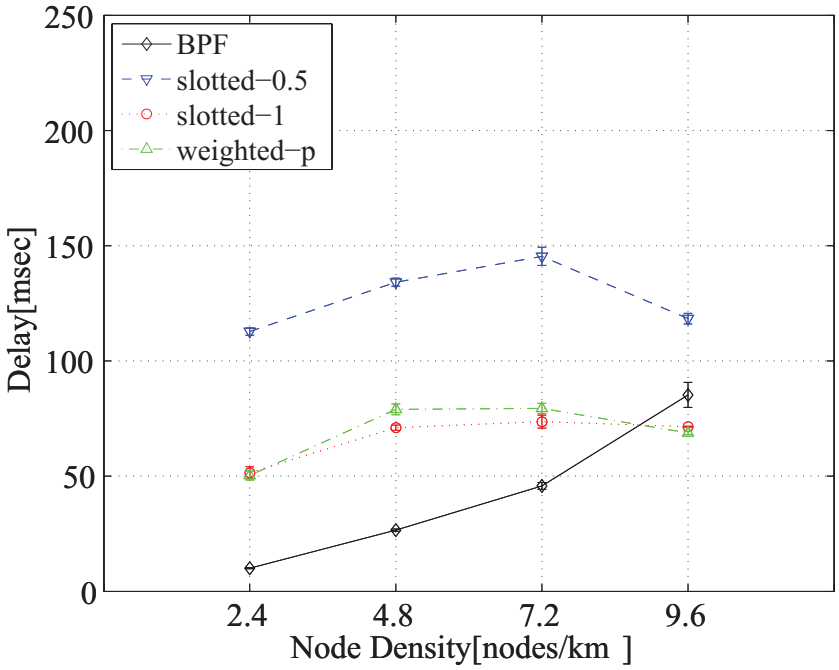


Figure 3.5: Delay for 4 different protocols with 8 sources (20$kbps$)

3.4.3 Number of Hops and Amount of Replicas

Figure 3.6 shows the number of hops from source to the final destination. The number of hops is the same for all the protocols for different node density because all of the protocols try to choose the nearest node to the final destination only in different ways.

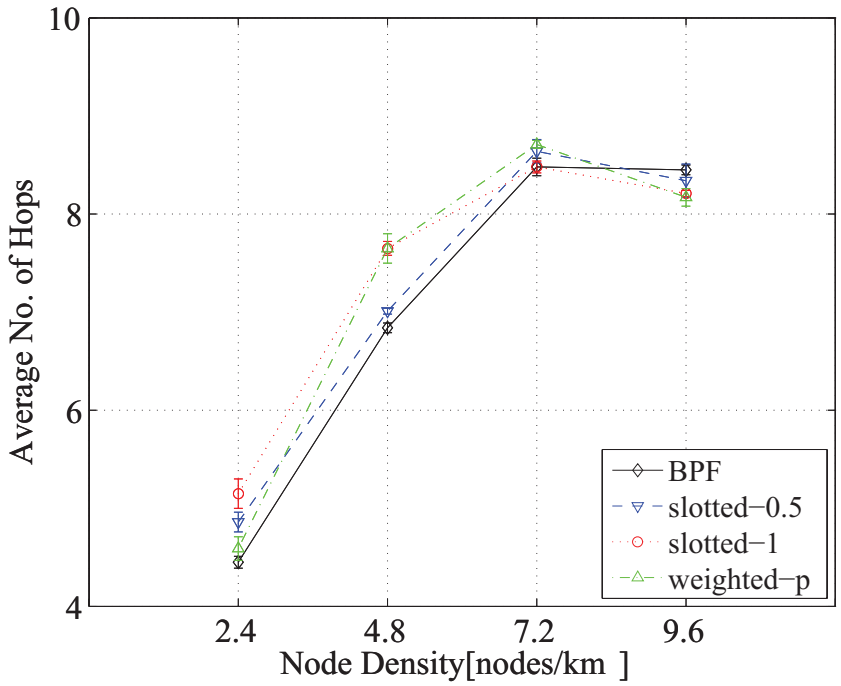


Figure 3.6: Number of hops for 4 different protocols with 8 sources (20$kbps$)

As discussed before, BPF distributes the back-off time and gives shorter back-off time to the nodes which are nearer to the final destination, trying to reduce the number of replicas through suppression. The other protocols do it by using WAIT_TIME and allowing for the reception of all possible packets with the same ID from neighbor nodes and then using one of those packets to calculate whether or when to forward the packet. As Figure 3.7 shows, the BPF without using WAIT_TIME produces the same number of replicas at the final destination, showing that our protocol achieves higher PDR% with the same redundancy as the other protocols, i.e. it is more efficient.

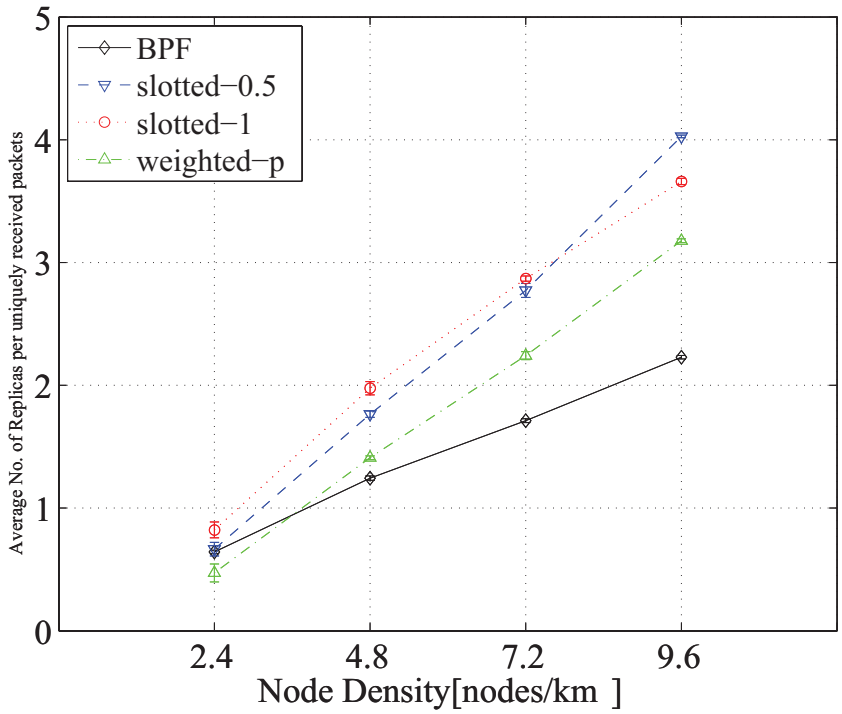


Figure 3.7: Average Number of replicas per uniquely received packets for 4 different protocols with 8 sources ($20kbps$)

3.4.4 Scaling Source Nodes

Since we envision a scenario where all nodes can be data sources, Figure 3.8 shows the PDR% for increasing percentage of nodes being network sources in the highest node density (9.6 nodes/km) scenario previously considered. As expected, as the number of source nodes in the network increases, the PDR% decreases due to increasing network congestion. Nevertheless, the proposed protocol shows a higher PDR% in all situations, showing a higher efficacy. As a future work, we will study a way to increase the PDR% for high node density having all nodes as source nodes by decreasing the number of useless forwarding.

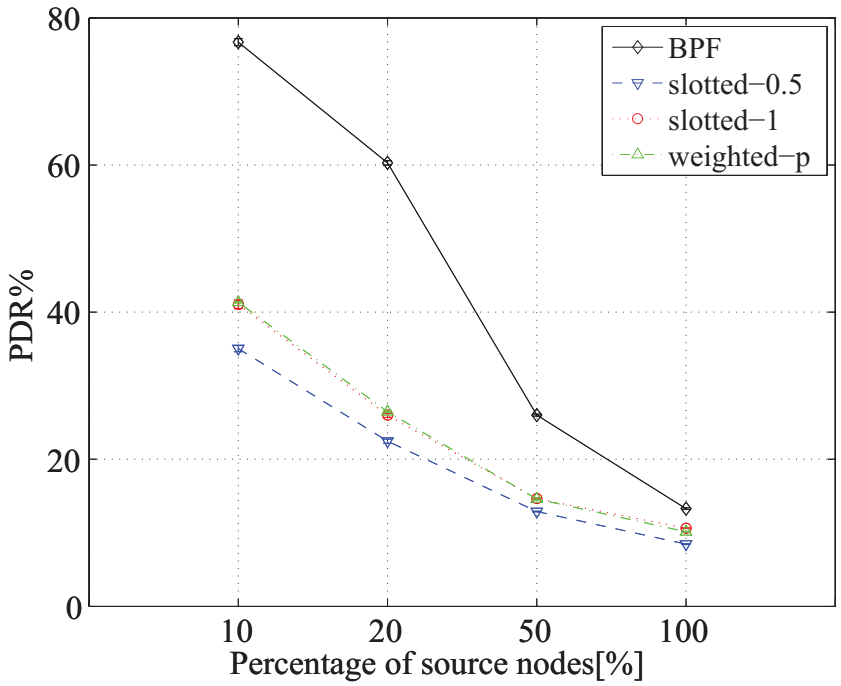


Figure 3.8: PDR% for different number of source with 9.6 nodes/km

3.5 Summary

We envision the usage of a VANET as an infra-structure for an urban cyber-physical system that makes available up-to-date data about various parameters of an urban area to services outside of the network. The main consumers of such data are applications like as traffic management or navigation using real-time traffic information for regular and for emergency vehicles, or personal mobility and environmental monitoring. The data gathering system is modeled as a many-to-one communication over VANET, a scenario that has not been previously addressed. In this scenario, broadcast storms are more likely to happen than in regular scenarios, since there are more source nodes regularly sending packets, so the amount of congestion in the network should be carefully monitored.

In this chapter, we propose the Back off-based Per-hop Forwarding (BPF), a broadcast- and receiver-based per-hop forwarding protocol that selects the forwarding order among the nodes receiving the packet by mapping it into back-off time, so that nodes likely to be nearer to the final destination have shorter back-off times. BFP has the following properties: 1) it does not require nodes to exchange periodic messages with their neighbors communicating their locations for keeping low the management message overhead; 2) it uses geographic information about the current sender and the final destination node in the header of each data packet to route it in a hop-by-hop basis; 3) it takes advantage of redundant forwarding to increase packet delivery to a destination.

We evaluated the proposed protocol and compared its performance to broadcast storm mitigation techniques for safety message dissemination using ns-3. The results show that the proposed protocol achieves higher packet delivery rates and uses on average the same number of hops and causes less redundant packets at the data sink. When subject to increasing load due to increasing number of nodes generating data, all studied protocols significantly reduce the PDR%, although BFP maintains a higher delivery efficacy.

However, the results also indicate that there is still room for improving the performance in higher load scenarios, which will be the focus of the next steps. Another matter of interest in this context is analyzing the effect of the length of the path between sources and destinations and limiting the amount of network congestion observed near the sink, a typical problem from sensor networks.

Chapter 4

Supporting Sensing Application in Vehicular Networks

4.1 Introduction

Vehicular ad-hoc networks (VANET) have moved into the spotlight driven mainly by the benefits expected from safety, traffic management, and infotainment applications Gerla and Kleinrock (2010). In this chapter, we propose a different utilization, namely to use a VANET as the infrastructure for an urban monitoring system, an approach that has not been explored so far. Vehicles equipped with a wide range of sensing devices and the ability to communicate with each other offer a unique opportunity for gathering real-time data about a city, like traffic conditions Hao (2010), environmental parameters, video and audio for surveillance Gerla and Kleinrock (2011), or physical condition of the drivers Rodrigues et al. (2010b). Knowing the updated state of relevant variables for a city is not only critical for applications such as navigation using real-time traffic information both for regular and for emergency vehicles, but also for personal mobility support and environmental monitoring Rodrigues et al. (2011a).

The IEEE has recently released the 802.11p standard for VANET IEE (2010), which is derived from the 802.11a and 802.11e standards. The 802.11p PHY uses the 5.9 GHz frequency band with channels of 10 MHz. Additionally, 802.11p MAC does not require node association for communication, and no handshakes or acknowledgments are foreseen on the control channel.

The purpose of urban sensing is to provide information about an urban area to entities outside the VANET in near real-time. Each node collects information about its environment and sends it periodically to the gateway, configuring a many-to-one network topology, where all nodes are simultaneously both data generators and forwarders and the ultimate destination of the data lies outside the VANET.

Several protocols for data collection in wireless sensor networks (WSN) and mobile ad-hoc networks (MANETs) have been previously proposed, but VANET links are more volatile and multi-hop paths have very short durations Viriyasitavat et al. (2011), in addition to the lack of reliability at the link level due to the highly dispersive environment. In this scenario, the protocols

developed for WSN incur a very high path management overhead Li et al. (2009) and are not adequate. On the other hand, existing VANET protocols focus on unicast or data dissemination, and the scenario described has not been previously addressed.

This chapter proposes and evaluates a data collection protocol over an urban VANET. Our main contributions are 1) a new data collection protocol for data gathering in an urban area; 2) the evaluation of different suppression levels to limit the number of forwarders at each hop; 3) the benchmarking of broadcast-based protocols using a metric appropriate to measure sensing performance.

Our results show that significant gains can be obtained from probabilistic forwarding to further reduce the replication of packets at each hop while there is a significant amount of packets from each road segment received at the gateway for a wide range of vehicle densities.

4.2 Data Collection Challenges

The envisioned scenario is a city environment with moving cars which are collecting sensor data about the city, e.g. noise or air pollution, and the VANET is used as sensing infrastructure for gathering and sending the data to a back-office outside the VANET for monitoring purposes. This configures a many-to-one communication pattern, and it is assumed each node knows its own geographic position and that of the gateway.

In a city, cars move along the streets with unpredictable patterns and buildings block the communication between vehicles in non-aligned streets, creating a very volatile environment, where a single direct link lasts on average 20 s Viriyasitavat et al. (2011). Data collection protocols for wireless sensor networks and MANETs foresee some movement and provide mechanisms for path re-establishment, but they would need to be used too often in such volatile scenarios causing too high an overhead.

On the other hand, the number of vehicles in each street segment is highly variable in time and across different streets, and not feasible to be estimated in real-time by vehicles in other streets with low overhead. This makes it difficult to use source routing to take the packet towards the destination, because it is not possible to know at the packet source, or anywhere along the way, whether there are vehicles that can serve as forwarders in the street segments between any vehicle and the gateway, causing packet die out with high likelihood.

Furthermore, during traffic congestion, communication can be very difficult due to an overly loaded shared medium. In this situation, which is common in urban areas, requiring additional message exchange for coordination purposes increases further the medium load.

A protocol for data collection in an urban scenario using vehicle-to-vehicle communication must take all these limitations into account.

4.3 Protocol Design

The proposed protocol is Urban Data Collector (UDC) protocol, a data gathering protocol that has the following features: 1) it does not require nodes to exchange periodic messages with their neighbors; 2) it uses the 802.11p MAC layer; 3) it takes advantage of redundant forwarding to increase packet delivery to the gateway; 4) it limits the amount of redundancy using suppression techniques.

The UDC protocol is a network layer protocol that can be run on top of the recently standardized 802.11p MAC. It is a broadcast- and receiver-based protocol, so that it can take advantage of any node that receives a forwarded packet without requiring an exchange of information about node present in the neighborhood. This is especially relevant because the MAC layer does not provide acknowledgments, and it is not easy for a forwarder node to know whether its packet has been received by another node. Next, we exemplify why some well-known solutions are not adequate for data gathering in urban areas, and explain how our protocol addresses those issues.

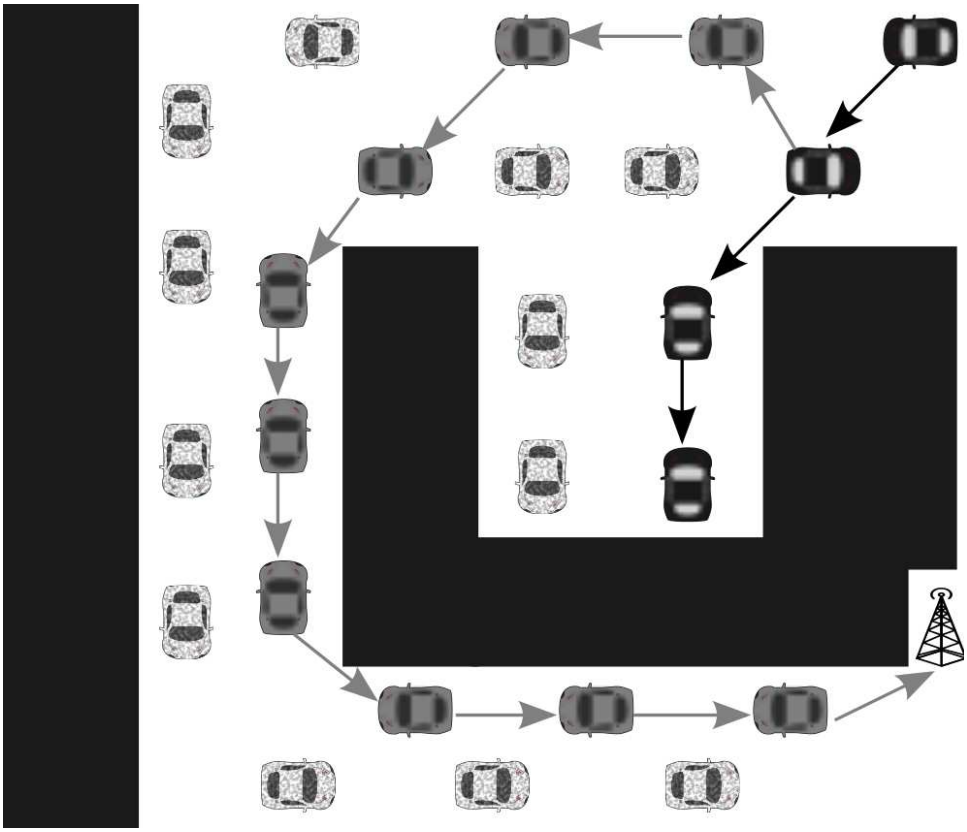


Figure 4.1: UDC help packets to get to the gateway when a path in the direction of gateway is blocked by a building.

Figure 4.1 illustrates a case where a greedy protocol cannot deliver a packet to the gateway (shown as an antenna on the right bottom) because a chosen forwarder does not have any neighbor in the direction of the gateway. Protocols like GPSR Karp and Kung (2000a) use perimeter routing technique to recover from the blocked path and find another path to the gateway, but these

introduce a high delay and cause a lot of additional transmissions. Because the proposed protocol follows two paths simultaneously (black and gray paths in Figure 4.1), although the black path is blocked, the gray path provides a redundant path which avoids packet die out.

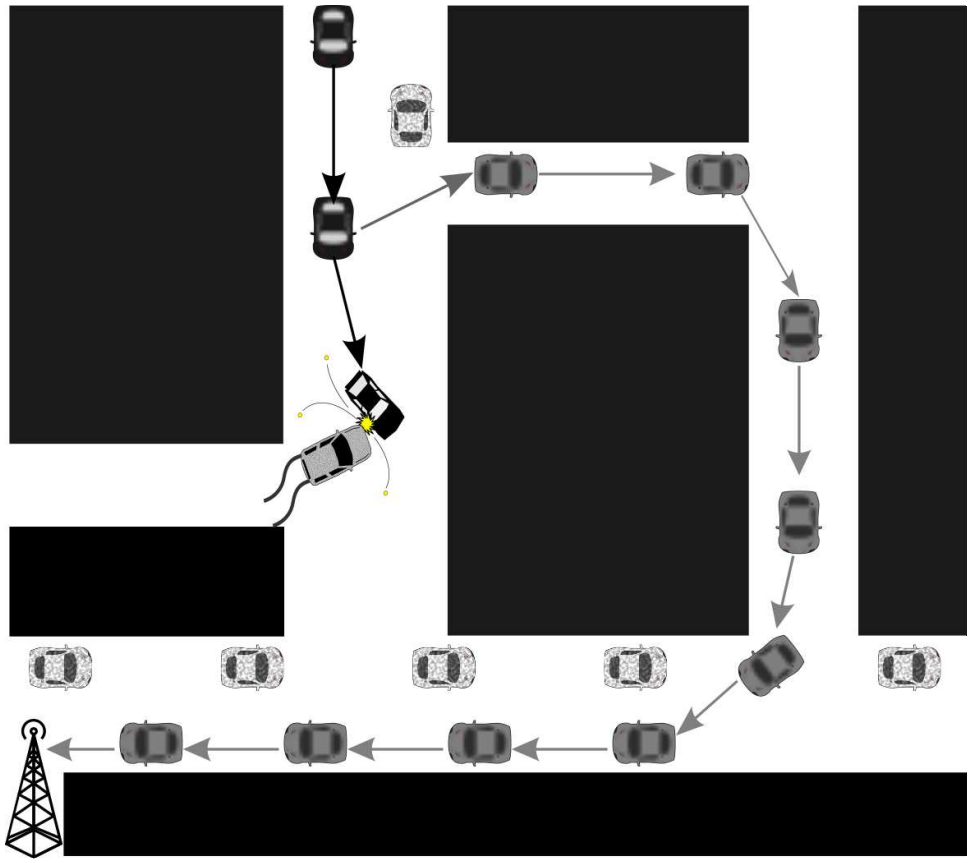


Figure 4.2: UDC avoid packet die out when an accident happens on the direction of the gateway.

In Figure 4.2, the main path (black path) to the gateway is blocked because the packet reaches a node that has no other nodes in range. This can happen because of the high volatility of the network, where the node that sent the packet is no longer in range after the routing dead-end has been detected. In this case, the perimeter technique will not be able to recover from this situation because there are no other nodes in the communication range of the single vehicle that has the packet. Because our protocol uses both the main path (black path) and the redundant path (gray path) for forwarding the packet the likelihood of both dying out is reduced.

Existing protocols get stuck in a blocked road or a road without forwarders toward the gateway because most of them take a binary decision for selecting the next forwarder. UDC gives the forwarding opportunity to several neighbor nodes to create redundant paths to avert packet die out. Finally, a fully distributed suppression technique that requires no coordination among nodes has been used to limit the number of redundant paths so as to reduce the overhead. This is accomplished by taking advantage of two different types of forwarding: directional forwarding and probabilistic forwarding. Directional forwarding means that each node, upon receiving a packet,

forwards it in the direction of the gateway after the expiration of a timer, which is used for giving forwarding priority to nodes that represent more progress towards the gateway. Probabilistic forwarding means that each node forwards the received packet with a probability which depends on difference distances between itself and the previous node to the gateway. The next sections describe in detail how these two techniques are combined to provide the desired functionality.

4.3.1 Directional Forwarding

UDC uses a network layer timer calculated as a function of the difference between the distance of the previous and potential forwarder to the gateway to prioritize forwarders nearer to the gateway. The timer is calculated at each node in a distributed manner based only on the information contained in the header of the received packet and of the node itself.

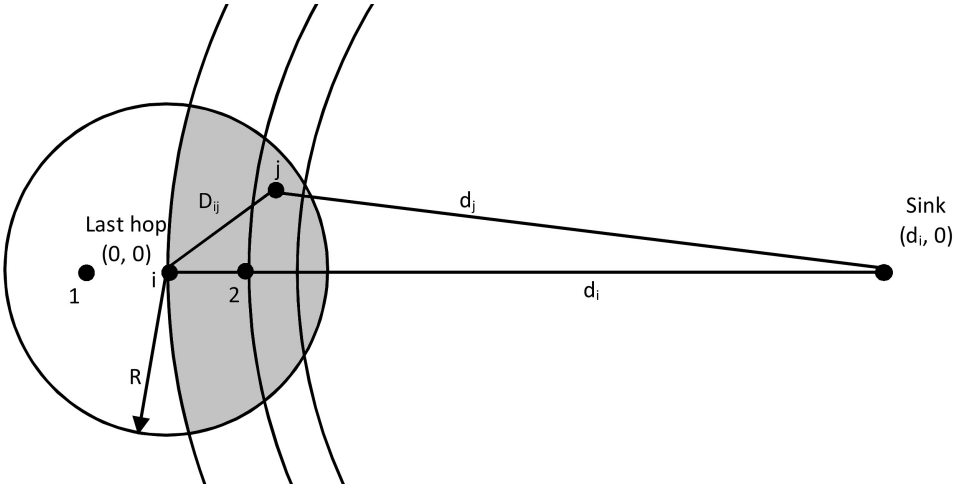


Figure 4.3: One-hop forwarding entities

Figure 4.3 illustrates the scenario used to explain the calculation of the timer: node i is the sender of the packet and nodes 1, 2 and j are receivers and potential forwarders. The communication range is divided into two different areas, marked white and gray in Figure 4.3. Nodes in the gray area are nearer to the gateway than the previous hop, whereas nodes in the white area are farther away. Nodes that represent the most progress towards the destination should be preferred forwarders. These are the nodes located closer to the end of the transmission range in the direction of the gateway, hence, they shall have shorter timers and forward earlier than nodes farther away from the gateway.

The time coefficient is calculated according to Equation 4.1. This timer coefficient lies in range [0, 1] and is then multiplied by the maximum timer T_{max} to deterministically calculate the actual timer value at all potential forwarders (Figure 4.3).

$$C = (1.0 + \frac{d_j - d_i - R}{2R}), \tag{4.1}$$

where d_j is the distance of node j to the gateway, d_i the distance between the previous hop and the gateway, and R is the communication range.

4.3.2 Suppression Techniques

Suppression techniques are mechanisms that effect the reduction of the number of packets re-forwarded at each hop. The simplest suppression mode is that any node that hears a packet being forwarded while waiting for timer expiration stops the timer and discards the packet. In this way, the node that correctly receives a packet and is closest to the gateway will be the first to forward, and any node that hears that forwarding will not double forward. But nodes on different sides of the previous hop may not hear each other, i.e., nodes in the white area do not hear nodes in the gray area in Figure 4.3 and cause duplicated forwarding. This problem could be solved by limiting the forwarder nodes to nodes that are nearer to the gateway than the previous hop. But this is likely to raise a problem for low node density, causing packet die out in the more likely event of lack of a forwarder nearer to the gateway.

Four different levels of suppression have been studied to overcome these problems: basic, weak, moderate and strong suppression. In the basic method, nodes discard a packet if they receive the same packet during the timer or wait time (see Section 4.3.3), meaning that it has been forwarded by a node which is closer to the destination.

In weak suppression, UDC reduces the number of forwarders by using probabilistic forwarding for the nodes with higher distance to the gateway (white area in Figure 4.3), while nodes closer to the destination (gray area in Figure 4.3) forward the packet with probability one. Nodes in the white area forward a packet with a probability that decreases with increasing distance to the gateway, calculated according to Eq. 4.2.

$$p_{fwd} = (1.0 - \frac{d_j - d_i}{R}) \tag{4.2}$$

In strong suppression, nodes in the gray area, which are ranked by a timer, forward the packet with a different probability, calculated according to Eq. 4.3 and nodes in the white area do not forward the packet at all.

$$p_{fwd} = (\frac{d_i - d_j}{R}) \tag{4.3}$$

An intermediate level, moderate suppression, is added for which the forwarding probability is increased linearly with decreasing distance of potential forwarders to the gateway, calculated using Eq. 4.4.

$$p_{fwd} = (\frac{d_i - d_j + R}{2R}) \tag{4.4}$$

Figure 4.4 shows the forwarding probability of different suppression levels where the x-axis shows the difference distance between the last hop and potential forwarders to the gateway ($d_i - d_j$) relative to the communication range (R).

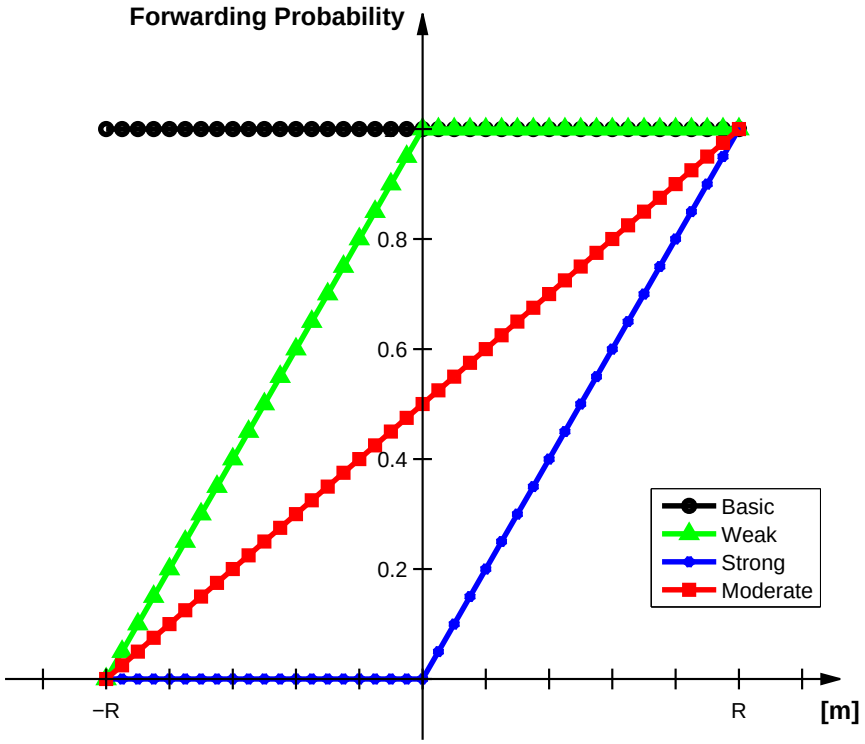


Figure 4.4: Forwarding Probability calculated for different suppression techniques

The complete network layer forwarding protocol algorithm is described in Figure 4.5. Upon receiving a packet, node checks if it is not a gateway and consequently if it did not receive the packet before, it calculates the timer (section 4.3.1) and in the end of the timer according to the probability (section 4.3.2) forwards the packet.

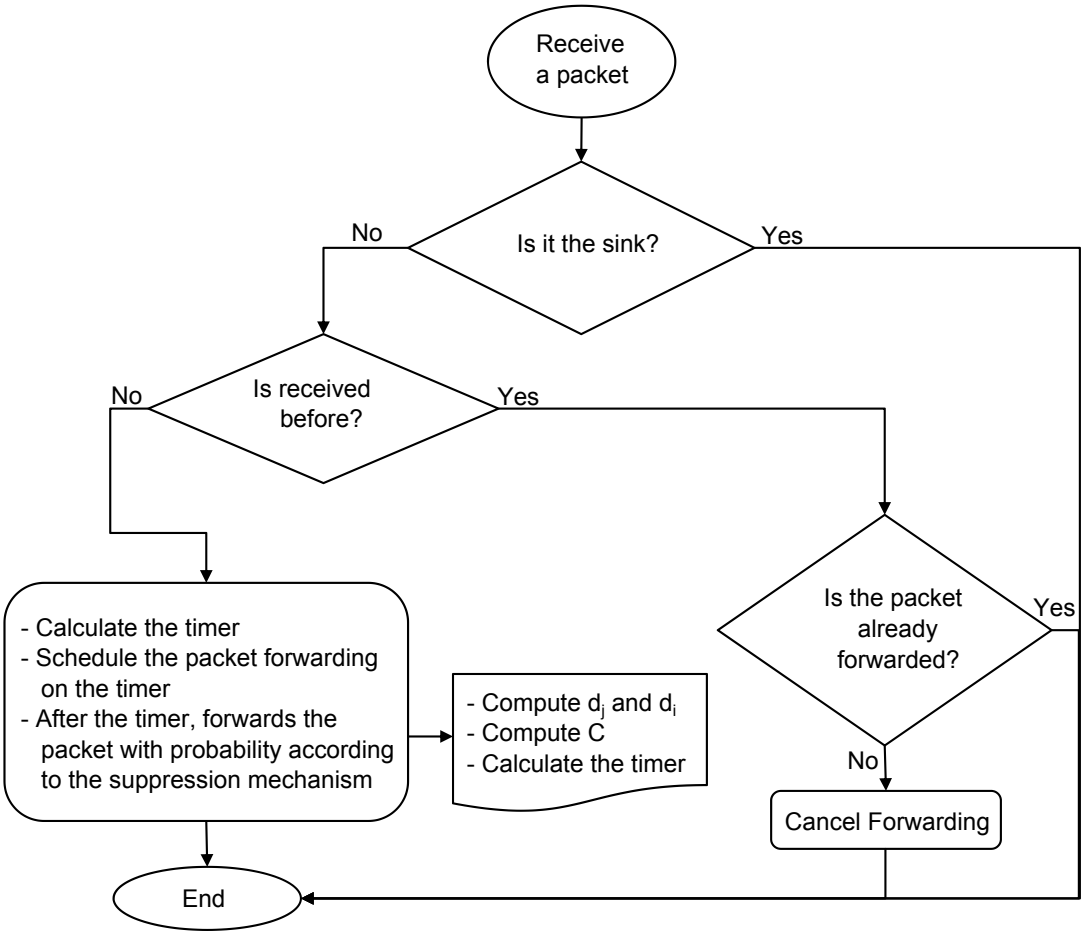


Figure 4.5: Protocol algorithm

4.3.3 Channel Access Time

The average CSMA/CA channel access time obtained from simulation of UDC protocol with different suppression techniques is illustrated in Figure 4.6. The average channel access time for all node densities for 4 different suppression techniques was 5 ms. This time is embedded in the protocol by letting each node wait for this duration before starting the timer at the network layer, to allow the suppression mechanisms to hear packets forwarded by other nodes.

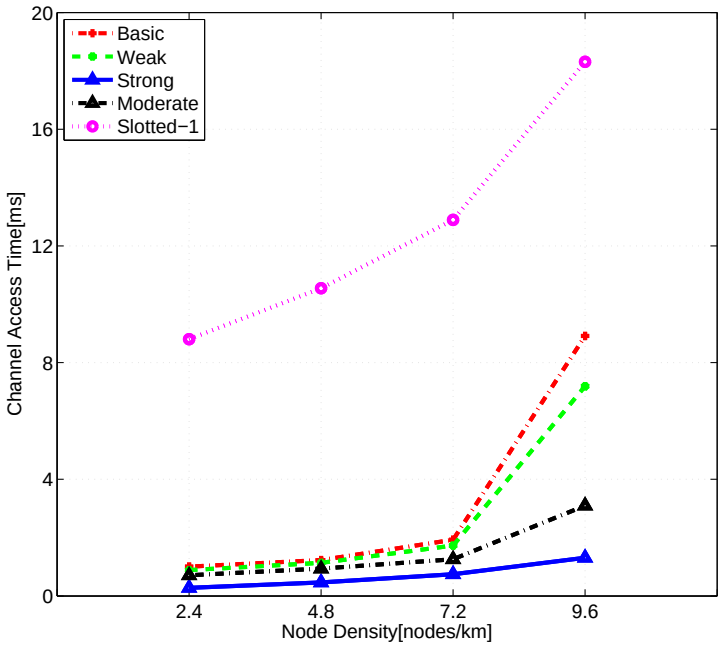


Figure 4.6: Average channel access time for 4 suppression techniques.The slotted 1-persistence protocol used for comparison is introduced in Chapter 2.

4.4 Simulation

The Network Simulator 3 (NS-3) ns3 version 3.9 is used to simulate a 2.5×2.5 km Manhattan Grid topology with 5×5 roads, adding up to a total of $30km$ road length. There is one gateway located in the center of the topology, at (1250 m, 1250 m), and all other nodes are data sources. The communication range is set to 500 m using a path-loss exponent of 2.5, and each node has on average at least four nodes and at most 84 nodes (crossroads with 42 nodes/km) within its communication range. The data rate is set to 1 kbps with100 Bytes packets, which is sufficient for envisioned sensing applications Rodrigues et al. (2011a). The other simulation parameters can be found in Table 4.1.

Table 4.1: Simulation Set up

Parameter Name	**Value**
Node Density	4, 8, 16 and 42 nodes/km
Average Speed	14 m/s
Minimum Speed	3 m/s
MAC protocol	802.11p
Channel Data rate	6 Mbps
Channel Bandwidth	10 MHz
Communication Range	500 m
Propagation Model	Nakagami-m (m=1.56)

Two seconds warm-up time is considered at the start of the simulation before reaching steady-state. The simulated time is set to guarantee that each node generates at least 20 packets. UDC with four different suppression levels (Section 4.3.2) and the maximum timer equal to $T_{max} = 5ms$ is used for simulation. The slotted-1 persistence protocol Wisitpongphan et al. (2007a) is employed for comparison because it showed the best performance in a benchmark of existing protocols for data gathering using VANET in Nozari Zarmehri and Aguiar (2011) (see Chapter 3).

4.5 Performance Evaluation

The performance of UDC is evaluated using the following metrics:

- **Sensing accuracy**: In the city environment for having an explicit view about the city, the urban sensing application needs having information about each road segment. Then the application can extract useful information about each road segment and expand it from road segment to entire city. So the sensing accuracy is defined to show how obtained information is accurate and can be applicable for the urban sensing. Therefore, it has been defined as a number of received packets from each road segment in a second from all nodes within that section. To calculate it, after analyzing the collected data at the gateway, the received packets are separated depending on the source‘s location.

- **Network efficiency**: The question of how well the protocol acts in the broadcast fashion is defined as network efficiency. This metric shows how many of the packet forwardings are used for the packets which get to the sink. The definition is: the number of hops for each packet receives at the sink, including both unique packets and their replicas, divided by the number of all forwarded packets in all nodes.

Additionally, extensive simulations were run to determine the different reasons for the packet discards at each node, and the evaluation of these results is shown in this section.

4.5.1 Sensing accuracy

Figure 4.7 shows the sensing accuracy for the UDC with different suppression levels. It plots the average number of received packets from each road segment at the gateway in one-second windows in the simulated road topology against varying node densities. It shows that the sensing accuracy is most dependent on the node density when using the strong suppression level. For the lowest node density, it discards more than 87% of the received packets at each hop and is not able to produce enough redundancy paths, causing a high probability of packet die out. As the node density increases, the strong suppression level represses about 97% of received packets at each hop and thus reduces the number of collisions, achieving the highest sensing accuracy for the highest node density.

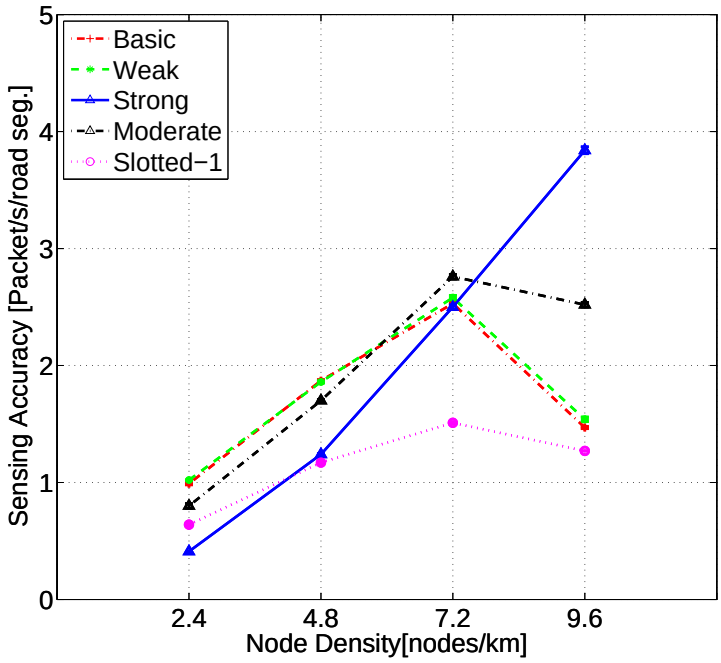


Figure 4.7: Sensing accuracy for different data collection mechanism in urban area

On the other hand, the weak and basic suppression level achieve similar sensing accuracy with less dependency to the node density. Since the sensing application needs to obtain information from each road segment independently of the node density, a protocol with fewer variations in the sensing accuracy for different node densities is a better choice for a sensing application. This fact and the fact that both provide the highest accuracy for the lowest node density, makes them a better choice for the envisioned application. The moderate suppression level has a sensing accuracy that lies midway between the performance of weak and strong, showing better performance than strong suppression at low node density and better performance than weak suppression at high node density.

4.5.2 Network Efficiency

Figure 4.8 shows that the UDC with strong suppression level uses broadcast forwarding in the most efficient way, since a larger percentage of the packets forwarded in the network end up reaching the sink. More interestingly, the weak suppression level shows a higher network efficiency than the basic level while achieving similar sensing accuracy (see the previous section) since the lower amount of forwarded packets causes a lower number of collisions. The moderate suppression level again lies midway between strong and weak suppression levels, as intended at its construction. These results also show that the efficiency of the protocol drops with increasing node density due to the high amount of collision caused by a higher medium load, as expected for a broadcast protocol in a shared medium.

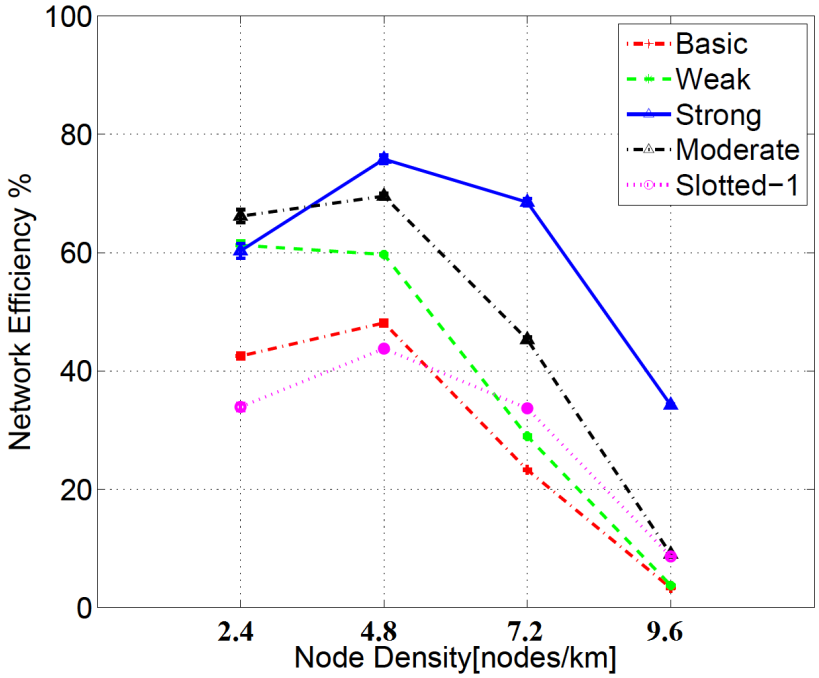


Figure 4.8: Efficiency for using different suppression techniques

4.5.3 Packet Drop Analysis

An extensive simulation is carried out to analyze the reasons of performance drop for increasing node density, with the aim of gaining insight into the possible ways to improve the performance of the protocol. Figure 4.9 identifies the different possible reasons for packet discards at the different communication layers.

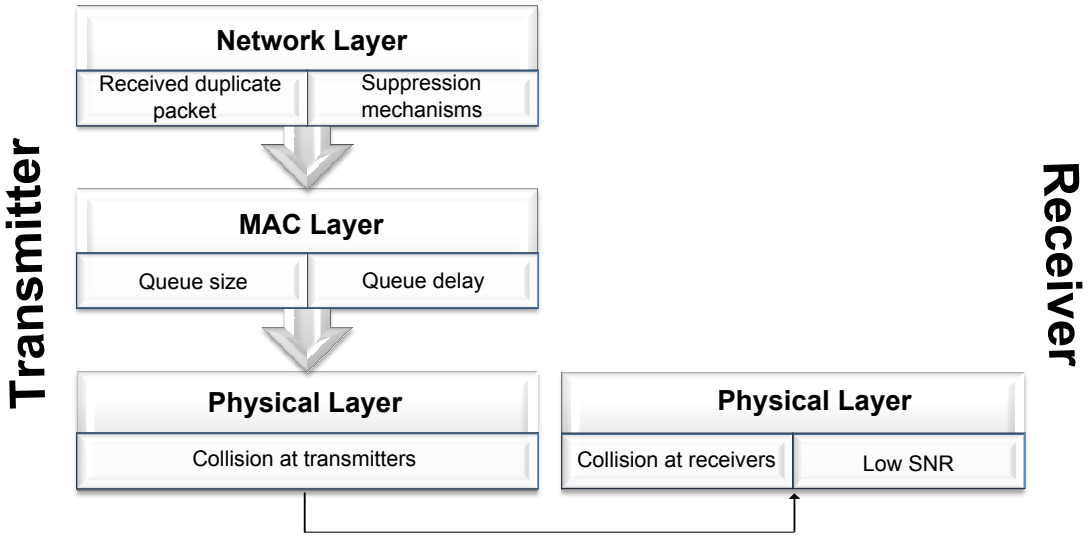


Figure 4.9: Different reasons for drop packets depend on different communication layers

In this analysis, only the packets that do not get to the sink is considered and the drop reasons for these packets is counted. Table 4.2 shows the reason why the last replication of a packet still alive in the network dies out before reaching the gateway. This table also shows that the main reason for dropping the packets is excessive suppression, i.e. suppression of packets that would have been useful for reaching a higher sensing accuracy.

Also, even for low node densities, suppression and collision are the main causes of packet die out, and not low SNR, i. e. packets do not die out because of lack of forwarders, but due to the broadcast nature of the protocol, which causes collisions and requires suppression. On the other hand, the single main cause for packet die out for the strong method is suppression, for any node density.

Moreover, we conclude that an intermediate solution between the weak and strong suppression techniques would achieve a better balance between the collision and suppression trade-offs. So UDC with moderate suppression level is proposed as the best compromise, in accordance with the results of the performance evaluation in Section 4.5.

Table 4.2: Percentage of each reasons in the protocol performance for different node densities

Density nod./km	**Sup. Method**	**Sup. %**	**Col. %**	**Low SNR%**	**PDR %**
4	Strong	81.06	0	0.46	18.48
	Weak	19.23	6.15	1.4	73.22
8	Strong	48.01	0.02	0.04	51.93
	Weak	15.73	6.92	0.21	77.14
16	Strong	39.11	0.22	0.02	60.65
	Weak	28.64	11.93	0	59.43
42	Strong	55.27	0.79	0	43.94
	Weak	53.79	33.59	0	12.62

4.6 Summary

We envision the usage of a VANET as infrastructure for an urban cyber-physical system that makes up-to-date data about various parameters of an urban area available to services outside of the network. We propose and evaluate the use of UDC protocol, a broadcast- and receiver-based forwarding protocol.

The effect of four different suppression techniques on the sensing accuracy and network overhead is evaluated using NS-3 large scale simulation. The results reveal that for supporting the sensing applications in the urban area, the weak suppression increases the sensing accuracy with lower dependency to the node density. Also, it shows that the weak method has less excessive suppression than other methods and is the best solution of those evaluated for urban sensing.

As part of ongoing efforts, the study of network limitation and its effect on the sensing accuracy will be investigated (see Chapter 5).

Chapter 5

Numerical Limits for Data Gathering in Wireless Networks

5.1 Introduction

Data collection is a major application of wireless sensor networks. Typically, each node in the network captures information about its environment through sensors and sends it towards the sink node, which has all resources and functionalities to store and/or process the data. All nodes are also relays, helping data from other nodes get to the sink. In this scenario, it is crucial to know the maximum amount of data that each node can produce without causing the network to overload.

In this chapter, we provide initial numeric results on the limit of wireless data gathering along a chain with respect to the probability of collisions due to hidden nodes. The scenario for this calculation is many-to-one communication from all nodes in the chain to the sink. To the best of our knowledge, there are no previous results that enable the calculation of a numeric limit for the data generated by each node as a function of the network parameters. The lack of such a calculation motivates us to show the limitation of the network in a chain of nodes with some numerical results. In this study, we focus on the maximum service rate at each node for achieving a certain packet delivery rate at the sink. For example for a chain of 15 nodes, for having guaranteed 90% packet delivery for each node at the sink, the service rate must be less than 25%.

The rest of the chapter is organized as follows. The network model which has been used for our calculation is illustrated in Section 5.2. The main contribution for limitation of the data gathering is shown in sections 5.3 and 5.4. Section 5.5 shows the results for two well-known performance metrics: packet delivery rate (PDR) and service rate. Results are discussed in section 5.6. In Section 5.7, we verify our calculation by simulating the same scenario. Finally, we conclude the chapter in Section 5.8.

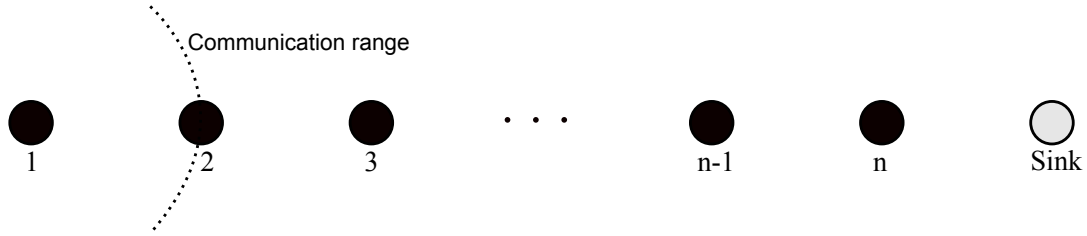


Figure 5.1: Network model

5.2 Network Model

According to our previous study for an urban scenario Nozari Zarmehri and Aguiar (2011), each source passes its packets through a chain of nodes to the sink. So we consider a network model deployed as a flat chain of n nodes, which are all data sources. Also, there is one sink node, located at the end of the chain, which we will consider to be at the right side without loss of generality. Each node is not only a source of data packets but also a relay for the data coming from all its left side neighbors on its way towards the sink. The network model used for the calculations is illustrated in Figure 5.1.

Each node has at most two neighbors located at the end of its transmission range. Moreover, nodes forward all packets received from the left side and ignore all packets received from the right side (implied ranking). The channel transmission rate is W [bps] and the packet transmission duration is τ, and can be calculated as $\tau = L/W$, where L is the packet size.

5.3 Maximum Service Rate without Collisions

In this section, we calculate the maximum service rate as the maximum amount of data that can be generated at each node. The actual generation rate should be less than the maximum achievable service rate to avoid overload. We make the simplifying assumption that nodes send their packets in a coordinated way from left to right, starting at the farthest node from the sink, and each node sends one packet of its own and forwards one packet from each node in its left, adding up to i packets at node i. All nodes use the same amount of resources, time, for sending each data packet, τ, so node i needs $i \cdot \tau$ to transmit its packets. We also assume that each node requires a certain time to access the channel for each transmission, the channel access time T_{CA}, which is on average equal for all nodes.

Assuming a total available time of T, the first node on the left has $T - T_{CA}$ time for its transmission. But the second node cannot send during the transmission of the first and third nodes to avoid collisions, where the second node has $T - T_{CA} - 2\tau$ time to send its data packets, which include his own data packet and the data packet forwarded from node one. This calculation can be continued to the end of the chain. Although node n is the last node in the chain, it is not the bottleneck because it must not wait for any transmission on its right side, since there is the sink.

Hence, the bottleneck of this scenario is node $n-1$ because it is the last node that must avoid collision with neighbors on both sides.

The total waiting time for the bottleneck in this scenario is given by the sum of the transmission times of the neighbor nodes, $n-2$ and n, with the average waiting time: $T_{CA}+(n-2)\tau+n\tau$. To accommodate all transmissions from all nodes without collisions, the total available time T should be at least equal to the waiting time plus the time needed for its own transmissions at the bottleneck:

$$T \geq T_{CA}+(n-2)\tau+n\tau+(n-1)\tau \tag{5.1}$$

From here, we can calculate the maximum duration of a single packet, τ, depending on the other network parameters as follows:

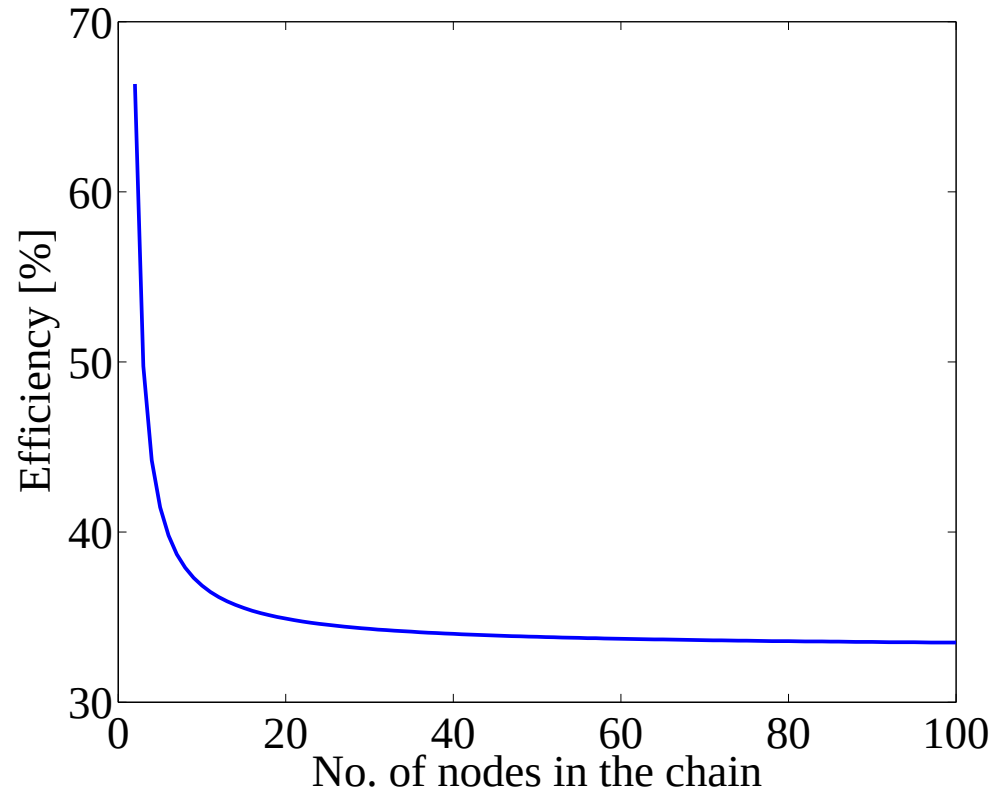
$$\tau \leq \frac{T-T_{CA}}{(3n-3)} \ [seconds], \tag{5.2}$$

And the maximum amount of data that can be generated by each node within each period T without causing congestion is:

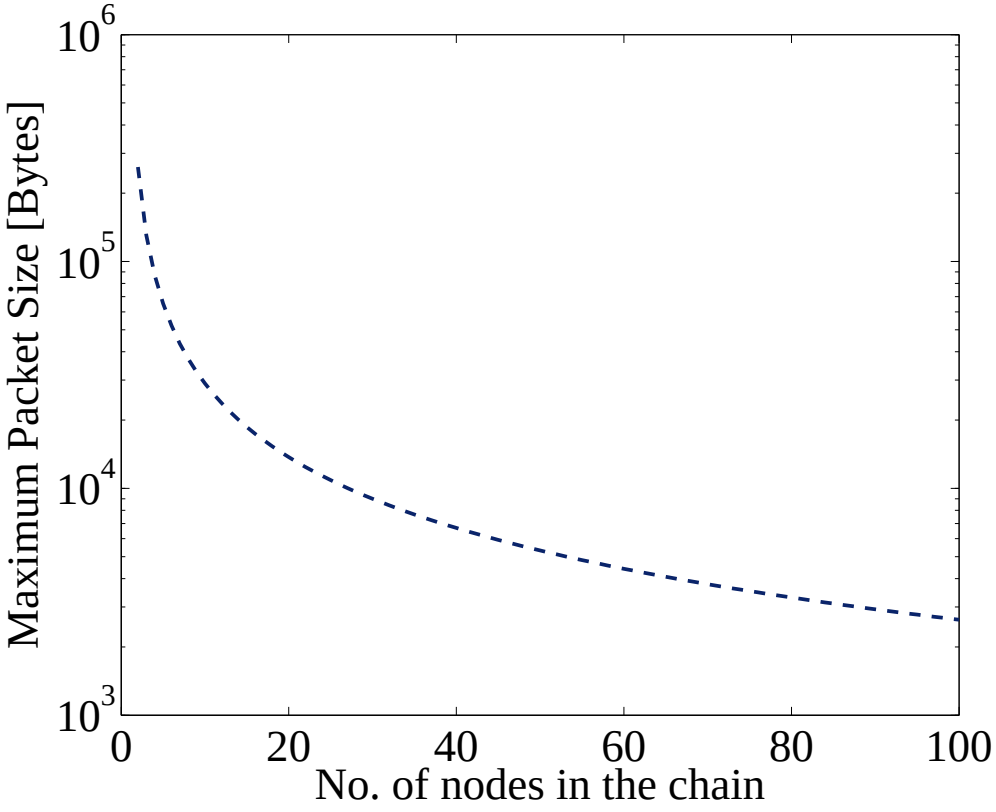
$$L \leq \frac{(T-T_{CA})W}{(3n-3)} \ [bits] \tag{5.3}$$

We also calculate the best-case resource usage efficiency, ρ, which can only be obtained through perfect scheduling, by dividing the maximum amount of data that can be transmitted by all nodes per unit time by the available channel bit rate (W):

$$\rho = \frac{L \cdot n}{T \cdot W} \tag{5.4}$$



(a) Efficiency



(b) Maximum packet size

Figure 5.2: Performance of a simple Chain

Figure 5.2a shows numeric values for the resource efficiency and Figure 5.2b the maximum amount of data per node for T=1 s, T_{CA}=5 ms, and available bandwidth W=6 Mbps. Both values decrease with increasing number of nodes in the chain, as expected. The bandwidth efficiency is below 70% for very short chains and stabilizes at around 35%, and we have not yet considered hidden node collisions.

5.4 Interference Model

In the previous section, we assumed that the transmission starts were fully coordinated and node i would only start to transmit after it received all packets from node $i-1$, in which case no collisions could occur. In this section, we extend the results to a more realistic scenario, in which transmission starts are not coordinated and packet receptions at any node can be impaired by collisions caused by simultaneous transmissions from both neighbor nodes, which cannot hear each other (hidden nodes). This only impairs the packet delivery rate from the left side neighbor, since the data flows from left to right. To estimate the number of collisions and its influence on the system performance we assume a time slotted system, in which the total time T is divided into N equal time slots.

The minimum number of time slots can be calculated by considering the bottleneck in the network according to Eq 5.5. Node $n-1$ should wait during the transmissions of nodes $n-2$ and n to avoid collision, and needs $n-1$ time slots for own transmission.

$$N = (n-2) + (n-1) + n = 3n-3 \tag{5.5}$$

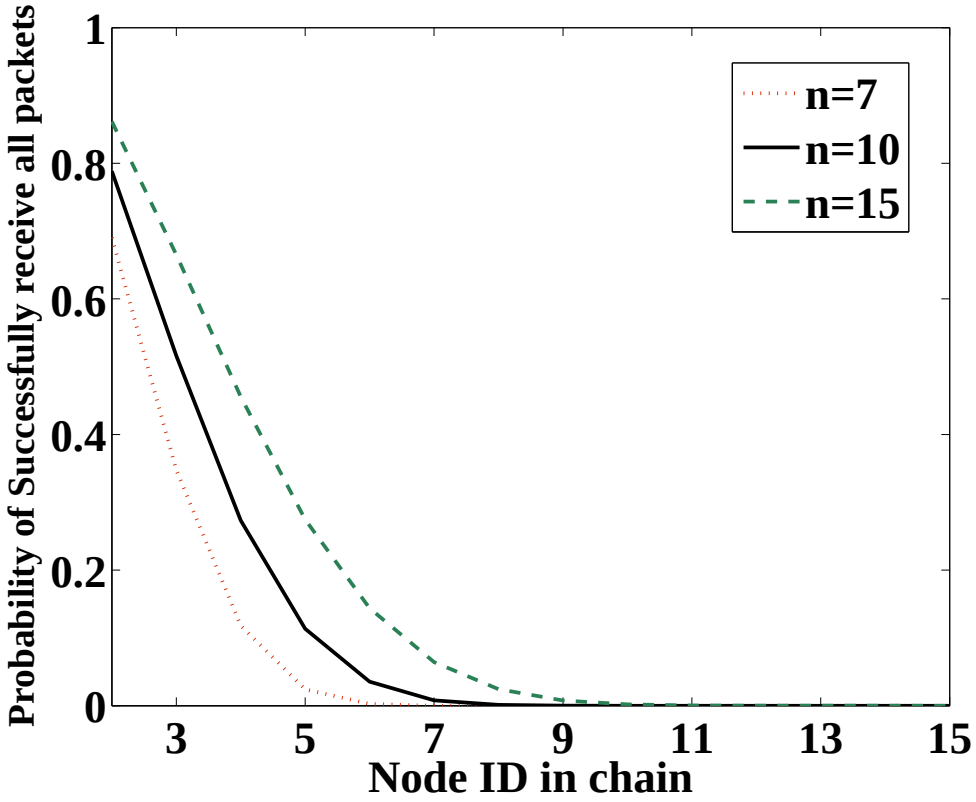
Even with this number of time slots, a collision can still happen when two hidden nodes select the same time slot for a transmission. For example, nodes 2 waits for node 1 and 3 but cannot hear the transmission of node 4 and so it is possible that nodes 2 and 4 select the same time slots for their transmission, causing a collision that impairs the reception of the packet from node 2 at node 3. So, nodes in the chain do not receive all packets generated to their left side, as part of those collide as a consequence of the hidden node problem. Equation 5.6 expresses the total received packets at node $n-1$, R_{n-1}, obtained by subtracting the total number of collisions ($E[Col]$) from node 2 to node $n-1$ from the total packets generated to the left of the node.

$$R_{n-1} = \sum_{i=1}^{n-2} k_i - \sum_{j=2}^{n-1} E_j[Col] \tag{5.6}$$

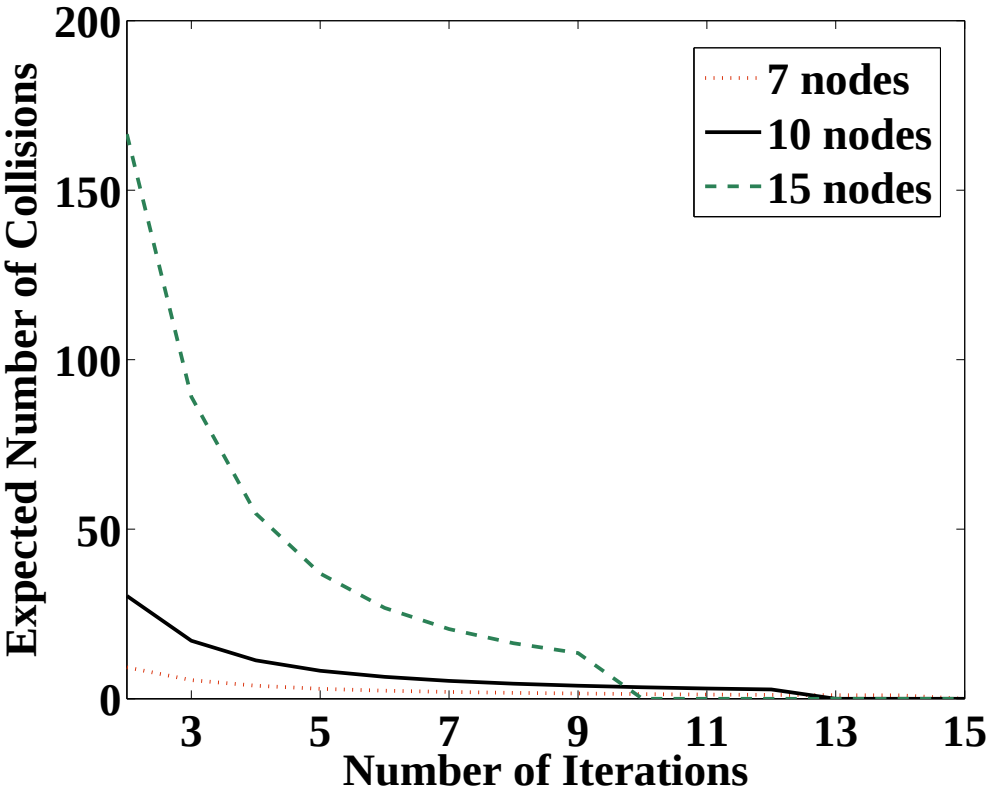
The best case scenario is when time slot selection is coordinated and no collisions occur, meaning that nodes receive correctly all packets generated at their left side. Therefore, the probability of receiving all packet successfully in node $n-1$ can be calculated by Equation 5.7. Node $n-1$ receives successfully all the packets only if it selects its time slots for transmission from all time slots excluding the ones chosen by nodes $n-2$ and n.

$$P_{n-1}[Successful] = \frac{\binom{N-k_{n-2}-k_n}{k_{n-1}}}{\binom{N}{k_{n-1}}} \tag{5.7}$$

This calculation is plotted in Figure 5.3a for each node in the chain and for three different chain lengths. The probability of successfully receiving all packets from all nodes decreases as the number of source nodes increases and goes to zero before reaching to the sink for all chain lengths, meaning that it is not possible to receive all packets at the sink without collision.



(a) Probability of receiving packets successfully at each node in a chain



(b) Effect of increasing N on the expected number of collision

Figure 5.3: Chain Performance

For considering possible collisions in Equation 5.7, which is more realistic, this equation has been extended. We use Y_i as a random variable that indicates the number of collisions at node i. For example $F_{Y_2}(0) = P(Y_2 = 0) = P_2[Successful]$ or $F_{Y_2}(1) = P(Y_2 = 1)$ is the probability of having one collision at node two. The probability of having y collision at node $(n-1)$ is:

$$F_{Y_{n-1}}(y) = P(Y_{n-1} = y) = \frac{\binom{N-k_{n-2}-k_n+y}{k_{n-1}-y} \times \binom{k_{n-1}}{y}}{\binom{N}{k_{n-1}}} \tag{5.8}$$

In Equation 5.8, the numerator has two factors. The first factor shows that node $n-1$ as a receiver has y collisions. So this node has y common time slots with its neighbors. The second part just shows the selection of these collisions from all received packets.

So, the expected number of collision can be calculated for node $(n-1)$ from Equation 5.9:

$$E_{n-1}[Col] = \sum_{i=1}^{kn-2} i \times F_{Y_{n-1}}(i) \tag{5.9}$$

In the presence of collisions, some of the time slots are used for unsuccessful transmissions. So, to evaluate the effect of increasing the number of time slots on system performance, we iteratively add the total number of expected collisions over all nodes (Equation 5.10) to the number of time slots (N). Thus, we do an iterative search for the number of total time slots required to transmit one packet from each node with high packet delivery rate at the sink.

$$E_{Total}[Col] = \sum_{i=2}^{n} E_i[Col] \tag{5.10}$$

$$N_{Updated} = 3n - 3 + E_{Total}[Col] \tag{5.11}$$

Figure 5.3-b shows how the expected number of collision is improved by increasing the number of time slots (N). In this figure, each node generates one packet. By increasing N, the expected number of collision decreases (Figure 5.3-b). However, it causes also a decrease in the service rate.

5.5 Results

5.5.1 End-2-End PDR%

Following our previous work Nozari Zarmehri and Aguiar (2012), the most interesting metric for an urban data collector (UDC) protocols is the number of packets received at the sink (PDR%). Moreover, this packet delivery ratio is considered between each source and the sink (End-to-End) because we assume that the sensing application needs information from each source to create a macro vision about the city. According to Equation 5.6, collisions are the main reason of PDR% reduction at the sink.

Figure 5.4 shows the PDR% at the sink for a chain with varying number of nodes. The average number of hops from each source node to the sink in our previous study Nozari Zarmehri and Aguiar (2012) was between 10 to 15 hops, so we focus on this chain length range. The plot shows the end-to-end PDR% for various chain lengths, calculated for the farthest node from the sink, as it is the worst case PDR.

Table 5.1 shows the number of time slots in each iteration for each configuration. For 10 nodes, after 12 iterations, there is no collision and the number of time slots remains the same. It also happens for 15 nodes after 10 iterations. Therefore, the curves for 10 and 15 nodes in figure 5.4 reach to the 100% and cross other curves.

Table 5.1: Number of time slots for each iteration

No. of Nodes	Iteration														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
7	18	40	49	54	58	61	63	65	67	69	70	71	72	73	74
10	27	97	127	144	155	163	169	174	179	183	186	189	189	189	189
15	42	422	588	677	731	767	794	814	830	843	843	843	843	843	843

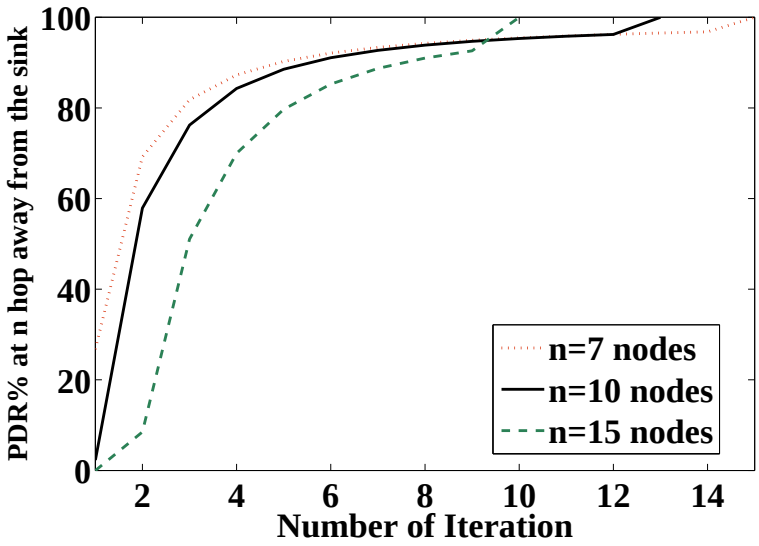


Figure 5.4: PDR% for different chain length

The PDR% with the smallest N is almost zero for the farthest node from the sink. By increasing N, giving more opportunities for transmission, the PDR% is improved to 100% after 12 and 10 iterations for 10 and 15 nodes, respectively.

5.5.2 Service Rate

To increase the PDR, we are increasing the amount of time slots necessary to transmit one packet from each node, hence decreasing the service rate and reducing the resource usage efficiency. So, now we evaluate the service rate for this scenario. The service rate is the amount of packets per unit time that each node can generate.

Figure 5.5 shows the service rate for different chain lengths in different iterations. Clearly, by increasing the number of nodes in the chain, the minimum total number of required time slots increases (see Eq. 5.5) and so the achievable service rate decreases. The difference in service rate between different chain lengths is due to the existence of a high number of collision in a longer chain. So, increasing the number of time slots improves the packet delivery rate at the cost of a sharp decrease in the service rate. This trade-off is discussed in the next section.

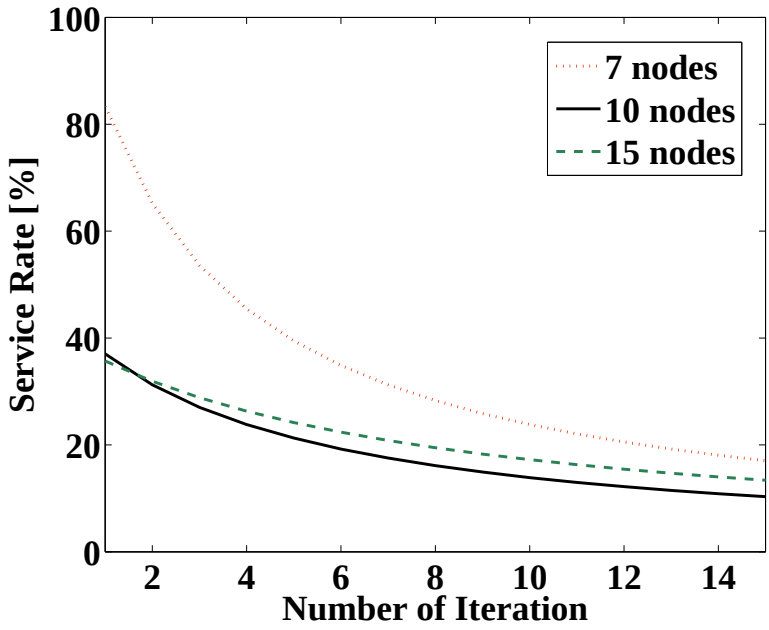


Figure 5.5: Service Rate for different chain length

5.6 Discussion

The previous results provide a quantification for the trade-off between the packet delivery rate and service rate. By increasing the number of time slots, the packet delivery rate increases due to the reduced number of collisions and the service rate falls because the total amount of generated packets is fixed. Figure 5.6 shows this trade-off for better illustration. For the shortest chain, the service rate falls very sharply by increasing the number of time slots because there are fewer time slots in this scenario.

As an example of our goal which is system design, if we have 10 nodes that need to generate three packets per unit time each and we want to guarantee a PDR of 85% at the sink, then we should design the system to have at least 432 time slots (fourth iteration: $3 * 144 = 432$), achieving a resource utilization of 7% at most.

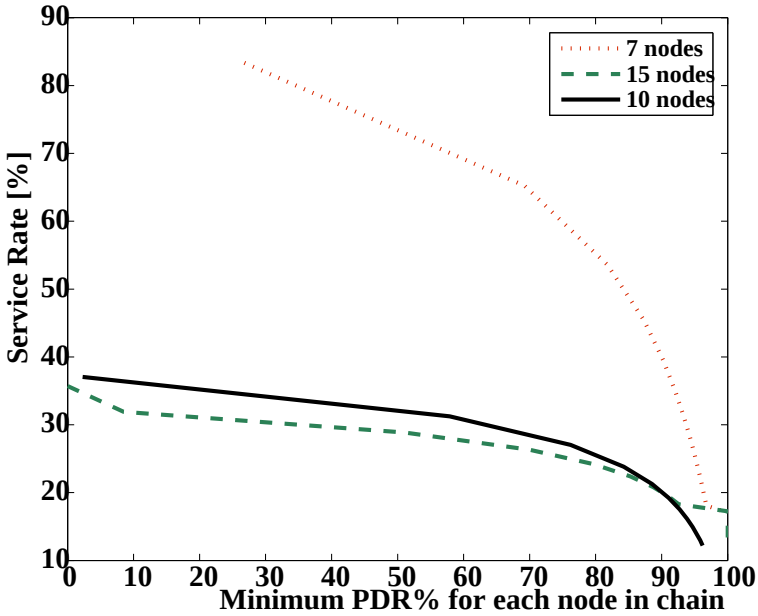


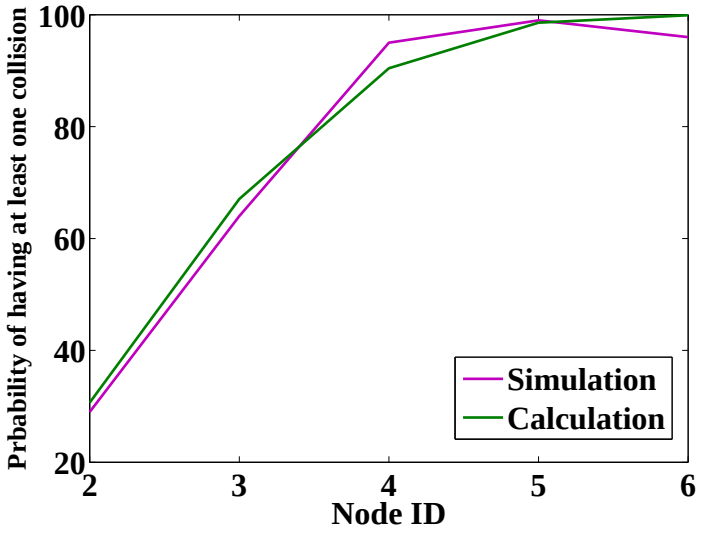
Figure 5.6: Trade-off between Service rate and PDR%

5.7 Verification of the results

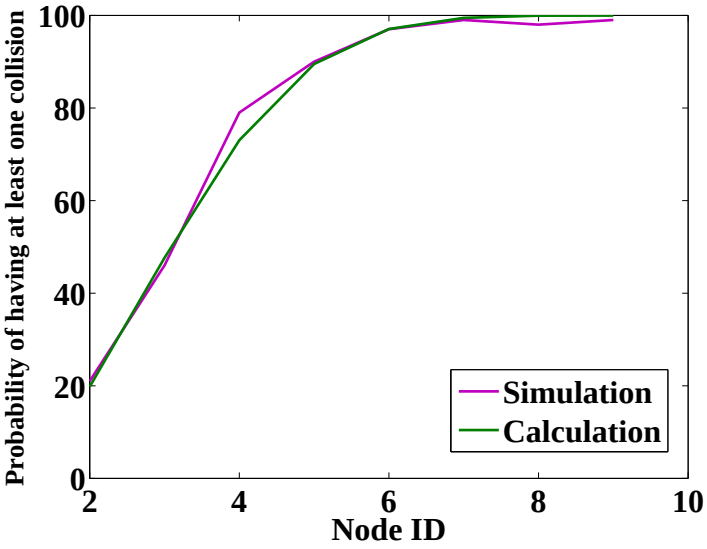
We verify the analytical results by simulating the same scenario. We design a time slotted system with $N = 3n - 3$ slots and each node randomly selects time slots for its transmission, meaning that node one selects one time slot, node two selects two time slots and so on.

After choosing the time slots for all nodes, we count the number of collisions at each receiver. As a result, at each receiver, the common time slots between the receiver and both its one-hop neighbor in two directions are counted. E.g. in node three, we count the common time slots between nodes (3, 2), (3,4), and (2, 4). It shows the number of collisions at node three (receiver).

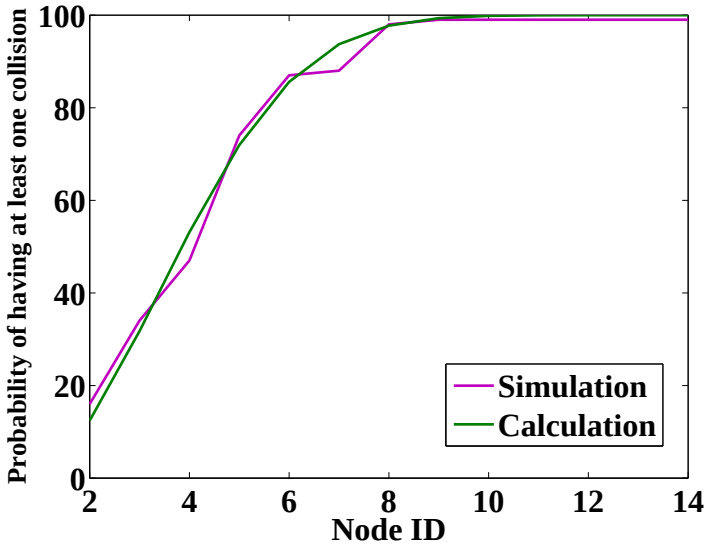
We start by verifying the probability of having at least one collision ($1 - P_i[Successful]$). We calculate this probability with both Equation 5.7 and simulation. In the simulation, we count the number of successful receptions and then reduce it from one. This gives the probability of having at least one collision at each node. Figure 5.7 shows the results for a chain with a different number of nodes and the analytical results match the simulation results.



(a) 7 nodes



(b) 10 nodes



(c) 15 nodes

Figure 5.7: Verification of probability of at least one collision

Then, we verify the PDR% at the sink, i.e. the results from Figure 5.4. Again, the PDR% is calculated for the packet generated at the first node in the chain. To calculate this metric in the simulation, we calculate the probability of having at least one collision at each node and then reduce it from one to obtain PDR_i. Then, we use Equation 5.12 to calculate the PDR% at the sink for the first node in the chain:

$$PDR_{min} = \prod_{i=2}^{n-1} PDR_i, \tag{5.12}$$

By comparing the Simulation (S) and analytical (A) results (Figure 5.8), we can see that the results are almost the same, although there is a small difference. This difference happens because in the simulation, most of the time, there are few time slots that are not selected by any node. So, the obtained results are for a lower effective number of time slots, which decreases the available resources. So, the number of time slots in the simulation is actually less than the number of time slots in the analytical calculation. As a result, the PDR% in the simulation is slightly less than the calculation.

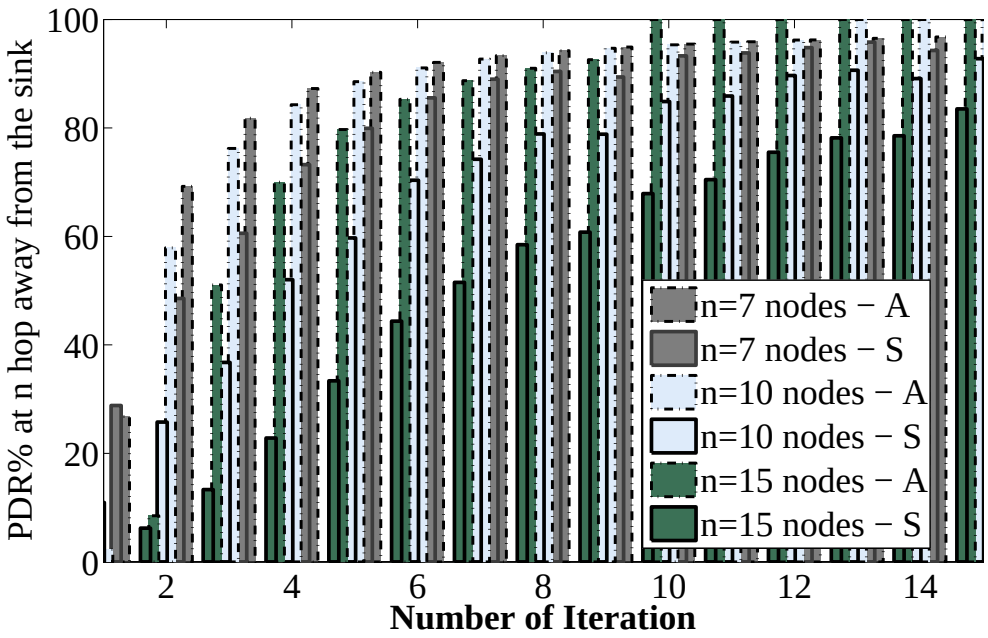


Figure 5.8: Comparison of Simulation (S) and analytical (A) results for the end-to-end PDR%

5.8 Conclusion and Future Work

In this chapter, we study the limitations of data gathering in the sense of many-to-one scenario and we provide a method to estimate the maximum amount of data that each node can generate to guarantee a pre-defined packet delivery ratio at the sink. Firstly, we calculate the maximum packet size for a scenario without considering the collision. Secondly, we add up collision to our calculation and calculate the end-to-end packet delivery ratio and also the service rate. The

results quantify the trade-off between these two metrics. Finally, we verify our analytical results by simulation of the same scenario.

The main line of future work will be to extend these results to more complex network topologies used in data gathering, especially trees. Furthermore, we will model more precisely the impact of contention, and verify the results in experimental settings.

Part II

Data Management for Modeling

Chapter 6

Improving Model Performance by Using Existing Hierarchies in the Data

Traditionally, Data Mining (DM) applications use a single model. This model is created by applying an algorithm on all the data available, or a carefully selected part of it. For example, a single model has been used to predict the trip duration in public transportations Mendes-Moreira et al. (2012).

However, in a VANET, data is collected in different settings and the transmission of data is costly. This discourages the construction of a single dataset with all the data. Furthermore, the phenomenon that is under analysis may also vary dramatically with vehicles (e.g. taxis that work regularly in an airport as opposed to the ones that work in the city center), which means that the best model may vary with the vehicle. Therefore, different learning processes must be used (i.e. different datasets, different algorithms and/or different parameter settings) for different vehicles.

On the other hand, although there is a cost associated with exchanging data, the use of data from different vehicles may, in some cases, yield gains in predictive performance that compensate for that cost. For instance, a taxi in a small town may not generate enough data to obtain reliable models and, thus, it may compensate to join its data with data obtained from others.

All of this means that, in order to maximize the quality of the results, the learning process involves selecting:

- the training data, i.e. local or collect data from other vehicles,
- the algorithm, i.e. the learning method with the most suitable bias for the data,
- the parameter settings of the algorithm, i.e. the values of the parameters of the selected algorithm that will yield the best model.

Although the number of data sources in traditional DM projects is significantly smaller than in the context of VANETs, a parallel can be established in some settings. Traditional projects often work on the raw data obtained from the transactional systems, such as the Enterprise Resource Planning (ERP) and Customer Relationship Management (CRM) systems. Many companies also

have Business Intelligence (BI) systems that collect and organize the data from the same systems Kimball et al. (2011). The data is used to compute metrics that represent the performance of the business processes (e.g. sales or new customers acquired). The metrics are stored in a Data Warehouse (DW) and are organized into dimensions (e.g., product taxonomy and store location). These dimensions have different granularity levels (e.g., store location can be analyzed at the street, city, region and country level). In this setting, DM models can be generated at different levels of granularity (e.g. customer, product, store). Traditionally, DM algorithms were applied at the global level. For instance, a single model [1] was learned to make sales predictions for all products. However, as more data is collected and the data characterizes objects at a finer level, there is a growing interest in more specific models, that represent groups or individual entities Soulié-Fogelman (2006). For instance, we may learn a direct marketing model for each segment of clusters or a sales prediction model for each individual product. Assuming that the data available is sufficient to generate models at different levels of granularity, the choice is usually based on some domain-level knowledge. For instance, in an application to detect errors in foreign trade transactions, approaches have been attempted where the modeling is done at the global level or at the local level Soares et al. (1999); Torgo and Soares (2010). However, this choice may not yield the best local models in every case. Thus, as in the case of VANETs, there is an opportunity to maximize the quality of results by selecting for each learning process, the training data, the algorithm and its parameters.

Before proposing a methodology to address this problem, we investigate empirically if a real application confirms the existence of such an opportunity. The hypothesis is whether the best results are obtained by learning different local models using different learning algorithms on training data obtained at different granularity levels.

However, the VANET data available for this project does not allow the testing of this hypothesis at a detailed level, due to the lack of timestamps in the observations. Therefore, we test the hypothesis on a traditional DM problem with the characteristics that were indicated earlier: observations are organized hierarchically using a simple DW dimension.

Thus, in this chapter we argue that the DW schema, namely the hierarchy associated with the dimensions, can be used to support the selection of the granularity that should be used to develop the DM models. Furthermore, we argue that for different parts of the problems (e.g. different products), the best granularity level may vary. The problem is concerned with the use of outlier detection methods to detect errors in foreign trade data Soares et al. (1999); Torgo and Soares (2010).

The description of the data and previously obtained results with this data are presented in Section 6.1. Section 6.2 describes our proposal for outlier detection using different granularity levels of the product hierarchy. We then present the evaluation of the proposal in Section 6.3. Finally, Section 6.4 concludes the chapter.

[1] For simplicity, we include multiple model approaches, such as ensemble learning, when we refer to "single model".

Table 6.1: Dataset features

Feature Name	Description
RN	Row Number
Origin (O)	Type of method used to insert the data into the dataset (1: Disk, 2: Manual, 3: Web)
In/Out (IO)	The flow (1: arrival, 2: dispatch)
Lot number (L)	Group of transactions that were shipped together
Document number (DN)	Official document number
Operator ID (OID)	Company that is responsible for the transaction
Month (M)	Month of the transaction
Line number (LN)	Line number in the document
Country (C)	Country of origin/destination
Product code (PC)	Code of the product
Weight (W)	Weight of the traded goods
Total cost (TC)	The total cost of the traded goods
Type	import/export: (1: Import, 2: Export)
Cost/Weight (TCW)	The ratio of Total cost per weight
Average Weight/Month (AvgWM)	Average weight of the transactions made in the same month of the product which the current transaction is from
Standard Deviation of Weight/Month (SDWM)	Standard deviation of AvgWM
Score (S)	normalized distance of the Total cost/weight value to the average value Soares et al. (1999)
Transaction number (TN)	Number of transactions made in the same month of the product which the current transaction is from
Error (E)	target value (1: error, 0: normal transaction)

6.1 Background

In this section, we start by describing the dataset of foreign trade transactions used in this work. Then, we describe the previous results obtained on that dataset.

6.1.1 Foreign Trade Dataset

The data which is used in this chapter is the foreign trade data collected by the Portuguese Institute of Statistics (INE). Users from Portuguese companies fill in forms about import/export transactions with other EU countries, containing several pieces of information about those transactions. They may insert incorrect data when filling the form. Some of the common errors are declaring the cost in Euro rather than kEuro; the weight in grams instead of kilos; and associating a transaction with the wrong item. At INE, the experts extend the dataset with a few other attributes. The most important of those attributes, in terms of error detection, is the Total Cost/Weight. This attribute represents the cost per kilo. The experts typically detect errors by analyzing this value first and then, eventually, other fields. The dataset contains the following information Soares et al. (1999)(see Table 6.1):

6.1.2 Goals and Previous results

The success criteria originally defined by the experts for the automated system is:

- The system should select less than 50% of transactions,
- The selected transactions should contain more than 90% of the errors.

In the first approach to this problem, four different outlier detection algorithms were applied Soares et al. (1999): box plot J.S. Milton, J.J. Corbet (1997), Fisher's clustering algorithm Fisher (1958), Knorr & Ng's cell-based algorithm Knorr and Ng (1998), and C5.0 Quinlan (1998). The best results were obtained with C5.0. A scoring approach was used, which orders the transactions according to the probability of being an outlier. The model obtained with C5.0 was able to identify 90% of the errors by analyzing 40% of the transactions. Fisher's algorithm using the Euclidean distance and identifying 6 clusters detected 75% of the errors by selecting 49% of the transactions.

The approach based on clustering was further explored more recently Loureiro et al. (2005). In this work, several hierarchical clustering methods were explored. The transactions allocated to small clusters are selected for manual inspection. The results show that the performance of the algorithm depends on the distance function used. The best performance was obtained with the Canberra distance.

6.2 Error Detection Methodology

In this section, an exploratory data analysis is done in Section 6.2.1. Then, we describe the use of an outlier detection algorithm for predicting the erroneous transactions in the foreign trade dataset in Section 6.2.2. Finally, the evaluation methodology and metrics use for the evaluation are described in Section 6.2.3.

6.2.1 Data Preparation and Exploration

In INE dataset, each product is presented by a unique 8-digits code. For better illustration, a small sample of the dataset is presented in Table 6.2. It contains one transaction from each of the months January, February, March, May, June, August, September, and October in 1998 and January and February in 1999.

The 8-digits code can be decomposed to create the data hierarchy. Level 1 is defined by 8-digits, which means that individual products are identified using the 8-digits code. All products that have the same 6-, 4-, and 2-digits belong to the same level 2, 3, and 4, respectively. For better illustration, the existing hierarchy in the product codes starting with 11 is shown in Figure 6.1. The categories for levels one and two are just shown for product codes starting with 1129 for simplicity. As expected, the number of different categories increases by going down in the hierarchy structure because each level contains multiple product categories/codes from the level beneath it.

Table 6.2: Illustrative sample of the dataset

RN	O	IO	L	DN	OID	M	LN	C	PC	W	TC	TCW	AvgWM	SDWM	S	TN	E
1	2	2	1001	10001	1727	1	1	11	85411650	10	7	739	2.435	3315.93	0.5114699	75	0
2	2	2	1001	10001	1727	1	4	3	49794616	2000	4497	2248	21.711	48369.10	0.4023847	12	0
3	2	2	1001	10001	1727	1	5	11	49794616	25	22	892	21.711	48369.10	0.4304191	12	0
4	2	2	1001	10001	1727	1	8	11	60786650	2	2	1117	27.904	17155.10	1.5614917	4	0
5	2	2	1001	10001	1727	1	10	11	60770400	1	1	1640	0.6697	418.278	2.3197148	7	0
6	2	2	1001	10002	5293	1	1	11	83134627	35	299	8572	15.754	8061.60	0.8908902	3	0
7	2	2	1001	10002	5293	1	2	11	83137027	50	609	12137	10.429	6026.43	0.2832658	50	0
8	2	2	1001	10002	5293	1	3	11	83788501	343	3685	10735	9.493	5167.60	0.2403615	33	0
9	2	2	1001	10002	5293	1	4	11	83784016	1891	19824	10482	13.201	7144.85	0.3806161	9	0
10	2	2	1001	10002	5293	1	5	11	83780035	1644	16492	10033	9.558	4724.45	0.1005249	40	0

RN: Row Number
O: Origin
IO: Input/Output
L: Lot Number
DN: Document Number
OID: Operator ID
M: Month
LN: Line Number
C: Country
PC: Product Code
W: Weight
TC: Total Cost
TCW: Total Cost / Weight
AvgWM: Average Weight per Month
SDWM: Standard Deviation of Weight per Month
S: Score
TN: Transaction Number
E: Error

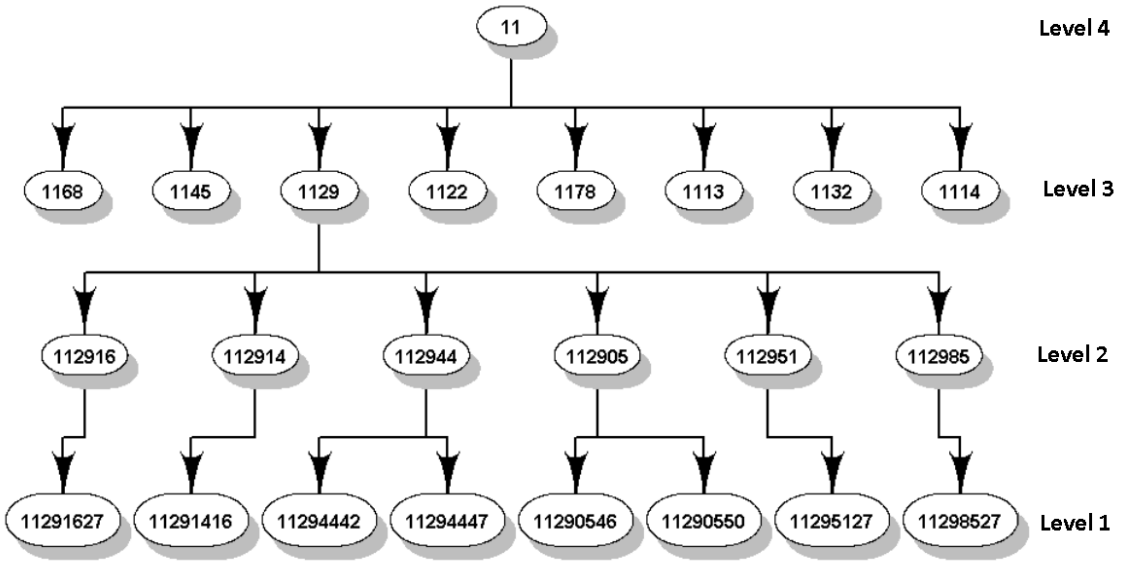


Figure 6.1: An example of existing hierarchical structure within the data set: Product codes starting with 11 and for simplicity only show the product codes starting with 1129.

An important fact about this problem is that it is very unbalanced, as is typical in anomaly detection problems Chandola et al. (2009). There are very few errors: less than 1% of the transactions are erroneous.

A summary of the datasets used for the experiment is shown in Table 6.3.

The number of categories in each month varies between 3411 to 4697 categories. Actually, these numbers are the number of unique product codes at the first level (common 8-digits). On average, there are around 9 to 16 observations for each category at the same level. Obviously, at the higher levels, the number os observations is higher than this level due to aggregation (Figure 6.2). Another fact about the data is that there are a few errors in each month which is less than 1% percentage for all months.

For example, the number of categories in August were 4522, 3032, 963, 94 for level 1 (8-digits), 2 (6-digits), 3 (4-digits), and 4 (2-digits) respectively. Considering all months, the maximum number of categories are 4697, 3109, 972, 95 for levels 1, 2, 3, and 4 respectively, and the corresponding minimum values are 3411, 2449, 833, 90.

6.2.2 Error detection method

To find the errors, an outlier detection algorithm is applied at each level of hierarchy for each product code. We used the LOF Breunig et al. (2000) algorithm. The LOF algorithm computes the density of each object in terms of the distance to its k-closest neighbors. If the local density of an object is less than of its neighbors, then the object is considered to be an outlier Breunig et al. (2000). Rather than a label of outlier or non-outlier, LOF computes for each observation an *outlierness score*.

Table 6.3: An exploratory data analysis for datasets

Month	Number of categories at the first level	No. of errors	The percentage of errors per total transactions	The average number of examples for each category
9801	3411	79	0.24	9.36
9802	3757	109	0.27	10.47
9803	3884	150	0.33	11.25
9805	4251	158	0.29	12.33
9806	3697	173	0.45	10.12
9808	4522	130	0.20	13.93
9809	4697	189	0.24	16.36
9810	4249	258	0.44	13.37
9901	3662	256	0.65	10.46
9902	3763	158	0.37	11.13

We used the implementation available from the DMwR package Torgo (2010) of R R Core Team (2014a). The number of neighbors that is used in the calculation of the local outlier factors is k=5.

Afterward, in our approach, we use the outlier scores and sort them in ascending order. The first (lower) quartile is the median of the first half of the data ($Q1$), the second quartile is the median, and the third (upper) quartile is the median of the second half of the data ($Q3$) J.S. Milton, J.J. Corbet (1997) (see Figure 6.3). The following equation shows the Interquartile Range (IQR) 6.1:

$$IQR = Q3 - Q1 \tag{6.1}$$

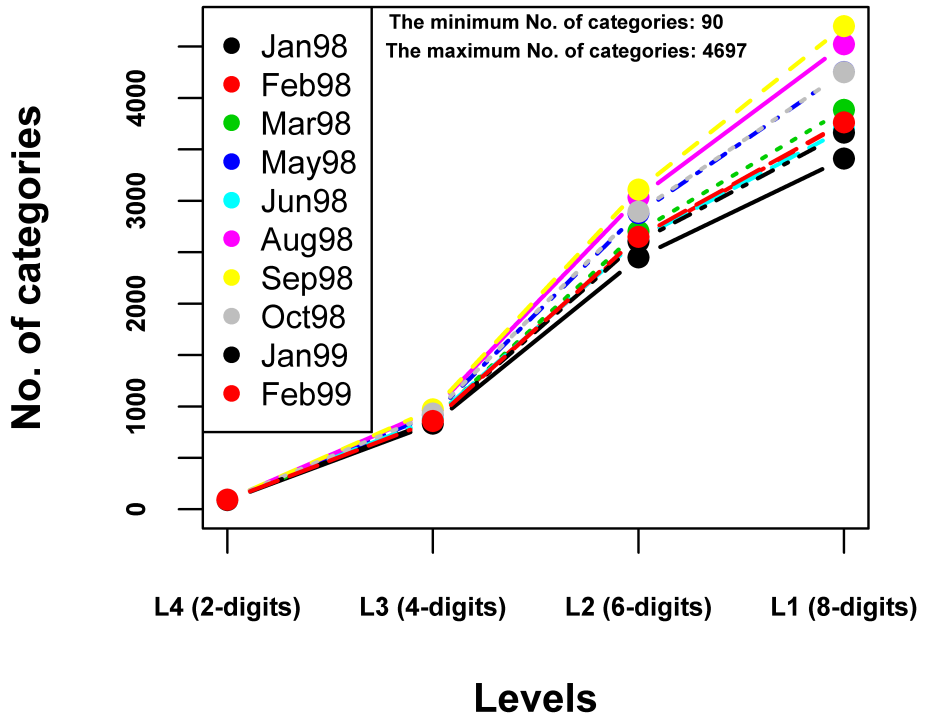


Figure 6.2: Number of catogories for each level in each month

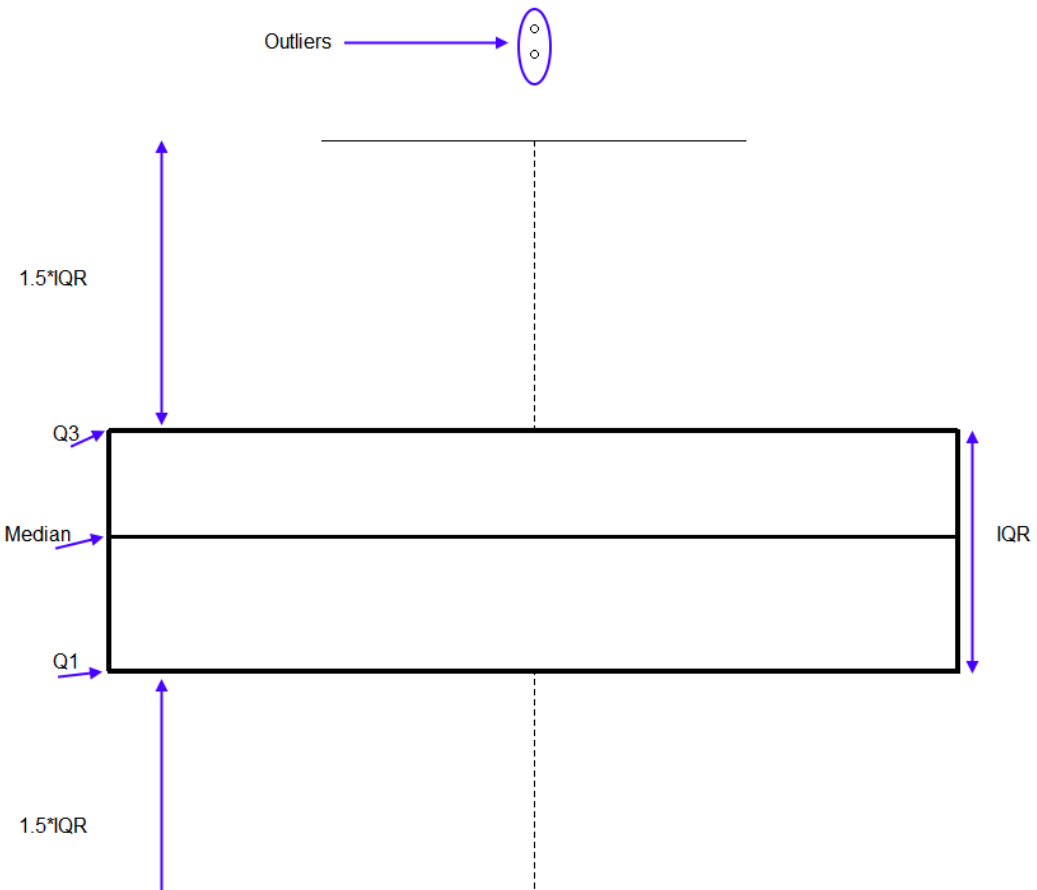


Table 6.4: Concepts used in evaluation metrics

		Ground Truth	
		True	False
Predictions	True	*tp*	*fp*
	False	*fn*	*tn*

The factors that are substantially larger or smaller than the other factors are referred to as outliers. The potential outliers are the ones that fall below Equation 6.2, or above Equation 6.3. However, for our purposes, the only values that we are interested are the largest ones, representing the transactions that may have more impact on the foreign trade statistics that are computed based on these data.

$$Q_1 - 1.5 * IQR \tag{6.2}$$

$$Q_3 + 1.5 * IQR \tag{6.3}$$

By applying this method to the data hierarchy at different levels of the product taxonomy, we are able to investigate our hypothesis, namely that the best results can be obtained at different granularity levels. For each product code, the LOF algorithm is applied on its associated data from four levels of hierarchy and then the performance of these four models are compared in the direction of our hypothesis.

It should be noted that, as in other approaches to this problem Loureiro et al. (2005); Torgo and Soares (2010), we have analyzed import and export transactions separately, as they are quite different in nature.

6.2.3 Evaluation

The goal of the outlier detection method is to meet the criteria defined by the experts. Accordingly, proper metrics as described in the following section are selected. Additionally, we note that to find the outliers, the class (field *Error*) is not used. It is used only to evaluate the observations selected by the methods as suspicious.

6.2.3.1 Metrics

Table 6.4 shows the basic concepts that are used in the evaluation measures. *True Positive (tp)* is the number of errors which are predicted correctly. If the method can not predict an error, then it is counted as *False Negative (fn)*. On the other hand, *True Negative (tn)* is the number of error-less transactions which is predicted appropriately. But if the method predicts an error-less transaction as an erroneous one, then it is counted as a *False Positive (fp)*.

The selected metrics for evaluating the model are *recall* Powers (2007) and *effort*. The *recall* is the fraction of relevant instances (e.g. errors) that are retrieved (Equation 6.4). Recall shows

the ratio of detected error per number of transactions and should be more than 90% to satisfy the expert criteria (See Section 6.1.2).

$$Recall = \frac{tp}{tp + fn} \tag{6.4}$$

Besides this common and well-known metric for model evaluation, we define another metric to introduce the cost of manually analyzing the outliers. *Effort* is the proportion of transactions that were selected by the method for manual inspection (Equation 6.5). The *effort* should be as low as possible and not higher than 50%, as indicated earlier (See Section 6.1.2).

$$\textbf{Effort} = \frac{\textit{Total number of transaction predicted as outliers}}{\textit{Total number of transactions}} = \frac{tp + fp}{tp + fp + tn + fn} \tag{6.5}$$

6.2.3.2 Methodology to obtain results for different levels of aggregation

As explained in Section 6.1.1, in our experiment, there are different products, P_i for $i = 1,...,n_1$ while n_1 is the number of unique products at the first level. Product's codes are organized in a four-levels hierarchy $(C_i^j, j = 1,...,k))$ where $k = 4$ is the number of levels. Each product is distinguished by an 8-digits code. Level one contains all the products with the same 8-digits code. Level two includes all the products with the same first 6-digits code. Similarly, levels three and four contain the products with the same 4- and 2-digits code, respectively.

The outlier detection method is used separately for the transactions aggregated at each level of the product hierarchy. To illustrate this, let us consider the product taxonomy in Figure 6.1, focusing in particular, on product code 11290550. In this case, the outlier detection is applied on data from one product code, 11290550 (level 1). When applied at level 2, the same algorithm would be applied on the transactions from all the products that have the same 6-digits product codes as the 11290550 product, i.e., 112905. This means joining the transactions from this product with the ones from products 11290550 and 11290546. At level 3, the number of product codes with the same 4-digits code (1129) is eight (the two bottom layers in Figure 6.1). So the outlier detection algorithm is applied to data from those eight product codes. Finally, all product codes that start with 11 are grouped at level 4 and the outlier detection method is applied to data from all of them. To ensure that results are comparable, the evaluation is also done at each aggregation level.

To illustrate the differences in the results provided by the method at different levels of the data hierarchy, we present the distribution of scores for the month of February, concerning import transactions for the product code *11681414* at the different levels of the hierarchy, in Figure 6.4. At level 4, around 99.7% of outlier scores are below 25. For level 3, around 99.6% of outlier scores are below 50. On the other hand, at levels 2 and 1, most of the transactions have very low outlier scores, namely below 2 (around 94% for level 2, 92.6% for level 1).

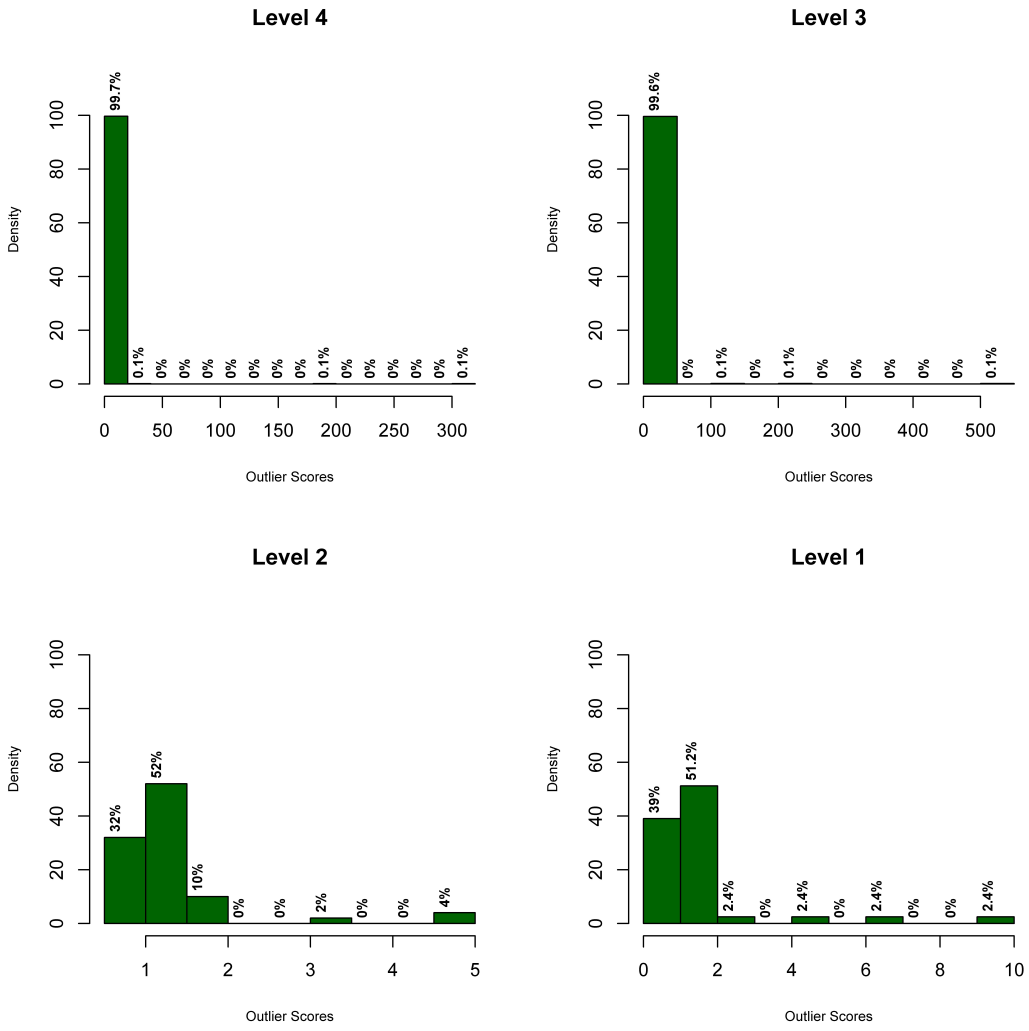


Figure 6.4: The distribution of scores obtained by the outlier detection method at the 4 levels for Product code starting with 11 in February

6.3 Results

In this section, the results of empirical evaluation of the hypothesis are investigated in this chapter are presented. In the first part (Section 6.3.1), the results for two illustrative examples are discussed. The main claim of this chapter, which is that the best model for different entities can be obtained with data from different levels of its hierarchy, is investigated in Section 6.3.2. Given that the results obtained in our experiments confirm our hypothesis, we then investigate how much improvement could be obtained if the best level could be selected individually for each entity (Section 6.3.3).

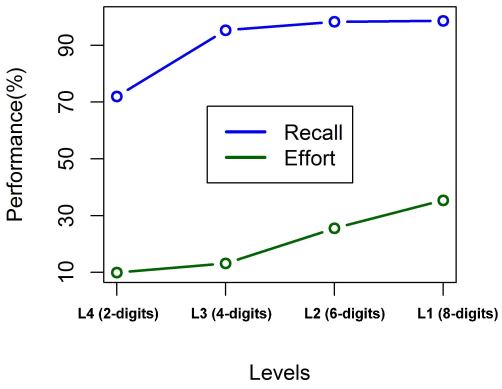


Figure 6.5: Import transactions (February)

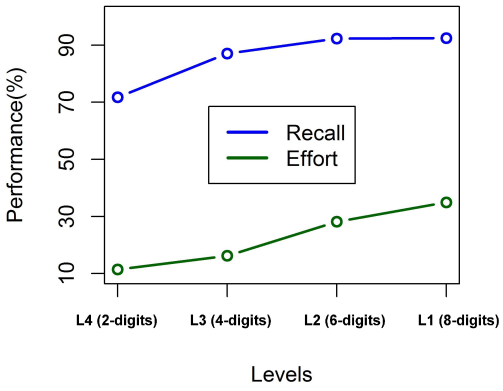


Figure 6.6: Export transactions (February)

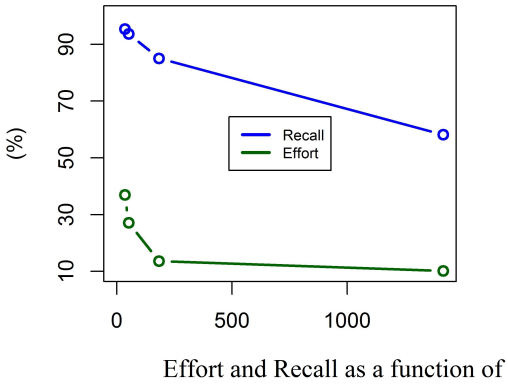


Figure 6.7: Import transactions (June)

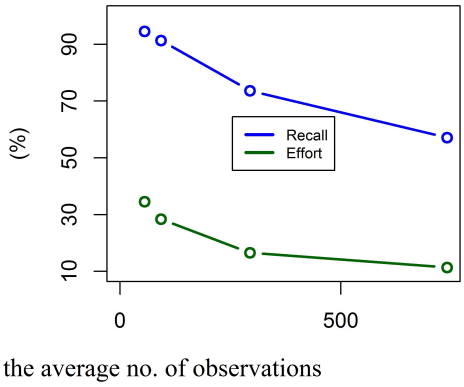


Figure 6.8: Export transactions (June)

6.3.1 Selected Examples

Here, we present the results obtained on all dataset (Section 6.1.1), analyzed according to the different levels of the hierarchy in the product codes used for data aggregation.

Figures 6.5 to 6.8 show the average results (both *effort* and *recall*) for two different months, February and June, separating import and export transactions. Analyzing the results of the method at the top level of the hierarchy for February (Figures 6.5 and 6.6), we observe that the model can predict more than 70% of the errors just by selecting 10% of the transactions. Despite a very significant reduction in the *effort*, the *recall* is not good enough to be accepted by the experts. On the other hand, for level three (4-digits), the model can predict more than 90% of the errors by analyzing just 12% of the transactions which is acceptable by the experts. So in this month, the best results are obtained by grouping the products at the four digits level of the product codes.

In June (Figures 6.7 and 6.8), the results show that grouping by two (fourth level) and four

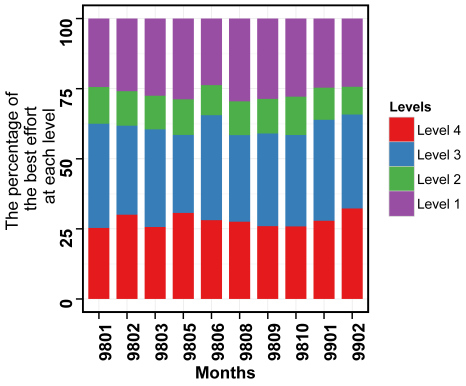


Figure 6.9: The best effort

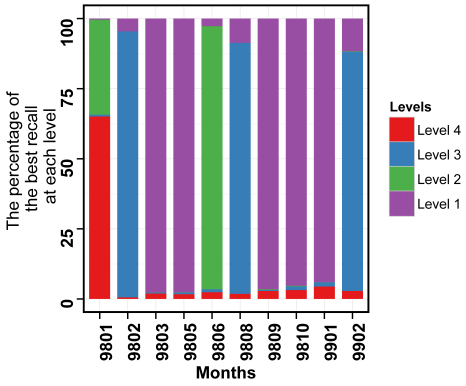


Figure 6.10: The best recall

(third level) digits of the product codes leads to just 10% of the *effort*. Although the x-axis shows the average number of observations per set of transactions analyzed, the levels are also shown from level 1, the points at the left-hand side, to the level 4, represented by the point at the right-hand side. However, this is not an acceptable result due to very low *recall* (less than 90%). The best results for this month are obtained at the second level (six digits) of product codes.

In summary, the results in these two months illustrate the claim of this work: analyzing the data at the third (four digits) and fourth levels (two digits) requires lower *effort*. But to obtain acceptable results, namely a *recall* higher than 90%, an analysis at the six digits level is the best.

6.3.2 Comparing Results Obtained with Data From Different Levels of the Product Hierarchy

To investigate the hypothesis that deciding the level of aggregation at which the data is analyzed affects the results significantly, we analyze the percentage of the products codes for which the best *effort* (the lowest *effort*) is obtained at each level of the product taxonomy for every month (Figure 6.9). A similar plot for the *recall* measure is shown in Figure 6.10.

For example, in February 1999, the best (lowest) *effort* is obtained at the level 1 (8-digits) in 24% of the product codes, while for 10% of the products, it is at level 2 (6-digits), 34% at level 3 (4-digits), and 32% at level 4 (2-digits).

In the same month, a similar scenario is observed with the *recall* measure: the best (highest) *recall* is obtained at level 1 for 11% of the products, while for 1% it is at level 2, 85% at level 3, and finally only 3% at level 4.

According to the graphs, the best results are obtained at different levels of the taxonomy for different products. These results confirm our hypothesis: *The best results are obtained by using data at different level of granularity.*

In other words, for some products, the best results are obtained at the product level (8-digits) while, for other products, the best results are obtained by aggregating transactions at the fourth (two digits), the third (four digits) or the second (six digits) levels of the taxonomy. This means that, for each product, it is important to decide the level of data aggregation to use, if any.

6.3.3 Personalized Data Selection is Required for Optimal Results

The average results presented in the previous section, clearly show that the best results are obtained by analyzing the data at different levels of granularity, as defined by the hierarchical structure which is used to organize the products. Thus, it can be expected that the global results can be improved by selecting for each individual entity (i.e. product in the current case) the level in which it should be analyzed.

6.3.3.1 Best Level According to Single Criterion

In fact, the illustrative examples presented in Table 6.5, where the *effort* for several product codes at different levels of the hierarchy is given, provide additional evidence supporting our hypothesis. The optimal selection for each product is the lowest obtained *effort* amongst all four levels. Clearly, the best selection is varied for different products. Therefore, modeling at the same level for all product codes does not lead to the best results and may cause a performance degradation.

Table 6.5: Illustrative results for effort obtained by applying the method at different levels of the product hierarchy. The ***optimal*** selection is the lowest effort amongst 4 levels.

Product Code	**Level 4 (2-digits)**	**Level 3 (4-digits)**	**Level 2 (6-digits)**	**Level 1 (8-digits)**
09751691	0.09	***0.07***	0.13	0.23
33292727	***0.08***	0.11	0.11	0.11
88065114	***0.13***	0.20	1.00	1.00
44325116	0.08	0.07	0.14	***0.05***
85224327	0.10	***0.09***	0.12	0.12
82225014	0.10	***0.06***	0.10	1.00
23290416	0.13	***0.13***	0.15	0.14
49391616	0.10	0.13	0.07	***0.06***
49395127	***0.10***	0.13	0.20	0.20
44224327	***0.08***	0.12	1.00	1.00
44321416	0.08	***0.07***	0.12	0.11
44751627	***0.08***	0.14	0.11	0.11
44142716	0.08	0.20	0.20	***0.08***
44531627	***0.08***	0.11	0.13	0.13
30680327	0.16	***0.11***	1.00	1.00
30294327	0.16	***0.14***	1.00	1.00

The average results for each month are also calculated and shown in Table 6.6. For each product, the optimal *effort* is calculated and then its related *recall* at the selected level is found. Suppose that the best *effort* for product code i is obtained from level 2. So the best level for this product code in our model is C_i^2. Then the *recall* for the same level (C_i^2) is selected. The process is repeated for all products in a given month and the average is presented the Table 6.6. This procedure is also done for the best *recall* and the related *effort* at the same level.

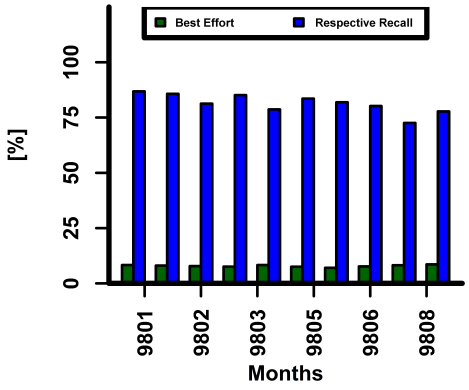


Figure 6.11: The average best effort with respective recall for each month

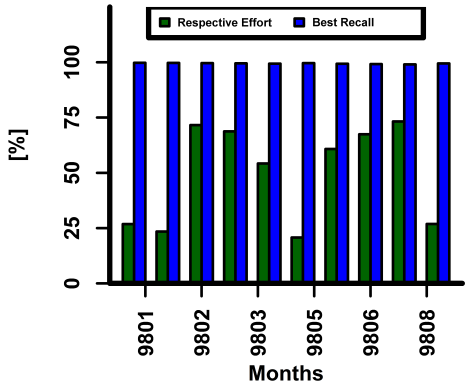


Figure 6.12: The average best recall with respective effort for each month

The average of best *effort* for all months is always below 10% (Figure 6.11) and the average of best *recall* is also over 99% (Figure 6.12). Both results are within the limits of acceptance that were defined by the experts. However, if we take both criteria into account, *effort* and *recall*, neither of the strategies presented here, i.e. focusing on either *effort* or *recall*, achieves acceptable results. Table 6.6 shows that in the optimal *effort* strategy, the corresponding average *recall* is always below 90%. The scenario for the optimal *recall* strategy is better, as there are several months with a corresponding average *effort* below 50%, but in the majority of the cases, the *effort* is above the established maximum. In summary, the results are not acceptable by the experts in both cases. Thus, a better strategy is needed.

Table 6.6: The average result for the best effort and its related recall and the best recall and its related effort.

Month	Optimal effort	Recall relative to the optimal effort	Optimal recall	Effort relative to the optimal recall
9801	8.31	86.85	99.73	26.83
9802	8.10	85.72	99.69	23.45
9803	7.85	81.27	99.60	71.61
9805	7.61	85.15	99.49	68.74
9806	8.33	78.68	99.34	54.23
9808	7.56	83.55	99.59	20.76
9809	7.08	81.93	99.31	60.81
9810	7.70	80.25	99.19	67.42
9901	8.22	72.58	99.01	73.25
9902	8.61	77.78	99.44	26.89

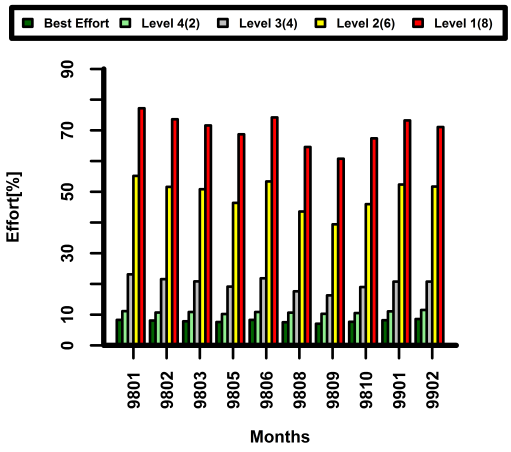


Figure 6.13: The average effort in each month when the training data is always selected from: The best level (Dark Green), Level 4 (Light Green), Level 3 (Gray), Level 2 (Yellow), or Level 1 (Red)

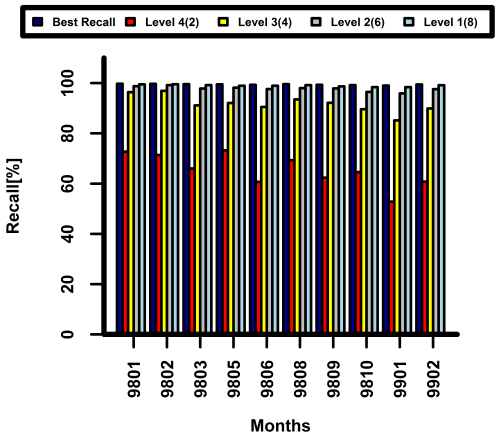


Figure 6.14: The average recall in each month when the training data is always selected from: The best level (Dark blue), Level 4 (Red), Level 3 (Yellow), Level 2 (Gray), or Level 1 (Light blue)

6.3.3.2 Best Level vs. Fixed Level

In the previous section, we evaluated the approach of choosing the granularity of the data for each individual product code, considering one evaluation criterion at a time. Significant gainings were obtained in terms of the corresponding criterion, even if for the other criterion they were not satisfactory. The question can be raised if a simpler approach, such as using the best overall level for all products, could not achieve comparable results.

Suppose that level four (aggregation with 2-digits product codes) is used for modeling all products. Figure 6.13 shows the average *effort* for all product codes in each month. It compares using a fixed level for modeling instead of using the best level for each product code. The figure shows that choosing the best level, the *effort* is always below 10% while by choosing any of the other levels systematically, the *effort* is always higher than 10%. In fact, because the best level is varied between different levels and it is not fixed, the average *effort* for the best level is lower than all individual levels.

The results for *recall* are also illustrated in Figure 6.14. The only difference between *recall* and *effort* is that the *recall* for lower levels is higher due to more specific model for each individual product. But the *effort* for the higher levels, where there is more data for modeling, is lower (better) than other levels. Figure 6.14 shows that the average *recall* for the best level is almost higher than other levels which is the same result as *effort*. Accordingly, this strategy also cannot satisfy expert advice.

6.3.3.3 Best Level Taking into Account the Expert Advice

Although the optimal strategies focusing on *effort*/*recall* are quite good concerning the measure they optimize, the corresponding *recall*/*effort* are not acceptable by experts. For that reason, two additional strategies are attempted. They consist of determining the best value for a single measure, ensuring that the value for the other is within the limits established by the experts. For instance, the best *effort* for each product code is selected by considering only the results with a valid *recall* (more than 90%). The same course of action is performed for calculating the best *recall*. When computing the best *recall*, only the results with an *effort* less than 50% are considered. Finally, an average over the acquired results for each month are computed and shown in Table 6.7.

Table 6.7: The average result for the best effort for acceptable recall and the best recall for acceptable effort by expert.

Month	Optimal effort w/ acceptable recall	Related recall to the optimal effort	Optimal recall w/ acceptable effort	Related effort to the optimal recall
9801	11.38	100.00	97.82	10.92
9802	11.23	99.91	98.11	10.45
9803	15.05	100.00	96.61	10.52
9805	12.92	100.00	97.23	10.09
9806	15.45	100.00	96.17	10.82
9808	11.06	100.00	98.01	10.27
9809	11.81	100.00	97.74	9.81
9810	15.61	100.00	95.72	10.35
9901	20.66	100.00	93.22	10.52
9902	14.45	100.00	96.74	10.89

The results obtained with these new strategies are very good. When focusing on *effort*, the largest value is 20.66%, which is much less than the limit of 50%. In fact, most of the values are close to 10%. Furthermore, the corresponding *recall* is, except in one case, 100%. When compared to the previous effort-based strategy (Table 6.5), we observe that with a small increase in the *effort* (typically 4 p.p.), we obtain a rather large gain in *recall* (typical 15 p.p.). When focusing on *recall*, the results are also good, with an *effort* which is always around 10% yielding *recall*s that are almost always above 95%. When compared to the previous recall-based strategy (Table 6.5), the improvements are not as impressive. However, it should be noted that, in the new strategy, the results are all withing the limits defined by the experts, unlike in the previous one.

Comparing this strategy with the selection of the best level taking into account a single criterion (Section 6.3.3.1), on average a significant gain for both *effort* and *recall* is obtained (comparing results in Tables 6.6 and 6.7). For example, in Table 6.6, the related *effort* to the optimal *recall* is 20.76% in the best case (in August) while in the same month and by taking into account the expert advice (Table 6.7), the *effort* is 10.27% (around 10% improvement/gain for *effort*). On the other hand, the best-related *recall* obtained in Section 6.3.3.1 is 86.85% for January 1998 while

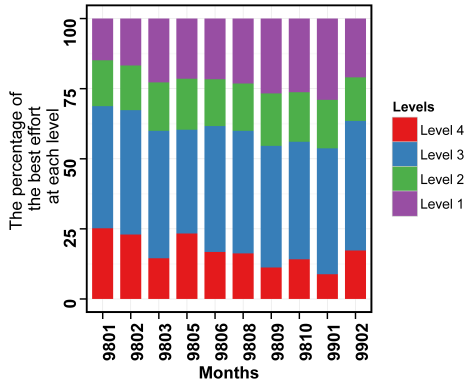


Figure 6.15: The optimum effort

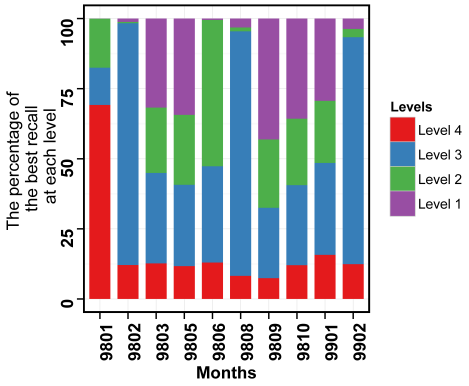


Figure 6.16: The optimum recall

for almost all months the *recall* is 100% when taking into account the expert advice. Therefore, around 14% gain is obtained with this strategy for *recall*.

Figures 6.15 and 6.16 show that for both strategies, namely effort-based (Figure 6.15) and recall-based (Figure 6.16), the selected level of granularity is still distributed amongst all possible values. Thus, this figures confirm our hypothesis, that the optimum level to choose the data for making a local model is varied between all levels.

6.4 Summary

In this chapter, we have investigated the effect of aggregating data at different levels of a product taxonomy in the performance of an outlier detection method applied to the problem of identifying erroneous foreign trade transactions collected by the Portuguese Institute of Statistics (INE). The approach is tested on 10 months of data and the results are evaluated in terms of the *recall* and the *effort* involved in the manual analysis of the selected transactions.

The results show that, depending on the product, the best results can be obtained at different levels of aggregation. In fact, in some cases, the best results are obtained at the lowest level where there is no grouping.

This means that different aggregation levels should be selected for different local models, i.e. products in this application. One approach that can be followed is to use a metalearning approach to mapping the characteristics of the data with the ideal level of aggregation Brazdil et al. (2009). This approach will be presented in Chapter 7 and is tested on the same problem in Chapter 8.

Although the results on the error detection problem are important to illustrate the generality of the proposed approach, our main goal is to test it on the generation of predictive local models for VANETs. This will be investigated in the Chapter 9.

Chapter 7

A Metalearning Framework for Model and Data Granularity Selection

Business Intelligence (BI) can help companies to implement an effective strategy or model reaching competitive market advantage and long-term stability. A BI application uses data from a Data Warehouse (DW) which is organized into dimensions. Taking into account the DW dimensions, model creation can be done in each dimension independently. Comparing the performance of these models reveal that the best performance can be obtained at different levels of hierarchy (Chapter 6).

The data management problem addressed in this thesis is concerned with settings in which multiple local models are developed, i.e. models that are applicable to a small part of the space of data (local models), instead of a single global model. This is true in the VANETs, for instance, in the problem of predicting the duration of taxi trips addressed in this project, in which a model is developed for each car (Chapter 9). But local models can also be useful in more traditional Data Mining (DM) problems, in which the spatial issues involved in VANETs are not applicable. For instance, in the foreign trade error detection problem (Chapter 8), separate models can be developed to detect errors in transactions of different products. Figure 7.1 shows this model development in the data hierarchy. Different models are created at different levels and the performance of them is compared to select the best one.

In these settings, the training data for a local model can be the local data, i.e. data from the part of the space that the model is associated with, or the learning process of a specific local model may use data from other parts of the data space. Despite potential gains in the quality of the models, the sharing of data may have costs. For instance, due to shared medium, transmitting data between vehicles has cost. Even in more traditional settings, such as in the foreign trade error detection problem, using more data for training the model increases the computational costs of that task.

The results in the previous chapter confirm that there is potential in sharing data in the process of learning local models. However, the results also show that, depending on the part of the data space (i.e. the local model), the best results can be obtained at different levels of aggregation. Additionally, the results show that, in some cases, the best results are obtained at the lowest level

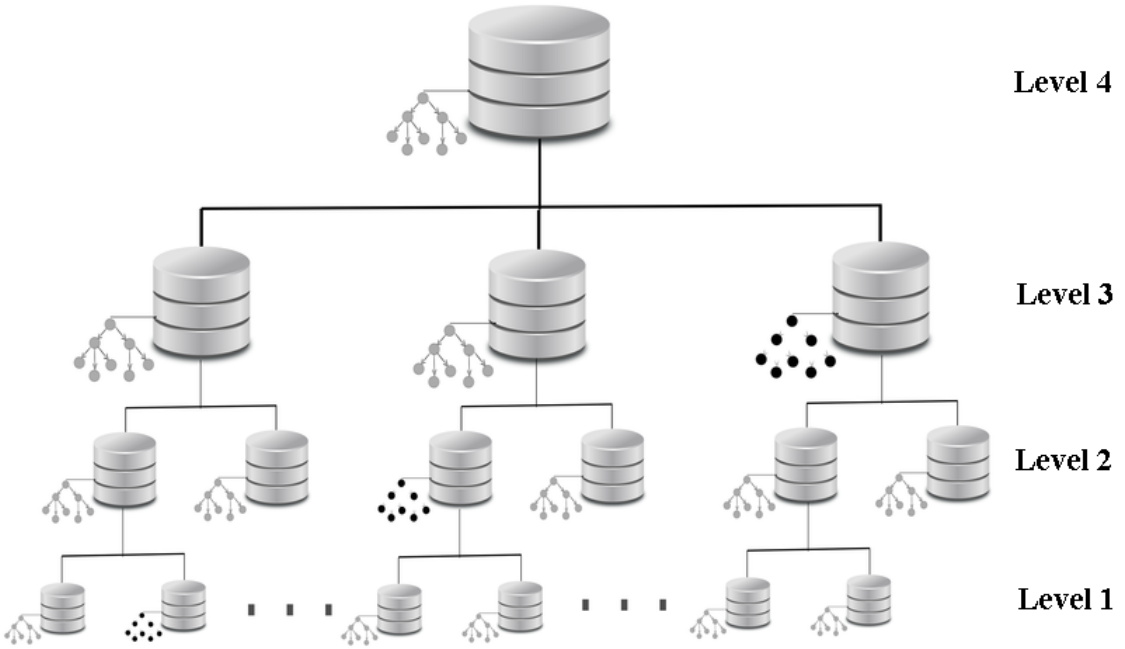


Figure 7.1: Illustration of a hierarchy in datasets: for each category the best performance (black model) is obtained at different levels

where there is no grouping and the local model is trained using only local data. This means that different aggregation levels should be selected for different local models.

One solution can be metalearning Vilalta and Drissi (2002). It models the relationship between the characteristics of the data with the performance of the algorithms. It is often used to select the best algorithm for a specific problem, such as classification or regression. Here, we use this approach to address the problem of selecting the right level of granularity, as defined by DW dimensions, to model a DM problem. In our approach, the characteristics of the data are mapped to the performance of the learning algorithms at different levels of granularity.

The methodology which is used in this chapter is introduced in Section 7.1.

7.1 Methodology

In this section, firstly the traditional method is presented (Section 7.1.1). Then the proposed framework is discussed by describing the definitions (Section 7.1.2), metafeatures (Section 7.1.2.1), and metadata (Section 7.1.2.2).

7.1.1 Traditional method

In the real world, data is gathered by individual entities (E_i), where each entity is potentially associated with multiple examples (local data). For instance, an entity can be a product code or a taxi that has associated data. In the traditional approaches, a global model is developed by

applying machine learning algorithms to the data from all entities to predict the future events for all entities.

Suppose that the available dataset consists of n entities, $\{E_i, \forall i \in \{1,...,n\}\}$. Each example associated with the entity i consists of a set of m features, $X_i = (x_1, x_2, \ldots, x_m)$, and a target variable, y_i $((X_i, y_i))$. Then, a global model that maps the features (X_i) to the target variable (y_i) is learned from a training set (Figure 7.2). Finally, its performance is evaluated on the test dataset. Therefore the dataset used in the traditional data mining approaches is like $DB = \{E_i, X_i, y_i\}, \forall i \in \{1,...,n\}$.

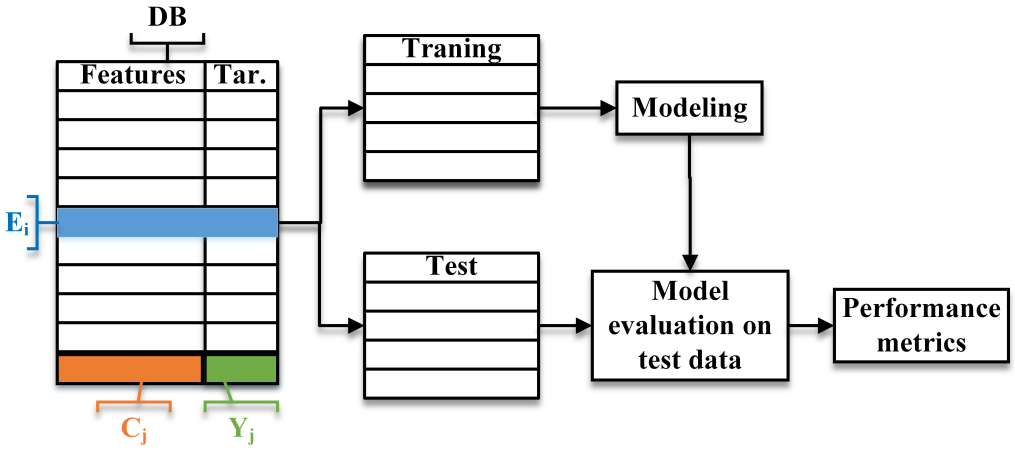


Figure 7.2: Traditional Data Mining approach

As an example, in the previous chapter (Chapter 6), in the INTRASTAT dataset, each product code represents an entity and all data associated with this product code is considered as local data. The set of features in this case are:

$$\begin{aligned} C_i = (&Origin, In/Out, Lotnumber, Documentnumber, OperatorID, Month, \\ &Linenumber, Country, Productcode, Weight, Totalcost, Type, \\ &Cost/Weight, AverageWeight/Month, StandardDeviationof \\ &Weight/Month, Score, Transactionnumber) \end{aligned} \tag{7.1}$$

The target variable for this dataset is *Error* which makes this a classification task. According to the Tables 6.2 and 6.3, at the lowest level, there are nine observations on average for each category which indicates the number of observations for the local data. Equation 7.1 shows the complete list of features for each product code in the traditional approach.

Another case study which is studied in this thesis is the taxi dataset taken from DRIVE-IN project Cmuportugal.org (2014). In this dataset and the lowest level, each taxi creates a category and all the observations for each taxi is the local data. Table 7.1 shows the number of taxis and

the average number of observations for each taxi in each month. There are around 440 taxis and at the lowest level, there are between 1238-1484 observations for local data.

Table 7.1: A simple statistics about Taxi dataset

Month Month	No. of Taxis No. of Taxis	Average number of observation per each taxi
201301	439.00	1328.20
201302	443.00	1238.50
201303	443.00	1356.60
201304	446.00	1302.70
201305	443.00	1484.90
201306	442.00	1385.70
201307	440.00	1416.90
201308	435.00	1253.90
201309	432.00	1382.10
201310	434.00	1478.50

In this dataset, the complete list of features is shown in Equation 7.2. The name of features is also provided in Table 7.2. The target variable is the trip duration. Therefore, the problem is a regression task. The objective is to predict the trip duration (dt).

$$C_i = (X\_id, driver, ts, st, id, pst, track, dd, src, dst, n, pos, dt) \tag{7.2}$$

Table 7.2: The description of Taxi dataset's features

Feature	Feature's description
X_id	Event identifier
driver	Taxi driver identifier
ts	Time stamp of the event (seconds)
st	Taxi state
id	Taxi identifier
pst	Previous state identifier
track	GPS track, encoded with polyline algorithm
dd	Distance between src and dst (meters)
src	GPS coordinates of the source position
dst	GPS coordinates of the destination position
n	Name of the taxi stand
pos	Location of the taxi stand
dt	Duration of the trip (seconds)

The traditional scheme is unidirectional scheme (See Figure 7.2). It maps a set of features to a target variable.

7.1.2 Proposed Metalearning Framework

In this section, we propose a generalized model that can be used in a broad range of applications. In the proposed model, there are k entities ($E_i, \forall i \in \{1,\ldots,k\}$) which are organized into several hierarchies (H_p). For instance, entities are taxis which are organized according to their geographical positions (hierarchy). Each hierarchy is organized in different levels (Equation 7.3). For example, taxi level, road-side unit level, and a central unit level where the data from all taxis is collected.

$$H_p = \{L_{p,q}\}, \forall q \in \{1,\ldots,l_p\} \tag{7.3}$$

where l_p is the number of levels in the hierarchy p. However, since, we have a single hierarchy in our case study, we ignore hierarchies in the notation, for simplicity, thus, $L_{p,q} \equiv L_q$.

In addition, each level in a hierarchy has a domain which is organized in categories, i.e., all taxis in the same geographical location or all taxi in the city. Another example are product codes which have 8-digits or 6-digits common product codes. This can be shown as:

$$L_q = \{C_{q,r}\}, \forall r \in \{1,\ldots,k_q\} \tag{7.4}$$

where k_q is the number of categories in the level q of the hierarchy p. Each category contains a set of entities (the ones that are associated with it). For instance, road side unit category contains all taxis within its communication range. In case of product codes, level two contains all the

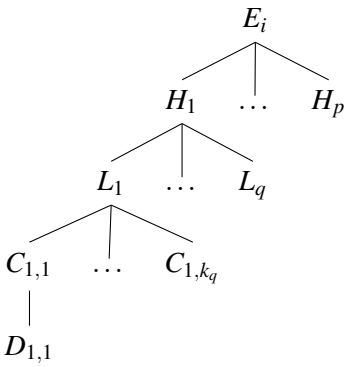


Figure 7.3: The proposed hierarchy structure

transactions that have the same 6-digits code. However, there is one particular category which is at the same level as the individual entity (i.e. Equation 7.5). For instance, in the taxi dataset, at the taxi level, each taxi makes a category and in the INTRASTAT dataset, each product at the first level makes a category.

$$C_{1,r} = \{E_r\} \tag{7.5}$$

where $C_{1,r}$ is the category r at level one and contains only one entity (E_r).

The relationship between levels is defined as a parent and children relationship. We define C_{q,r_1} (category r_1 at level q) as the parent of C_{q+1,r_2} (category r_2 at level $q+1$) if C_{q+1,r_2} is a subcategory of C_{q,r_1} (Equation 7.6). In the other words, all entities contained in C_{q+1,r_2} are contained in C_{q,r_1} and also that the set of entities in C_{q,r_1} is the union of the set of entities of all C_{q+1,r_2} which are subcategories of C_{q,r_1} (Equation 7.7).

$$C_{q+1,r_2} \text{ is a subcategory of } C_{q,r_1} \Leftrightarrow C_{q,r_1} \text{ is the parent of } C_{q+1,r_2} \tag{7.6}$$

$$C_{q,r_1} = \bigcup_{C_{q+1,r_2}} \{E_j\}, \forall C_{q+1,r_2} \in \{\text{subcategories of } C_{q,r_1}\} \tag{7.7}$$

For better illustration, Figure 7.3 shows this hierarchical structure in a tree structure.

On the other hand, for modeling, the data is also needed to follow the same structure. As mentioned previously, suppose that the available dataset for a given supervised learning problem consists of k entities, $E_i, \forall i \in \{1,\ldots,k\}$. There are also several examples associated with each entity, $\{e_{i,1},\ldots,e_{i,o}\}$. We refer to this set of examples associated with an entity E_i as its *local data*. Each example associated with the entity i consists of a set of m features, $X_i = (x_1, x_2, \ldots, x_m)$, and a target variable, y_i ($e_{i,j} = (X_{i,j}, y_{i,j})$). The features ($x$'s) have fixed meaning with different values for each example. In taxi dataset, the examples are the GPS data obtained from taxis and the features are the characteristics of each observation (see Equation 7.2).

Therefore, the dataset associated with the entity i can be defined like $D_i = \{e_{i,1},\ldots,e_{i,m}\}$. For

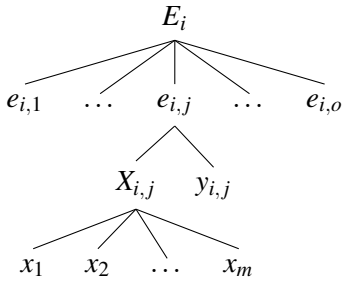


Figure 7.4: The data structure used in the proposed model

example, in Figure 7.3, $D_{1,1}$ is the dataset related to the category one at the level one. In this level, each identical entity build a category and therefore, $D_{1,1}$ is the data associated with the entity one at the level one (at this level, C_1 contains only E_1). Thus:

$$D_{1,1} = \{e_{1,1}, \ldots, e_{1,o}\} \tag{7.8}$$

Which means the dataset for the category one at the level one has only one entity and contains all the examples associated with this entity. For simplicity, $D_{1,1} \equiv D_1$.

The dataset structure for the entity i is shown in the Figure 7.4.

Generally, $D_{q,r}$ is the dataset for the category r at the level q and is contained $n_{q,r}$ examples associated with the category r at the level q. According to the Formula 7.6, $D_{q,r}$ can be defined as:

$$D_{q,r} = \bigcup_{r_j} D_{q+1,r_j}, \forall r_j \in \{\text{children of } C_{q,r}\} \tag{7.9}$$

which means that the data for an upper level's category is the union of the data from all of its children. In other words, if D_j is considered as the dataset for the entity j at the first level, then we can conclude that the dataset for the category r at the level q is:

$$D_{q,r} = \bigcup_{j} D_j, \forall j \in \{\text{indices of entities in } C_{q,r}\} \tag{7.10}$$

This structure in the data and the whole methodology is depicted in the Figure 7.5. The data is organized according to the above structure on the left side of the figure. Then several algorithms are applied to the data from different levels of hierarchy for each entity (Performance Evaluation block). In parallel, the values of a set of metafeatures (Section 7.1.2.1) are also calculated based on the data for each entity and at different levels (Metafeatures calculation block). Having a set of values for metafeatures and the best performance obtained from the performance evaluation block, the metadata (Section 7.1.2.2) is created for each entity. Therefore, the metadata has k rows equals to the number of entities. In the final step, several machine learning algorithms are applied to the metadata to find a recommendation for each entity which is a combination of a level and an

algorithm (see Equation 7.11).

$$\begin{aligned} Input &: \{\underbrace{E_i}_{entity}\} \\ Output &: \{\underbrace{\quad l \quad}_{recommended\,level}, \underbrace{\quad g \quad}_{recommended\,algorithm}\} \end{aligned} \tag{7.11}$$

Having this methodology in mind, instead of using a global model for all entities, it is possible to develop a local model for each entity. The advantage of using a local model for prediction is that the model may be more accurate than the global model due to the having only related examples to the entity. For example, the data from a taxi in a highway may not be useful for prediction of trip duration of a taxi in the city center. On the other hand, the disadvantages of creating a local model for each entity, E_i, is that the training data associated with it, D_i, may be insufficient to learn a reliable model.

7.1.2.1 Metafeatures

The values of metafeatures are also calculated for each entity and at all levels. In general, mf_i^q is the values of metafeatures for entity i at the level q. In result, there are q sets of values of metafeatures for each entity: $mf_i^1, \ldots, mf_i^q$. For example, mf_i^1 is obtained by calculating the values of metafeatures for the entity i and at the level 1.

Metalearning uses a set of features, called metafeatures (mf_i^q), to depict the dataset characteristics. Then it tries to find a correlation between these metafeatures and the performance of the base-level algorithms Pavel Brazdil, João Gama (2005); Christian Köpf, Charles Taylor (2000); Lagoudakis and Littman (2000). The effectiveness of metalearning highly depends on these metafeatures. Depending on the learning tasks, different sets of metafeatures are selected. In the next two chapters, two different sets of metafeatures suitable for classification (Section 8) and regression (Section 9) tasks are introduced. The full lists of these metafeatures are also presented in Appendix A.

7.1.2.2 Metadata

The dataset used for metalearning is called metadata which is data about the original dataset. The metadata for each entity consists of the values of metafeatures for different levels plus the best performance obtained from the Equation 7.12.

For each entity, the best performance ($P_{best\,i}$) is obtained from the performance evaluation block by comparing the performance of all algorithms at different levels of hierarchy and selecting the best one (see Equation 7.12):

$$P_{best\,i} = \max_{w,j}(P_{iw}^j)\ ,\ \forall w \in \{1,...,g\}, \forall j \in \{1,...,k\} \tag{7.12}$$

As an example, Equation 7.13 shows the general form of the metadata which is used in metalearning block:

$$Row\ i \rightarrow E_i, mf_i^1, mf_i^2, \ldots, mf_i^l, P_{best\,i}, \tag{7.13}$$

For a better explanation, suppose we have only one metafeature which is the number of observations (*n.examples*). Then the metadata is: the entity, the number of observations at the first level, the number of observations at the second level, ..., the number of observations at the level k, and the best performance obtained from performance evaluation block.

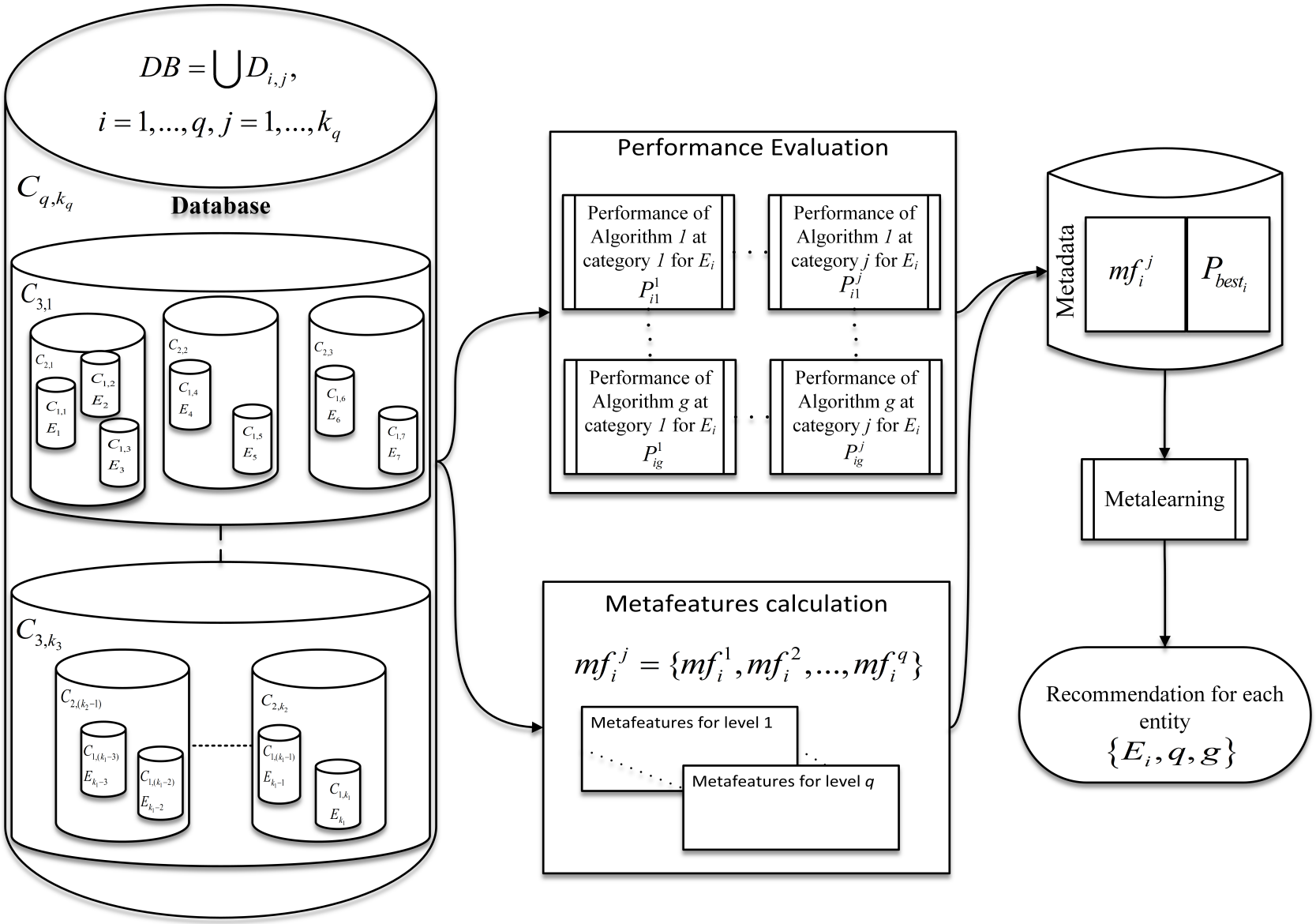


Figure 7.5: Proposed methodology used for metalearning

7.2 Summary

Metalearning approaches aim at assisting users to select appropriate learning algorithm for the particular data mining task. This problem is even worse when considering the existing hierarchy in the datasets. In this chapter, we have proposed a new metalearning framework to predict the best level of granularity to apply the recommended algorithm. The basic idea is to reduce the computational costs for applying different algorithms at different levels of granularity to reach the best performance. Our model recommends an algorithm and a level of granularity to obtain the best performance.

The next two chapters include the validation of the framework by applying it to two different datasets: Statistical dataset (Chapter 8) and Taxi dataset (Chapter 9). These two datasets are selected to evaluate the framework on two different problems: classification and regression, respectively.

Chapter 8

Validation of the Metalearning framework: A Case Study on Error Detection in Foreign Trade Statistics

In the previous chapter, we introduced a metalearning-based framework to address the problem of selecting the best level of data granularity to learn local models. This framework is useful for problems in which 1) learning processes are carried out at the local level (e.g. different vehicles in a VANET) and 2) there are costs in sharing data between learning processes (e.g. transmitting data between vehicles in a VANET), although 3) the quality of the models obtained by learning processes that combine different subsets of data may be better than the models obtained just with the local data.

In Chapter 6, we showed that such a framework may also be useful in a more traditional data mining (DM) setting, in which similar issues are relevant. For instance, if the data is organized according to a data warehouse-like (DW) structure, with hierarchies of dimensions, then the local models can be obtained on the lowest level of the hierarchy (e.g. at the product level) but data sharing can be done based on that hierarchy (e.g. the model for a given product may use data from other products in the same category).

In this chapter, we empirically test the framework on the problem addressed in Chapter 6, the detection of errors in foreign trade transactions.

The chapter is organized as follows. Section 8.1 summarizes the case study description. Section 8.2 describes our customized methodology for data analysis and metalearning to find the best level of granularity for this case study. The obtained results are presented in Section 8.6. Finally, a conclusion is presented in Sections 8.7.

8.1 Case study: error detection in foreign trade statistics

Foreign trade statistics are important to describe the state of the economy of countries Soares et al. (1999). They are usually estimated by the different national statistics institutes based on

data provided by companies. The problem in INTRASTAT dataset (See Tables 6.1 and 6.2 for the dataset description) is already describe at chapter 6, Section 6.1.1. The goal is to detect as many errors as possible – to maximize the quality of the statistics – with as little manual effort as possible – to minimize the cost.

Some of the previous work on error detection has used outlier detection, classification and clustering approaches (e.g., Soares et al. (1999); Nozari Zarmehri and Soares (2014)). In general, satisfactory results have been obtained as some approaches were able to detect most of the erroneous transactions by choosing a small subset of suspicious transactions for manual inspection. However, this was not true for all products. This is partly due to the fact that some products have very few transactions. Given that each product is analyzed individually, the decision can be based on a very small set of data.

In Chapter 6, the products are organized in a 4-levels taxonomy. An example of such a taxonomy can be Food (Level 4), Bread (Level 3), Sliced bread (Level 2), Pack of 16 slices (Level 1) (Figure 8.1).

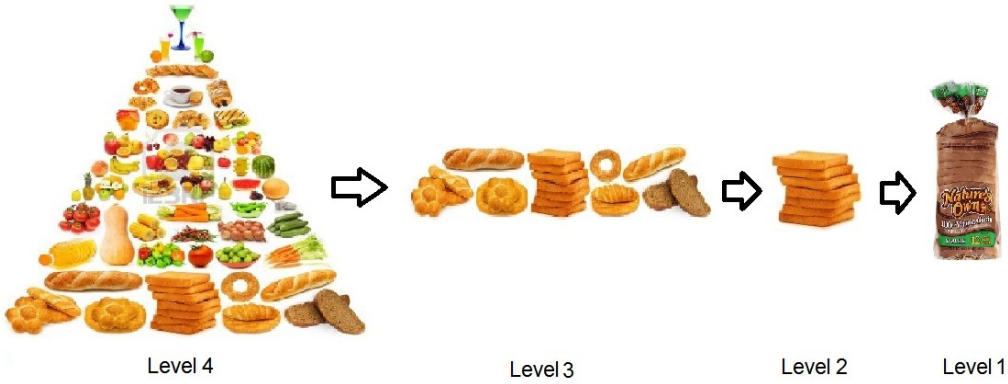


Figure 8.1: An example of existing taxonomy in foods

Each product is presented with a unique 8-digits product code (Level 1). By going up in the product taxonomy, the number of products in each level increases. Therefore, by grouping the transactions at a higher level of the product taxonomy may help to obtain better results when compared to an analysis at the product level (Level 1) itself, especially in the cases where the number of observation at this level is too low.

According to Chapter 6, the best results are obtained at different levels of the taxonomy for different products (Figure 7.1). For example, the best results for the products on the right leaf are obtained by training with the data from the third level of product taxonomy while for the products at the middle leaf, the best results are obtained by training with the data from the second level (black models in Figure 7.1). In spite of the fact that the results show that the aggregation is generally useful, they also show that the best results for different products are obtained at different levels of granularity (see Section 6.3.1).

8.2 Methodology

In Chapter 7, we introduce the framework used in this project. To obtain the best model for each entity, the training data can be used at different levels of hierarchy in the data taxonomy, including the local data, i.e. the data related to one product code, or the data from other levels, i.e. the data from level two where the data is aggregated by 6-digits product code(see Section 6.2).

In this section, the proposed framework (see Chapter 7) is customized for the problem of error detection in foreign trade transactions. The entity here is a product code with 8-digits. For example instead of E_i, P_i is used in the framework. Figure 8.2 shows the customized framework. The problem is to use the training data from the best level of granularity to apply an outlier detection algorithm. Therefore, the framework recommends a level with the best performance amongst all levels.

In our experiment, there are different products, P_i for $i = 1,...,n_1$ while n_1 is the number of unique products at the first level. The products are organized in a four-levels hierarchy ($C_i^j, j = 1,...,k$)) where $k = 4$. Each product is distinguished by an 8-digits code. Level one contains all the products with the same 8-digits code. Level two includes all the products with the same first 6-digits code. Similarly, levels three and four contain the products with the same 4- and 2-digits code, respectively. For example, all the dried fruits have a product code starting with 11 while raisins, which are a specific type of dried fruits, have code 1155.

The best model for a given product depends on the product itself and the data available. For instance, given a very specific product with a lot of transactions, the best results can be obtained by learning a model on training data from that product only. On the other hand, if the product is very general behavior or has very few transactions, the best model can be obtained by training with the full dataset. It is expected that in other cases the best model can be obtained by training on data from intermediate levels of the hierarchy. Therefore, the question is, given a product, what subset of the data should be used for training. By mapping the characteristics of the different subsets to the performance of learning algorithms, a metalearning approach can be used in this problem.

However, another question is raised. For large datasets, the number of candidate subsets is very large, so it is not feasible to consider all of them in a metalearning approach. When available, such as in our case study, a hierarchy of the observations (e.g. products, product families, ...) can be used to reduce the number of candidate sets of training data to be considered. Therefore, for product P_i the datasets considered contain the transactions concerning the levels $C_i^1, C_i^2, \ldots, C_i^k$ (where $k = 4$ in our case study). A metalearning approach is then used to choose, for a given product P_i, the level of the hierarchy, $j = 1, \ldots, k = 4$, that contains the data which is expected to generate the best model for that product.

For each unique product, metafeatures (see Section 8.2.1) are calculated at all levels, $j = 1, \ldots, k = 4$. mf_i^j is a vector containing the calculated metafeatures for the product i at the level j. The metadata consists of algorithm performance data and metafeatures. Algorithm performance data is obtained by running base-level experiments. In these experiments, an outlier detection method (LOF Torgo (2010); Nozari Zarmehri and Soares (2014)) is applied at each level, for each

product. The data from a month is used for training and the accuracy of the model is tested with the data from the next month. The performance of outlier detection for the product i on the level j is indicated by PI_i^j.

Having PI_i^j and mf_i^j enables us to create the metadata set. Each metadata row includes the product code (P_i), metafeatures for all levels of the product code ($mf_i^j, \forall j \in \{1,\ldots,k\}$), and the best performance obtained by outlier detection methods for that product among all the levels ($PI_{best\,i}$, see Equation 8.1). Equation 8.2 shows the format of the metadata sets in our experiment.

$$PI_{best\,i} = \max_j(PI_i^j)\,, \forall j \in \{1,\ldots,k\} \tag{8.1}$$

$$\{P_i, mf_i^1, mf_i^2, mf_i^3, mf_i^4, PI_{best\,i}\},$$
$$\forall i \in \{1,\ldots,n_1\} \tag{8.2}$$

By applying a learning algorithm on the metadata, we obtain a meta-model that can be used to predict the best level of granularity for each product code.

8.2.1 Metafeatures

As discussed previously (Chapter 7, Section 7.1.2.1), a set of metafeatures is selected related to the problem. In this case study and according to our dataset, we select 15 metafeatures to describe the characteristics of INTRASTAT dataset. The extracted metafeatures are shown by mf_i^j notation in the customized framework. A list of all metafeatures which are used in this study with a brief description is provided in the Table 8.1 and is also described in Appendix A.

Table 8.1: Extracted features used in metalearning - INTRASTAT dataset

Feature Name	**Description**
n.examples	Number of examples
n.attrs	Number of attributes
prop.symbolic.attrs	Proportion of symbolic attributes
prop.missing.values	Proportion of missing values
class.entropy	Class entropy
avg.mutual.information	Average mutual information
prop.h.outlier	Proportion of continuous attributes with outliers
avg.attr.entropy	Average attribute entropy
avg.symb.pair.mutual.infor	Average mutual information between pairs of symbolic attributes
avg.abs.attr.correlation	Average absolute correlation between continuous attributes
avg.skewness	Mean skewness of attributes
avg.abs.skewness	Mean absolute skewness of attributes
avg.kurtosis	Mean kurtosis of attributes
canonical.correlation.best.linear.combination	Canonical correlation of the best linear combination of attributes to distinguish between classes
relative.prop.best.linear.combination	Proportion of the total discrimination power explained by the best linear combination

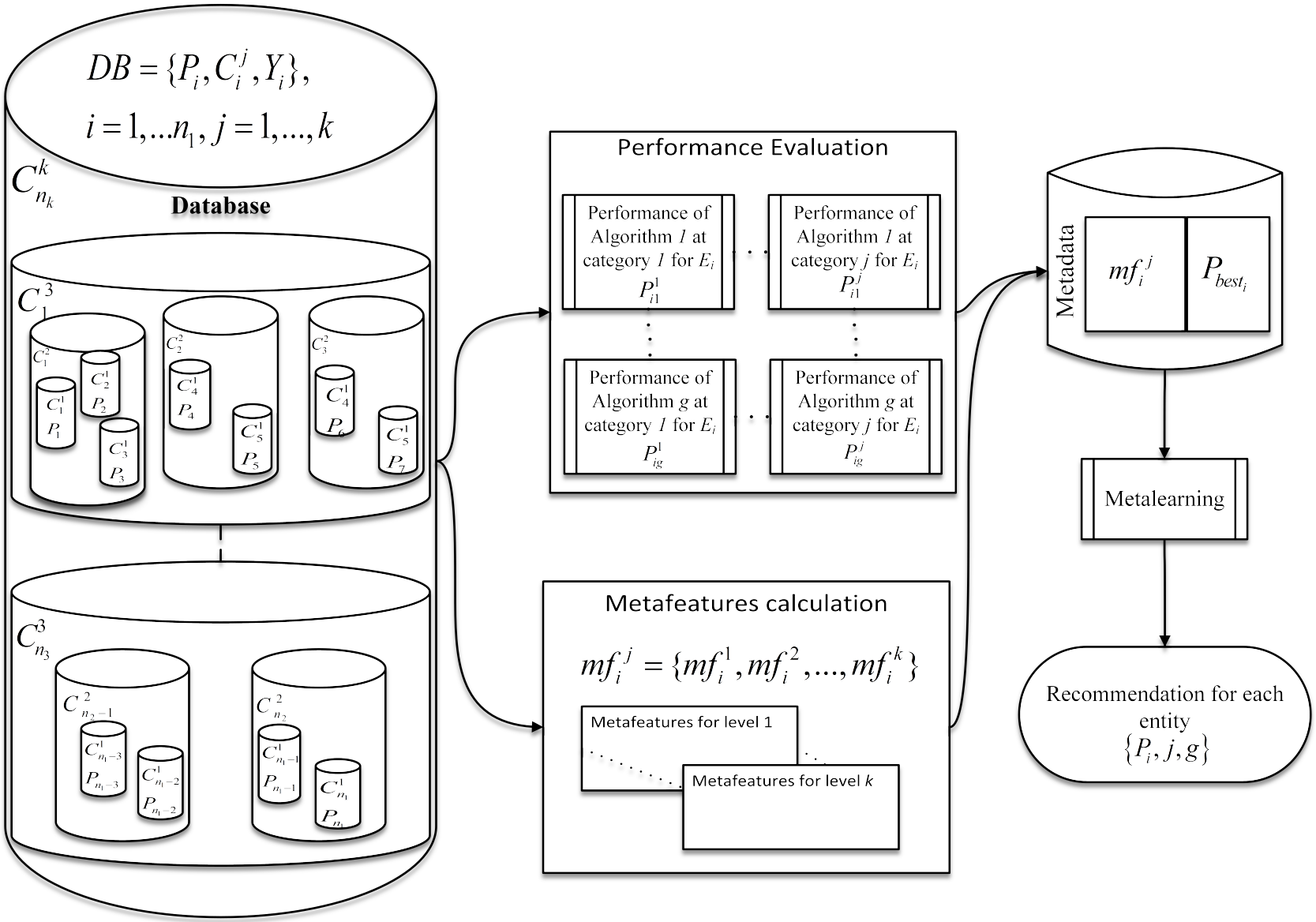


Figure 8.2: Methodology used for metalearning

8.3 Experiments Setup

In this section, the experimental setup at both base-level (Section 8.4) and meta-levels (Section 8.5) are explained.

8.4 Base-level

In chapter 6 we introduce our base-level experimental setup. There are ten months of INTRASTAT data. The LOF algorithm Torgo (2010); Nozari Zarmehri and Soares (2014) is trained on the data obtained from each level of granularity (see Section 8.2). In this case study, the number of algorithms at the base-level equals to one($g = 1$). Therefore, the output of the framework is the best level of granularity for each entity. As a result, for each product code, there are four performance indicators (PI_i^j) obtained from all possible levels (four levels, in this case, $j = 1, \ldots, k = 4$).

In the results section (Section 8.6), the levels are indicated by: level 1 ($OD8$), level 2 ($OD6$), level 3 ($OD4$), and level 4($OD2$). While "OD" stands for the outlier detection method and the suffixes (8, 6, 4, and 2) show the number of common digits among the product codes used for aggregation.

The evaluation part is done by testing each model obtained from above scenario against the data from the same level and the same product code in the next month. To be clearer, assume that for product code i, a model is trained using the data from month m at level two. Then this model is evaluated by the data taken from the level two for product i but at the month $m+1$.

Thereafter, by knowing the performance of outlier detection algorithm at all levels, the best level for each entity is selected as its actual value at the base-level. Then the majority level [1] is selected as the prediction level for all entities at the base-level. Based on the actual and prediction values, the accuracy of the base-level is calculated.

Two metrics are also used for the base-level evaluation: *recall* and effort. The explanations of the metrics are presented in Chapter 6, Section 6.2.3.1.

8.5 Meta-level

Two machine learning algorithms were used in our experiments at the meta-level: Random Forest (RF) Amit and Geman (1997); Breiman; Liaw and Wiener (2002) and Decision Tree (DT) Olshen et al. (1984); Safavian and Landgrebe (1991); Ripley (2014). A DT is a tree-like structure, splitting a dataset into branch-like segments, by evaluating a condition on a feature in each node($F1$, ..., $F4$ in Figure 8.3). The origin of the DT is a root node at the top of the tree (Figure 8.3).

[1] For example, if there are 10 entities in the dataset and the best level for these entities are levels 2, 3, 2, 1, 2, 4, 2, 1, 2, and 2 at the base-level, then the majority level is the second level (2).

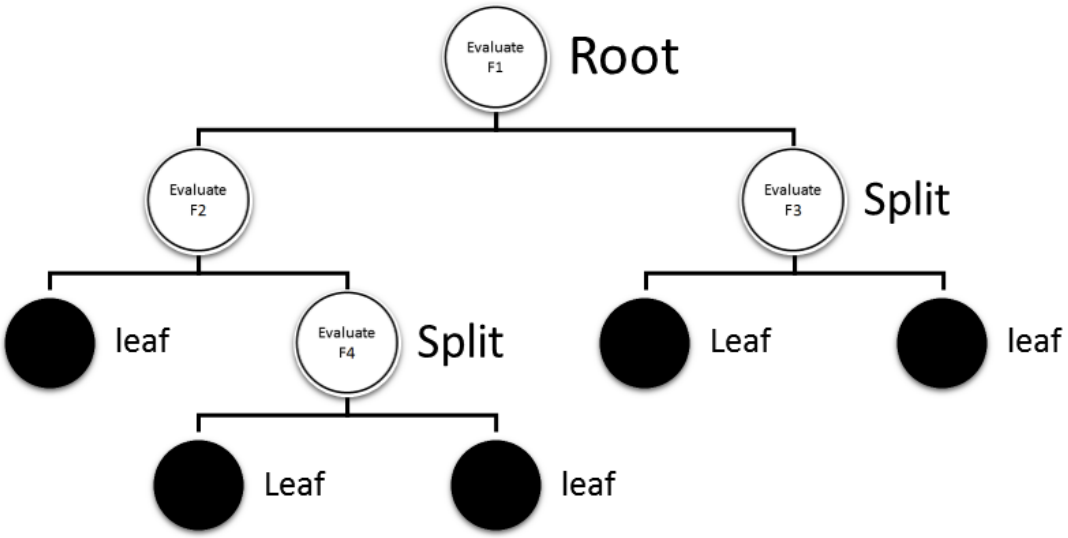


Figure 8.3: Decision Tree structure: at each split one feature ($F1$ to $F4$) is evaluated

The decision was taken after computing all features downward. The classification rules are the path from the root to the leaf. The choice of DT is related to the good results obtained on this problem previously by this algorithm Friedl and Brodley (1997).

Random forest is an ensemble learning approach for classification, regression and other tasks Breiman. RFs are the multitude of decision trees (forest). The output is the mode von Hippel (2005) of the classes (classification) or mean von Hippel (2005) prediction (regression) of the individual trees.

RFs are trying to overcome the problem of overfitting in DTs Hastie et al. (2009). In particular, trees overfit their training sets due to having low bias and high variance. Random forests are averaging multiple decision trees which are trained on different parts of the same training set with the goal of reducing the variance of DTs. Generally, this leads to great improvement in the performance of the final model Hastie et al. (2009).

These algorithms are applied to the metadata (Equation 8.2) to train a model and predict the best level of granularity for each product code which is called *predicted level*.

Following that, for evaluation, the accuracy of the meta-level is obtained by comparing the *predicted level* at the meta-level with the best level at the base-level. The accuracy is calculated using the Equation 8.3. The definition of all the elements of this equation is described in Table 6.4 and in the Section 6.2.3.1. This prediction accuracy is our primary evaluation measure for meta-level.

$$Accuracy = \frac{tp + tn}{tp + fp + fn + tn} \tag{8.3}$$

8.6 Results

As mentioned previously, the main performance evaluation metric is the accuracy of meta-level versus base-level. This metric shows that how well the meta-level can predict the best level for

training at the base-level. In this section, the base-level results (Section 8.6.1) from previous Chapter 6 is presented briefly. Then the meta-level results (Section 8.6.2) is compared with the base-level results.

8.6.1 Base-level Results

The base-level results show that the best level for different product codes is varied. In this section the result of a selected example (product codes starting with *11*) is presented in Section 8.6.1.1. Then the average results for different months (namely February and June) are shown in Section 8.6.1.2. Finally, the optimum results is illustrated in Section 8.6.1.3.

8.6.1.1 Selected Example

The distribution of outlier scores for the month of February, concerning import transactions for product code *11681414* at different levels of hierarchy, is shown in Figure 6.4. The variation in the scores happens due to the accuracy of models in the different levels for different product codes.

This score is used to distinguish the errors in the dataset. The transactions with high outlier score (namely more than the $1.5 * (Q3 - Q1)$ J.S. Milton, J.J. Corbet (1997) (see Section 6.2.2) are selected as outliers (they contain an error). These results show that for this particular product code, at the higher levels where there are more transactions, the number of detected errors (with high outlier scores) is higher than the lower levels. Using these results, the obtained results for evaluation metrics (Section 6.2.3.1) are presented in the next section.

8.6.1.2 Monthly Results

In this section, we present the previous results for two months, analyzed according to the different levels of the hierarchy in the product codes used for data aggregation.

Looking at Figures 6.5 to 6.8, it is observed that the base-level can predict around 70% and less than 70% of the errors just by selecting 10% of the transactions. Despite a very significant reduction in the *effort*, the *recall* is not good enough to be accepted by the experts.

On the other hand, for level three (4-digits), the model can predict more than 90% of the errors on February by analyzing just 12% of the transactions which is acceptable by the experts. So in this month, the best results are obtained by grouping the products at the four-digits level of the product codes (level three).

In June (Figures 6.7 and 6.8), the results show that grouping by two (fourth level) and four (third level) digits of the product codes leads to just 10% of the *effort*. Although the x-axis shows the number of observations, the levels are also shown from level 1, the most left points, to the level 4, the most right points, in this figures. However, this is not an acceptable result due to very low *recall* (less than 90%). The best results for this month are obtained at the second level (six digits) of product codes.

The percentage of the products codes for which the best *effort* (the lowest *effort*) and the best *recall* (the highest *recall*) are obtained at each level of the product taxonomy for every month is illustrated in Figures 6.9 and 6.10, respectively.

In summary, analyzing the data at the third (four digits) and fourth levels (two digits) requires lower *effort*. But to obtain acceptable results, namely a *recall* higher than 90%, an analysis at the six digits level is the best. On that account, in Section 8.6.1.3, we investigate the best way to choose the best level for training a model to meet the expert advice.

8.6.1.3 Optimal level selection

There are three different proposals for the optimum selection of the best level for modeling to obtain the best performance (see Section 6.3.3). We introduce them briefly as follow:

1. Using the best performance for either *recall* or *effort*. In this case, results show that it causes a drop on the other metric (Figures 6.11 and 6.12).

2. Using a fixed level for training a model which leads to the best performance for most of the product codes at the base-level. The result for this investigation also show that it has lower performance comparing to the previous selection (Figures 6.13 and 6.14).

3. Using the best performance for either *recall* or *effort* after considering the expert advice (*recall* more than %90 and *effort* less than %50). The best *effort*/*recall* is chosen, in which, the related *recall*/*effort* is acceptable by an expert. This strategy is selected as the most advantageous way to select the best level for training a model (Figures 6.15 and 6.16).

In overall, the best level is varied between different levels. Therefore, there is a need for a strategy that can help to find the best level of hierarchy amongst the existing levels. We use the metalearning for the problem of level selection in this case study. These results are discussed in the next section (Section 8.6.2).

8.6.2 Base-level vs. Meta-level

In this section, we evaluate the meta-level accuracy based on the best performance obtained at the base-level. Figure 8.4 shows the boxplot of the *effort*s obtained at the base-level and the meta-level. It is clear that the meta-level has a lower *effort* (better performance) in comparison with the base-level. On the other hand, the performance of both base-level and meta-level satisfies the expert advice (*effort* less than 50%).

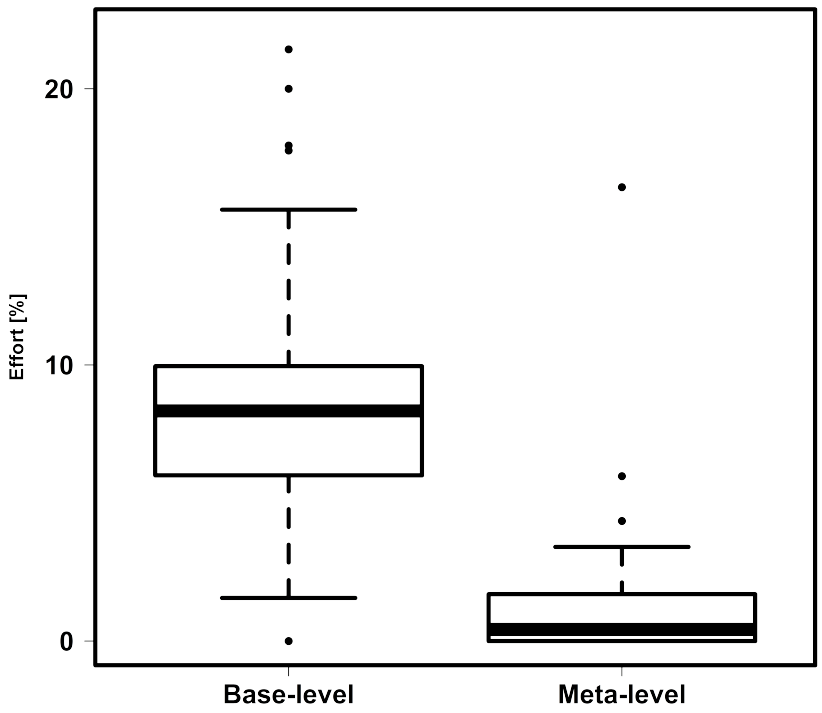


Figure 8.4: The effort obtained from base-level vs. meta-level

As previously mentioned, the main performance evaluation metric at the meta-level is its accuracy in comparison with the base-level. This metric shows that how well the meta-level can predict the best level obtained at the base-level. The results are plotted in Figure 8.5. The accuracy of applying the RF algorithm on the metadata is labeled as *ML-RF*. The accuracy of DT is also plotted as *ML-Tree*. The average accuracy is calculated in each month for all the product codes.

The accuracy of base-level shows that for around 55% of product codes, a specific level of granularity is the best level while in other cases, another level is the best level. Therefore, the base-level itself is not accurate enough because the best level of granularity is different from one product code to another and therefore a fixed level can not be always the best level of granularity. Obviously, the accuracy of the RF model has outperformed the accuracy of the base-level. The performance of the RF algorithm at the meta-level is almost two times of the base-level. On the other hand, the performance of the DT is not satisfactory. It only beats the base-level for September 1998. Figure 8.5 is shown that the RF at the meta-level is more suitable for modeling.

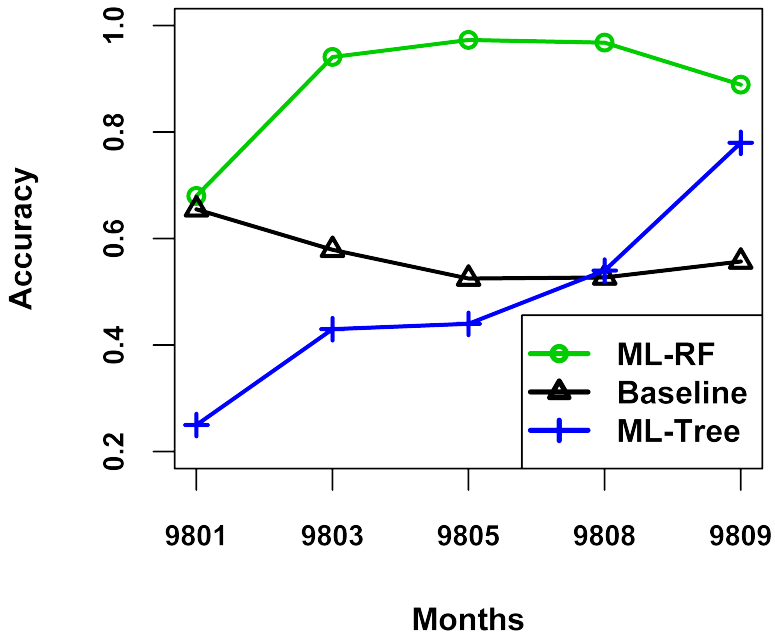


Figure 8.5: Comparing the accuracy of metalearning approaches with the baseline

The comparison of the Recall obtained by the meta-level and the base-level is also illustrated in Figure 8.6. While the base-level sustains significantly more than 50% in *recall* degradation for the higher levels, the RF incurs around 60% or more decrease in *recall* at the lowest level (*OD*8). In addition, the DT algorithm performs variably at different levels.

Nevertheless, for the lower levels, where the number of observations is very low, the base-level obtains better *recall* than the meta-level approaches while for higher levels the RF at the meta-level is outperformed other approaches.

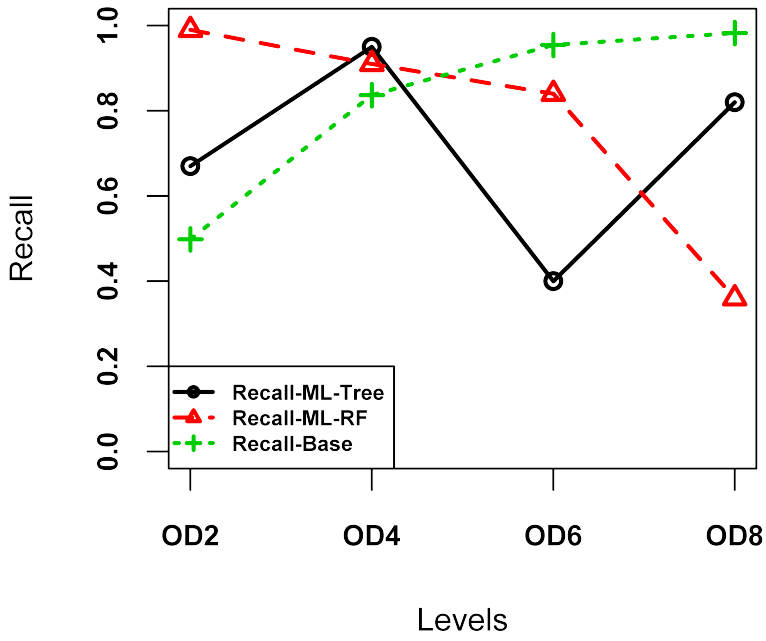


Figure 8.6: Comparing metalearning with different algorithms with baseline: Recall

Figure 8.7 compares the average *effort* obtained from meta-level with the base-level. At the higher levels where there are more observations for modeling, the base-level has a lower *effort* than the meta-levels approaches. But for the lower levels where the number of observations is low, both meta-level algorithms outperform the base-level approach. This is due to the inaccurate models built on top of a few training examples at the base-level.

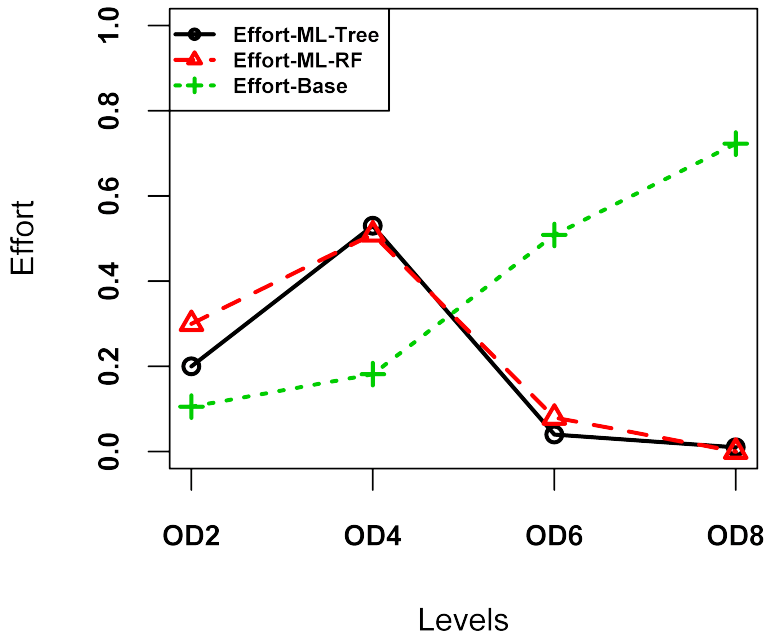


Figure 8.7: Comparing metalearning with different algorithms with the baseline: Effort

From our experiments, our metalearning model compares favorably with the base-level results for the lower levels of granularity. However, the base-level for the higher levels performs better because there is more data to build a more accurate model. It should be noted that the best performance of the base-level is chosen in the comparison which needs a lot of computational costs to be obtained. But then in the lower levels, the metalearning approaches show better results due to the lack of data at these levels at the base-level.

8.7 Summary

In this chapter, we present the first test of the framework that was proposed in the previous chapter. The framework addresses the problem of selecting the best level of data granularity to learn local models.

Although it was developed with models for VANETs in mind, the framework may also be useful in a more traditional data mining (DM) setting, particularly if the data is organized according to a data warehouse-like (DW) structure. The first test, described here, is one such problem, error detection in foreign trade transactions, which was discussed in Chapter 6.

The proposed framework was applied on transactions from several months, described using a set of metafeatures based on 15 basic measures that were used in previous metalearning research.

Extensive experimental results have illustrated the improvement of accuracy of the metalearning approaches when compared to the baseline. In particular, the results suggest that RF algorithm obtains very good results at the meta-level.

In the next chapter, the proposed framework is tested for a VANETs related problem. Knowing the travel duration beforehand can help drivers, passengers, and even the travel companies to better manage their trips, routes, and time. Unlike the problem already discussed in this chapter, the trip duration is a regression problem. Therefore, the framework is also tested for different problems including classification (this chapter) and regression (the next chapter).

Chapter 9

Using Metalearning for Prediction of Taxi Trip Duration Using Different Granularity Levels

In the previous chapter, we tested the metalearning-based framework (Chapter 7) to address the problem of selecting the best level of data granularity to learn local models (Chapter 8) on the problem of detecting errors in foreign trade transactions (Chapter 6).

In this chapter, the framework is tested on another problem which is more related to this project. A taxi application introduces an interesting challenge for metalearning. Each taxi generates enough data to learn its own model. However, it can be expected that, in some cases, the quality of the global model generated from the full set of data, i.e. concerning all taxis, can be better than the model generated solely with "local" data. Therefore, besides selecting an algorithm to learn the best model for a taxi, a decision can be made also concerning whether only local data or global data.

In this chapter, we close the circle on the data management goals of this project, by applying the same methodology to the problem of taxi trip time duration. More specifically, we investigate the use of a metalearning approach to the problem of algorithm selection in a case study of predicting trip duration for a taxi company. The taxi dataset is obtained from the Carnegie Mellon Portugal project, DRIVE-IN (Distributed Routing and Infotainment through Vehicular Inter-Networking) Cmuportugal.org (2014).

The experiment is done on the data from five months in 2013, from February to June. In each month, the data is collected by around 440 taxis.

The approach is evaluated at the meta-level (i.e. the ability to choose the most accurate base-level algorithm) and at the base-level (i.e. the base-level performance of the algorithm selected by the metalearning approach). The results obtained are positive at both levels.

The chapter is organized as follows. Section 9.1 summarizes the case study description. Section 9.2 describes our methodology for data analysis and metalearning to find the best level of

granularity. The evaluation strategies are discussed in Section 9.3. The obtained results are presented in Section 9.4. Finally, Section 9.5 concludes the chapter.

9.1 Motivation

With the fast growth of Intelligent Transportation Systems (ITS) and Advanced Travelers Information Systems (ATIS), data collected by those systems can be useful to understand and improve processes in taxi companies and other organizations dealing with transportation, i.e. public transportation companies, logistics companies, and local government Rodrigues et al. (2011a).

An example of a problem that can benefit from the analysis of data is trip duration in taxi companies Zhang and Rice (2003b); Rashed and Jürgens (2010); Lee and Gerla (2010a). Especially knowing the estimated trip time duration beforehand can be very informative for taxi companies, drivers, and passengers to make the right decision for the scheduling and route planning. Data concerning the taxi trips (essentially GPS data) collected by taxis can be used for that purpose.

Data mining approaches can be used for the prediction of the trip duration. Using the data collected by taxis, these approaches relate trip duration with several variables describing the trip like origin, destination, time of day, day of week, and the weather.

The prediction of trip duration may vary for different taxis, due to differences in the brand of the vehicle, its usage, and driving habits. Therefore, algorithm selection should be made not at the global level but at a lower one, such as the taxi itself.

On the other hand, in applications with multiple sources of data in which the data schema is the same, it is possible that the quality of the model for a given source can be improved by training it with data from other sources. Therefore, the problem of algorithm selection is also extended to the dataset granularity selection. For the purpose of trip duration prediction, each taxi can use its data, data from its neighbors, data collected at the nearest road-side unit, or the whole dataset which is collected centrally throughout the city.

9.2 Methodology

In this section, the methodology used in this chapter is described. The dataset is introduced briefly in Section 9.2.1. Section 9.2.2 explains the methodology which is applied on the base-level. The approach for meta-level is discussed in Section 9.2.3. The metafeatures used in the metalearning are also presented in Section 9.2.4.

9.2.1 Dataset

The dataset is obtained from a large-scale scenario Cmuportugal.org (2014), one of the taxi companies in the city of Porto. Porto is the second largest city in Portugal, with an area of 41.3 km^2, and comprises 965 km of roads. It is the central city in a metropolitan area with more than one million inhabitants. There are 63 taxi stands in the city and the main taxi union has 441 vehicles.

Each taxi has an on-board unit with a GPS receiver and collects the travel log. The dataset provided by the project Cmuportugal.org (2014) consists of five months in 2013 for all the vehicles. The list of all features with their descriptions is already presented in Table 7.2.

As an example of the scenario where the data is collected, Figure 9.1 shows a snapshot of the taxis's placements in the city of Porto. The green dots show the taxi positions in the city. The communication range for two taxis (red and black) is also shown by purple circles.

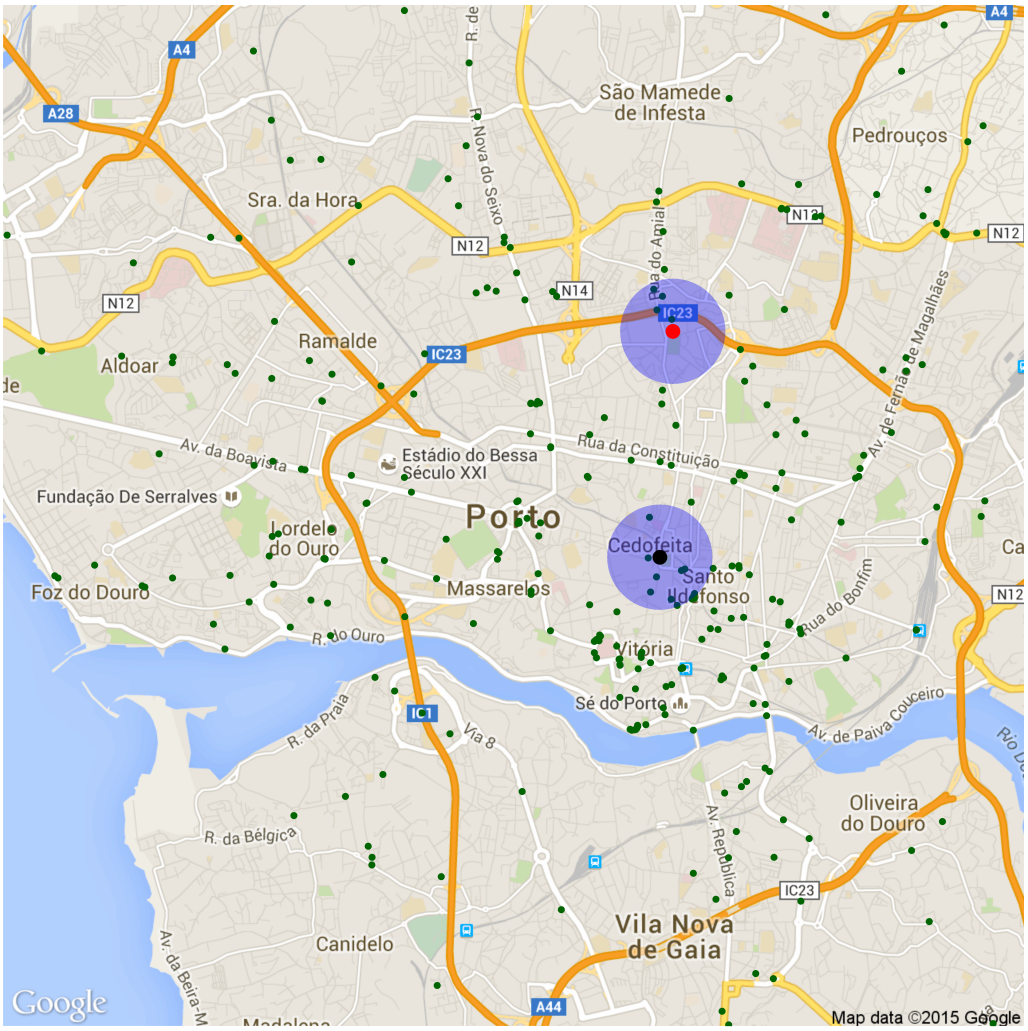


Figure 9.1: Illustrative map of Porto, Portugal. The green dots are the taxi placement. The neighboring area for red and black taxi is shown by purple circle around them.

9.2.2 Base-level Approach

In this section, the traditional data mining method which is described in Section 7.1.1 is customized for the base-level experiment and this dataset. At the base-level, the same scheme is used. Each taxi is represented by an entity in the scheme, $E_i = T_i$. The target variable is the trip duration ($Y_i = DT_i$). So the base-level scheme is like $DB = \{T_i, X_i, DT_i\}, \forall i \in \{1, ..., n\}$, where C_i is a set

of the features which are used at the base-level (Equation 7.2). And the objective is to predict the trip duration (dt). All the observations for each taxi is considered as the local data.

There are 5 months data in 2013: February to June. Two levels of granularity are selected: taxi itself and the data for whole month, C_i^1, C_i^2 (see Equation 7.2). Four algorithms are applied on different levels of the dataset (DB) at the base-level to predict the target variable (Trip Duration):

- Decision Tree (DT) Olshen et al. (1984); Safavian and Landgrebe (1991); Ripley (2014),
- Random Forests (RF) Amit and Geman (1997); Breiman; Liaw and Wiener (2002),
- Support Vector Machines (SVM) Hearst et al. (1998); Schölkopf and Smola (1998); Steinwart and Christmann (2008),
- Linear Regression (LM) Seber and Lee (2012); Montgomery et al. (2012).

The results of this experiment is used for comparison at the base-level as well and to create the meta-dataset at the meta-level.

9.2.3 Meta-level Approach

In terms of the metalearning approach, the possibility of generating meta-examples at different levels of granularity of the data, adds another dimension to the meta-dataset. So for each entity, instead of having just one set of C_i, other feature sets can be generated for different levels or categories of the data, $C_i^1, C_i^2, C_i^3, ..., C_i^k$, where k is the number of levels or categories. Therefore the meta-dataset for the metalearning process is $DB = \{T_i, C_i^j, Y_i\}, \forall i \in \{1, ..., n\}, \forall j \in \{1, ..., k\}$. The customized model for this case study used in this chapter is shown in Figure 9.2.

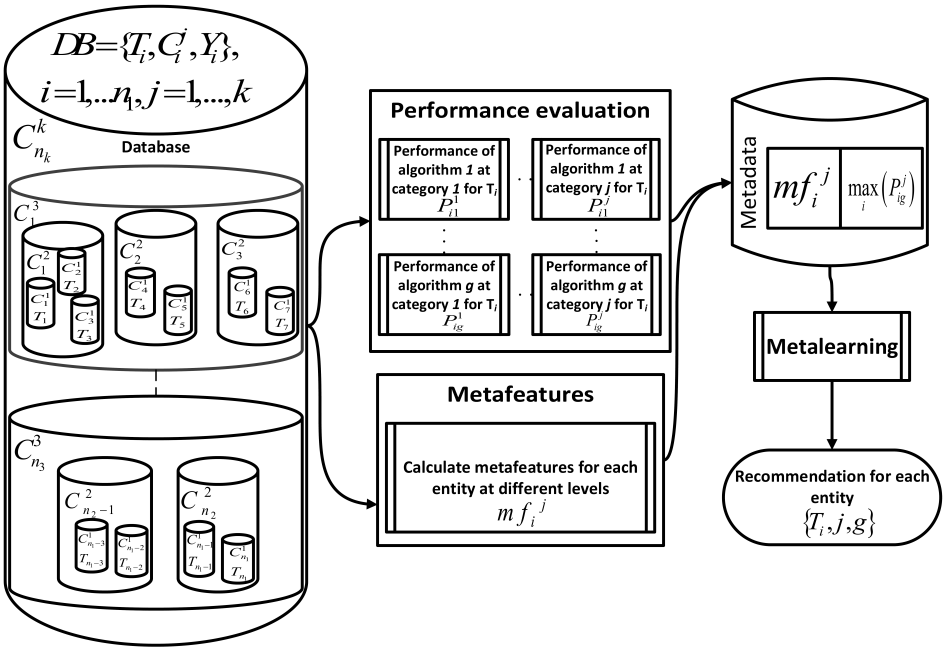


Figure 9.2: Proposed methodology used for metalearning

In the proposed model, there are two different levels: the taxi itself (local model) and the data for the whole month (global model). At the level one, each taxi (T_i) creates a unique category, $C_i^1, \forall i \in \{1,...,n_1\}$ where n_1 is the number of taxis. The level two data has only one category containing all the data from all taxis. At the meta-level, to create the meta-dataset, we need the best performance obtained at the base-level for each taxi. Therefore, the best base-level performance is selected as $P_{best\,i}$ for taxi i (see Equation 9.1).

$$P_{best\,i} = \max_{w,j}(P_{iw}^{j}) \ , \ \forall w \in \{1,\ldots,4\}, \forall j \in \{1,2\} \tag{9.1}$$

Where w is the number of algorithms and j is the number of levels at base-level.

In addition, the metafeatures (see Section 9.2.4) are also calculated for each taxi and at different levels. In general mf_i^j is the calculated metafeatures for taxi i at the level j. As a result, there are two sets of metafeatures for each taxi: mf_i^1 and mf_i^2.

Finally the metadata (see Section 9.2.5) structure for each taxi consists of the taxi identification, metafeatures for the two levels and the best performance at the base-level obtained from Equation 9.1. A general form of metadata is shown in Equation 9.2.

$$T_i, mf_i^1, mf_i^2, P_{best\,i} \tag{9.2}$$

In the experiment setup, two algorithms are applied on the metadata at the meta-level to predict the best performance ($\hat{P}_{best_i}$):

- Decision Tree (DT),
- Random Forests (RF).

The target variable at the meta-level is $P_{best\,i}$. The predicted best performance at the meta-level is called $\hat{P}_{best_i}$. Using $\hat{P}_{best_i}$, the meta-model recommends the best level and the best algorithm for each taxi to have the best performance at the base-level.

The performance of metalearning model is evaluated using the predicted best performance ($\hat{P}_{best_i}$) and the actual value obtained from Equation 9.1 ($P_{best\,i}$). The results of this evaluation are presented in Section 9.4.

9.2.4 Metafeatures

The extracted metafeatures noted above, are described briefly in this section. A comprehensive study was done by Peng et al. Peng et al. (2002) for feature selection. Also the reasons to select these metafeatures are also discussed in Section 8.2.1. For this problem, in total, 31 metafeatures were selected to describe the structure of the dataset. These metafeatures are selected based on the regression problem. Their effectiveness through extensive experiments was evaluated.

A list of all metafeatures that we used for this study with a brief description is provided in Table 9.1. The detailed description of each metafeature is explained in Peng et al. (2002) and is also described in Appendix A.

Table 9.1: The description of metafeatures used for metalearning

Feature	Feature's description
1	Number of examples
2	$log(10)$ of the number of examples
3	Number of attributes
4	Ratio of number of examples by number of attributes
5	$log(10)$ of the ratio of number of examples by number of attributes
6	Number of continuous attributes
7	Number of symbolic attributes
8	Number of binary attributes
9	Proportion of continuous attributes
10	Proportion of symbolic attributes
11	Proportion of binary attributes
12	Correlation between continuous attributes
13	Average absolute correlation between continuous attributes
14	Minimum absolute correlation between continuous attributes
15	Maximum absolute correlation between continuous attributes
16	The ratio between the standard deviation and the standard deviation of alpha trimmed mean
17	Number of continuous attributes with outliers
18	Proportion of continuous attributes with outliers
19	Correlation matrix between attributes and target
20	Average correlation continuous attribute/target
21	Minimum correlation continuous attribute/target
22	Maximum correlation continuous attribute/target
23	Check if standard deviation is larger than mean
24	Ratio of the standard deviation and the mean of the target attribute
25	Sparsity based on the coefficient of variation
26	Sparsity based on the absolute coefficient of variation
27	Standard deviation of the proportions of a histogram with 100 bins of target values
28	textith.outlier value, as calculated for the continuous attributes
29	Outlier detection based on the notion of outliers used for continuous attributes
30	Mean distance between each target value and its two neighbors (sorted by value)
31	Average mean distance between each target value and its two neighbors (sorted by value)

9.2.5 Metadata

The dataset used for metalearning is called metadata. For each entity, the best performance obtained from the performance evaluation part is selected according to the Equation 9.1.

The metadata for each entity is consisted of metafeatures for different levels plus the best performance obtained from the Equation 9.1. Equation 9.3 shows the general form of the metadata which is used for metalearning:

$$Row\ i \rightarrow T_i, mf_i^1, mf_i^2, \ldots, mf_i^k, P_{best\,i} \tag{9.3}$$

As an example, suppose we have only one metafeature which is the number of observations. Then the metadata is: the entity, the number of observations at level one (mf_i^1), the number of observations at level 2 (mf_i^2), and the best performance obtained amongst two levels ($P_{best\,i}$). Equation 9.4 shows this example:

$$T_i, mf_i^1, mf_i^2, P_{best\,i} \tag{9.4}$$

The main idea of metalearning is to find out the best algorithm and the best level of hierarchy to apply the algorithm depending on the metafeatures obtained at different levels. Consequently, the metalearning maps the extracted metafeatures from the original datasets to the best performance obtained at different levels by applying different algorithms to the original dataset. Our metalearning model recommends a level and an algorithm for each taxi in which, applying the recommended algorithm on the recommended level produces the best performance (see Equation 9.5).

$$Output : \{\underbrace{T_i}_{entity}, \underbrace{j}_{recommended\,level}, \underbrace{g}_{recommended\,algorithm}\} \tag{9.5}$$

9.3 Evaluation

In this section, the evaluation process is discussed for both base-level 9.3.1 and meta-level 9.3.2.

9.3.1 Base-level evaluation

At the base-level, the problem of prediction of the trip duration is a regression problem. Each algorithm is applied on the training dataset and the model obtained is applied to new examples to predict the corresponding trip duration. This prediction is evaluated by the Normalized Root-Mean-Square-Error (NRMSE). This measure is based on the popular RMSE (Equation 9.6), which is based on the differences between the predicted and the observed values.

$$RMSE = \sqrt{\frac{\sum(\hat{Dt}_i - Dt_i)^2}{n_1}} \tag{9.6}$$

where Dt_i is the observed trip duration, $\hat{Dt}_i$ is the predicted trip duration and n_1 is the number of predicted values.

The NRMSE is the RMSE divided by the standard deviation of the variable being predicted (Equation 9.7). Using R R Core Team (2014b), the package hydroGOF Zambrano-Bigiarini (2014) is used to compute the NRMSE.

$$NRMSE = 100 * \frac{RMSE}{\sigma} \tag{9.7}$$

where σ is the standard deviation of the predicted variable. Having the NRMSE for all the possible runs, the algorithm with the best NRMSE (the lowest one) is selected as the best algorithm for each taxi to be used at the meta-level.

9.3.2 Meta-level evaluation

As previously mentioned, at the meta-level, two algorithms were applied on the meta-data (Equation 9.2): random forest and decision tree. The meta-level approach predicts a base-level algorithms along the level of granularity which will have the best performance (lowest NRMSE) for a given taxi and month.

The performance of the proposed metalearning model is evaluated by the accuracy of the prediction. This is done by comparing the predicted best performance ($\hat{P}_{best_i}$) and the actual value at the base-level ($P_{best\,i}$). In addition, we also evaluate the performance of the proposed model relative to the possible range of base-level performance. $Scaled_{error}$ shows the relative NRMSE of the metalearning model with respect to the best and the worst NRMSE of the base-level. It is shown in the following equation:

$$Scaled_{error} = \frac{NRMSE_{ML} - NRMSE_B}{NRMSE_W - NRMSE_B} \tag{9.8}$$

where $NRMSE_{ML}$ is the NRMSE of the proposed metalearning model, $NRMSE_B$ and $NRMSE_W$ are the best and the worst NRMSE obtained by the base-level algorithms, respectively. The metalearning approach can behave as well as the best performance at the base-level ($NRMSE_B$) or as worse as the worst performance at the base-level ($NRMSE_W$). Therefore, the range of $Scaled_{error}$ is between 0 and 1. If the prediction of the metalearning model equals to the best performance of the base-level, then $Scaled_{error} = 0$. In the worst case when the performance of the meta-model equals to the worst performance of the base-level, then $Scaled_{error} = 1$. As a result, the lower the $Scaled_{error}$ the better performance is expected for the meta-level experiment.

9.4 Results

In this section, the use of metalearning for algorithm selection and level of granularity is investigated. The performance of different algorithms is evaluated for local and global data for each taxi at the base-level and the meta-level. Section 9.4.1 shows the base-level results while the performance of the meta-level is discussed in Section 9.4.2. The comparison of base-level and meta-level results is also illustrated in Section 9.4.3.

9.4.1 Base-level results

As previously mentioned (Section 9.3.1), the performance of the base-level is evaluated by NRMSE. Figure 9.3 shows the box-plot of the average NRMSE for different taxis in five months. It can be seen that the NRMSE for all months is less than 5%. The average NRMSE for each month is around 1%. Therefore, on average, the base-level error is 1% which sounds considerably good. This means that the base-level algorithms can predict the trip duration very precisely.

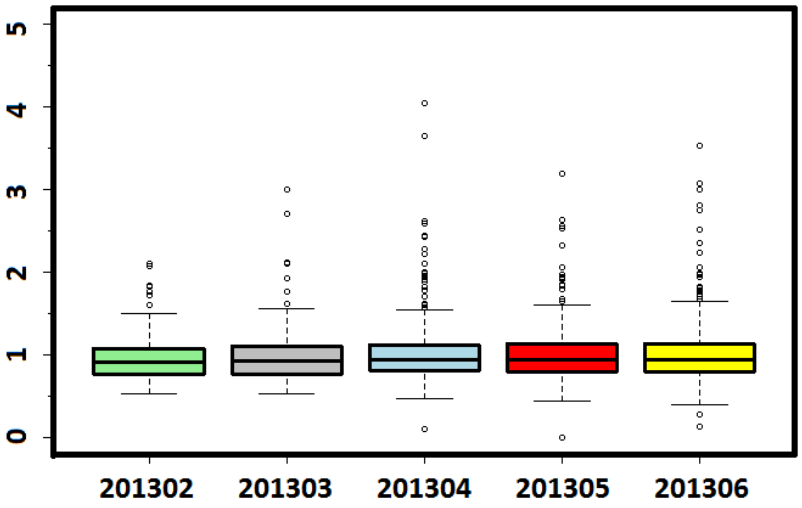
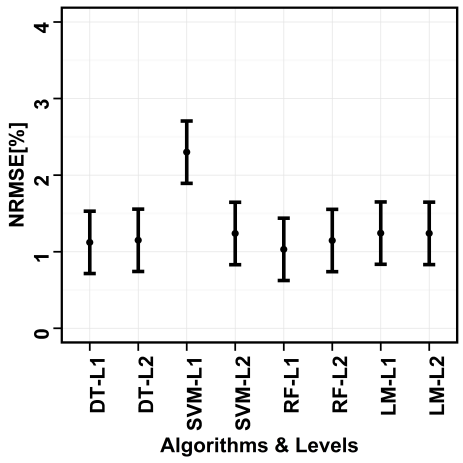
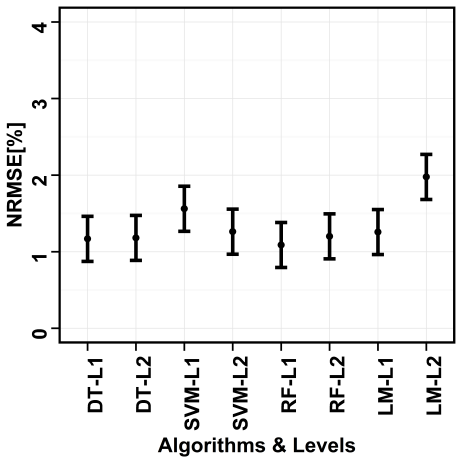


Figure 9.3: NRMSE[%] for different months

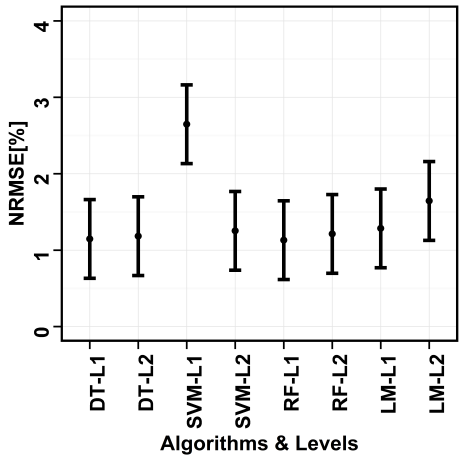
The performance of individual algorithms are also shown in Figure 9.4 which shows the NRMSE for each algorithm at each level for all months. It also shows that the NRMSE at the base-level is around 1% and the variation for month April, 2013 is more than other month. In general, the SVM algorithm at the first level (the local mode trained only by the data from taxi itself) has the worst performance while the random forest algorithm at the level one has the best performance on average. Although it is not true for all taxis and the best performance may vary between different algorithms and levels (Figure 9.5).



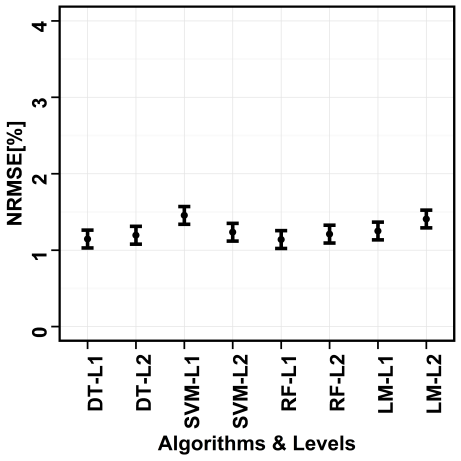
(a) 201302



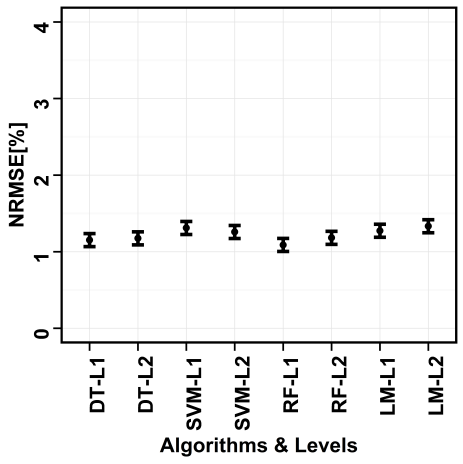
(b) 201303



(c) 201304



(d) 201305



(e) 201306

Figure 9.4: NRMSE for all algorithms at each level in different months

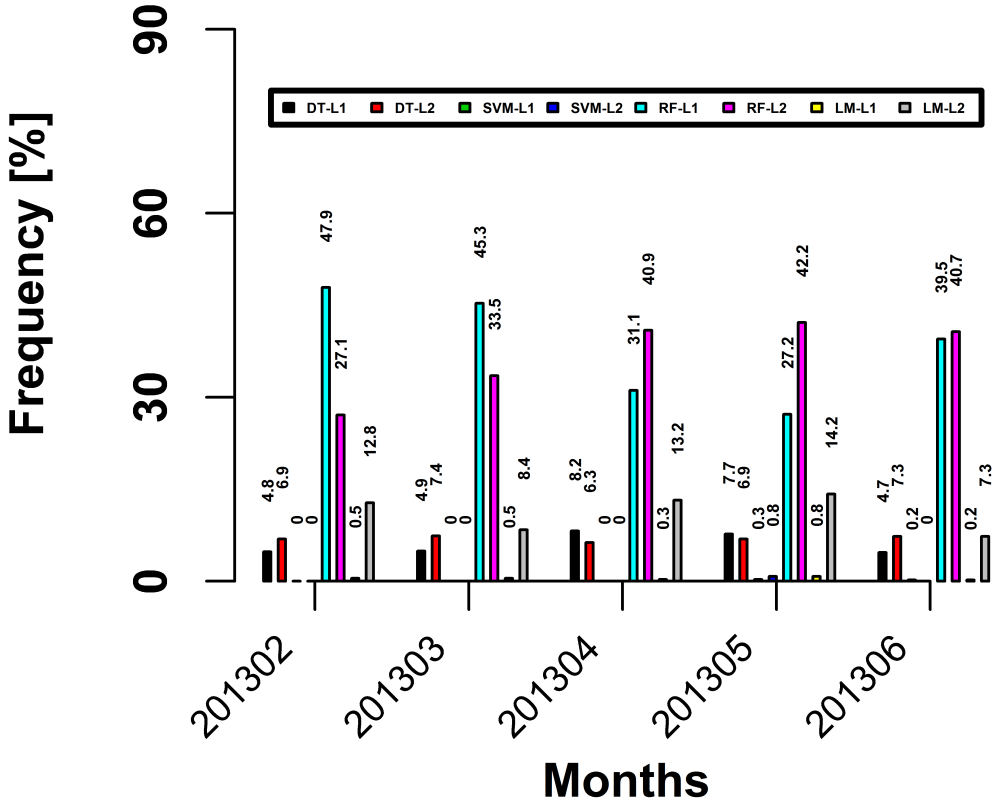


Figure 9.5: The percentage of the best algorithm in all months: base-level

9.4.2 Meta-level results

At the meta-level, the best algorithm and the best level of granularity have a less variation than the base-level.Table 9.2 shows the results of the best algorithm and the best level for each month at the meta-level. Clearly, the SVM algorithm is never selected as the best at the level one in our experiments. The linear regression algorithm at the first level is also selected for only 0.24% of taxis on June which can be illuminated from the algorithm space at the base-level. The SVM algorithm is also selected as the best only for less than 1% of taxis at the level two, on June. So we may remove it from the algorithm space in the base-level and decrease the computational cost. Obviously, the RF algorithm is the algorithm that is selected as the best algorithm in both levels in most of the time.

The meta-level results is also shown that the best level varies. For February, March, and April, the best level is the level one. This is probably due to terrible weather in Porto and as a result, there are more trip for each taxi in these months. So an accurate model can be trained by using only the data from taxi itself. But in months May and June, while the weather is better and people start using other public transportations like metro and buses, the number of trips for each taxi is decreased. Therefore, the best level of granularity for taxis is changed from level one to level two where there is enough data to train an accurate model (global model).

Table 9.2: The meta-level results (the percentage of occurrence) for the best algorithm and the best level of granularity on different months. Columns: meta-levels algorithms. Rows: base-level algorithms

		Year 2013									
		February		March		April		May		June	
		RF	DT	RF	DT	RF	DT	RF	DT	RF	DT
Level 1	DT	0	0	0.49	0	0	0	0.53	9.74	0.94	0
	SVM	0	0	0	0	0	0	0	0	0	0
	RF	88.3	69.68	71.92	99.01	78.93	100	36.94	0.26	29.41	21.65
	LM	0	0	0	0	0	0	0	0	0	0.24
Level 2	DT	0.53	1.6	0	0.99	0.31	0	0	31.24	0	0
	SVM	0	0	0	0	0	0	0	0	0	0.94
	RF	11.17	23.94	27.59	0	20.75	0	60.16	58.76	66.35	72.24
	LM	0	4.79	0	0	0	0	2.37	0	3.29	4.94

As discussed previously (Section 9.3.2), our metric to evaluate the meta level results is $Scaled_{error}$. Figure 9.6 shows this metric on average for all months. The $Scaled_{error}$ is about 30% in the worst scenario and around 5% for the best one.

The overall results of the calculated $Scaled_{error}$ seem interesting while the $Scaled_{error}$ is very low. It shows that the performance of the algorithms selected by the meta-model is close to the best performance obtained by the base-level by any of the algorithms. This satisfies our objectives.

The $Scaled_{error}$ is increased with time. This is because the difference between the worst and the best NRMSE for the months May, and June is decreasing and therefore the denominator in Equation 9.8 is decreased. The comparison of the worst, the best and the meta-level results is illustrated in Figure 9.7. In addition, this figure also confirms that the meta-level results are almost followed the best base-level results.

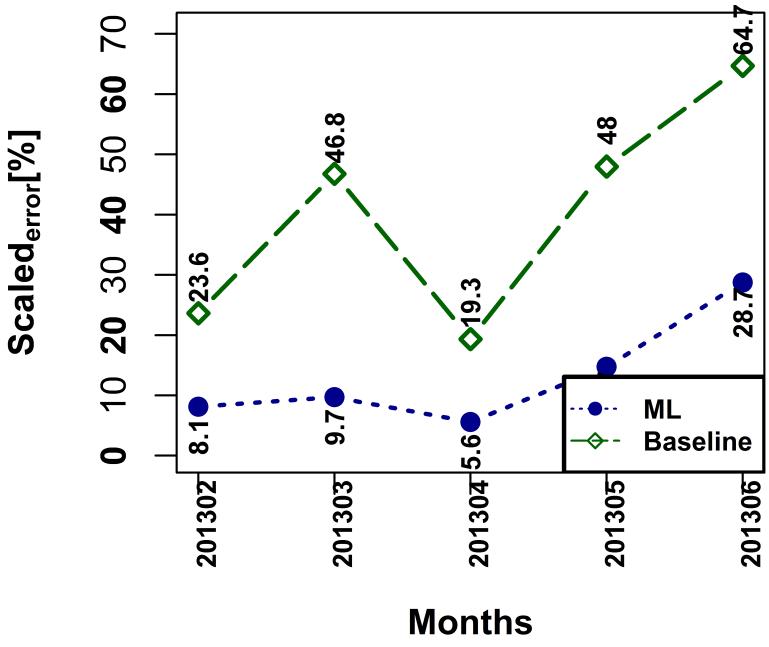


Figure 9.6: The average $Scaled_{error}[\%]$ over all taxis for each month

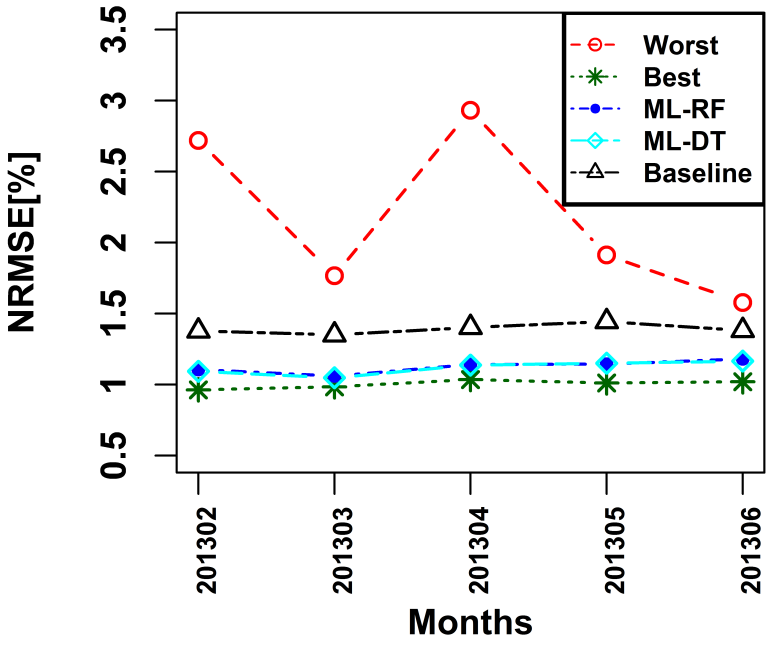
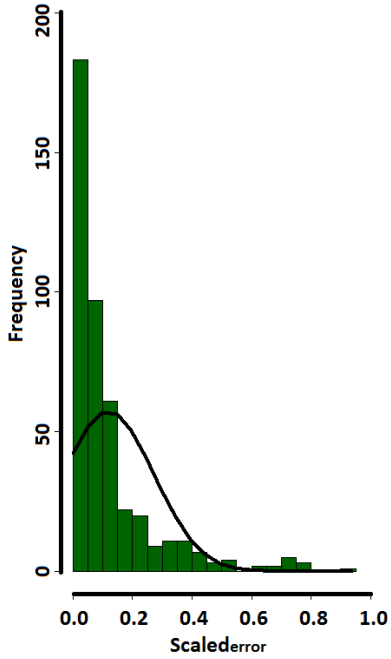


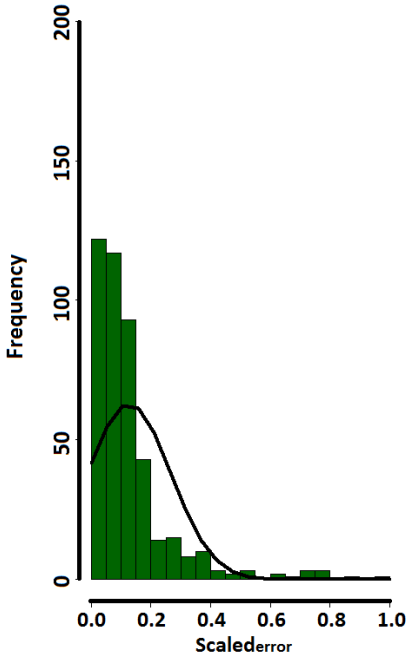
Figure 9.7: The NRMSE for the worst, the best and the meta-level results: ML-RF and ML-DT

The distribution of calculated $Scaled_{error}$ for each taxi is shown in Figure 9.8. As we expected, the density is concentrated around zero. The distribution for RF and DT algorithms show that on average the $Scaled_{error}$ is less than 0.2 in both cases. Although the density of $Scaled_{error}$ for RF algorithm has higher concentration near the origin.

In general, the results show that during the months with lots of rain (February to April) in Porto, when there are more taxi trips in the city, the local model for each taxi is more accurate than the global one. But on May and June while there is less taxi trips in the city of Porto, taxis should the global model for prediction.



(a) RF algorithm



(b) DT algorithm

Figure 9.8: Distribution of $Scaled_{error}$ over each taxi

9.4.3 Base-level vs. meta-level results

In metalearning one of the most important metric for evaluation is the accuracy. The comparison of accuracy between the base-level and the meta-level is presented in Fig. 9.9. According to this re-

sult, the performance of the meta-level outperforms the base-level for most of the months. In April 2013, due to the lack of enough observations for calculating the metafeatures, the performance of metalearning is dropped.

The accuracy of the base-level is calculated based on the majority algorithm and level with the best performance at the base-level. Imagine that on February, applying the random forest algorithm (Figure 9.5 shows that for 47.9% of the time, the random forest algorithm is the best choice) at the level one is selected as the prediction of the best choice for all taxis in this month. Then the accuracy of base-level for this month is calculated by knowing the actual best choice and this prediction. At the meta-level, the accuracy is calculated by comparing the prediction of meta-level with the actual best choice at the base-level. On average, the meta-level accuracy is 32% higher than the base-level accuracy.

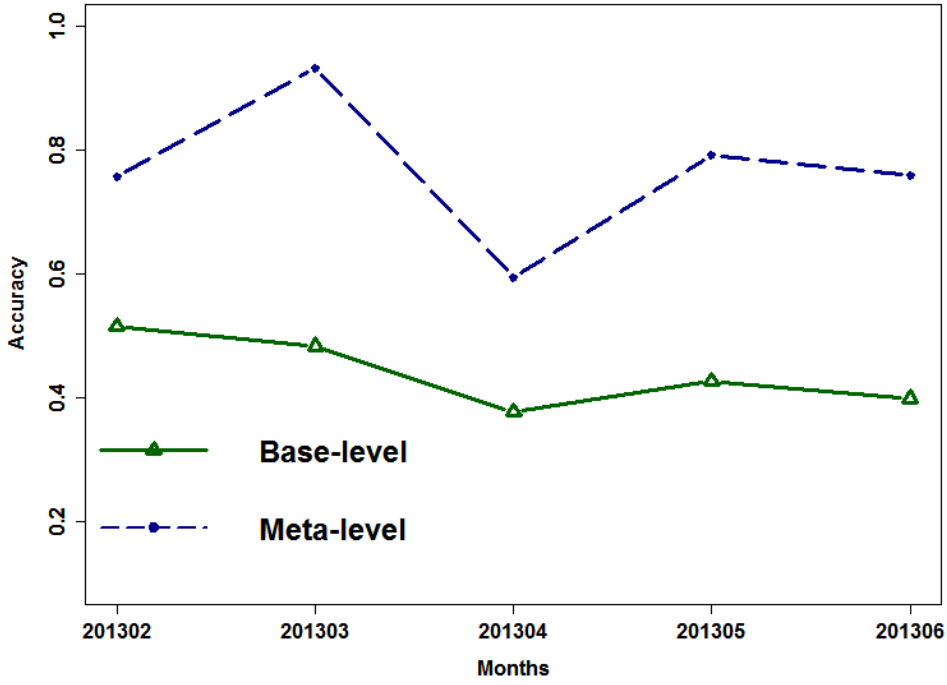


Figure 9.9: Accuracy: base-level vs. meta-level

9.5 Summary

We proposed the use of metalearning for prediction of trip duration. The experiments are performed on the taxi dataset from Drive-In project. The machine learning algorithms are performed at two different levels of granularity: taxi and month levels. The results show that the metalearning can help predicting the algorithm with the best performance at the base-level with high accuracy.

Furthermore the performance of the base-level itself is also considerably applicable. In overall, the results show that the metalearning predicts the trip duration with the error rate less than 5%.

Chapter 10

Conclusions and Future Work

10.1 Conclusion

Vehicular networks, which are part of the communication infrastructure for ITS, offer several applications like safety, traffic management, and infotainment applications. Vehicles move all over the city and sense events from the city environment. This data can be processed and sent to a central location where it can be aggregated and used for various applications from real-time to offline applications. Therefore, using in-car sensors as data sources and cars as data carriers, a macro vision of the city can be obtained. In addition, vehicular networks can be a cost-efficient infrastructure for urban sensing making an urban monitoring system without actually deploying connected sensors.

In this project, a solution is proposed for the broadcast storm problem in the massive urban sensing application. The proposed solution uses a back-off timer to distribute the forwarders according to their distance to the final destination which is a data collection point in the network. It also uses different suppression techniques for further suppressing the unnecessary packet forwarding. The protocol is compared to other existing protocols that may be used for the purpose of the data collection in VANETs. In most of the cases, it outperforms other protocols with high end-to-end packet delivery ratio and low end-to-end delay.

Another contribution of the project is that the sensing capacity is estimated. This sensing capacity indicates the maximum amount of data that a node can generate before saturating the shared medium. The results of this theoretical limit are also compared to the simulation results for validation.

In detail, the following contributions related to data communication are done:

- A new protocol for urban sensing and data collection is designed and evaluated (Chapter 3). The protocol is a broadcast- and receiver-based per-hop forwarding protocol. It selects the forwarding order among the nodes receiving the packet by mapping it into back off time, in which, nodes that are nearer to the final destination have shorter back-off times. The protocol does not require nodes to exchange periodic messages with their neighbors communicating their locations for reducing the message overhead and it uses geographic

information about the current sender and the final destination node in the header of each data packet to route the packet in a hop-by-hop basis by taking advantage of redundant forwarding to increase packet delivery to a destination. The performance of the protocol is compared with the broadcast storm mitigation techniques using NS-3. The proposed protocol achieves higher packet delivery rates and uses on average the same number of hops and causes less redundant packets at the data sink.

- The performance of the proposed protocol is compared with the performance of different broadcast suppression techniques (Chapter 4). The effect of four different suppression techniques on the sensing accuracy and network overhead is evaluated using NS-3 large scale simulation.

- The sensing capacity is calculated and validated with the simulation. In Chapter 5, the limitations of data gathering protocol in the sense of many-to-one scenario are studied and a method to estimate the maximum amount of data that each node can generate to guarantee a pre-defined packet delivery ratio at the sink is also proposed. The maximum packet size for a scenario without considering the collision is calculated. Then, the collision is added to the calculation and the end-to-end packet delivery ratio and the service rate are calculated. The results quantify the trade-off between these two metrics. Finally, the analytical results are validated by simulation of the same scenario.

The data management part of the project is concerned with the choosing of the right piece of data and the right algorithm to build an accurate model for a particular entity in a specific area of the city. Taking advantage of existing hierarchy in the collected data, a general data mining framework is proposed. The proposed framework is used for improving the performance of traditional data mining and machine learning algorithms for building predictive models. This approach uses metalearning to relate the performance of the different machine learning algorithms with the data characterization to reduce the computational costs as well as to improve the model performance. The performance of the framework is evaluated on two datasets: VANETs dataset and unrelated dataset to VANETs but with common properties.

The contributions related to data management are:

- Use of existing hierarchies in the data is investigated to improve the performance of the learning process. In Chapter 6, the effect of aggregating data at the different levels of a product taxonomy in the performance of an outlier detection method applied to the problem of identifying erroneous foreign trade transactions collected by the Portuguese Institute of Statistics (INE) is investigated. The evaluation results show that, depending on the product, the best results can be obtained at different levels of aggregation. However, in some cases, the best results are obtained with no grouping. This means that different aggregation levels should be selected for different local models.

- In Chapter 7, a metalearning framework is proposed for data hierarchy level and algorithm selection. The basic idea for this proposal is to reduce the computational costs for applying

different algorithms at different levels of granularity to reach the best performance. The proposed model recommends an algorithm and a level of granularity to obtain the best performance.

- The framework is evaluated by applying it to different datasets (Chapters 8 and 9). Although the proposed framework is developed for VANETs-related models in mind, the framework may also be useful in a more traditional data mining (DM) setting, particularly if the data is organized according to a data warehouse-like (DW) structure. In Chapter 8, the proposed framework is applied on the INTRASTAT dataset to detect erroneous transactions. Extensive experimental results have shown the improvement of accuracy of the metalearning approaches, when compared to the baseline. In Chapter 9, the proposed framework is tested for a VANETs related problem. Knowing the trip duration beforehand can help drivers, passengers, and even the travel companies to better manage their trips, routes, and time. At the base-level, the machine learning algorithms are performed at two different levels of granularity: taxi and month levels. The results show that the metalearning can help predicting the algorithm with the best performance at the base-level with high accuracy. In overall, the results show that the metalearning predicts the trip duration with the error rate less than 5%. Unlike the problem already discussed in Chapter 8, the trip duration is a regression problem.

10.2 Future Work

The proposed approach in this project covers the whole process from data collection to the data model management for different ITS applications. Considering the hyperconnected world, this proposal can be used by different applications including manufacturing of products by forecasting of product demands, supply chain management by predicting of the best location of stocks, and real-time optimization of supply chain networks by prediction of the best route for fleets, through networking machinery, sensors and control systems together.

All manufacturers have a desire to give their consumers exactly what they want. This can be done by using historical data collected from sales and production lines and forecasting the demand for products. Our proposed approach can be implemented in this case by organizing the past data into hierarchy structure. One of possible solution can be using three different hierarchy levels: the data associated with a product, the data related to a group of products which have the same type, i.e. foods, clothes, and so on, and all the data.

This proposal can be used in a supply chain management system where a poor stock's location can give low productivity, unreliable deliveries of materials, high costs, and poor customer service. The same approach can be done using the data that can be collected from the logistics and also stocks movement within the supply chain. The approach can help to deal with the uncertain and non-stationary demand with minimum cost.

Bibliography

"Bonnmotion." [Online]. Available: http://net.cs.uni-bonn.de/wg/cs/applications/bonnmotion/

"IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specificatio," IEEE Standard 802.11p, 2010.

"Intelligent transport systems," http://www.etsi.org/technologies-clusters/technologies/intelligent-transport?highlight=YToxOntpOjA7czozOiJpdHMiO30=, accessed: 2015-05-25.

"ns3::nakagamipropagationlossmodel class reference," July 13, 2011.

"802.11a-1999 - Supplement to IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Phy," 1999. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=815305

"IEEE 802.11e Wireless LAN for Quality of Service," *IEEE Standard*, 2002.

"IEEE 1609 - Family of Standards for Wireless Access in Vehicular Environments (WAVE)," *IEEE Standard*, 2013.

"Network simulator 3." [Online]. Available: http://www.nsnam.org/

A. Abdrabou and W. Zhuang, "On a Stochastic Delay Bound for Disrupted Vehicle-to-Infrastructure Communication with Random Traffic," in *GLOBECOM 2009 - 2009 IEEE Global Telecommunications Conference*. IEEE, Nov. 2009, pp. 1–6.

——, "Probabilistic Delay Control and Road Side Unit Placement for Vehicular Ad Hoc Networks with Disrupted Connectivity," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 1, pp. 129–139, Jan. 2011.

A. K. K. Aboobaker, "Performance Analysis of Authentication Protocols in Vehicular Ad Hoc Networks (VANET)," 2010.

M. Abuelela, S. Olariu, and I. Stojmenovic, "OPERA: Opportunistic Packet Relaying in Disconnected Vehicular Ad Hoc Networks," *The Fifth IEEE International Conference on Mobile Adhoc and Sensor Systems*, pp. 285–294, 2008.

D. W. Aha, "Generalizing from Case Studies: A Case Study," in *Proceedings of the Ninth International Workshop on Machine Learning*, ser. ML92. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1992, pp. 1–10.

——, "Generalizing from Case Studies: A Case Study," in *Proceedings of the Ninth International Workshop on Machine Learning*, ser. ML92. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1992, pp. 1–10.

R. Ahlswede, N. Cai, R. W. Yeung, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

F. Al-Obeidat, A. T. Al-Taani, N. Belacel, L. Feltrin, and N. Banerjee, "A fuzzy decision tree for processing satellite images and landsat data," *Procedia Computer Science*, vol. 52, pp. 1192–1197, 2015.

K. ALEXANDROS and H. MELANIE, "Model selection via meta-learning: a comparative study," *International Journal on Artificial Intelligence Tools*, vol. 10, no. 04, pp. 525–554, 2001.

S. Ali and K. A. Smith, "Kernal Width Selection for SVM Classification," *International Journal of Data Warehousing and Mining*, vol. 1, no. 4, pp. 78–97, Jan. 2005.

Y. Amit and D. Geman, "Shape Quantization and Recognition with Randomized Trees," *Neural Computation, Massachusetts Institute of Technology*, vol. 9, no. 7, pp. 1545–1588, Oct. 1997.

M. H. Arbabi and M. Weigle, "Using vehicular networks to collect common traffic data," in *Proceedings of the sixth ACM international workshop on VehiculAr InterNETworking - VANET '09*. New York, New York, USA: ACM Press, Sep. 2009, p. 117.

K. Ashokkumar, B. Sam, R. Arshadprabhu *et al.*, "Cloud based intelligent transport system," *Procedia Computer Science*, vol. 50, pp. 58–63, 2015.

S. Badal, S. Ravela, B. Draper, and A. Hanson, "A Practical Obstacle Detection and Avoidance System," in *IEEE Workshop on Applications of Computer Vision*, 1994.

R. K. Balan, K. X. Nguyen, and L. Jiang, "Real-time trip information service for a large taxi fleet," in *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '11. New York, NY, USA: ACM, 2011, pp. 99–112.

H. Bensusan and C. Giraud-Carrier, "Casa batló is in passeig de gràcia or landmarking the expertise space," pp. 29–46, 2000.

J. Bernsen and D. Manivannan, "Greedy routing protocols for vehicular ad hoc networks," *Wireless Communications and Mobile Computing Conference*, pp. 632–637, Aug 2008.

——, "Greedy Routing Protocols for Vehicular Ad Hoc Networks," in *2008 International Wireless Communications and Mobile Computing Conference*. IEEE, Aug. 2008, pp. 632–637. [Online]. Available: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4600008

Y. Bi, H. Zhao, and X. Shen, "A Directional Broadcast Protocol for Emergency Message Exchange in Inter-Vehicle Communications," in *2009 IEEE International Conference on Communications*. IEEE, Jun. 2009, pp. 1–5.

Y. Bi, L. X. Cai, X. Shen, and H. Zhao, "A Cross Layer Broadcast Protocol for Multihop Emergency Message Dissemination in Inter-Vehicle Communication," in *2010 IEEE International Conference on Communications*. IEEE, May 2010, pp. 1–5.

J. Blum, A. Eskandarian, and L. Hoffman, "Mobility management in IVC networks," in *IEEE IV2003 Intelligent Vehicles Symposium. Proceedings (Cat. No.03TH8683)*. IEEE, 2003, pp. 150–155. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1212900

M. Boban, J. Barros, and O. Tonguz, "Geometry-based vehicle-to-vehicle channel modeling for large-scale simulation," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 9, pp. 4146–4164, Nov 2014.

L. Bononi and M. Di Felice, "A Cross Layered MAC and Clustering Scheme for Efficient Broadcast in VANETs," *2007 IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems*, pp. 1–8, 2007. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4428735

P. Brazdil, C. Giraud-carrier, C. Soares, and R. Vilalta, *Metalearning: Applications to Data Mining*, ser. Cognitive Technologies, C. Sammut and G. I. Webb, Eds. Springer, 2009.

P. B. Brazdil, C. Soares, and J. P. Da Costa, "Ranking learning algorithms: Using ibl and meta-learning on accuracy and time results," *Machine Learning*, vol. 50, no. 3, pp. 251–277, 2003.

P. Brazdil, C. Soares, and J. D. Costa, "Ranking learning algorithms: Using IBL and meta-learning on accuracy and time results," *Machine Learning*, pp. 251–277, 2003.

——, "Ranking learning algorithms: Using IBL and meta-learning on accuracy and time results," *Machine Learning*, pp. 251–277, 2003.

P. Brazdil, "Data transformation and model selection by experimentation and meta-learning," in *Workshop Notes–Upgrading Learning to the Meta-Level: Model Selection and Data Transformation, number CSR-98-02 in Technical Report*. Citeseer, 1998, pp. 11–17.

L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32.

M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *ACM sigmod record*, vol. 29, no. 2. ACM, 2000, pp. 93–104.

L. Briesemeister and G. Hommel, "Role-Based Multicast in Highly Mobile but Sparsely Connected Ad Hoc Networks," in *Proc of the first Annual Workshop on Mobile Ad Hoc Networking and Computing*, 2000.

L. Briesemeister, L. Schafers, and G. Hommel, "Disseminating messages among highly mobile hosts based on inter-vehicle communication," in *Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE*. IEEE, 2000, pp. 522–527.

C. Brodley, "Recursive automatic bias selection for classifier construction," *Machine Learning*, vol. 20, no. 1-2, pp. 63–94, 1995.

D. J. Chadwick, J. W. Marshall, B. L. Hinton, V. M. Patel, and H. W. Lam, "Communications concepts to support early implementation of ivhs in north america," *I V H S Journal*, vol. 1, no. 1, pp. 45–61, 1993.

C. Chan and S. Liew, "Data-Collection Capacity of IEEE 802.11-like Sensor Networks," in *2006 IEEE International Conference on Communications*. IEEE, 2006, pp. 3339–3346.

C.-Y. Chan and D. Marco, "Traffic monitoring at signal-controlled intersections and data mining for safety applications," in *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*. IEEE, 2004, pp. 355–360.

P. Chan and S. J. Stolfo, "Experiments on multistrategy learning by meta-learning," in *In Proc. Second Intl. Conference on Info. and Knowledge Mgmt*, 1993, pp. 314–323.

V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 15:1–15:58, Jul. 2009. [Online]. Available: http://doi.acm.org/10.1145/1541880.1541882

P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth, *{CRISP}-{DM} 1.0: Step-by-Step Data Mining Guide*, SPSS, 2000.

N. Cheifetz, A. Same, P. Aknin, and E. de Verdalle, "A pattern recognition approach for anomaly detection on buses brake system," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, Oct 2011, pp. 266–271.

S. Chen, M. Huang, S. Tang, and Y. Wang, "Capacity of Data Collection in Arbitrary Wireless Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 1, pp. 52–60, Jan. 2012.

S. I.-J. Chien and C. M. Kuchipudi, "Dynamic travel time prediction with real-time and historic data," *Journal of transportation engineering*, vol. 129, no. 6, pp. 608–616, 2003.

J. K. Christian Köpf, Charles Taylor, "Meta-Analysis: From Data Characterisation for Meta-Learning to Meta-Regression," *Proceedings of International Symposium on Data Mining and Statistics*, 2000. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.26.8159

Cmuportugal.org, "Drive-in: Distributed routing and infotainment through vehicular internetworking," 2014. [Online]. Available: http://www.cmuportugal.org/tiercontent.aspx?id=1552

F. C. Commission, "Fcc allocates spectrum in 5.9 ghz range for intelligent transportation systems uses," January 2015. [Online]. Available: http://transition.fcc.gov/Bureaus/Engineering_Technology/News_Releases/1999/nret9006.html

C. . C. C. Consortium, "Car 2 car communication consortium," September 2008. [Online]. Available: https://www.car-2-car.org

M. Di Francesco, S. K. Das, and G. Anastasi, "Data Collection in Wireless Sensor Networks with Mobile Elements," *ACM Transactions on Sensor Networks*, vol. 8, no. 1, pp. 1–31, Aug. 2011.

M. Dikaiakos, A. Florides, T. Nadeem, and L. Iftode, "Location-Aware Services over Vehicular Ad-Hoc Networks using Car-to-Car Communication," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 8, pp. 1590–1602, 2007.

——, "Location-aware services over vehicular ad-hoc networks using car-to-car communication," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 8, pp. 1590 –1602, oct. 2007.

P. Domingos, "Knowledge discovery via multiple models," *Intelligent Data Analysis*, vol. 2, no. 3, pp. 187–202, 1998.

L. Du, S. Ukkusuri, W. F. Yushimito Del Valle, and S. Kalyanaraman, "Optimization models to characterize the broadcast capacity of vehicular ad hoc networks," *Transportation Research Part C: Emerging Technologies*, vol. 17, no. 6, pp. 571–585, Dec. 2009.

E. J. Duarte-Melo and M. Liu, "Data-gathering wireless sensor networks: organization and capacity," *Computer Networks*, vol. 43, no. 4, pp. 519–537, Nov. 2003.

W. Duch and K. Grudzinski, "Meta-learning: searching in the model space," *Proceedings of the International Conference on Neural Information Processing*, vol. 1, pp. 235–240, 2001.

M. Durresi, A. Durresi, and L. Barolli, "Emergency Broadcast Protocol for Inter-Vehicle Communications," in *11th International Conference on Parallel and Distributed Systems (ICPADS'05)*, vol. 2. IEEE, 2005, pp. 402–406.

——, "Emergency broadcast protocol for inter-vehicle communications," in *Parallel and Distributed Systems, 2005. Proceedings. 11th International Conference on*, vol. 2. IEEE, 2005, pp. 402–406.

M. Efatmaneshnik, A. T. Balaei, A. Dempster, and J. Marczyk, "A Channel Capacity Perspective on Cooperative Positioning Algorithms for VANET," *ION GNSS 2009*, pp. 1034–1041, 2009.

F. El-Moukaddem, "Maximizing data gathering capacity of wireless sensor networks using mobile relays," in *Mobile Adhoc and Sensor Systems (MASS), 2010 IEEE 7th International Conference on*, 2010, pp. 312–321.

H. ElGamal, "On the Scaling Laws of Dense Wireless Sensor Networks: The Data Gathering Channel," *IEEE Transactions on Information Theory*, vol. 51, no. 3, pp. 1229–1234, Mar. 2005.

C. Englund, L. Chen, A. Vinel, and S. Lin, "Future applications of vanets," in *Vehicular ad hoc Networks*, C. Campolo, A. Molinaro, and R. Scopigno, Eds. Springer International Publishing, 2015, pp. 525–544. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-15497-8_18

F. ETSI Headquarters, Sophia Antipolis, "Cars 'talking and hearing in harmony' - a smart move for etsi!" September 2008. [Online]. Available: http://www.etsi.org/index.php/news-events/news/226-press-release-30th-september-2008

H. Fernandes, A. Boukerche, R. Pazzi, and S. Samarah, "Efficient data gathering and position dissemination protocols for heterogeneous vehicle ad hoc and sensor networks," in *Exhibition*. IEEE, Mar. 2009, pp. 1–4.

R. Fernandes, P. M. D'Orey, and M. Ferreira, "DIVERT for realistic simulation of heterogeneous vehicular networks," in *The 7th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE MASS 2010)*. IEEE, Nov. 2010, pp. 721–726.

W. D. Fisher, "On Grouping for Maximum Homogeneity," *Journal of the American Statistical Association*, vol. 53, no. 284, p. 789, Dec. 1958.

C. Fragouli, J.-Y. L. Boudec, and J. Widmer, "Network coding: an instant primer," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 1, 2006.

M. Franceschetti, O. Dousse, D. N. C. Tse, and P. Thiran, "Closing the Gap in the Capacity of Wireless Networks Via Percolation Theory," *IEEE Transactions on Information Theory*, vol. 53, no. 3, pp. 1009–1018, Mar. 2007.

M. Friedl and C. Brodley, "Decision tree classification of land cover from remotely sensed data," *Remote Sensing of Environment*, vol. 61, no. 3, pp. 399 – 409, 1997. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0034425797000497

H. Fusler, J. Widmer, M. Kasemann, M. Mauve, and H. Hartenstein, "Contention-based forwarding for mobile ad hoc networks," *Ad Hoc Networks*, vol. 1, no. 4, pp. 351–369, 2003.

H. Fusler, J. Widmer, M. Kaumlsemann, M. Mauve, and H. Hartenstein, "Contention-based forwarding for mobile ad hoc networks," *Ad Hoc Networks*, vol. 1, no. 4, pp. 351–369, 2003.

H. Füß ler, H. F. Uler, H. Hartenstein, M. Mauve, M. Käsemann, D. Vollmer, and M. M. M. K. Asemann, "Location-Based Routing for Vehicular Ad-Hoc Networks," Atlanta, Georgia, 2002. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.19.6287

——, "Location-Based Routing for Vehicular Ad-Hoc Networks," Atlanta, Georgia, 2002.

H. Füßler, M. Mauve, H. Hartenstein, M. Kasemann, and D. Vollmer, "Location based routing for vehicular ad-hoc networks," *ACM SIGMOBILE Mobile Computing and Communications Review (MC2R)*, vol. 7, no. 1, pp. 47–49, Jan 2003.

J. Gama and P. Brazdil, "Characterization of classification algorithms," in *Proceedings of the 7th Portuguese Conference on Artificial Intelligence: Progress in Artificial Intelligence*, ser. EPIA '95. London, UK, UK: Springer-Verlag, 1995, pp. 189–200.

M. Gastpar and M. Vetterli, "On the capacity of wireless networks: the relay case," in *Proceedings.Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3. IEEE, 2002, pp. 1577–1586.

M. Gerla and L. Kleinrock, "Vehicular networks and the future of the mobile internet," *Computer Networks*, vol. 55, no. 2, pp. 457–469, 2010.

——, "Vehicular networks and the future of the mobile internet," *Comput. Netw.*, vol. 55, pp. 457–469, February 2011. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2010.10.015

C. Giraud-Carrier, R. Vilalta, and P. Brazdil, "Introduction to the special issue on meta-learning," *Machine Learning*, vol. 54, no. 3, pp. 187–193, 2004.

S. Gr and P. M, "Performance Evaluation of IEEE 1609 WAVE and IEEE 802 . 11p for Vehicular Communications," *Performance Evaluation*, pp. 344–348, 2010.

M. Grossglauser and D. Tse, "Mobility increases the capacity of ad hoc wireless networks," *IEEE/ACM Transactions on Networking*, vol. 10, no. 4, pp. 477–486, Aug. 2002.

Y. Gunter, B. Wiegel, and H. P. Grossmann, *Cluster-based Medium Access Scheme for VANETs*. IEEE, 2007. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4357651

P. Gupta and P. Kumar, "Internets in the sky: capacity of 3D wireless networks," in *Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No.00CH37187)*, vol. 3. Sydney, Australia: IEEE, 2000, pp. 2290–2295.

——, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, Mar. 2000.

Z. J. Haas, M. R. Pearlman, and P. Samar, "The zone routing protocol (zrp) for ad hoc networks," *draft-ietf-manet-zone-zrp-04. txt*, 2002.

F. C. Haerri, Jérôme;Filali, "On Meaningful Parameters for Routing in VANETs Urban Environments under Realistic Mobility Patterns," in *AutoNet 2006, 1st IEEE Workshop on Automotive Networking and Applications (in conjuction with IEEE Globecom 2006)*, San Francisco, 2006.

B. Han and G. Simon, "Capacity Of Wireless Ad Hoc Networks , A Survey," *Network*, pp. 1–23, 2007.

J. Hao, "Traffic information aggregation and propagation scheme for vanet in city environment," in *2010 3rd IEEE International Conference on Broadband Network and Multimedia Technology (IC-BNMT)*. IEEE, Oct. 2010, pp. 619–623.

V. Harinarayan, A. Rajaraman, and J. D. Ullman, "Implementing data cubes efficiently," *SIGMOD Rec.*, vol. 25, no. 2, pp. 205–216, Jun. 1996. [Online]. Available: http://doi.acm.org/10.1145/235968.233333

S. Harsola, P. Deshpande, and J. Haritsa, "Targeted association rule mining in data cubes," Tech. Rep. TR-2012, Tech. Rep., 2012.

M. Hasan, D. Cuneo, and A. Chachich, "Analysis of traffic video to develop driver behavior models for microscopic traffic simulation," in *Intelligent Transportation System, 1997. ITSC '97., IEEE Conference on*, Nov 1997, pp. 747–752.

T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, 2nd ed. Springer-Verlag New York, 2009. [Online]. Available: http://www.bookmetrix.com/detail/book/44aea3a8-3253-4796-8244-700f19c708eb#downloads

T. A. Hauser and W. T. Scherer, "Data mining tools for real-time traffic signal decision support & maintenance," in *Systems, Man, and Cybernetics, 2001 IEEE International Conference on*, vol. 3. IEEE, 2001, pp. 1471–1477.

W. He, T. Lu, and C. Q. Yu, "A novel traffic flow forecasting method based on the artificial neural networks and intelligent transportation systems data mining," *Advanced Materials Research*, vol. 842, pp. 708–711, 2014.

Y. He, S. Blandin, L. Wynter, and B. Trager, "Analysis and real-time prediction of local incident impact on transportation networks," in *Data Mining Workshop (ICDMW), 2014 IEEE International Conference on*. IEEE, 2014, pp. 158–166.

M. A. Hearst, S. T. Dumais, E. Osman, J. Platt, and B. Scholkopf, "Support vector machines," *Intelligent Systems and their Applications, IEEE*, vol. 13, no. 4, pp. 18–28, 1998.

A. H. Ho, Y. H. Ho, and K. A. Hua, "A connectionless approach to mobile ad hoc networks in street environments," *IEEE Proceedings Intelligent Vehicles Symposium 2005*, pp. 575–582, 2005.

T. Ho, R. Koetter, M. Medard, D. R. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," *IEEE International Symposium on Information Theory 2003 Proceedings*, vol. pages442, pp. 442–442, 2003.

A. Host-Madsen, "On the capacity of wireless relaying," in *Proceedings IEEE 56th Vehicular Technology Conference*, vol. 3. Honolulu, HI, USA: IEEE, 2002, pp. 1333–1337.

B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, and S. Madden, "CarTel: a distributed mobile sensor computing system," *Proceedings of the 4th international conference on Embedded networked sensor systems*, p. 125, 2006.

——, "Cartel: a distributed mobile sensor computing system," in *Proceedings of the 4th international conference on Embedded networked sensor systems*, ser. SenSys '06. New York, NY, USA: ACM, 2006, pp. 125–138. [Online]. Available: http://doi.acm.org/10.1145/1182807.1182821

P. Jacquet and G. Rodolakis, "Multicast Scaling Properties in Massively Dense Ad Hoc Networks," in *11th International Conference on Parallel and Distributed Systems (ICPADS'05)*, vol. 2. IEEE, pp. 93–99.

N. Jankowski, W. Duch, and K. Grabczewski, *Meta-learning in computational intelligence*. Springer Science & Business Media, 2011, vol. 358.

L. Jeng-Wei, L. Chun-Chih, T. Shih-Pu, H. Mong-Fong, and K. Yau-Hwang, "A hybrid traffic geographic routing with cooperative traffic information collection scheme in VANET," *Advanced Communication Technology (ICACT), 2011 13th International Conference on*, pp. 1496–1501, 2011.

M. Jerbi, S. M. Senouci, Y. G. Doudane, and A.-L. Beylot, "Geo-localized virtual infrastructure for urban vehicular networks," in *2008 8th International Conference on ITS Telecommunications*. IEEE, Oct. 2008, pp. 305–310.

M.-F. Jhang and W. Liao, "On Cooperative and Opportunistic Channel Access for Vehicle to Roadside (V2R) Communications," in *IEEE GLOBECOM 2008 - 2008 IEEE Global Telecommunications Conference*. IEEE, 2008, pp. 1–5.

H. L. Jie Wu, "A Dominating-Set-Based Routing Scheme in Ad Hoc Wireless Networks," *Telecommunication Systems Journal*, 1999. [Online]. Available: http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.31.8425

D. Johnson, Y. Hu, and D. Maltz, "The dynamic source routing protocol (dsr) for mobile ad hoc networks for ipv4," Tech. Rep., Feb 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4728.txt

D. Johnson, Y. Hu, D. Maltz *et al.*, "The dynamic source routing protocol (dsr) for mobile ad hoc networks for ipv4," RFC 4728, Tech. Rep., 2007.

P. M. J.S. Milton, J.J. Corbet, *Introduction to statistics*. Mc Graw Hill, 1997.

M. Kamber, J. Han, and J. Chiang, "Metarule-guided mining of multi-dimensional association rules using data cubes." in *KDD*, vol. 97, 1997, p. 207.

S. Kamijo, Y. Matsushita, K. Ikeuchi, and M. Sakauchi, "Traffic monitoring and accident detection at intersections," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 1, no. 2, pp. 108–118, 2000.

B. Karp and H. Kung, "Gpsr: Greedy perimeter stateless routing for wireless networks," *in Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, 2000.

B. Karp and H. T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking - MobiCom '00*. New York, New York, USA: ACM Press, Aug. 2000, pp. 243–254. [Online]. Available: http://portal.acm.org/citation.cfm?id=345910.345953

S. Katti, H. Rahul, W. Hu, D. Katabi, and M. Médard, "XORs in the Air: Practical Wireless Network Coding," *IEEE/ACM Transactions on Networking*, vol. 16, no. 3, pp. 497–510, 2008.

A. Kchiche and F. Kamoun, "Centrality-based Access-Points deployment for vehicular networks," in *2010 17th International Conference on Telecommunications*. IEEE, 2010, pp. 700–706.

R. L. Keirstead, "Vehicle safety monitoring system for viewing blind spots," Feb. 17 2004, uS Patent 6,693,519.

J. Keller, I. Paterson, and H. Berrer, "An integrated concept for multi-criteria ranking of data-mining algorithms," *Meta-Learning: Building Automatic Advice Strategies for Model Selection and Method Combination*, 2000.

A. Keshavarz-Haddad and R. H. Riedi, "Bounds for the capacity of wireless multihop networks imposed by topology and demand," in *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing - MobiHoc '07*. New York, New York, USA: ACM Press, Sep. 2007, p. 256.

A. Khan, S. Sadhu, and M. Yeleswarapu, "A comparative analysis of DSRC and 802.11 over Vehicular Ad hoc Networks," *Ad Hoc Networks*, 2009.

R. Kimball, M. Ross, W. Thornthwaite, J. Mundy, and B. Becker, *The data warehouse lifecycle toolkit*. Wiley, 2011.

E. M. Knorr and R. T. Ng, "Algorithms for Mining Distance-Based Outliers in Large Datasets," in *Proceedings of the 24rd International Conference on Very Large Data Bases*, ser. VLDB '98. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998, pp. 392–403.

D. E. Knuth, "Big Omicron and big Omega and big Theta," *ACM SIGACT News*, vol. 8, no. 2, pp. 18–24, Apr. 1976.

Y. Kodratoff, D. Sleeman, M. Uszynski, K. Causse, and S. Craw, "Building a machine learning toolbox," 1992.

G. Korkmaz, E. Ekici, and F. Ozguner, "An Efficient Fully Ad-Hoc Multi-Hop Broadcast Protocol for Inter-Vehicular Communication Systems," in *IEEE International Conference on Communications*, vol. 1, 2006.

G. Korkmaz, E. Ekici, F. Özgüner, and U. Özgüner, "Urban multi-hop broadcast protocol for inter-vehicle communication systems," in *Proceedings of the first ACM workshop on Vehicular ad hoc networks - VANET '04*. New York, New York, USA: ACM Press, Oct. 2004, p. 76. [Online]. Available: http://portal.acm.org/citation.cfm?id=1023875.1023887

G. Korkmaz, E. Ekici, F. Ozguner, and U. Ozguner, "Urban multi-hop broadcast protocol for inter-vehicle communication systems," *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, Oct 2004.

D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO - Simulation of Urban MObility," *International Journal On Advances in Systems and Measurements*, vol. 5, no. 3&4, pp. 128–138, December 2012.

G. Kramer, M. Gastpar, and P. Gupta, "Cooperative Strategies and Capacity Theorems for Relay Networks," *IEEE Transactions on Information Theory*, vol. 51, no. 9, pp. 3037–3063, Sep. 2005.

J. Kwon, B. Coifman, and P. Bickel, "Day-to-day travel-time trends and travel-time prediction from loop-detector data," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1717, no. 1, pp. 120–129, 2000.

M. G. Lagoudakis and M. L. Littman, "Algorithm selection using reinforcement learning," in *Proceedings of the Seventeenth International Conference on Machine Learning*, ser. ICML '00. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, pp. 511–518. [Online]. Available: http://dl.acm.org/citation.cfm?id=645529.657981

B. Lauwens, B. Scheers, and A. Van de Capelle, "Throughput Analysis of Multi-Hop CSMA/CA Wireless Networks," in *2008 IEEE Sarnoff Symposium*. IEEE, Apr. 2008, pp. 1–6.

J. LeBrun, C.-N. Chuah, D. Ghosal, and M. Zhang, "Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks," *Vehicular Technology Conference*, vol. 4, pp. 2289–2293, May 2005.

J. Lebrun, C. N. Chuah, D. Ghosal, and M. Zhang, "Knowledge-Based Opportunistic Forwarding in Vehicular Wireless Ad Hoc Networks," in *Ieee Vehicular Technology Conference*, vol. 61, no. 4. Citeseer, 2005, p. 2289.

J. Lee, "Design of a Network Coverage Analyzer for Roadside-to-Vehicle Telematics Networks," in *2008 Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*. IEEE, 2008, pp. 201–205.

U. Lee, E. Magistretti, M. Gerla, P. Bellavista, and A. Corradi, "Dissemination and harvesting of urban data using vehicular sensing platforms," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 2, pp. 882 –901, feb. 2009.

U. Lee and M. Gerla, "A survey of urban vehicular sensing platforms," *Computer Networks*, vol. 54, no. 4, pp. 527–544, Mar. 2010. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2009.07.011http://linkinghub.elsevier.com/retrieve/pii/S1389128609002382

——, "A survey of urban vehicular sensing platforms," *Computer Networks*, vol. 54, no. 4, pp. 527–544, Mar. 2010.

U. Lee, E. Magistretti, B. Zhou, M. Gerla, P. Bellavista, and A. Corradi, "Efficient Data Harvesting in Mobile Sensor Platforms," *Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops PERCOMW06*, pp. 352–356, 2006. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1599002

U. Lee, E. Magistretti, M. Gerla, P. Bellavista, and A. Corradi, "Dissemination and Harvesting of Urban Data Using Vehicular Sensing Platforms," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 2, pp. 882–901, 2009. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4573261

D. Li, H. Huang, X. Li, M. Li, and F. Tang, "A Distance-Based Directional Broadcast Protocol for Urban Vehicular Ad Hoc Network," in *2007 International Conference on Wireless Communications, Networking and Mobile Computing*. IEEE, Sep. 2007, pp. 1520–1523.

F. Li and Y. Wang, "Routing in vehicular ad hoc networks: A survey," *IEEE Vehicular Technology Magazine*, vol. 2, no. 2, pp. 12–22, 2007. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4450627

J. Li, C. Blake, D. S. J. D. Couto, H. I. Lee, and R. Morris, "Capacity of wireless ad hoc networks," in *Proceedings of the 7th annual international conference on Mobile computing and networking MobiCom 01*, vol. 1, no. 2. ACM Press, 2001, pp. 61–69.

J. Li, C. Blake, D. S. J. De Couto, H. I. Lee, and R. Morris, "Capacity of Ad Hoc Wireless Networks," *Proceedings of the 7th annual international conference on Mobile computing and networking MobiCom 01*, vol. 01, no. 1, pp. 61–69, 2001.

T. Li, Y. Li, and J. Liao, *A Contention-Based Routing Protocol for Vehicular Ad Hoc Networks in City Environments*. IEEE, Jun. 2009.

X.-Y. Li, S.-J. Tang, and O. Frieder, "Multicast capacity for large scale wireless ad hoc networks," in *Proceedings of the 13th annual ACM international conference on Mobile computing and networking - MobiCom '07*. New York, New York, USA: ACM Press, Sep. 2007, p. 266.

A. Liaw and M. Wiener, "Classification and regression by randomforest," *R News*, vol. 2, no. 3, pp. 18–22, 2002.

H. Liimatainen, "Utilization of fuel consumption data in an ecodriving incentive system for heavy-duty vehicle drivers," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, no. 4, pp. 1087–1095, Dec 2011.

C. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 7, pp. 1265–1275, 1997. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=622910

——, "Adaptive clustering for mobile wireless networks," *IEEE Journal of Selected Areas in Communications*, vol. 15, no. 7, pp. 1265–1275, 1997.

J. Lin and D. Yu, "Traffic-related air quality assessment for open road tolling highway facility." *Journal of Environmental Management*, vol. 88, no. 4, pp. 962–969, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/B6WJ7-4P18BGG-4/2/fcf721b6150a27e7035f2f6d83616130

G. Lindner and R. Studer, "Ast: Support for algorithm selection with a cbr approach," in *Proceedings of the Third European Conference on Principles of Data Mining and Knowledge Discovery*, ser. PKDD '99. London, UK, UK: Springer-Verlag, 1999, pp. 418–423. [Online]. Available: http://dl.acm.org/citation.cfm?id=645803.669655

B. Liu, P. Thiran, and D. Towsley, "Capacity of a wireless ad hoc network with infrastructure," in *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing - MobiHoc '07*. New York, New York, USA: ACM Press, Sep. 2007, p. 239.

B. Liu, D. Towsley, and A. Swami, "Data gathering capacity of large scale multihop wireless networks," in *2008 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems*. IEEE, Sep. 2008, pp. 124–132.

C. G. Liu, G. Liu, B.-s. Lee, B.-c. Seet, C.-h. Foh, and K.-k. Lee, "A Routing Strategy for Metropolis Vehicular," pp. 533–542, 2004. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.60.1587

G. Liu, B.-S. Lee, B.-C. Seet, C. Foh, K. Wong, and K.-K. Lee, "A routing strategy for metropolis vehicular communications," *in International Conference on Information Networking (ICOIN)*, pp. 134–143, 2004.

C. Lochert, M. Mauve, H. Fussler, and H. Hartenstein, "Geographic routing in city scenarios," *ACM SIGMOBILE'05*, vol. 9, no. 1, pp. 69–72, 2005.

C. Lochert, M. Mauve, H. Füß ler, and H. Hartenstein, "Geographic routing in city scenarios," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 9, no. 1, p. 69, Jan. 2005. [Online]. Available: http://portal.acm.org/citation.cfm?id=1055959.1055970

C. Lochert, B. Scheuermann, C. Wewetzer, A. Luebke, and M. Mauve, "Data aggregation and roadside unit placement for a vanet traffic information system," in *Proceedings of the fifth ACM international workshop on VehiculAr InterNETworking VANET 08*. ACM Press, 2008, p. 58. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1410043.1410054

A. Loureiro, L. Torgo, and C. Soares, "Outlier Detection Using Clustering Methods: a Data Cleaning Application," in *Proceedings of the Data Mining for Business Workshop*, D. C. Soares C Moniz L, Ed., Oporto, Portugal, 2005, pp. 57–62.

D. E. Lucani, F. H. P. Fitzek, M. Medard, and M. Stojanovic, "Network coding for data dissemination: it is not what you know, but what your neighbors don't know," *2009 7th International Symposium on Modeling and Optimization in Mobile Ad Hoc and Wireless Networks*, pp. 1–8, 2009.

C. Ma, "Single Path Flooding Chain Routing in Ad Hoc Networks," in *2005 International Conference on Parallel Processing (ICPP'05)*. IEEE, Jun. 2005, pp. 303–310.

X. Ma, X. Chen, and H. H. Refai, "Performance and Reliability of DSRC Vehicular Safety Communication: A Formal Analysis," *EURASIP Journal on Wireless Communications and Networking*, vol. 2009, pp. 1–13, 2009.

O. Maimon and L. Rokach, *Data mining and knowledge discovery handbook*. Springer, 2005, vol. 2.

S. Mangold, S. Choi, P. May, O. Klein, G. Hiertz, L. Stibor, and Q. Bss, "IEEE 802 . 11e Wireless LAN for Quality of Service," *New York*, vol. 11, pp. 32–39, 2002.

V. N. Manish, "A Study on the Feasibility of Mobile Gateways for Vehicular Ad-hoc Networks," 2004. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.61.290

D. Marco, E. J. Duarte-Melo, M. Liu, and D. L. Neuhoff, "On the many-to-one transport capacity of a dense wireless sensor network and the compressibility of its data," pp. 1–16, Apr. 2003.

F. Martelli, M. E. Renda, and P. Santi, "Measuring IEEE 802.11p Performance for Active Safety Applications in Cooperative Vehicular Systems," in *2011 IEEE 73rd Vehicular Technology Conference (VTC Spring)*. IEEE, May 2011, pp. 1–5.

F. J. Martinez, J. C. Cano, C. T. Calafate, and P. Manzoni, "CityMob: A Mobility Model Pattern Generator for VANETs," in *ICC Workshops 2008 IEEE International Conference on Communications Workshops*. Ieee, 2008, pp. 370–374.

F. J. Martinez, C. K. Toh, J. C. Cano, C. T. Calafate, and P. Manzoni, "A Street Broadcast Reduction Scheme (SBR) to Mitigate the Broadcast Storm Problem in VANETs," *Wireless Personal Communications*, 2010.

F. J. Martinez, C.-K. Toh, J.-C. Cano, C. T. Calafate, and P. Manzoni, "Realistic Radio Propagation Models (RPMs) for VANET Simulations," in *2009 IEEE Wireless Communications and Networking Conference*. IEEE, Apr. 2009, pp. 1–6.

F. J. Martinez, M. Fogue, M. Coll, J.-C. Cano, C. T. Calafate, and P. Manzoni, "Assessing the Impact of a Realistic Radio Propagation Model on VANET Scenarios Using Real Maps," in *2010 Ninth IEEE International Symposium on Network Computing and Applications*. IEEE, Jul. 2010, pp. 132–139.

J. a. Mendes-Moreira, A. M. Jorge, J. F. de Sousa, and C. Soares, "Comparing state-of-the-art regression methods for long term travel time prediction," *Intell. Data Anal.*, vol. 16, no. 3, pp. 427–449, May 2012. [Online]. Available: http://dx.doi.org/10.3233/IDA-2012-0532

J. Mendes-Moreira, A. M. Jorge, J. F. de Sousa, and C. Soares, "Improving the accuracy of long-term travel time prediction using heterogeneous ensembles," *Neurocomputing*, vol. 150, Part B, pp. 428 – 439, 2015, special Issue on Information Processing and Machine Learning for Applications of EngineeringSolving Complex Machine Learning Problems with Ensemble MethodsVisual Analytics using Multidimensional ProjectionsSelected papers from the {IEEE} 17th International Conference on Intelligent Engineering Systems (INES'13)Selected papers from the Workshop on Visual Analytics using Multidimensional Projections, held at EuroVis 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231214012351

D. Michie, D. J. Spiegelhalter, C. C. Taylor, and J. Campbell, Eds., *Machine Learning, Neural and Statistical Classification*. Upper Saddle River, NJ, USA: Ellis Horwood, 1994.

D. Michie, D. J. Spiegelhalter, C. C. Taylor, and J. Campbell, "Machine learning, neural and statistical classification," Jun. 1995.

J. Miller, "Vehicle-to-vehicle-to-infrastructure (V2V2I) intelligent transportation system architecture," in *2008 IEEE Intelligent Vehicles Symposium*. IEEE, Jun. 2008, pp. 715–720.

J. Mittag, F. Schmidt-Eisenlohr, M. Killat, J. Härri, and H. Hartenstein, "Analysis and design of effective and low-overhead transmission power control for VANETs," in *Proceedings of the fifth ACM international workshop on VehiculAr Inter-NETworking - VANET '08*. New York, New York, USA: ACM Press, Sep. 2008, p. 39.

J. Mittag, S. Papanastasiou, H. Hartenstein, and E. G. Strom, "Enabling Accurate Cross-Layer PHY/MAC/NET Simulation Studies of Vehicular Communication Networks," *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1311–1326, Jul. 2011.

S. Mohajer, S. N. Diggavi, C. Fragouli, and D. N. C. Tse, "Capacity of deterministic Z-chain relay-interference network," in *2009 IEEE Information Theory Workshop on Networking and Information Theory*. IEEE, Jun. 2009, pp. 331–335.

M. E. Mohammad Nozari-Zarmehri, Kamal Shahtalebi, "Determination of MIMO Systems Capacity in Uniformly Distributed Channel Error," *JOURNAL OF ELECTRICAL ENGINEERING MAJLESI*, vol. 2, pp. 23–28, 2009.

D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to linear regression analysis*. John Wiley & Sons, 2012, vol. 821.

I. S. Mumick, D. Quass, and B. S. Mumick, "Maintenance of data cubes and summary tables in a warehouse," *SIGMOD Rec.*, vol. 26, no. 2, pp. 100–111, Jun. 1997. [Online]. Available: http://doi.acm.org/10.1145/253262.253277

B. Murphy, M. Lebold, J. Banks, and K. Reichard, "Diagnostic end to end monitoring amp;amp; fault detection for braking systems," in *Aerospace Conference, 2006 IEEE*, 2006, pp. 1–8.

D. Myr, "Real time vehicle guidance and forecasting system under traffic jam conditions," Nov. 12 2002, uS Patent 6,480,783.

V. Namboodiri, M. Agarwal, and L. Gao, "A study on the feasibility of mobile gateways for vehicular ad-hoc networks," *in Proceedings of the First International Workshop on Vehicular Ad Hoc Networks*, pp. 66–75, 2004.

M. Nekoui, A. Eslami, and H. Pishro-Nik, "The Capacity of Vehicular Ad Hoc Networks with Infrastructure," *2008 6th International Symposium on Modeling and Optimization in Mobile Ad Hoc and Wireless Networks and Workshops*, pp. 267–272, 2008.

S. C. Ng, W. Zhang, Y. Yang, and G. Mao, "Analysis of Access and Connectivity Probabilities in Infrastructure-Based Vehicular Relay Networks," in *2010 IEEE Wireless Communication and Networking Conference*. IEEE, Apr. 2010, pp. 1–6.

D. Nguyen, T. Tran, T. Nguyen, and B. Bose, "Wireless Broadcast Using Network Coding," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 2, pp. 914–925, 2009.

J. Ni, "Connectivity properties of a random radio network," *IEE Proceedings - Communications*, vol. 141, no. 4, p. 289, 1994.

D. Niu and B. Li, "Topological Properties Affect the Power of Network Coding in Decentralized Broadcast," *2010 Proceedings IEEE INFOCOM*, pp. 1–9, 2010.

H. Noshadi, E. Giordano, H. Hagopian, G. Pau, M. Gerla, and M. Sarrafzadeh, "Remote medical monitoring through vehicular ad hoc network," *Vehicular Technology Conference, 2008. VTC 2008-Fall. IEEE 68th*, pp. 1–5, Sep 2008.

——, "Remote Medical Monitoring Through Vehicular Ad Hoc Network," in *2008 IEEE 68th Vehicular Technology Conference*. IEEE, Sep. 2008, pp. 1–5. [Online]. Available: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4657288

M. Nozari Zarmehri and A. Aguiar, "Data Gathering for Sensing Applications in Vehicular Networks," in *2011 IEEE Vehicular Networking Conference*. Amsterdam: 2011 IEEE Vehicular Networking Conference, 2011, pp. 139–156.

M. Nozari Zarmehri and A. Aguiar, "Supporting Sensing Application in Vehicular Networks," in *ACM MobiCom Workshop on Challenged Networks*. Istanbul: ACM Press, Aug. 2012.

M. Nozari Zarmehri and C. Soares, "Improving Data Mining Results by taking Advantage of the Data Warehouse Dimensions: A Case Study in Outlier Detection," in *Encontro Nacional de Inteligência Artificial e Computacional*. São Carlos, Brazil: UFMG, LBD, 2014.

T. J. Oechtering, C. Schnurr, I. Bjelakovic, and H. Boche, "Broadcast Capacity Region of Two-Phase Bidirectional Relaying," *IEEE Transactions on Information Theory*, vol. 54, no. 1, pp. 454–458, Jan. 2008.

L. Olshen, C. J. Stone *et al.*, "Classification and regression trees," *Wadsworth International Group*, vol. 93, no. 99, p. 101, 1984.

L. Pan, H. Xiaoxia, F. Yuguang, and L. Phone, "Optimal Placement of Gateways in Vehicular Networks," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 6, pp. 3421–3430, Nov. 2007.

G. L. Pappa and A. Freitas, "Automating the Design of Data Mining Algorithms: An Evolutionary Computation Approach," Nov. 2009.

B. H. Pavel Brazdil, João Gama, "Characterizing the applicability of classification algorithms using meta-level learning," pp. 83–102, 2005. [Online]. Available: http://www.bookmetrix.com/detail/chapter/fecc6a44-eaee-4a4a-9ffe-ab0da6f41b66#downloads

Y. Peng, P. A. Flach, C. Soares, and P. Brazdil, "Improved dataset characterisation for meta-learning," in *Discovery Science*. Springer, 2002, pp. 141–152.

C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (aodv) routing," Tech. Rep., Feb 2007. [Online]. Available: http://www.ietf.org/rfc/rfc3561.txt

C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA'99. Second IEEE Workshop on*. IEEE, 1999, pp. 90–100.

B. Pfahringer, H. Bensusan, and C. Giraud-Carrier, "Meta-learning by landmarking various learning algorithms," in *In Proceedings of the Seventeenth International Conference on Machine Learning*. Morgan Kaufmann, 2000, pp. 743–750.

H. Pishro-Nik, A. Ganz, and D. Ni, "The Capacity of Vehicular Ad-hoc Networks," in *Proc. of the 45th Annual Allerton Conference*, Sep. 2007.

H. Pishro-Nik, "Vehicular Ad Hoc Networks," *EURASIP Journal on Advances in Signal Processing*, vol. 2010, pp. 1–1, 2010.

D. M. W. Powers, "Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation," School of Informatics and Engineering, Flinders University, Adelaide, Australia, Tech. Rep. SIE-07-001, 2007.

A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint relaying for flooding broadcast messages in mobile wireless networks," in *System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on*. IEEE, 2002, pp. 3866–3875.

Y. Qian and N. Moayeri, "Design Secure and Application-Oriented VANETs," *Proceedings of IEEE VTC*, vol. 11, pp. 2794–2799, 2008. [Online]. Available: http://www.antd.nist.gov/pubs/Yi-Paper7.pdf

Y. Qiong and S. Lianfeng, "A Multi-Hop Broadcast scheme for propagation of emergency messages in VANET," in *2010 IEEE 12th International Conference on Communication Technology*. IEEE, Nov. 2010, pp. 1072–1075.

R. Quinlan, "C5.0: An Informal Tutorial," 1998. [Online]. Available: http://www.rulequest.com/see5-unix.html

K. N. Qureshi and A. H. Abdullah, "A survey on intelligent transportation systems," *Middle-East Journal of Scientific Research*, vol. 15, no. 5, pp. 629–642, 2013.

R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2014. [Online]. Available: http://www.R-project.org/

——, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2014. [Online]. Available: http://www.R-project.org/

T. Rashed and C. Jürgens, *Remote Sensing of Urban and Suburban Areas*, ser. Remote Sensing and Digital Image Processing, T. Rashed and C. Jürgens, Eds. Springer, 2010, vol. 10, no. 42. [Online]. Available: http://www.springerlink.com/index/10.1007/978-1-4020-4385-7

L. Rendell and H. Cho, "Empirical learning as a function of concept character," *Machine Learning*, vol. 5, no. 3, pp. 267–298, Aug. 1990.

L. Rendell, R. Sheshu, and D. Tcheng, "Layered concept-learning and dynamically variable bias management," pp. 308–314, Aug. 1987.

J. R. Rice, "The algorithm selection problem," ser. Advances in Computers, M. Rubinoff and M. C. Yovits, Eds. Elsevier, 1976, vol. 15, pp. 65 – 118.

B. Ripley, *tree: Classification and regression trees*, 2014, r package version 1.0-35. [Online]. Available: http://CRAN.R-project.org/package=tree

G. P. Rodrigues, F. Vieira, and T. T. V. Vinhoza, "A Non-Intrusive Multi-Sensor System for Characterizing Driver Behavior," *Transportation*, pp. 1620–1624, 2010.

J. Rodrigues, F. Vieira, T. Vinhoza, J. Barros, and J. Cunha, "A non-intrusive multi-sensor system for characterizing driver behavior," Sep 2010, pp. 1620–1624.

J. Rodrigues, F. Vieira, A. Aguiar, and J. Barros, "A mobile sensing architecture for massive urban scanning," *14th International IEEE Annual Conference on Intelligent Transportation Systems*, Oct 2011.

J. G. P. Rodrigues, A. Aguiar, F. Vieira, J. Barros, and J. P. S. Cunha, "A mobile sensing architecture for massive urban scanning," in *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, Oct. 2011, pp. 1132–1137. [Online]. Available: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=6082958

U. T. Rosi, C. S. Hyder, and T.-h. Kim, "A Novel Approach for Infrastructure Deployment for VANET," in *2008 Second International Conference on Future Generation Communication and Networking*. IEEE, Dec. 2008, pp. 234–238.

A. L. D. Rossi, A. C. P. de Leon Ferreira de Carvalho, C. Soares, and B. F. de Souza, "MetaStream: A meta-learning based method for periodic algorithm selection in time-changing data," *Neurocomputing*, vol. 127, no. 0, pp. 52–64, 2014.

L. Rubio, J. Reig, and N. Cardona, "Evaluation of nakagami fading behaviour based on measurements in urban scenarios," *International Journal of Electronics and Communications*, vol. 61, pp. 135–138, Feb 2007.

——, "Evaluation of Nakagami fading behaviour based on measurements in urban scenarios," *International Journal of Electronics and Communications*, pp. 135–138, 2007.

H. Rudolf, G. Wanielik, and A. Sieber, "Road condition recognition using microwaves," in *Intelligent Transportation System, 1997. ITSC '97., IEEE Conference on*, Nov 1997, pp. 996–999.

S. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 21, no. 3, pp. 660–674, 1991.

F. D. Salim, S. W. Loke, A. Rakotonirainy, B. Srinivasan, and S. Krishnaswamy, "Collision pattern modeling and real-time collision detection at road intersections," in *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*. IEEE, 2007, pp. 161–166.

S. Sarawagi, R. Agrawal, and N. Megiddo, *Discovery-driven exploration of OLAP data cubes*. Springer, 1998.

F. Schmidt-Eisenlohr and H. Hartenstein, "Simulation-based capacity estimates for local broadcast transmissions," in *Proceedings of the seventh ACM international workshop on VehiculAr InterNETworking - VANET '10*. New York, New York, USA: ACM Press, Sep. 2010, p. 21.

B. Schölkopf and A. Smola, "Support vector machines," *Encyclopedia of Biostatistics*, 1998.

N. Schweighofer and K. Doya, "Meta-learning in reinforcement learning," *Neural Networks*, vol. 16, no. 1, pp. 5–9, 2003.

G. A. Seber and A. J. Lee, *Linear regression analysis*. John Wiley & Sons, 2012, vol. 936.

M. Sepulcre, J. Gozalvez, and H. Hartenstein, "Application-Based Congestion Control Policy for the Communication Channel in VANETs," *October*, vol. 14, no. 10, pp. 951–953, 2010.

M. Sepulcre, J. Gozalvez, J. Hä andrri, and H. Hartenstein, "Contextual communications congestion control for cooperative vehicular networks," *Wireless Communications, IEEE Transactions on*, vol. 10, no. 2, pp. 385 –389, Feb 2011.

M. Sepulcre, J. Gozalvez, J. Harri, and H. Hartenstein, "Contextual Communications Congestion Control for Cooperative Vehicular Networks," *IEEE Transactions on Wireless Communications*, vol. 10, no. 2, pp. 385–389, Feb. 2011.

S. Shakkottai, X. Liu, and R. Srikant, "The Multicast Capacity of Large Multihop Wireless Networks," *IEEE/ACM Transactions on Networking*, vol. 18, no. 6, pp. 1691–1700, Dec. 2010.

T. J. Shepard, "A channel access scheme for large dense packet radio networks," in *Conference proceedings on Applications, technologies, architectures, and protocols for computer communications - SIGCOMM '96*. New York, New York, USA: ACM Press, 1996, pp. 219–230.

W.-Y. Shieh, W.-H. Lee, S.-L. Tung, B.-S. Jeng, and C.-H. Liu, "Analysis of the Optimum Configuration of Roadside Units and Onboard Units in Dedicated Short-Range Communication Systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 4, pp. 565–571, Dec. 2006.

O. Shigehiko, "Nakagami-m fading channel," *Journal of the Institute of Electronics, Information and Communication Engineers*, vol. 86, no. 12, pp. 969–971, 2003.

M. Slavik and I. Mahgoub, "On the scalability of wireless multi-hop broadcast protocols with respect to density in VANET," in *2011 International Conference on Communications and Information Technology (ICCIT)*. IEEE, Mar. 2011, pp. 92–95.

N. Smavatkul and S. Emeott, "Voice Capacity Evaluation of IEEE 802.11a with Automatic Rate Selection," in *GLOBECOM '03. IEEE Global Telecommunications Conference (IEEE Cat. No.03CH37489)*, vol. 1. IEEE, 2003, pp. 518–522.

C. Soares, P. Brazdil, J. Costa, V. Cortez, and A. Carvalho, "Error Detection in Foreign Trade Data using Statistical and Machine Learning Algorithms," in *Proceedings of the 3rd International Conference and Exhibition on the Practical Application of Knowledge Discovery and Data Mining*, N. Mackin, Ed., London, UK, 1999, pp. 183–188.

C. Soares, P. B. Brazdil, and P. Kuba, "A meta-learning method to select the kernel width in support vector regression," *Machine learning*, vol. 54, no. 3, pp. 195–209, 2004.

S. Y. Sohn, "Meta analysis of classification algorithms for pattern recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 21, no. 11, pp. 1137–1144, Nov 1999.

F. Soulié-Fogelman, "Data Mining in the real world: What do we need and what do we have?" in *Proceedings of the KDD Workshop on Data Mining for Business Applications*, R. Ghani and C. Soares, Eds., 2006, pp. 44–48.

I. Steinwart and A. Christmann, *Support vector machines*. Springer Science & Business Media, 2008.

I. Stojanovic, Z. Wu, M. Sharif, and D. Starobinski, "Data dissemination in wireless broadcast channels: Network coding versus cooperation," *IEEE Transactions on Wireless Communications*, vol. 8, no. 4, pp. 1726–1732, 2009.

J.-C. Thill, "Geographic information systems for transportation in perspective," *Transportation Research Part C: Emerging Technologies*, vol. 8, no. 1, pp. 3–12, 2000.

S. Thrun, "Lifelong learning algorithms," pp. 181–209, May 1998.

L. Todorovski and S. Dzeroski, "Experiments in meta-level learning with ilp," in *In Proceedings of the Third European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, 1999, pp. 98–106.

L. Todorovski and S. Džeroski, "Combining classifiers with meta decision trees," *Machine learning*, vol. 50, no. 3, pp. 223–249, 2003.

O. Tonguz, N. Wisitpongphan, F. Bai, P. Mudalige, and V. Sadekar, "Broadcasting in vanet," *Mobile Networking for Vehicular Environments*, pp. 7–12, May 2007.

O. Tonguz, N. Wisitpongphan, and F. Bai, "Dv-cast: A distributed vehicular broadcast protocol for vehicular ad-hoc networks," *IEEE Wireless Communications*, 2010.

O. Tonguz, N. Wisitpongphan, J. Parikh, F. Bai, P. Mudalige, and V. Sadekar, "On the broadcast storm problem in ad hoc wireless networks," *3rd International Conference on Broadband Communications, Networks and Systems*, pp. 1–11, Oct 2006.

O. Tonguz*, N. Wisitpongphan*, F. Bait, P. Mudaliget, and V. Sadekart, "Broadcasting in VANET," in *2007 Mobile Networking for Vehicular Environments*. IEEE, 2007, pp. 7–12. [Online]. Available: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4300825

O. Tonguz, N. Wisitpongphan, and F. Bai, "DV-CAST: A distributed vehicular broadcast protocol for vehicular ad hoc networks," *Ieee Wireless Communications*, vol. 17, no. 2, pp. 47–57, 2010. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5450660

O. K. Tonguz, N. Wisitpongphan, J. S. Parikh, F. Bai, P. Mudalige, and V. K. Sadekar, "On the Broadcast Storm Problem in Ad hoc Wireless Networks," *3rd International Conference on Broadband Communications Networks and Systems*, pp. 1–11, 2006. [Online]. Available: http://www.scopus.com/inward/record.url?eid=2-s2.0-51749108008&partnerID=40&md5=849026c4bc963408a40c1f65dbed4fd6

L. Torgo, *Data Mining with R, learning with case studies*. Chapman and Hall/CRC, 2010.

L. Torgo and C. Soares, "Resource-bounded Outlier Detection Using Clustering Methods," in *Data Mining for Business Applications*, ser. Frontiers in Artificial Intelligence and Applications, R. G. Carlos Soares, Ed. IOS Press, 2010, pp. 84–98.

L. Torgo, W. Pereira, C. Soares, and C. Torgo, Luis; Pereira, Welma; Soares, "Detecting Errors in Foreign Trade Transactions: Dealing with Insufficient Data," in *EPIA '09: Proceedings of the 14th Portuguese Conference on Artificial Intelligence*, ser. Lecture Notes in Computer Science, L. S. Lopes, N. Lau, P. Mariano, and L. M. Rocha, Eds., vol. 5816. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 435–446.

M. Torrent-Moreno, J. Mittag, P. Santi, and H. Hartenstein, "Vehicle-to-Vehicle Communication: Fair Transmit Power Control for Safety-Critical Information," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 7, pp. 3684–3703, Sep. 2009.

J. van Rijn, JanN. and Holmes, Geoffrey and Pfahringer, Bernhard and Vanschoren, "Algorithm Selection on Data Streams," *Lecture Notes in Computer Science, Springer International Publishing*, vol. 8777, pp. 325–336, 2014.

R. Vilalta and Y. Drissi, "A perspective view and survey of meta-learning," *Artificial Intelligence Review*, no. 1997, pp. 77–95, 2002.

W. Viriyasitavat, F. Bai, and O. Tonguz, "Dynamics of network connectivity in urban vehicular networks," *Selected Areas in Communications, IEEE Journal on*, vol. 29, no. 3, pp. 515 –533, march 2011.

P. T. von Hippel, "Mean, median, and skew: Correcting a textbook rule," *Journal of Statistics Education*, vol. 13, no. 2, p. n2, 2005.

F.-Y. Wang, "Parallel control and management for intelligent transportation systems: Concepts, architectures, and applications," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, no. 3, pp. 630–638, 2010.

S. Wang, C. Lin, Y. Hwang, K. Tao, and C. Chou, "A practical routing protocol for vehicle-formed mobile ad hoc networks on the roads," in *Proceedings. 2005 IEEE Intelligent Transportation Systems, 2005.* IEEE, 2005, pp. 161–166. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1520040

——, "A practical routing protocol for vehicle-formed mobile ad hoc networks on the roads," *in Proceedings of the 8th IEEE International Conference on Intelligent Transportation Systems*, pp. 161–165, 2005.

N. Wisitpongphan, F. Bai, P. Mudalige, and O. K. Tonguz, "On the Routing Problem in Disconnected Vehicular Ad-hoc Networks," *IEEE INFOCOM 2007 26th IEEE International Conference on Computer Communications*, pp. 2291–2295, 2007. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4215849

N. Wisitpongphan, O. Tonguz, J. Parikh, P. Mudalige, F. Bai, and V. Sadekar, "Broadcast storm mitigation techniques in vehicular ad hoc networks," *IEEE Wireless Communications*, 2007.

——, "Broadcast storm mitigation techniques in vehicular ad hoc networks," *IEEE Wireless Communications*, vol. 14, no. 6, pp. 84–94, Dec. 2007. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4407231

D. Wolpert and W. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, Apr. 1997.

C. Wu, K. Kumekawa, and T. Kato, "A novel multi-hop broadcast protocol for vehicular safety applications," *Journal of Information Processing*, vol. 51, pp. 930–944, 2010.

C. Wu, S. Ohzahata, and T. Kato, "Fuzzy logic based multi-hop broadcast for high-density vehicular ad hoc networks," in *Vehicular Networking Conference (VNC), 2010 IEEE*. IEEE, 2010, pp. 17–24.

C.-H. Wu, J.-M. Ho, and D.-T. Lee, "Travel-time prediction with support vector regression," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 5, no. 4, pp. 276–281, 2004.

H. Wu, R. Fujimoto, R. Guensler, and M. Hunter, "Mddv: a mobility centric data dissemination algorithm for vehicular networks," *ACM international workshop on VANETs*, Oct 2004.

——, "MDDV," in *Proceedings of the first ACM workshop on Vehicular ad hoc networks - VANET '04*. New York, New York, USA: ACM Press, Oct. 2004, p. 47. [Online]. Available: http://portal.acm.org/citation.cfm?id=1023875.1023884

J. Wu and H. Li, "A dominating-set-based routing scheme in ad hoc wireless networks," *the special issue on Wireless Networks in the Telecommunication Systems Journa*, vol. 3, pp. 63–84, 2001.

H. Xiong, R. Li, A. Eryilmaz, and E. Ekici, "Delay-Aware Cross-Layer Design for Network Utility Maximization in Multi-hop Networks," *Arxiv preprint arXiv10121681*, vol. 29, no. 5, p. 14, 2010.

Y. Xu, Y. Wu, G. Wu, J. Xu, B. Liu, and L. Sun, "Data Collection for the Detection of Urban Traffic Congestion by VANETs," in *2010 IEEE Asia-Pacific Services Computing Conference*. IEEE, Dec. 2010, pp. 405–410.

K. Yamada, H. Okada, and K. Fujimura, "GPS-based message broadcasting for inter-vehicle communication," in *Proceedings 2000 International Conference on Parallel Processing*. IEEE Comput. Soc, 2002, pp. 279–286. [Online]. Available: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=876143http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=876143

T. Yamamoto, R. Kitamura, and J. Fujii, "Drivers' route choice behavior: analysis by data mining algorithms," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1807, no. 1, pp. 59–66, 2002.

M. Zambrano-Bigiarini, *hydroGOF: Goodness-of-fit functions for comparison of simulated and observed hydrological time series*, 2014, r package version 0.3-8. [Online]. Available: http://CRAN.R-project.org/package=hydroGOF

S. Zhang, "Hot topic: physical-layer network coding," *in Proc. of ACM Mobicom*, 2006.

X. Zhang and J. A. Rice, "Short-term travel time prediction," *Transportation Research Part C: Emerging Technologies*, vol. 11, no. 3–4, pp. 187–210, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0968090X03000263

——, "Short-term travel time prediction," *Transportation Research Part C: Emerging Technologies*, vol. 11, no. 3, pp. 187–210, 2003.

H. Zheng, S. Xiao, X. Wang, and X. Tian, "On the Capacity and Delay of Data Gathering with Compressive Sensing in Wireless Sensor Networks," pp. 1–5, 2011.

B. Zhou, J. Cao, X. Zeng, and H. Wu, "Adaptive traffic light control in wireless sensor network-based intelligent transportation system," in *Vehicular Technology Conference Fall (VTC 2010-Fall), 2010 IEEE 72nd*. IEEE, 2010, pp. 1–5.

J. Zhou, D. Gao, and D. Zhang, "Moving vehicle detection for automatic traffic monitoring," *Vehicular Technology, IEEE Transactions on*, vol. 56, no. 1, pp. 51–59, 2007.

X. Zhou, J. Huang, R. Jing, and D. Li, "Fuel consumption estimate based on meso traffic characteristics and typical road environment," in *ITS Telecommunications (ITST), 2013 13th International Conference on*, Nov 2013, pp. 169–174.

M. Zorzi and R. Rao, "Geographic random forwarding (geraf) for ad hoc and sensor networks: multihop performance," *IEEE Transactions on Mobile Computing*, vol. 2, no. 4, pp. 337–348, Oct. 2003.

Appendix A

Description of the Metafeatures

The detail description about the metafeatures for INTRASTAT dataset (Section A) and taxi dataset (Section A) is presented in this Appendix.

Metafeatures for INTRASTAT Dataset

Totally, fifteen metafeatures are calculated for INTRASTAT dataset which are described in Table A.1.

Table A.1: Extracted features used in metalearning - INTRASTAT dataset

Feature Name	**Description**
n.examples	Number of examples
n.attrs	Number of attributes
prop.symbolic.attrs	Proportion of symbolic attributes
prop.missing.values	Proportion of missing values
class.entropy	Class entropy
avg.mutual.information	Average mutual information
prop.h.outlier	Proportion of continuous attributes with outliers
avg.attr.entropy	Average attribute entropy
avg.symb.pair.mutual.infor	Average mutual information between pairs of symbolic attributes
avg.abs.attr.correlation	Average absolute correlation between continuous attributes
avg.skewness	Mean skewness of attributes
avg.abs.skewness	Mean absolute skewness of attributes
avg.kurtosis	Mean kurtosis of attributes
canonical.correlation.best.linear.combination	Canonical correlation of the best linear combination of attributes to distinguish between classes
relative.prop.best.linear.combination	Proportion of the total discrimination power explained by the best linear combination

Metafeatures for Taxi Dataset

The description of 31 metafeatures which are calculated for creating metadata for taxi dataset are presented in Table A.2.

Table A.2: The description of metafeatures used for metalearning - Taxi dataset

Feature	**Feature's description**
1	Number of examples
2	$log(10)$ of the number of examples
3	Number of attributes
4	Ratio of number of examples by number of attributes
5	$log(10)$ of the ratio of number of examples by number of attributes
6	Number of continuous attributes
7	Number of symbolic attributes
8	Number of binary attributes
9	Proportion of continuous attributes
10	Proportion of symbolic attributes
11	Proportion of binary attributes
12	Correlation between continuous attributes
13	Average absolute correlation between continuous attributes
14	Minimum absolute correlation between continuous attributes
15	Maximum absolute correlation between continuous attributes
16	The ratio between the standard deviation and the standard deviation of alpha trimmed mean
17	Number of continuous attributes with outliers
18	Proportion of continuous attributes with outliers
19	Correlation matrix between attributes and target
20	Average correlation continuous attribute/target
21	Minimum correlation continuous attribute/target
22	Maximum correlation continuous attribute/target
23	Check if standard deviation is larger than mean
24	Ratio of the standard deviation and the mean of the target attribute
25	Sparsity based on the coefficient of variation
26	Sparsity based on the absolute coefficient of variation
27	Standard deviation of the proportions of a histogram with 100 bins of target values
28	textith.outlier value, as calculated for the continuous attributes
29	Outlier detection based on the notion of outliers used for continuous attributes
30	Mean distance between each target value and its two neighbors (sorted by value)
31	Average mean distance between each target value and its two neighbors (sorted by value)