Thesis National Number : 2024ENTP0007

A THESIS OF ENTPE
Member of the Université de Lyon

Doctoral school N° 162
Ecole Doctorale MEGA - Mécanique, énergétique, génie civil, acoustique
To obtain the graduation of
PhD in Mobility and network engineering

Defended on Mai 16th 2024 by:
BENZAGOUTA Mohamed Lamine

Vehicular networks data for traffic management

In front of the following examination committee:

GAITI, Dominique	Professor	UTT	Committee chair
RENAULT, Eric	Professor	ESIEE	Reviewer
ESSEGHIR, Moez	HDR	UTT, MCF	Reviewer
EL HILLALI, Yassin	Professor	UPHF	Examiner
EL FAOUZI, Nour-Eddin	Research director	UGE	Director
FOUCHAL, Hacène	Professor	URCA	Co-director
ANISS, Hasnaa	Dr, Research engineer,	UGE	Supervisor

Numéro national de thèse (NNT) : 2024ENTP0007

THÈSE DE DOCTORAT DE L'ENTPE
Membre de l'Université de Lyon

École Doctorale N° accréditation 162
Ecole Doctorale MEGA - Mécanique, énergétique, génie civil, acoustique

Spécialité / discipline de doctorat : Ingénierie de la mobilité et des réseaux

Soutenue publiquement le 16/05/2024, par :
BENZAGOUTA Mohamed Lamine

Données des réseaux véhiculaires pour la gestion du trafic

Devant le jury composé de :

GAITI, Dominique	Professeur	UTT	Président
RENAULT, Eric	Professeur	ESIEE	Rapporteur
ESSEGHIR, Moez	HDR	UTT, MCF	Rapporteur
EL HILLALI, Yassin	Professeur	UPHF	Examinateur
EL FAOUZI, Nour-Eddin	Directeur de recherche	UGE	Directeur de thèse
FOUCHAL, Hacène	Professeur	URCA	Co-directeur de thèse
ANISS, Hasnaâ	Dr, Ingénieur de recherche	UGE	Encadrante

"Everything we hear is an opinion, not a fact. Everything we see is a perspective, not the truth."

Marcus Aurellius

Abstract

The field of modern transportation is changing dramatically as vehicle communications continue to develop and evolve. The integration of advanced communication technologies is reshaping the way vehicles interact with each other and with the surrounding infrastructure. The dynamic interchange of real-time information between vehicles and the transportation infrastructure characterizes this progression, which is categorized under the umbrella of Cooperative Intelligent Transport Systems (C-ITS).

Actors of the C-ITS generate various amounts of data from communications achieved through vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications schemes. Useful information on various issues such as anomalies, failures, road profiles, etc., could be revealed from the analysis of these data. The analysis, could be managed by operators and vehicles, and its output could be very helpful for future decision making.

In this thesis, a data analysis work was done on two sorts of data sources. The first is made up of PCAP files that were taken from a test car and provide vital information on vehicle communication. The second source consists of information collected via the CoopITS smartphone application which is an application that simulates C-ITS, where a smartphone plays the role of a C-ITS station.

In our first contribution, we analyzed the PCAP data to confirm if any unusual behaviors were observable in the data. The primary aim was a particularly sensitive C-ITS failure,which is transmission failure of a road-side unit (RSU). End users (vehicles) are responsible for detecting such failures, after which they notify road operators, who then recover the problem.

In our second contribution, we utilized data extracted from the CoopITS application. Our focus revolved around two particular recorded actions by the application: the transmission of a CAM message and Traffic Light Management (TLM) actions. We conducted a matching process between CAMs and TLM records, utilizing the results to examine vehicle behavior at signalized intersections. This analysis led to the observation and classification of four distinct patterns.

In our third contribution, we examined data within CAMs sourced from several participants utilizing the CoopITS application to establish road profiles. The trajectories of individual vehicles undergo classification into diverse classes employing four distinct algorithms: K-means, Agglomerative Clustering, DBSCAN, and BIRCH. At the end, the results of these clustering algorithm are then compared.

Keywords: V2X, V2V, V2I, C-ITS, ETSI, Anomaly detection, Intelligent Transportation Systems, RSU, GLOSA.

Résumé court

Le domaine des transports modernes subit une transformation significative à mesure que les communications entre véhicules continuent de se développer. L'incorporation de technologies de communication avancées transforme la manière dont les véhicules interagissent entre eux et avec l'infrastructure environnante. Cette avancée est caractérisée par l'échange dynamique d'informations en temps réel entre les véhicules et l'infrastructure de transport, laquelle est catégorisée sous le terme de systèmes de transport intelligents coopératifs (C-ITS).

Les acteurs du C-ITS génèrent diverses quantités de données par le biais des communications effectuées dans le cadre des systèmes de communication de véhicule à véhicule (V2V) et de véhicule à infrastructure (V2I). L'examen de ces données peut fournir des informations utiles sur divers sujets, notamment les anomalies, les défaillances, les profils routiers, etc. L'analyse peut être gérée par les opérateurs et les véhicules, et les résultats peuvent être très utiles pour prendre des décisions futures.

Dans cette thèse, un travail d'analyse des données a été effectué sur deux types de sources de données. La première est constituée de fichiers PCAP provenant d'une voiture d'essai et fournissant des informations essentielles sur la communication du véhicule. La seconde source est constituée d'informations collectées via l'application CoopITS pour smartphone, qui est une application simulant le C-ITS, où un smartphone joue le rôle d'une station C-ITS.

Dans notre première contribution, nous avons analysé les données PCAP pour déterminer si des comportements anormaux étaient observable dans les données. L'objectif principal était de cibler une défaillance particulièrement critique du C-ITS, à savoir la défaillance de transmission d'une unité de bord de route (RSU). Les utilisateurs finaux (véhicules) sont chargés de détecter de telles défaillances, après quoi ils informent les opérateurs routiers, qui prennent ensuite les mesures nécessaires pour résoudre le problème.

Dans notre deuxième contribution, nous avons utilisé des données extraites

de l'application CoopITS. Nous nous sommes concentrés sur deux actions particulières enregistrées par l'application : la transmission d'un message CAM et les actions du gestionnaire des feux de circulation (TLM). Nous avons mené un processus de correspondance entre les messages CAM et les enregistrements TLM, en utilisant les résultats pour examiner le comportement des véhicules aux intersections signalisées. Cette analyse a conduit à l'observation et à la classification de quatre patterns distincts.

Dans notre troisième contribution, nous avons examiné les données des CAMs provenant de plusieurs participants utilisant l'application CoopITS pour établir des profils routiers. Les trajectoires des véhicules individuels sont classées en diverses catégories à l'aide de quatre algorithmes distincts : K-means, Agglomerative Clustering, DBSCAN, et BIRCH. À la fin, les résultats de ces algorithmes de clustering sont comparés.

Mots clés: V2X, V2V, V2I, C-ITS, ETSI, Anomaly detection, Intelligent Transportation Systems, RSU, GLOSA.

Acknowledgements

I would like to take the opportunity to thank all those who helped me during my thesis' process.

First, I would like to thank my supervisors, I thank Pr Hacene FOUCHAL and Hasnaâ ANISS for their moral support, encouragement, and their meticulous following of my work's progress, and I thank Pr Nour-Eddin EL FAOUZI for his valuable advices. My supervisors were the key to the successful completion of my thesis.

Second, I would like to thank the reviewers for accepting to evaluate my thesis, and I would also like to thank the examinators for accepting to be part of the jury.

Third, I would like to thank all my colleagues at ERENA, I would like to thank Mr HADHELI AbdelMename for his help in administrative and organizational matters, I thank Romain DULOUT, Léo MENDIBOURE and Imed GHNAYA for the moral uplifting attitude and the interesting discussions we always had. And I would thank Sassi MAALOUL for his help in scientific matters. And I also pay special attention to all my friends and university colleagues

In the end, I would like to say that I am humbly grateful towards both my parents, and very thankful towards my brother and sister. My family has always been a motivation, and a source of continuous support and encouragement.

Contents

Abstract iii

Acknowledgements vii

List of Figures xiii

List of Tables xvii

Abbreviations xix

Synthèse en français xxi
0.1 Introduction . . . xxi
0.1.1 Cooperative Intelligent Transport Systems . . . xxii
0.1.2 Application CoopITS . . . xxii
0.1.3 GLOSA . . . xxiii
0.2 Synthèse des chapitres . . . xxiii
0.3 Conclusion . . . xxv

1 Introduction 1
1.1 Context . . . 1
1.1.1 Communication Technologies . . . 2
1.1.2 Vehicular applications . . . 3
1.1.3 Cooperative Intelligent Transport Systems . . . 6
1.1.4 Communication standards . . . 8
1.2 CoopITS application . . . 10
1.3 Contributions . . . 11
1.4 Outline . . . 13

2 State of the art 15
2.1 Introduction . . . 15
2.2 Cooperative Intelligent Transport Systems . . . 16
2.3 Clustering Algorithms . . . 18
2.3.1 Hierarchical-Based Algorithms . . . 19
2.3.1.1 Agglomerative clustering . . . 20
2.3.1.2 BIRCH Clustering . . . 20
2.3.1.3 Divisive hierarchical clustering . . . 20

2.3.2 Partitioning-Based Algorithms . . . 21
2.3.2.1 k-means . . . 21
2.3.2.2 K-Medoids . . . 22
2.3.3 Density-based clustering algorithms . . . 22
2.3.3.1 DBSCAN Clustering . . . 22
2.3.3.2 DENCLUE Clustering . . . 23
2.3.3.3 OPTICS Clustering . . . 23
2.4 Anomaly detection on complex systems . . . 24
2.4.1 Analysis . . . 27
2.5 Traffic prediction on intersections . . . 27
2.5.1 Analysis . . . 29
2.6 Profiling over C-ITS . . . 29
2.6.1 Analysis . . . 31

3 Signal/Range Correlation and RSU Failure Detection 33
3.1 Motivation . . . 33
3.2 Antenna theory . . . 34
3.2.1 Bases of electromagnetism . . . 34
3.2.2 Antennas radiation patterns . . . 36
3.3 Study environment . . . 38
3.3.1 RSU packets characterization . . . 39
3.3.2 RSU coverage . . . 39
3.4 RSU failure detection . . . 40
3.4.1 Alarm message . . . 44
3.5 Conclusion . . . 49

4 Road profiles 51
4.0.1 Introduction . . . 51
4.1 Data visualisation using T-SNE . . . 51
4.2 Trajectory analysis . . . 52
4.2.1 k-means analysis . . . 54
4.2.2 Agglomerative clustering analysis . . . 56
4.2.3 BIRCH Clustering analysis . . . 58
4.2.4 DBSCAN Clustering analysis . . . 60
4.2.5 Combination of four algorithms experimentation . . . 62
4.2.5.1 Combination of Agglomerative Clustering and K-means . . . 62
4.2.5.2 Combination of BIRCH and Agglomerative Clustering . . . 63
4.2.5.3 Combination of BIRCH and Dbscan . . . 64
4.2.5.4 Combination of BIRCH and K-means . . . 65
4.2.5.5 Combination of Agglomerative Clustering and Dbscan . . . 65
4.2.5.6 Combination of Dbscan and K-means . . . 66
4.3 Conclusion . . . 67

5 Impact of speed limitation on traffic **69**
5.1 GLOSA 69
5.2 Smartphone data 70
5.3 Analysis 72
5.4 Traffic jam prediction using a single vehicle in a stop 80
5.4.1 Traffic jam prediction using a moving vehicle 82
5.5 Conclusion 84

6 Conclusion **85**

Bibliography **87**

List of Figures

1.1 A General Scheme for Vehicle-to-Vehicle Communications. 4
1.2 A General Scheme for Vehicle-to-Infrastructure Communications. . 5
1.3 The C-ITS protocol stack. 7
1.4 Comparison between ITS-G5 and C-V2X. 9
1.5 The architecture of the CoopITS application. 10
1.6 Screen shots of the CoopITS application, source : official playstore home page. 11

3.1 Radiation diagram for an isotropic antenna : a- vertical plane, b- horizontal plane. 36
3.2 Radiation diagram for an omnidirectional antenna : a- vertical plane, b- horizontal plane. 37
3.3 Radiation diagram for a directional antenna : a- vertical plane, b- horizontal plane. 37
3.4 The path that the test vehicle traveled in the city of Reims, France 38
3.5 The scheme of the correlation between signal and distance for a normal RSU in Reims 42
3.6 The scheme of the correlation between signal and distance for a normal RSU in Saint Maurice, Paris 42
3.7 The scheme of the correlation between signal and distance for a failing RSU in Reims 43
3.8 Clustering using range, signal, speed, and Pearson's coefficient in function of range and signal with normalized values 44
3.9 The structure of the alarm packet 45
3.10 Treatment of the alarm message by the vehicles, a) at the coverage of an RSU, the vehicle starts building the alarm message, b) the vehicle finishes building the alarm message after leaving the coverage of an RSU, c) the message is broadcast in multi-hop fashion by the vehicles to inform each other of the status of the RSU, d) the vehicle sends the alarm message to the next RSU it encounters . . 50

4.1 An example of the trajectory that was extracted from CAM messages, plotted on top of an OSM map 52
4.2 The determined segments, plotted on Google maps 53
4.3 The rectangle over segments, plotted on an OSM map 53
4.4 An example of a normalized trajectory points (60 points) 54
4.5 Samples of trajectories from cluster 0 for the kmeans method. . . . 55
4.6 Samples of trajectories from cluster 1 for the kmeans method. . . . 55

4.7 Samples of trajectories from cluster 2 for the kmeans method. . . . 55
4.8 Samples of trajectories from cluster 3 for the kmeans method. . . . 55
4.9 Samples of trajectories from cluster 4 for the kmeans method. . . . 56
4.10 K-means clusters visualized using T-SNE. 56
4.11 Samples of trajectories from cluster 0 for the Agglomerative clustering method. 57
4.12 Samples of trajectories from cluster 1 for the Agglomerative clustering method. 57
4.13 Samples of trajectories from cluster 2 for the Agglomerative clustering method. 57
4.14 Samples of trajectories from cluster 3 for the Agglomerative clustering method. 58
4.15 Samples of trajectories from cluster 4 for the Agglomerative clustering method. 58
4.16 Agglomerative Clustering clusters visualized using T-SNE. 58
4.17 Samples of trajectories from cluster 0 for the BIRCH clustering method. 59
4.18 Samples of trajectories from cluster 1 for the BIRCH clustering method. 59
4.19 Samples of trajectories from cluster 2 for the BIRCH clustering method. 59
4.20 Samples of trajectories from cluster 3 for the BIRCH clustering method. 60
4.21 Samples of trajectories from cluster 4 for the BIRCH clustering method. 60
4.22 BIRCH clusters visualized using T-SNE. 60
4.23 DBSCAN clusters visualized using T-SNE 61
4.24 Samples of trajectories from clusters for the Dbscan clustering method : a) aberrant observation, b) cluster 0, c) another sample from cluster 0, d) cluster 1, e) cluster 2, f) cluster 3, g) cluster 4, h) cluster 5. 62

5.1 An example of the trajectory that was extracted from CAM messages, plotted on top of an OSM map 71
5.2 Trajectory data in the segment that was chosen, plotted on top of an OSM map 72
5.3 An example of some TLM records 72
5.4 A trajectory with free flow at a green light (a) velocity in function of the distance (b) velocity in function of time 74
5.5 A trajectory with free flow at a green light (a) velocity in function of the distance (b) velocity in function of time 75
5.6 A trajectory with a stop at a green light (a) velocity in function of the distance (b) velocity in function of time 76
5.7 A trajectory with a stop at a green light (a) velocity in function of the distance (b) velocity in function of time 77
5.8 A trajectory with free flow at a red light (a) velocity in function of the distance (b) velocity in function of time 78

5.9 A trajectory with free flow at a red light (a) velocity in function of the distance (b) velocity in function of time 79
5.10 A trajectory with a stop at a red light (a) velocity in function of the distance (b) velocity in function of time 80
5.11 A trajectory with a stop at a red light (a) velocity in function of the distance (b) velocity in function of time 81
5.12 A trajectory with combined pattern, velocity in function of the distance 82

List of Tables

3.1 Recorded communications with the RSUs 44

4.1 Kmeans: number of trips for each cluster 54
4.2 Agglomerative clustering: number of trips for each cluster 57
4.3 BIRCH: number of trips for each cluster 59
4.4 DBSCAN: number of trips for each cluster 61
4.5 Comparison between the AC and k-means 63
4.6 Comparison between the BIRCH and AC 64
4.7 Comparison between the BIRCH and DBSCAN 64
4.8 Comparison between the BIRCH and K-means 65
4.9 Comparison between the AC and Dbscan 66
4.10 Comparison between the DBSCAN and K-means 67

5.1 Recorded communications with the RSUs 83

Abbreviations

V2X	Vehicle-to-Everything communications
V2I	Vehicle-to-Infrastructure communications
V2V	Vehicle-to-Vehicle communications
V2P	Vehicle-to-Pedestrian
ICT	information and communication technologies
IoT	Internet of Things
M2M	machine-to-machine
SAE	Society of Automotive Engineers
C-ITS	Cooperative Intelligent Transport Systems
GPS	global positioning systems
CA	Cooperative Awareness
CAM	Cooperative Awareness Message
DEN	Decentralized Environmental Notifications
DENM	Decentralized Environmental Notifications Message
RSU	Road Side Unit
TLM	Traffic Light Maneuver
SPATEM	Signal Phase and Timing Message
RLT	Road and Lane Topology
MAPEM	MAP (topology) Extended Messages
IVI	Infrastructure to Vehicle Information
IVIM	Infrastructure to Vehicle Information Message
C-V2X	Cellular Vehicle-to-Everything
V2N	vehicle-to-network
V2P	vehicle-to-vulnerable user (pedestrian, cyclist)

GLOSA	Green Light Optimal Speed Advisory
SOA	Service-Oriented Architecture
ITS	Intelligent Transport Systems
WSN	Wireless Sensor Networks
SDN	Software Defined Networking
PC	Pseudonym Certificate
CPC	Common Pseudonym Certificate
GAN	Generative Adversarial Networks
AGNESS	Agglomerative Nesting
AC	Agglomerative Clustering
BIRCH	Balanced Iterative Reducing and Clustering using Hierarchies
CF	Clustering Feature
WCSS	within-cluster sum of squares
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DENCLUE	DENsity-based CLUstEring
OPTICS	Ordering Points To Identify the Clustering Structure
LOS	line-of-sight
NLOS	non line of sight
ID	in-distance
OD	out-distance
VEC	vehicular edge computing
MNF	Mobile phone Network
CV	Connected Vehicles
MLE	maximum likelihood estimator
EM	Expectation Maximization
FCD	Floating Car Data
GNSS	Global Navigation Satellite Systems
POI	Point of Interest
OBU	On-Board Unit
T-SNE	t-distributed Stochastic Neighbor Embedding

Synthèse en français

0.1 Introduction

Les villes intelligentes représentent une réponse révolutionnaire aux défis urbains, intégrant les technologies de l'information et de la communication (TIC) dans tous les aspects de l'infrastructure et des services. Axées sur des problèmes écologiques, sociales et économiques, elles s'efforcent de créer des écosystèmes urbains durables, connectés et résilients, améliorant ainsi la gestion des ressources et la qualité de vie des citoyens.

Les technologies de communication, allant des réseaux sans fil à la 5G et à l'IoT, font partie intégrante de la révolution des villes intelligentes, remodelant l'urbanisation en fournissant des données en temps réel pour les systèmes intelligents. Ces innovations améliorent les fonctions urbaines, telles que la gestion du trafic rendant les villes réactives aux besoins des habitants. Malgré le potentiel, des défis tels que la protection de la vie privée, la cybersécurité et l'accès équitable doivent être relevés avec soin.Diverses technologies de communication sans fil, notamment le Wi-Fi, les réseaux cellulaires, Bluetooth, NFC, RFID, Zigbee, Z-Wave, la communication par satellite et le Li-Fi, jouent un rôle crucial dans cette intégration.

Les communications entre véhicules, manifestées dans les technologies Vehicle-to-Everything (V2X), transforment l'expérience de la route en permettant aux véhicules de communiquer entre eux et avec l'infrastructure routière. Le V2X sert de moteur à la mobilité intelligente, offrant des avantages en matière de sécurité, d'efficacité du trafic, de réduction des émissions, d'atténuation des embouteillages et de préparation aux véhicules autonomes. Ces systèmes sans fil, y compris

les communications de véhicule à véhicule (V2V) et de véhicule à infrastructure (V2I), créent un réseau sophistiqué pour l'échange d'informations en temps réel, la prévention des collisions, l'optimisation du flux de trafic et la mise en place des bases de la future conduite autonome.

0.1.1 Cooperative Intelligent Transport Systems

Les systèmes de transport intelligents coopératifs (C-ITS) représentent une avancée transformatrice dans le domaine du transport, en favorisant la coopération entre les véhicules, l'infrastructure routière, et le réseau routier dans son ensemble. En tirant parti de la connectivité, les C-ITSs créent un écosystème dans lequel les véhicules interagissent intelligemment pour améliorer la sécurité routière, optimiser le trafic, réduire les émissions et améliorer l'expérience de la conduite. Les technologies clés, notamment la connectivité sans fil, les capteurs, le GPS et les algorithmes de traitement des données, permettent aux véhicules de partager des informations vitales sur l'état des routes et les incidents. Les C-ITS servent des objectifs essentiels tels que l'amélioration de la sécurité routière, l'amélioration de l'efficacité des transports, la gestion du trafic et la réduction de l'impact sur l'environnement, en s'alignant sur les applications critiques qui nécessitent une faible latence et une grande fiabilité de communication. L'accès à des informations en temps réel et à diverses sources d'information est crucial pour le bon fonctionnement des C-ITS.

0.1.2 Application CoopITS

CRoads est l'un des projets d'initiative commune des États membres de l'Union européenne qui vise à développer des solutions C-ITS innovantes. L'architecture de CRoads est très riche et comprend à la fois des communications à longue et courte portée. Dans le cadre du projet, l'application CoopITS a été développée, dans laquelle un smartphone joue le rôle d'une station (un véhicule). Elle permet l'envoi et la réception de messages C-ITS en utilisant le réseau cellulaire. L'application a été lancée en janvier 2021 et a fonctionné principalement dans la région de la Nouvelle-Aquitaine en France.

L'application enregistre tous les messages C-ITS qui sont stockés, tels que les CAM, DENM, SPATEM, MAPEM et IVIM, qui sont enregistrés dans leur état brut codé en ASN.1 UPER, puis sont décodés et stockés dans une base de données.

0.1.3 GLOSA

Green Light Optimal Speed Advice (GLOSA) est un cas d'usage du C-ITS. Il concerne les intersections équipées d'un feu de circulation et indique au conducteur la phase en cours, le temps de la phase suivante ainsi qu'un avis de vitesse qui permet au conducteur d'éviter de s'arrêter en cas de feu rouge. Il est basé sur deux services C-ITS [1] qui sont le Traffic Light Management (TLM) et le Road and Lane Topology (RLT). Le TLM envoie des informations de sécurité aux véhicules se trouvant à proximité d'un carrefour, les informant de l'état en temps réel du feu de circulation et de son état futur, ainsi que de la marge de temps entre les deux.

0.2 Synthèse des chapitres

Le Chapitre 2 donne un aperçu de l'état de l'art des travaux de recherches sur les C-ITS. Il explore les travaux académiques importants qui ont grandement accéléré le développement des C-ITS. Le chapitre se concentre ensuite sur les méthodes d'apprentissage automatique, en particulier celles liées aux algorithmes de classification. Les travaux relatifs à la détection des anomalies dans les systèmes complexes sont ensuite examinés, en mettant l'accent sur l'identification des anomalies dans les RSU. Ensuite, les efforts consacrés au domaine de la prédiction du trafic aux intersections sont présentés. Ensuite, le chapitre présente quelques travaux concernant le profilage dans le cadre du C-ITS, en se concentrant sur les travaux concernant les profils des conducteurs. L'objectif de ce chapitre est d'offrir une vue d'ensemble de l'état de l'art influençant la thèse et la recherche sur les C-ITS et les sujets connexes.

Dans le chapitre 3, la détection des défaillances des RSU opérationnels par les utilisateurs finaux est étudiée. Cette problématique est critique et cruciale pour les C-ITS. Nous avons montré que la maintenance des RSUs pouvait être

réalisée de manière simple, sans investissement supplémentaire ni équipement. Les véhicules recueillent les observations des RSU pendant leurs trajets et les analysent à l'aide de notre mécanisme de détection des défaillances. Ce mécanisme compare le comportement observé au comportement correct connu, et lorsqu'une anomalie est détectée, les véhicules envoient des alertes aux opérateurs routiers afin de les informer de l'existence d'une RSU suspecte. Les profils de comportement des RSU sont également étudiés et une différence entre les zones urbaines et interurbaines est établie et soigneusement examinée dans cette étude.

Nous avons utilisé des données constitué de fichiers PCAP provenant d'un véhicule d'essai. Le véhicule a suivi plusieurs trajectoires en France, en Allemagne et en Italie et est entré en contact avec un total de 23 RSU en ITS-G5. Nous avons observé que, dans le cas d'une RSU qui fonctionne, il existe une corrélation inverse acceptable entre la portée et l'intensité du signal de ses transmissions. Nous avons également observé que, dans le contexte interurbain avec des communications en visibilité directe, la distance de communications en entrée dans la couverture a tendance à être plus grande que celle de la sortie pour une RSU fonctionnelle. Ces deux propriétés ont été utilisées pour notre évaluation des RSU.

Chapitre 4 représente un travail effectué sur la base de données de l'application smartphone CoopITS. Les profils routiers sont dérivés en observant et en analysant les informations contenues dans les messages CAM envoyés par les différents utilisateurs. Ensuite, quatre algorithmes de clustering sont appliquées aux trajectoires des voitures individuelles. Un profil de route est défini par la vitesse du véhicule en fonction du temps.

La première étape de notre processus consiste à utiliser les profils de route pour suivre la vitesse de chaque véhicule sur un certain segment de route. Ensuite, nous classons les trajectoires en utilisant des méthodes de clustering et en les organisant en fonction de caractéristiques et de modèles communs. La méthode de clustering améliore la compréhension de l'ensemble de l'analyse en facilitant une compréhension plus approfondie des nombreux schèma de mouvement inclus dans les données routières. Les résultats des algorithmes de clustering sont ensuite comparés. Afin d'être proactifs, nous émettons des alertes lorsque nous détectons des trajectoires anormales. Cela nous permet d'avertir rapidement les usagers et

l'opérateur routier de toute anomalie potentielle dans le mouvement des véhicules.

Chapitre 5 concerne un travail effectué sur le service TLM dans le système C-ITS. Les données utilisées appartiennent à la base de données contenant les logs de l'application CoopITS. L'application conserve les messages du C-ITS et toutes les actions (y compris TLM) qui y sont enregistrées. Dans notre cas, les messages CAM ont été utilisés pour extraire les trajectoires des véhicules, les trajectoires ont ensuite été analysées et découpées en segments de route, et un segment avec des feux de circulation a été choisi pour l'étude. Nous avons ensuite observé les variations de vitesse en fonction du temps et de la distance sur le segment, selon l'état des feux de circulation. Nous avons classé les observations en quatres catégories, la première étant un feu vert avec un trafic fluide, la deuxième étant un feu vert avec un arrêt dû à une file d'attente, la troisième étant un arrêt dû à un feu rouge et la dernière étant un feu rouge sans arrêt dû au respect de l'avis de vitesse donné par le service GLOSA.

0.3 Conclusion

Cette thèse explore le domaine des C-ITS, en se concentrant dans l'exploration de l'état de l'art sur les travaux sur la C-ITS, la détection des anomalies, les algorithmes de clustering, les profils des conducteurs, le clustering des trajectoires et les méthodologies de prédiction du trafic. Elle utilise l'analyse de données pour analyser deux ensembles de données : Des fichiers PCAP provenant de véhicules d'essai en France, en Allemagne et en Italie, et des données provenant de l'application smart phone CoopITS en France (Nouvelle Aquitaine). Les travaux de la thèse se concentrent sur la détection d'anomalies pour les unités de bord de route (RSU), en définissant deux profils : les RSU en fonctionnement et les RSU en panne.Le deuxième travail essaye de classifier différents trajets du même chemin avec quatre algorithmes de clustering, et les compare. Le troisième travail explore les méthodologies de prédiction du trafic concernant les intersections routières, révélant quatre modèles de conduite distincts.

Chapter 1

Introduction

1.1 Context

The emergence of the concept of smart cities means a revolution in today's urban environment, they provide an advanced technological solution to the challenges that are brought by the increasing urbanization worldwide. These technologically future oriented cities go beyond the actual limits of urban planning by using Information and Communication Technologies (ICT) into every facet of urban infrastructure and services.

At the crossroad of ecological, social, and economic requirements, the smart cities' goal is to create more resilient, connected, and sustainable urban environments. In this process of transformation of cities, the judicious use of sensors, networks, and big data facilitates the way for the upcoming of more efficient resource management, a significant improvement in citizens' quality of life, and optimized municipal services.

This technological revolution centers around a particular technological advancement which is connectivity. Ubiquitous connectivity enables data exchange between the various urban actors such as citizens, vehicles, infrastructure, and institutions. From intelligent transport systems to waste management, to energy saving, and improving utilities, data is fueling more informed and thus more intelligent decision making processes and greater cooperation (coordination) between the various urban actors.

Furthermore, smart cities prioritize also environmental sustainability. They seek to reduce the ecological footprint. Advanced strategies such as smart energy management, efficient resource usage and the adoption of more renewable and environmental friendly sources of energy, all contribute to the creation of

environmentally friendly cities and environment.

In a futuristic scenario, and in terms of managing public affairs, the citizen is involved at the heart of a smart city concept. Citizens would be involved in the usage of participatory technologies (which encourage the participation of actors) and digital platforms, which encourages the cooperation between residents and local authorities, enabling a more efficient and transparent decision making mechanisms and adaptation to changing need of the population.

With the emergence of automation, a decisive step into the evolution of smart cities has been marked. It defines a new era where technology redefines the way we design and live in our cities. That is because automation is gradually playing the role of a catalyser in the development of our urban environment.

As technologies evolve, automation becomes an enabler for creating more responsive and smart cities. In our case Automated and connected cars are revolutionizing the road transportation niche. Other automation centered technologies contribute to proactive management of city resources, more efficient services, and reduced environmental impact.

1.1.1 Communication Technologies

As mentioned before, communication technologies are playing a crucial role in the development of smart cities. From WiFi adhoc networks to 5G and from connected sensors to the internet of things (IoT), these technologies are reshaping the way our cities work and evolve. Future cities draw their current dynamism from these technologies.

Smart sensors and the IoT [2] are integrating into the urban environment, providing a multitude of real-time data that feeds intelligent systems for traffic management, adaptive street lighting, waste collection and many other services, creating a city responsive to the changing needs of its inhabitants.

However, the integration of these technologies is not without its challenges. Issues of privacy, cybersecurity, and equitable access to digital services need to be carefully addressed if smart cities are to be truly inclusive and beneficial to all.

There are a variety of wireless communication technologies that exist. Below is a summary of some of the most commonly used:

1. **Wi-Fi:** For "Wireless Fidelity", and based on the IEEE 802.11 standard (802.11a, 802.11g, 802.11n, 802.11ac, and more), Wi-Fi enables wireless connection to the Internet and other local networks. It is widely used in homes, businesses, public spaces and mobile devices (LAN networks).

2. **Cellular Networks (3G, 4G, 5G):** Cellular networks, based on 3GPP standards, such as 3G, 4G and 5G, enable long-distance wireless communications. They are essential for mobile communications and high-speed Internet access.

3. **Bluetooth:** This short range wireless technology is used for data transfer between devices such as smartphones, headphones, speakers and electronic peripherals.

4. **NFC (Near Field Communication):** NFC enables the exchange of information between compatible devices over very short distances. It is often used for contactless transactions, information sharing between smartphones, etc.

5. **RFID (Radio-Frequency Identification):** RFID systems use electronic tags to store and retrieve data remotely. They are used in inventory management, parcel tracking, access cards, etc.

6. **Zigbee :** is a wireless communication protocol developed to enable low-cost, low power machine-to-machine (M2M) communications. It is usually used in home automation by enabling sensor networks and IoT.

7. **Z-Wave:** Similar to Zigbee, Z-Wave is a wireless communication protocol designed for home automation applications.

8. **Satellite Communication:** Satellite communication technologies enable the transmission of data over long distances, notably for global communications, satellite navigation, and satellite TV services.

9. **Li-Fi (Light Fidelity):** Li-Fi uses visible light to transmit high-speed data.

1.1.2 Vehicular applications

Vehicular communications are becoming an essential aspect of our road experience. The communications are grouped under the acronym Vehicle-to-Everything (V2X), and are setting novel ways where the road actors interact with each other. Actors such as vehicles, the infrastructure and even vulnerable users such as pedestrians and cyclists. In this context, V2X applications are becoming a catalyst to the evolution of our road experience. They bring significant benefits in terms of safety, traffic efficiency, reduced emissions, reduced congestion, and preparation for the coming of automated vehicles.

Vehicular communications use a wide range of wireless connectivity technologies that enable road actors to share crucial information between them in real time. This information is used for a variety of purposes as stated before, such as to warn the actors of road events such as a coming danger, to coordinate vehicles and vulnerable users movements through an intersection or simply to optimize traffic flow.

The most recurrent and important V2X communications are the Vehicle-to-Vehicle (V2V) [3], they enable vehicles to communicate directly with each other, creating ways of instantaneous exchanges of information between them. This would help prevent collisions, provide real-time alerts on driving conditions, form a set of cooperation medium between the vehicles which would pave the way for automated vehicles that are aware of their environment.

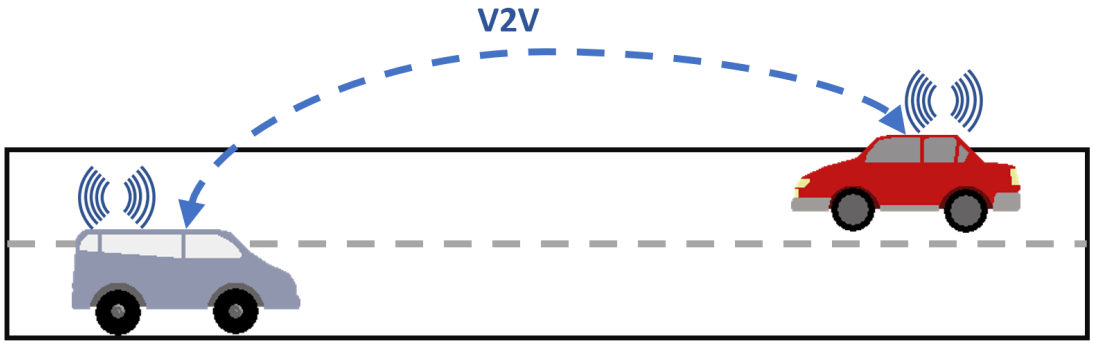

FIGURE 1.1: A General Scheme for Vehicle-to-Vehicle Communications.

Vehicle-to-Infrastructure (V2I)[3] communications extend this interconnection to road infrastructures actors. Traffic lights, road signs and other infrastructure elements become active participants in traffic management, creating an ecosystem where coordination between vehicles and infrastructure promises smoother, safer traffic.

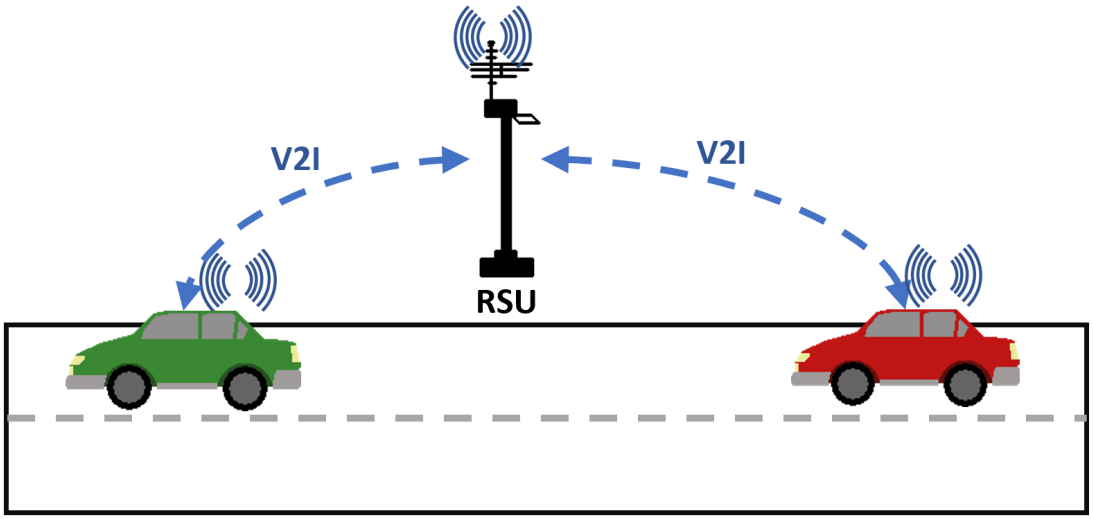

FIGURE 1.2: A General Scheme for Vehicle-to-Infrastructure Communications.

However, the impact of vehicular communications goes beyond simply improving the road experience. These technologies play an essential role in preparing for the forthcoming of automated and connected vehicles, this is done by the medium of cooperation and collaboration through data sharing between the road actors.

Connected vehicles:

Connected vehicles incorporate advanced communication technologies to establish a connection between the vehicle, road infrastructure and other vehicles. As stated before, this connectivity is often referred to as V2X, encompassing V2V, V2I, V2P (Vehicle-to-Pedestrian), and other communications.

Connected vehicles use wireless networks, such as 4G, 5G, or adhoc wifi connections to exchange data in real time. This enables vehicles to share information on their states (position and velocity for example) road conditions, traffic incidents, optimal routes, and much more.

automated vehicles:

Automated vehicles, often referred to as driverless cars, are cars equipped with an automated driving system capable of functioning without human intervention (adaptive cruise control for lower automation levels for example). These vehicles use a variety of sensors, such as cameras, lidars, radars, and ultrasonic sensors to perceive their environment in real-time. Combined with advanced processing, this sensory data enables the vehicles to make complex navigation tasks.

Autonomy levels vary on a scale of 0 to 5, as defined by the Society of Automotive Engineers (SAE). At level zero, the driver assumes all the driving while at level five, the vehicle can fully autonomously drive in all conditions without

human intervention.

Connected and Automated Vehicles:

Automated vehicles can benefit from connectivity to receive information from their surrounding which enhances their perception of the environment. In turn, connected vehicles can leverage autonomous capabilities to optimize driving, including use cases such as automated parking, driving assistance, etc. The terms are often used together as we usually speak of automated and connected vehicles.

1.1.3 Cooperative Intelligent Transport Systems

Cooperative Intelligent Transport Systems (C-ITS) are a revolutionary step in the evolution of technologies (mainly ICT) applied to the transport sector. They are designed to encourage the cooperation between road actors (hence the name). A new age of connected and intelligent mobility is being created by the C-ITS.

Key technologies that are laying the foundation of C-ITS include wireless connectivity, advanced sensors, global positioning systems (GPS), and intelligent data processing algorithms. Thanks to these innovations, connected and automated vehicles can share data about their environment and act according to the data they receive.

The C-ITS offer vehicular communication mechanisms that enable the exchange of messages between road traffic actors, in particular vehicles and road infrastructure. The final objectives of such a system can be summarized as follows:

- Improving road safety;
- Improving road transport efficiency;
- Traffic management and congestion reduction;
- Protecting the environment and reducing air pollution.

Given the requirements (Quality of Service) associated with C-ITS, these systems fall within the scope of mission-critical applications, i.e. those requiring very low latency and high communications reliability. In addition, vehicles need real-time information about their environment. The availability of information and the multiplication of information sources are therefore essential to the operation of a vehicular communication system.

Figure 1.3 represents the protocol stack of the C-ITS system. The application layer consists of services destined for the user such as the UI. The C-ITS services are implemented in the Facilities layer.

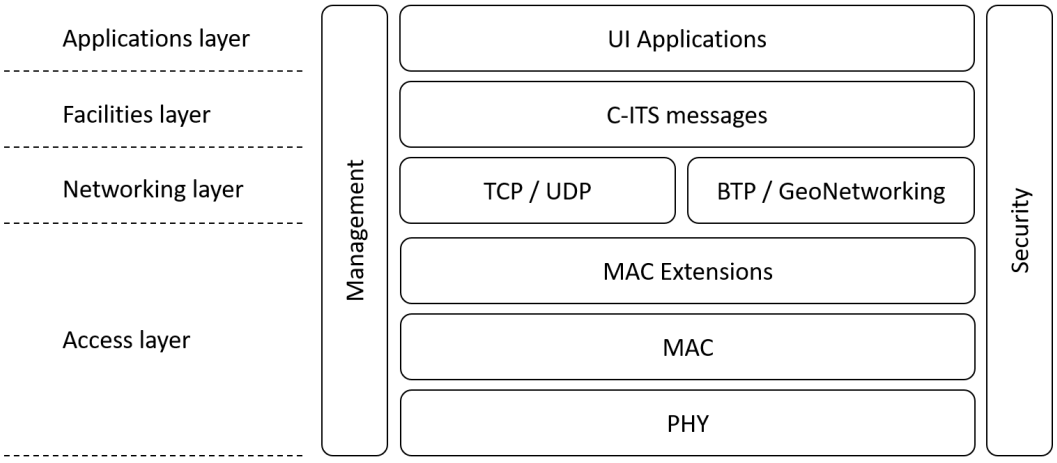

FIGURE 1.3: The C-ITS protocol stack.

C-ITS are based on several services implemented in the Facilities layer. Each service is linked to a specific message type. These "Day 1" services use V2I and/or V2V communications. Some services use only messages transmitted from infrastructure to vehicles, while others use both communication schemes. In terms of services that use both V2V and V2I, the most notable are Cooperative Awareness (CA) and Decentralized Environmental Notifications (DEN):

CA [4] is a service that controls the transmission and reception of Cooperative Awareness Messages (CAM). CAM is a periodic message sent by vehicles and Road Side Units (RSU) at broadcast frequencies between 1 and 10 Hz. The message is sent to all neighboring vehicles and RSUs and contains information about the original sender, such as its position, speed and other information. The aim is to create an awareness between vehicles of their surroundings, to encourage cooperation between them.

DEN [5] is the service responsible for broadcasting Decentralized Environmental Notifications Messages (DENM). A DENM is a message indicating an event on the road, such as an accident or roadworks. When an event on the road is detected, the vehicle or RSU can transmit a DENM to the zone concerned. Unlike the CAM, a vehicle receiving a DENM can re-transmit it to other nodes in the network. A DENM message can remain in persistent broadcast mode for as long as the event continues.

For V2I communications, in addition to CA and DEN, there are services whose messages are generated solely by the infrastructure. Messages are transmitted from the infrastructure to the vehicle and not vice versa. We cite the following services [1]:

Traffic Light Maneuver (TLM) is a service that manages the generation,

transmission and reception of Signal Phase and Timing Messages (SPATEM). Its aim is to control user access to intersections and conflict zones. It sends safety information to vehicles present at an intersection, informing them of the real-time status of the traffic light and its future status, as well as the time margin between the two. The SPATEM message is then sent periodically to all participants for as long as they are in the intersection to drive or cross (pedestrian).

Road and Lane Topology (RLT) is the service responsible for generating, transmitting and receiving MAP (topology) Extended Messages (MAPEM). A MAPEM is a message containing a digital topology map of the geometry of an area, describing lanes, crosswalks, conflict zones (intersections) and authorized maneuvers. A lane is made up of several connected landmarks. At the connection of these crossing points with a conflict zone, a set of authorized maneuvers is modeled as a connection between the two points that connect the two lanes with the zone. The MAPEM is transmitted at the same time as a SPATEM when the vehicle approaches an intersection.

Infrastructure to Vehicle Information (IVI) is a road signage service that uses the Infrastructure to Vehicle Information Messages (IVIM) to provide information on physical or virtual road signs, such as contextual speeds or road warnings, as well as the presence of roadworks.

1.1.4 Communication standards

As far as radio communications are concerned, the two most commonly used standards for vehicular communications are ITS-G5 (and its equivalent DSRC) and Cellular Vehicle-to-Everything (C-V2X). The former is a standard based on Wi-Fi technology and is mainly short-range. C-V2X, based on the cellular network, has a greater communication range and possesses the same mechanisms as cellular technologies, such as the scheduling of radio channel occupancies for transmission.

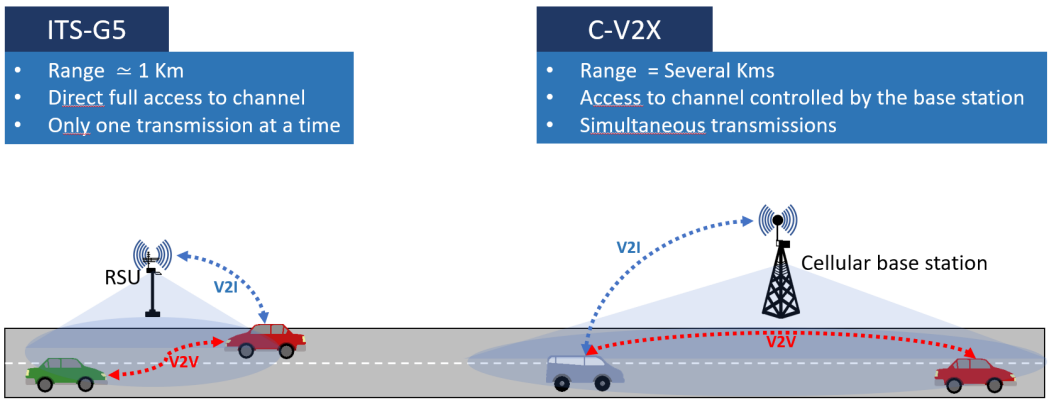

FIGURE 1.4: Comparison between ITS-G5 and C-V2X.

C-V2X was standardized by 3GPP in Release 14 [6]. In this release, 4 types of communication in broadcast mode were specified: a) V2V or vehicle-to-vehicle, b) V2I or vehicle-to-infrastructure (RSU), c) V2N or vehicle-to-network, when communication is with a server from the cellular network, and d) V2P or vehicle-to-vulnerable user (pedestrian, cyclist).

Two modes of operation have been specified for these communication schemes:

- Direct V2V communications via the PC5 air interface, also known as Sidelink. This is a type of direct communication between users, without passing through the network.

- V2V communications via the Uu air interface, where each communication is made with an initial link to the network (Uplink) and then another to the destination (Downlink).

ITS-G5 [7], on the other hand, is based on Wi-Fi and operates in the 5.9GHz band. It implements the IEEE 802.11p communication standard and its extension 802.11bd . It has a shorter communication range and supports only ad hoc links between vehicles and also RSUs. The sender occupies the entire available channel during transmissions.

As far as the protocol stack is concerned, ITS-G5 differs from other standards with the presence of the GeoNet layer [8]. To relay a message in a dynamic, mobile environment on an ITS-G5 network, geographic routing is used. The GeoNetworking protocol is an ETSI standard and enables packets to be relayed from the source to a zone called the relevance zone. The aim of this protocol is to continuously build a list of neighbors and then choose the most suitable neighbor to which the message will be sent to reach the destination. The idea is therefore to make a

sequence of hops that will eventually enable the message to be transmitted to the final destination.

1.2 CoopITS application

CRoads is one of the joint initiative projects of European member states that aim to developing innovative C-ITS solutions. The architecture of CRoads is very rich and comprises of both long and short range communications. As part of the project, the CoopITS application was developed, in which a smartphone plays the role of a station (a vehicle). It allows the sending and reception of C-ITS messages using the cellular network. The application was launched in January 2021 and it functioned mainly in the region of "La Nouvelle-Aquitaine" in France.

The application records all C-ITS messages which get stored, such as CAMs, DENMs, SPATEMs, MAPEMs, and IVIMs, which are recorded in their raw ASN.1 UPER encoded state, and then are decoded and stored into a data base. Figure Fig.1.5 shows the architecture of the application, whereas Fig.1.6 shows some screen shots of the app from the playstore official homepage.

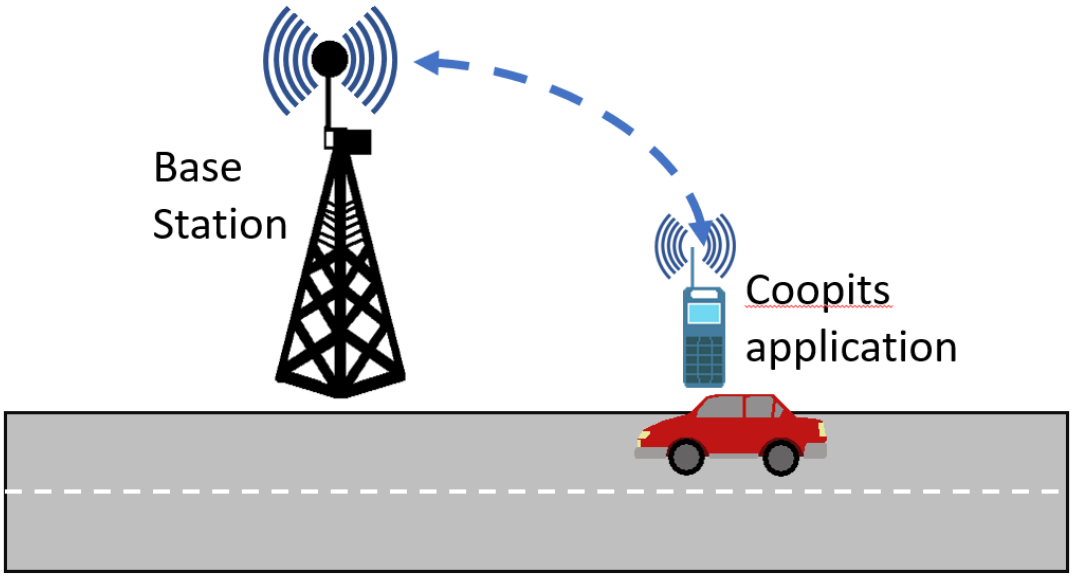

FIGURE 1.5: The architecture of the CoopITS application.

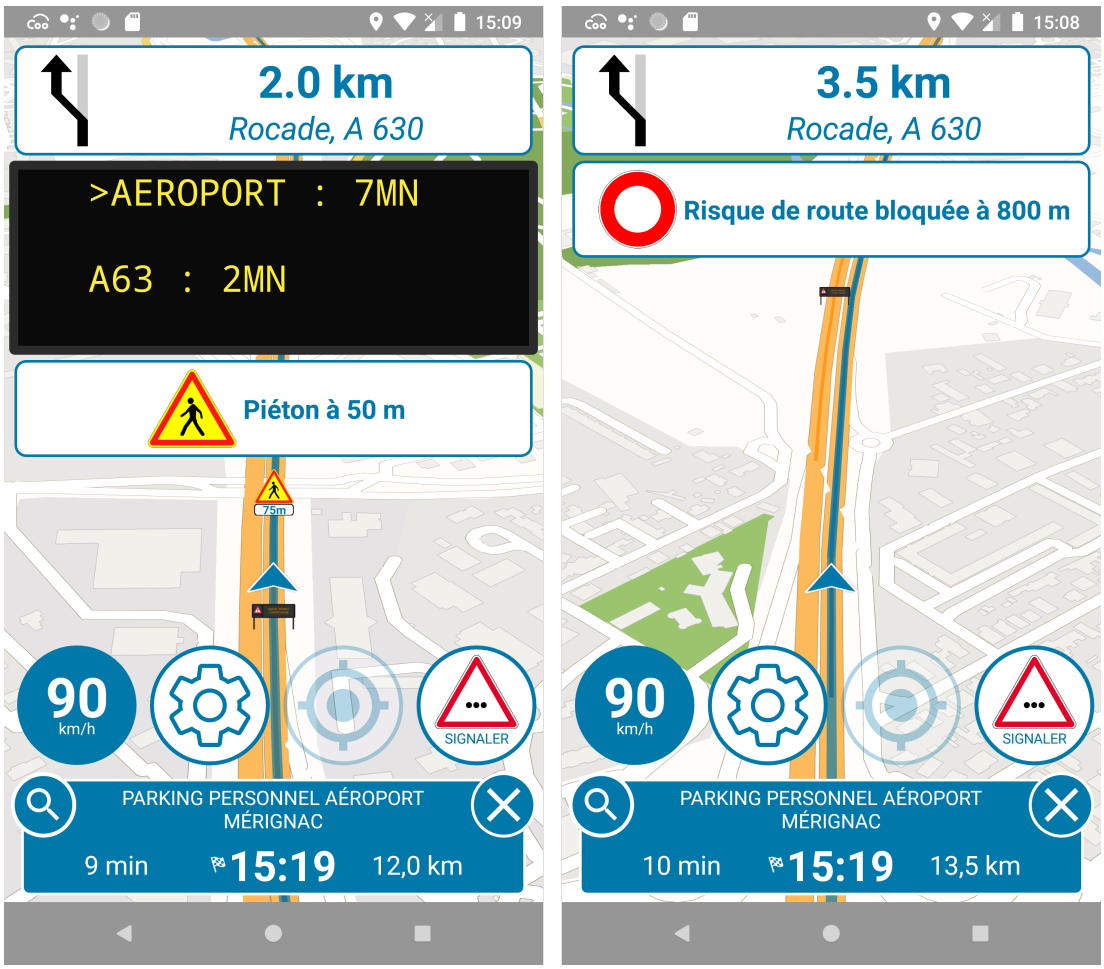

FIGURE 1.6: Screen shots of the CoopITS application, source : official playstore home page.

1.3 Contributions

Within the field of the C-ITS, technological developments are constantly reshaping the urban mobility landscape to provide connected and intelligent mobility. Each piece in the following contributions explores a crucial area, leveraging data analysis and profiling for furthering our knowledge on vehicular communications :

1. Road-Side Unit Anomaly Detection : is a work that concentrates on RSUs, The research explores RSU profiles, looking particularly at communication range and signal strength. We find characteristic profiles linked to failing RSUs, providing a foundation for anomaly detection. A key innovation lies in the proposed mechanism for detecting RSU failures, where end users or vehicles play an active role. These vehicles can communicate the RSU failure to approaching vehicles and the infrastructure by sending alarm messages in the event of a malfunction.

2. Impact of Speed Limitation on Urban Road Traffic : is a paper in which we examine the use case of Green Light Optimal Speed Advisory (GLOSA) in the context of the C-ITS. To establish a correlation between traffic lights and the state of the relevant vehicles (velocity), the study entails a matching process between GLOSA data, specifically TLM records, and CAM messages. By means of this analysis, the study pinpoints and groups four unique driving patterns.

3. Towards Road Profiling with Cooperative Intelligent Transport : Is a work that involves the segmentation of a single trajectory that is traversed multiple times, either by the same or different drivers. Each segment is defined as the portion of the road situated between two road crossings. A set of 601 segmented trajectories was produced as a result of the study. Following segmentation, these trajectories have been processed by a number of clustering algorithms in order to compare performance and results and identify road profiles.

   In addition to the 3 previous papers, during the process of this thesis, another paper was published alongside a book chapter, which are respectively:

   1- Leo Mendiboure, Mohamed Lamine Benzagouta, Dominique Gruyer, Tidiane Sylla, Morayo Adedjouma and Abdelmename Hedhli. Operational Design Domain for Automated Driving Systems: Taxonomy Definition and Application. 2023 IEEE Intelligent Vehicles Symposium (IV), Anchorage, AK, USA, 2023, pp. 1-6, doi: 10.1109/IV55152.2023.10186765.

   2- Mohamed Benzagouta, Ramzi Boutahala, Secil Ercan, Sassi Maaloul, Hasnaâ Aniss, Léo Mendiboure, Marwane Ayaida and Hacène Fouchal. Towards an Optimization of Data Transmission in Cooperative Intelligent Transport Systems. Cooperative Intelligent Transport Systems: Control and Management Wiley 2024.

   The rests of chapters 3, 4, 5, and 6 are covered by the following articles (which are the same in the contributions subsection) :

   1- Benzagouta, M.-L.; Aniss, H.; Fouchal, H.; El Faouzi, N.-E. Road-Side Unit Anomaly Detection. Vehicles 2023, 5, 1467-1481. https://doi.org/10.3390/vehicles5040080.

2- Benzagouta, Mohamed Lamine ; Aniss, Hasnaa ; Fouchal, Hacène ; El Faouzi, Nour-Eddin. (2023). Towards Road Profiling with Cooperative Intelligent Transport Systems.

3- Benzagouta, M. L., Aniss, H., Fouchal, H., ; El-Faouzi, N. E. (2023, September). Impact of Speed Limitation on Urban Road Traffic. In International Conference on Innovations for Community Services (pp. 89-102). Cham: Springer Nature Switzerland.

1.4 Outline

The thesis content is organized as follows, Chapter 2 synthesizes some works related to topics we used in our contributions, topics such as C-ITS related works, clustering algorithms, anomaly detection in complex systems, traffic prediction on intersections and profiling over C-ITS.

Chapter 3 focuses on work carried out in the analysis of pcap data. The pcap data were collected in a naturalistic environment using a test vehicle and several RSUs encountered on the road. The analysis revealed a correlation between signal strength from the RSUs and communication range, mainly an inverse relationship. Fault detection is also based on another observation, namely that communication range is higher when entering an RSU's coverage area than when leaving it. These observations were used to determine RSU failures, with failure detection to be carried out by the vehicles themselves, which should transmit an alarm message to the road operator concerning the failure.

Chapter 4 deals with work carried out on the CoopITS smartphone application database. The data contained in the CAM messages of various participants are observed and analyzed in order to extract road profiles. The trajectories of individual vehicles are then grouped together using the K-Means algorithm and three other clustering algorithms. Each road profile is characterized by its speed profile and its temporal profile (the day and time of the trajectory).

Chapter 5 covers a work done on the TLM service in the C-ITS system. In our case, the CAM messages were used to extract vehicle trajectories, the trajectories were then analyzed and cut into road segments, and a segment with traffic lights was chosen for the study. We then observed speed variations as a function of time and distance on the segments according to traffic light status. We classified the observations into several categories.

Last, In Chapter 6 is the conclusion of the thesis.

Chapter 2

State of the art

2.1 Introduction

An overview of the present state of the art on C-ITS research is given in this chapter. It explores important academic works that have greatly accelerated the development of C-ITS. The chapter then shifts its focus to machine learning methods, particularly those related to classification algorithms. Then, works related to complex system anomaly detection are discussed, emphasizing the identification of anomalies in RSUs. Next, efforts that are devoted to the field of traffic prediction at intersections are presented. Next, the chapter presents some works concerning profiling within the C-ITS framework, emphasising on works concerning driver profiles. The goal of this chapter is to offer a comprehensive overview of the state-of-the-art developments influencing C-ITS research and related topics.

This chapter takes important parts of the following articles:

1- Benzagouta, M.-L.; Aniss, H.; Fouchal, H.; El Faouzi, N.-E. Road-Side Unit Anomaly Detection. Vehicles 2023, 5, 1467-1481. https://doi.org/10.3390/vehicles5040080.

2- Benzagouta, Mohamed Lamine ; Aniss, Hasnaa ; Fouchal, Hacène ; El Faouzi, Nour-Eddin. (2023). Towards Road Profiling with Cooperative Intelligent Transport Systems.

3- Benzagouta, M. L., Aniss, H., Fouchal, H., El-Faouzi, N. E. (2023, September). Impact of Speed Limitation on Urban Road Traffic. In International Conference on Innovations for Community Services (pp. 89-102). Cham: Springer Nature Switzerland.

2.2 Cooperative Intelligent Transport Systems

C-ITS is a new paradigm in modern transportation, a paradigm change made possible by the incorporation of modern ICT. These systems make use of contemporary communication technologies to completely re-imagine the present mobility scene while also improving transportation's sustainability, safety, and efficiency. This review of the state-of-the-art looks at significant achievements made in the C-ITS field. This section attempts to give a comprehensive perspective of the critical role that C-ITS will play in transforming transportation in the future by traveling through significant works.

With an emphasis on Service-Oriented Architecture (SOA), Grid Computing, and Cloud Computing, the paper [9] undertakes a survey that particularly investigates the application of distributed architectures in Intelligent Transport Systems (ITS). The authors classify the most important and basic applications in the ITS domain and provide a reference architecture for distributed ITS. The article [10] discusses integration challenges that are essential for enabling an intelligent transportation system to address problems facing the transportation industry, such as rising fuel prices, increased CO2 emissions, increasing traffic jams, and improved road safety. The writers draw attention to relevant new technologies that can improve ITS. Connected vehicles are presented as a way to share relevant data by utilizing wireless technologies for V2V and V2I communications.

WITS, A system for gathering and exchanging transportation-related data is presented in [11] using Wireless Sensor Networks (WSNs). Both software and hardware WSN module design and prototyping are covered. WSN technology is integrated into the ITS to improve its efficiency. Its advantages include wireless connectivity, low energy consumption, and extensive road coverage. CarTel is a mobile distributed sensor computing system developed by Hull et al in [12]. It offers software solutions for the collection, processing, delivery, and visualization of data obtained from sensors situated on mobile devices. The system facilitates the transfer of sensor data to a centralized portal for further analysis and interpretation.

The system requirements for setting up an intelligent traffic management data center are examined in [13]. It provides a design methodology based on SOA, explains the implementation strategy in detail, and makes a case for the SOA framework. Using the Beijing Traffic Data Center as an example, it highlights how SOA is used in its implementation. The shared characteristics of SOA, such as its fine-grainedness, platform independence, distributed architecture, and service-based

applications, support the choice. The goal of [14] is to investigate, from several angles, how 5G will affect the ITS. It starts by giving a summary of the technological background and financial advantages of 5G, focusing on how it will affect important vertical industries within smart cities, such as manufacturing, energy, healthcare, entertainment, automotive, and public transportation. According to the paper, 5G has the potential to improve automated transportation systems' safety and efficiency over current networks, which could lead to an increase in city intelligence.

The article [15] provides an overview of wireless technologies and discusses potential challenges to V2X connectivity, with a focus on self-driving and connected cars. These technologies mark the advent of robots that directly impact the daily lives of millions of people. It explores the role of Software Defined Networking (SDN) in Heterogeneous Networks and goes into detail about the 5G architecture built with SDN. According to the paper [16], there is great potential for developing V2X communication technologies, particularly V2I. The goal is to find out how V2I technology can improve commuters' travel times and solve typical issues. The suggested method entails informing oncoming cars about the traffic light cycle via roadside infrastructure, such as traffic light controllers at intersections. Vehicles cooperate to decide on the best speeds and course of action based on this data in order to reduce delays and prevent needless stops when crossing intersections. The evaluation's findings show that commuters who use this strategy experience a notable decrease in their average travel time.

Using edge infrastructures, Gupta et al. [17] presented a safe and reliable method for V2V and V2I communication. Instead of using direct peer-to-peer communication, this approach uses trusted cloudlets to authorize, examine, and verify the integrity, anonymity, and authenticity of messages exchanged within the system. Moving vehicles or roadside equipment dynamically connects to neighboring cloudlets so that security controls can be put in place to verify and filter communications, stop fraud, and stop rogue vehicles from interacting with other vehicles.

The C-ITS infrastructure on the Budapest Ring Road (expressway M0) was successfully modeled by the authors in [18], and they also simulated V2X messaging for Day 1 and Day 2 cooperative services. Real-world traffic data measurements were processed through a comprehensive implementation of the complete V2X communication chain, including facilities, applications, and services, in order to incorporate them into the simulations. A traffic model was developed in [19] to evaluate the possible capacity increase on the Italian A22 highway in comparison

to current values. Different percentages of automated and manual vehicles were taken into account by the model. Theoretical findings imply that automated cars are capable of safely navigating the highway. Furthermore, based on experimental results using traffic data from four highway sections and Drake's flow model, the expected increase in lane capacity is up to 2.5 times higher than the current capacities.

The purpose of the study in [20] is to evaluate the precision of parameters sent in CAM messages by the initial generation of commercial V2X-enabled cars on the European market. Applications like digital accident data for accident analysis could benefit from the CAM data. The results of driving tests at a specified test site are reported in detail. Based on standardized message-triggering rules, these tests systematically produced CAMs in particular scenarios and described the statistical characteristics of the resulting CAM data. A novel method for handling the Pseudonym Certificates (PC) switching periods between vehicles is presented in [21]. The suggested approach makes use of a Common PC (CPC) for a brief amount of time before moving on to a new PC. During this time, vehicles sign their messages using the same shared PC.

CyberSAGE, a framework utilizing simulators for data generation, collection, and synthetic data creation through Generative Adversarial Networks (GANs), is presented in the paper [22]. CyberSAGE outlines future research directions, such as the addition of complex attacks, improvement of deep generators, and integration of new AI architectures to advance cybersecurity techniques for connected vehicles, even though it is still in its early stages as a proof of concept. A lightweight, open-source library that complies with standards and is application-independent is presented in the work [23]. The library is deployed in a small-scale test bed where it is used to send DENMs to a vehicle in order to cause it to slow down during bad weather. The vehicle uses CAMs to communicate its status information, which includes its current velocity. Using the LEVEL communication stack, the experiment demonstrates the location-dependent information exchange between an RSU and the vehicle.

2.3 Clustering Algorithms

Unsupervised learning algorithms are powerful tools in the vast field of machine learning that can handle unlabeled data and extract meaningful insights. This section delves deeply into a variety of unsupervised approaches, shedding light on the various approaches used to negotiate the challenges presented by unlabeled

datasets. The variety of techniques discussed here, which range from traditional clustering algorithms like k-means to hierarchical clustering, reveals a vast toolkit. The important tasks of preprocessing data, locating hidden structures, and exposing patterns that might otherwise go undetected are all accomplished by these techniques. The purpose of this section is to give a deeper understanding of the techniques of unsupervised learning.

2.3.1 Hierarchical-Based Algorithms

Unsupervised machine learning techniques like hierarchical clustering arrange data points into a hierarchy of clusters or a tree-like structure. This clustering technique's main objective is to combine related data points according to a predetermined similarity metric. Agglomerative and divisive hierarchical clustering are the two basic categories into which it can be broadly divided.

The technique used to determine the separation or dissimilarity between clusters at each stage of the clustering process is known as linkage in hierarchical clustering. The structure and form of the resulting hierarchical tree (dendrogram) are influenced by the linkage criterion, which dictates how the distance between clusters is measured as they are successively merged.

Different linkage methods are frequently employed, and they all define the distance between two clusters in a different way. The hierarchical tree's structures can vary depending on the linkage method used, which can also affect the clustering results. Among the primary linking techniques are:

- Single Linkage (Nearest-Neighbor Linkage): Defines the distance between two clusters as the shortest distance between any two members of the two clusters. Tends to produce elongated and chain-like clusters

- Complete Linkage (Farthest-Neighbor Linkage): Defines the distance between two clusters as the longest distance between any two members of the two clusters. Tends to produce compact, spherical clusters.

- Average Linkage: Defines the distance between two clusters as the average distance between all pairs of members from the two clusters. A compromise between single and complete linkage.

- Centroid Linkage: Defines the distance between two clusters as the distance between their centroids (mean vectors). Can be sensitive to outliers.

2.3.1.1 Agglomerative clustering

Agglomerative Clustering or Agglomerative Nesting (AGNESS) is a type of hierarchical clustering. The term "agglomerative" refers to the process of progressively merging or "agglomerating" groups of data points. Agglomerative clustering is a bottom-up approach, starting with individual data points and building up to larger clusters. It is in contrast to divisive clustering, which is a top-down approach that starts with all data points in one cluster and splits them into smaller clusters.

First, each observation is considered as a single element cluster, so for n observations we have n clusters. Second, each clusters that are close to each other are merged into one cluster, this process is repeated until one cluster remains containing all the n points. Third, a dendrogram is obtained which is like a tree diagram that shows the sequence in which the clusters were merged. Last a number of clusters k is chosen and the dendrogram is cut in the middle so it contains the required number of clusters.

The distance between the clusters can be defined using several methods. Some common methods -as stated before- include single-linkage (minimum pairwise distance), complete-linkage (maximum pairwise distance), average-linkage (average pairwise distance), and Ward's method (minimize the increase in variance within clusters).

2.3.1.2 BIRCH Clustering

Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) is a hierarchical clustering algorithm that is efficient for large datasets [24], it works in an online incremental manner allowing it to handle datasets that might not fit completely into memory. BIRCH maintains compact summary of data using Clustering Feature (CF), CFs contain statistical information about subsets of data points. CFs are organized in a hierarchical data structure called the CF Tree. The algorithm incrementally inserts data points into the tree, selecting the appropriate leaf nodes based on criteria such as minimizing the increase in subcluster diameter. If inserting a point would exceed a specified diameter threshold, a new leaf node is created. As the process continues, clusters are merged and the tree is rebalanced.

2.3.1.3 Divisive hierarchical clustering

Divisive hierarchical clustering is a top down approach in unsupervised machine learning. It starts by attributing all data points into a single global cluster and

recursively divides it into smaller clusters. Thus, the division continues until each data point is in a separate cluster or until a stop criterion is met. At each step, the algorithm takes the cluster with the highest intra-cluster dissimilarity and divides it into two new clusters. This dissimilarity can be measured using various distance metrics, and the choice of distance metric significantly influences the clustering results. Divisive clustering provides a hierarchical structure, often represented as a dendrogram, illustrating the successive divisions. While divisive clustering can reveal the natural hierarchy in the data, it tends to be computationally expensive, especially for large datasets, and is sensitive to the choice of dissimilarity metric and the stopping criterion. Hierarchical clustering employs the linkage techniques Slink [25] and Clink The single linkage (nearest-neighbor) approach is the basis for both Slink and Clink.

2.3.2 Partitioning-Based Algorithms

Partitioning based clustering algorithms are a class of unsupervised machine learning methods that aim to divide a dataset into discrete clusters. By maximizing the similarity of data points within each cluster and reducing the dissimilarity between clusters, these algorithms aim to producing partitions. Several well-known algorithms are as follows :

2.3.2.1 k-means

K-means [26] is a non supervised machine learning algorithm of non hierarchic clustering. It permits the clustering of n given points in K given clusters. Thus, similar observations end up in the same cluster so as to minimize a given function.

Given a set of n observations $(x_1, x_2, x_3, ..., x_n)$, k-means aims to partition the n observations into $k(k \leq n)$ set $S = \{s_1, s_2, ..., s_k\}$ by minimizing the distance, and the within-cluster sum of squares (WCSS) :

$$\arg\min_{s} \sum_{i=1}^{k} \sum_{x_j \in s_i} \|x_j - \mu_i\|^2$$

where μ_i is the centroid of the cluster s_i.

For the k-means, k centroids are randomly chosen from the n observations at first, each centroid forms a cluster and then each observation is attributed to the nearest (mostly euclidean distance) centroid/cluster. Then the centroids of clusters are recalculated based on the mean of all the data points assigned to each cluster. The n observations are assigned again to the nearest centroids and the process is repeated until convergence. Convergence occurs when the centroids

no longer change significantly, indicating that the algorithm has found a stable solution.

2.3.2.2 K-Medoids

K-Medoids [27] is a partitioning-based clustering algorithm that forms clusters around representative points (points in the dataset) called medoids in order to find naturally occurring clusters within a dataset. Since K-Medoids employs real data points as the centroids instead of the more widely used centroids of K-Means algorithm, it is more resilient to noise and outliers in the dataset. In order to minimize the total dissimilarity between data points and their assigned medoids, the algorithm iteratively assigns data points to the nearest medoid and updates the medoids. Clusters with central points that most accurately reflect the general properties of the data are produced as a result of this process, which is continued until convergence. Applications for K-Medoids can be found in a number of domains, such as pattern recognition, where the identification of representative data points is essential to comprehending underlying structures and patterns in the data.

2.3.3 Density-based clustering algorithms

Density-based clustering algorithms are a class of unsupervised machine learning methods that locate clusters Based on the density of data points in a dataset. Density-based techniques are able to detect clusters of any shape and can deal with noise in the data better than partitioning-based algorithms. The following are a few well-known density-based clustering algorithms:

2.3.3.1 DBSCAN Clustering

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [28] is a density-based clustering algorithm. It does not require the number of clusters to be specified beforehand. Instead, it identifies clusters based on the density of data points in the feature space.

For n observations, they can be classified as :

- Core Points: A data point is considered a core point if it has at least a specified number of neighboring points (minPts) within a certain radius (ϵ).

- Border Points: A data point is considered a border point if it is within the $\epsilon - radius$ of a core point but does not have enough neighboring points to be considered a core point itself.

- Noise Points (Outliers): Data points that are neither core points nor border points are considered noise points or outliers.

The DBSCAN algorithm is given three parameters, the data set, ϵ the distance which makes a cluster and $minPts$ as the minimum points that can constitute a cluster. It then selects randomly points that were not visited before, If the selected point has at least $minPts$ neighbours with ϵ distance, then it is considered a core point and the same process is done for its neighbouring points and a cluster is created with the core and border points that were visited.

2.3.3.2 DENCLUE Clustering

DENCLUE (DENsity-based CLUstEring) [29] is a density-based clustering algorithm that employs a mathematical model based on kernel density estimation. Unlike traditional clustering algorithms, DENCLUE views clusters as high-density regions separated by saddle points. It assesses the probability density function of the data using kernel functions, allowing it to identify clusters of varying shapes and sizes. The algorithm iteratively refines cluster boundaries by shifting points towards regions of higher density. One of DENCLUE's strengths lies in its ability to capture complex structures in the data and effectively handle datasets with irregularly shaped clusters. Its reliance on density estimation makes it robust to noise and outliers, making it suitable for applications such as spatial data analysis, where clusters may exhibit intricate spatial relationships and varying densities. DENCLUE provides a flexible approach to density-based clustering, contributing to its effectiveness in uncovering patterns in diverse datasets.

2.3.3.3 OPTICS Clustering

By expanding on the ideas presented by DBSCAN, the density-based clustering algorithm Ordering Points To Identify the Clustering Structure (OPTICS) [30] aims to reveal the underlying structure of data. OPTICS introduces a reachability distance metric to evaluate the connectivity between data points, thereby overcoming constraints associated with cluster shapes and sizes. OPTICS represents the density-based clustering structure by "ordering" data points according to their reachability distances, in contrast to traditional clustering algorithms that

yield a fixed number of clusters. Through the representation of the hierarchical connections among data points, OPTICS offers a more adaptable viewpoint on cluster identification.

2.4 Anomaly detection on complex systems

In order to guarantee the dependability, robustness, and continuous functioning of complex systems in a variety of contexts, failure detection is essential. It is essential to promptly identify and mitigate problems in a world where technology and interconnected systems are becoming more and more important. System failures can result in a variety of repercussions, including minor interruptions, major disruptions, financial losses, and even safety issues, depending on the industry, transportation, telecommunications, manufacturing, or information technology. This emphasizes how important it is to have reliable and effective failure detection systems that can quickly spot anomalies, irregularities, or departures from expected behavior. The field of failure detection has greatly improved with the advent of sophisticated sensors, data analytics, and machine learning approaches, opening up new possibilities for proactive problem-solving and system improvement.

A key component of the C-ITS designed to guarantee the stable operation of connected vehicle networks is RSU failure detection. RSUs are crucial for enabling communication between infrastructure and vehicles and for delivering necessary services for safety and traffic control. It is essential to identify RSU failures in order to keep information flowing smoothly and avoid interruptions of vital C-ITS functions.

This section of the state of the art explores the challenging field of failure detection, looking at approaches, difficulties, and the crucial part it performs in guaranteeing the durability and dependability of modern systems, while we focus mainly on RSU failures. The review focuses on the techniques and tools used to quickly detect and handle possible RSU anomalies, improving the dependability and efficiency of interconnected transportation networks.

In [31] an anomaly detection approach is the process of defining a region of normal behavior within a set of data and declare whichever data points that do not belong to this region as anomalies. Moreover, anomalies are classified into three categories : a) point anomalies which is when a data point is considered aberrant from the rest of the data, b) contextual anomalies which is when a data instance is anomalous in a specific context, c) collective anomalies which is when a collection

of related data instances is anomalous while the individual data instances may not be anomalous themselves.

Vehicular communication applications are known for their high mobility, which qualifies them as critical applications that require a reliable infrastructure and communication scheme. RSUs are one of the key components of a C-ITS system, they mainly ensure the V2I communication type. An omni-directional antenna of an RSU is assumed to have equal radio propagation in all directions, but the presence of obstacles such as buildings, tunnels, rivers or ground elevations can cause signal attenuation in certain directions. Such an RSU is not technically defective. However, high signal attenuation in some areas of effective coverage could indicate failure.

Some works utilize unsupervised machine learning for anomaly detection, The Isolation Forest or iForest algorithm, introduced by Liu, Ting, and Zhou in [32] The method employs an ensemble of isolation trees to isolate anomalies efficiently within a dataset.The key idea behind the Isolation Forest algorithm is that anomalies are likely to be isolated faster than normal instances when using a random partitioning strategy.

In [33], antenna failure detection was implemented using a probabilistic model. The model was constructed using real data from field tests, whereas the health assessment of RSUs was done through comparison between their behaviors in terms of radio propagation. In [34] a failure detector of VANET systems is proposed, which has some notions of signal attenuation. In [35] numerical simulations were performed in various radio propagation scenarios, where it was shown that various factors affect the attenuation of a signal. Signal tends to attenuate over large distances in line-of-sight (LOS) contexts, whereas in non line of sight (NLOS), the presence of a building or obstacle can greatly affect signal intensity (in dBm). Thus, the behavior of radio propagation is highly dependant on the geographical context be it rural or urban.

In terms of range, we have observed in our work (in the upcoming chapter) that in the rural context, and with the absence of obstacles, the in-distance (ID) of communication is greater than the out-distance (OD) which are respectively the range of the first point where the RSU signal is detected and the range of the last point. This was observed also in [36] where the range of communication between a vehicle and an RSU tends to be larger when the vehicle is moving towards the RSU. Moreover, they have observed a relationship between vehicle velocity and range where they explain it as a consequence of the Doppler effect. It can also be observed in [37]. The inverse case is however observed in [38], where the OD is

greater than the ID, the authors however explain this by the time needed for the mechanism of signal validation in the WiFi access points during a first connection.

DeepADV [39] is an anomaly detection framework for VANETS based on deep neural networks. A threshold is calculated based on the difference between a genuine and an anomalous message and then used to classify them. The algorithm is to be deployed on RSU and its aim is to detect faulty messages.

In [40], an anomaly detection scheme on VANETs using edge computing was proposed. The faults concerned transmission omission and were detected using RSU-based edge network and vehicular edge computing (VEC). These RSUs guard information about a number of vehicles and a number of collected packets, which are then used to determine whether an anomaly (a change in the numbers) has occurred. They tested their anomaly detection strategy on a simulation and found that the strategy was highly effective at high rates of fault ratio (25%). EVAD is a method proposed in [41] to detect anomalies in vehicles using edge computing, wherein a correlation between sensor variables is drawn and used for anomaly detection where, for an observation, if two supposedly correlated variables do not correlate in reality, an anomaly is detected.

In [42], a CNN is used to extract the spatial, temporal, and spatio-temporal traffic features then used for anomaly detection by the means of employing thresholds. In [43], an anomaly detection approach that takes into account the spatio-temporal features of VANET traffic is proposed. The approach consists of two phases; first, deep learning based on a CNN architecture is used for network traffic estimation; second, a decision-making approach based on reinforcement learning is used to identify the normal and anomalous traffic entries.

In [44] an anomaly detection approach that takes into account the spatio-temporal features of VANET traffic is proposed. The approach consists of two phases, first a deep learning based on CNN architecture is used for network traffic estimation, second a decision making approach based on reiforcement learning is used to identify the normal and anomalous traffic entries. When it comes to security anomalies, DAMASCO [45] is a security based system that aims to detect DoS attacks. A statistical approach was used to detect anomalies in vehicular communications where the MAC layer was addressed and assessed to identify potentially malicious nodes by the number of sent packets and block their activity. In [46], a certification-less authentication method was adopted, wherein the RSUs are only trusted partially and are granted the vehicle's information only partially. Moreover, anomaly detection was performed through clustering; precisely, the agglomerative clustering was used on the traffic data represented as time series

using the dynamic time wrap distance [47].

In [48] a methodology is proposed to analyse data collected from agriculture areas during many years and checks the anomalies on the productions and theire relationship with weathers conditions. In [49] a Mobile phone Network Data (MNF) based framework to detect anomalies in real time is proposed. The framework is based on two steps, the first one is an offline unsupervised learning done on the MND, the second step is the online real time detection of anomalies.

2.4.1 Analysis

Most of these works have notions of thresholding and variable correlation. To the best of our knowledge, no work has considered doing anomaly detection by end users (vehicles). No study also observes the range difference; meaning the comparison between the range of first contact of an RSU and last contact of the RSU. There is also no exploitation of signal intensity and range correlation to determine RSU profiles.

2.5 Traffic prediction on intersections

An important aspect of C-ITS is traffic prediction at intersections, which provides a proactive method of streamlining traffic and improving urban mobility. Intersections become focal points where vehicular movements intersect as urbanization accelerates, so precise prediction is essential for efficient traffic management. Traffic prediction at intersections uses advanced data analytics, machine learning, and sensor technologies to predict traffic patterns, predict congestion, and support adaptive signal control strategies. This review explores the importance of traffic prediction in the context of intersections, discussing the difficulties, approaches, and potential game-changers for developing more adaptable and efficient urban transportation systems.

Several works focus on traffic estimation from data concerning intersections and probe Connected Vehicles (CVs). In [50] data from traffic lights and CVs was used in order to estimate traffic volume. Based on the position of stop of a CV before traffic lights, the size of the queue is estimated. If another CV comes after it, the upper bound of vehicle arrivals between the two CVs can be calculated based on the trajectory of the second vehicle. They modeled traffic volume in traffic light intersections, then they modeled queue arrivals as a time-dependent

Poisson process, the traffic volume parameters are determined using maximum likelihood estimator (MLE) with the Expectation Maximization (EM) algorithm.

Predestination is a method proposed in [51], it aims at predicting the final destination of a trip in progress. The authors gathered data through GPS from 169 drivers doing around 7000 trips. They represented the origin and destination in form of tiles and three probabilistic methods were developed then combined. In the first method a driver's destination is assumed to be belonging to the list of previously visited tiles. The second one considers every possible destination and the last one examines the ground type and assumes the probability of it being a destination (a tile where there is only water is unlikely to be a destination for example).

A directional counts at intersections using Floating Car Data (FCD) approach was described in [52] The paper discusses a methodology for compiling directional counts, or traffic flow data at intersections by integrating data from multiple sources, including FCD and section counts. In [53] a statistical method has been used for real time estimation of queue lengths at signalized intersections using probe vehicle location data. The location of the last probe vehicle in the queue is used to estimate its length. They assume that the marginal probability distribution of the queue length is known and they present an analytical formulation that relies on this prior knowledge. The results show that their method can accurately estimate queue lengths.

A stochastic learning framework to estimate the index of a vehicle at signalized intersections has been proposed in [54], multiple experiments have been conducted with data sets from microscopic traffic simulations and field experiments. The framework involves constructing a three layer Bayesian model that models the relation between vehicle indices and the arrival and departure processes. The results showed that the proposed method can accurately estimate vehicle indices at signalized intersections using sample travel times. [55] proposes a method to predict turn directions of drivers at intersections based on past behaviors, likely destinations and the number of possible destinations by each turn, where it is assumed that drivers tend to take roads that offer them more destination options.

[56] presents a probabilistic method to predict the next road segment a driver will take based on his past traveled segments using GPS data. The experimentation used GPS data from 100 drivers from Seattle area and the results show that the developed markov model can make predictions that are more accurate than random guessing. In [57] a data fusion methodology for traffic flow estimation that combines multiple data sources is proposed. The data sources include FCD from

probe vehicles, Detector data, and historic data, and the three are fused using a Kalman filter and then is used to estimate turning volumes at intersections.

2.5.1 Analysis

Most mentioned studies concern several kinds of predictions on intersections. To the best of our knowledge there were no studies on the impact of GLOSA or traffic lights on intersections. There were also no works that detect driving patterns in intersections.

2.6 Profiling over C-ITS

The complex relationship that exists in the modern transportation landscape between vehicles and the road environment necessitates a nuanced understanding in order to develop intelligent and adaptive systems. Road and trajectory profiling is a key field in the C-ITS that offers deep insights into the dynamics of vehicle movements and the properties of road infrastructure itself. Analysis of road profiles and driver trajectories is crucial to optimizing traffic management, enhancing safety, and bringing in a new era of sustainable and efficient transportation networks as urbanization soars and technological advancements reshape the mobility paradigm.

The amalgamation of trajectory and road profiling makes use of a wide range of technologies, including cameras, sophisticated sensor networks, Global Navigation Satellite Systems (GNSS), and V2X communication. Together with advanced data analytics and machine learning algorithms, this abundance of data enables researchers, urban planners, and transportation experts to gain practical insights that go beyond traditional traffic management. The comprehensive viewpoint obtained from profiling not only helps with traffic optimization and infrastructure design, but it also opens the door for the creation of intelligent cars that can move through intricate urban environments with greater efficiency and safety. This review delves deeply into the niche of road and trajectory profiling, covering the techniques used, and the technological foundations influencing these evaluations.

In [58] an algorithm that classifies driving style is proposed. It utilizes the statistical information from jerk profiles to classify the driving into three categories. The first one is aggressive driving with high alternating jerk, the second one is the calm driving with smooth jerk profiles, and the third one is normal driving which has moderate braking and acceleration. In [59] a framework and methodology for

developing composite driver risk profiles is proposed. It uses empirical data from GPS data collected from 106 drivers in Sydney.

In [60] k-means and hierarchical clustering algorithms were run on a data set of 70 samples from a driving simulator containing number of left and right turns, number of left and right indicators, number of brakes, horns and gear change with speed. Results show that k-means performs better at classifying the data the way the authors intended, meaning three clusters slow, normal, and fast driving. In [61] GPS data (around 373 million records and 40 821 calculated speed profiles) of Taxis in the city of Zagreb was used and fed into G-means (Gaussian means) algorithm which is based on k-means. 770 clusters were determined.

In [62] smartphone sensor data was used, the data was obtained from IEEE Data port and was used to analyse driver behavior. It was clustered using the k-means method and 3 clusters were obtained were cluster 1 corresponds to aggressive/risky behavior, cluster 2 corresponds to normal driving and cluster 3 corresponds to calm driving. In [63] 4 different driver profiles were detected; safe, very safe, aggressive, and very aggressive. The profiling was done with data obtained from smartphones where a driver safety index was used to classify the profiles.

In [64] data from diverse motion sensors, including the accelerometer, gyroscope, and magnetometer, undergoes noise elimination using Kalman filter. It was then used to identify various driving events by applying time window for data extraction. The outcome of this comparison enables the classification of behavior into aggressive and non-aggressive driving profiles. In [65] they used four different machine learning algorithms; Artificial neural networks, Support Vector Machine, Random Forest, and Bayesian network on different smartphone sensor data where they identify 7 driving events; Aggressive breaking, Aggressive acceleration, Aggressive left turn, Aggressive right turn, Aggressive left lane change, Aggressive right lane change, and non-aggressive event.

In [66] the authors study the variable of vehicle Heading around some points of interest in the city of Reims. The study describes the variable in terms of min, max, mean, and median, and compares it to the heading of the point of interest. Some trajectories were extracted and clustered using different clustering approaches, the purity index of each clustering method is calculated and only the DBScan showed low scores.

The study in [67] utilized a driving simulator involving approximately 45 drivers with diverse characteristics such as gender, age, and driving experience. The simulation focused on driving toward and beyond a specified point of interest, in this case, a tunnel. The tunnel's position is regarded as the distance origin,

and the starting point for recording data towards the distance origin is termed the distance length, which remains consistent after passing the point of interest. The distance length is subdivided into equivalent segments known as distance gaps, each serving as a research unit. The authors employed these distance gaps to construct a driving ethogram, from which various characteristics related to statistics on longitudinal speed, acceleration, lateral movement, and deviations from the center lanes were derived. Numerous observations and remarks were documented at the conclusion of the paper.

In [68] the focus is on extracting driving profiles from real log data collected near 22 Points of Interest (POI). The main challenge addressed is the presence of incomplete data, where some vehicles recorded data for only one POI. Various data completion approaches, including interpolation, similar case analysis, and Bayesian methods, were employed to fill the missing data. The complete dataset was utilized for training and clustering, and it was also used to impute missing data across different ratios. The chosen ratio for analysis was 60% missing data and 40% complete data.

The work [69] examines a segment of Route National N118 in France, focusing on vehicles equipped with On-Board Units (OBUs) traveling in both directions. Data from these vehicles was extracted and analyzed using statistical methods, with a particular emphasis on speed profiles. The study provides insights into the speed profiles of C-ITS vehicles within a specific road segment.

Smartphone sensor data are used in [70], including gyroscope, accelerometer, and magnetometer readings, to distinguish between safe and unsafe driving behaviors. The sensor data undergoes preprocessing with filters to smooth acceleration and speed signals. A maneuver detection algorithm, specifically the Endpoint detection algorithm, is applied. Dynamic time warping is employed for calculating distances, and the processed signal is then subjected to a Bayesian classifier for the classification of driving instances as safe or unsafe.

2.6.1 Analysis

Most studies here concern driver profiles, with several types of classification such as Aggressive driving, Normal driving, Calm driving. To the best of our knowledge, there is no work concerning road profiles.

Chapter 3

Signal/Range Correlation and RSU Failure Detection

In this section we detail our failure detection technique based mainly on checking if the coverage which should be ensured by an RSU is effective or not. For this issue, we propose a new C-ITS message denoted "Alarm Message" which is sent by each vehicle entering in a zone covered by an RSU. The idea is to insert in this message some information about the observed signals by the vehicle mainly in the In-distance (ID) and the Out-distance (OD). The receiving RSU collects all received data and runs an on-the-fly analysis in order to observe if an anomaly has occurred.

3.1 Motivation

RSUs are one of the key actors in the C-ITS environment. They are stationary units installed on the side of roads, and act as access points for the vehicles to the infrastructure ensuring V2I communications. They are best useful and become essential in the case of a low density of C-ITS equipped vehicles [71]. In addition to the CAM and DENM transmissions, RSUs provide the vehicles with services that are implemented on the infrastructure. Services such as [1] :

1. TLM which is a service that administrates the generation and sending of SPATEM messages. Its objective is to control vehicle access to intersections and conflict zones. It sends safety information to the vehicles present in an intersection and informs them of the real time status of the traffic light and its future status as well as the time margin between the two.

2. RLT which is a service that manages the generation and reception of MAPEM messages. A MAPEM is a message containing a digital topology map of the geometry of an area. This topology describes lanes, crosswalks, conflict zones and permitted maneuvers.

3. IVI is a road sign service; it uses the IVIM message to provide information about physical or virtual road signs such as contextual speeds or road warnings, as well as the presence of roadworks.

Thus, an RSU in failure would mean that the infrastructure services would not be available in its area of supposed coverage (or partially unavailable), and it would also mean a degradation of the whole C-ITS system in the case of low density scenarios. Thus, to ensure a proper functioning of the system, RSUs must be monitored and their failures detected and reported in time.

3.2 Antenna theory

3.2.1 Bases of electromagnetism

In telecommunications, electromagnetic phenomenons are used to transmit signals. The signals are encoded using various standards. Electromagnetism in itself is the interaction of electric currents or fields, and magnetic fields. A charged particle produces an electric field, whereas a moving charged particle produces an electromagnetic field. When a charged moving particle is undergoing acceleration it produces an electromagnetic wave that propagates through space with the speed of light. electromagnetic waves have two characteristics wave length and frequency, the relation between the two is given by:

$$c = \lambda / f$$

Where c is the speed of light, λ the wave length and f its frequency, thus the larger the frequency is, the smaller the wavelength is and vice versa.

The electric field is a vector field that surrounds charged particles, and is defined by the force (in Newton) per unit charge. The electric field created by a charged point is given as [72]:

$$E = \frac{F}{Q} = \frac{Q}{4\pi\epsilon r^2}\hat{r}$$

Where E is the electric field (V/m), F is the electric force, Q is the single point charge, r is the distance, $\hat{r}$ is a vector along the r direction, and ϵ is the electric permittivity which is a constant.

Whereas a magnetic field is a closed loop vector field that surrounds moving electrical charges, meaning electrical currents. The magnetic field of a current is given as [72]:

$$H = \frac{I \times \hat{r}}{4\pi r^2}$$

Where H is the magnetic field (A/m), I is the current vector, $\hat{r}$ is the unit displacement vector from the current element to the field point, r is the distance of $\hat{r}$.

All electromagnetic phenomenons can be summarized in four simple equations regrouped by James Clerk Maxwell (1831–1879) [72]:

$$\nabla \times E = -\frac{dB}{dt} \tag{3.1}$$

$$\nabla \times H = J + \frac{dD}{dt} \tag{3.2}$$

$$\nabla \bullet D = \rho \tag{3.3}$$

$$\nabla \bullet B = 0 \tag{3.4}$$

where ρ is the charge density, D is the electric flux density, B is the magnetic flux density, and ∇ is a vector operator. Equation 3.1 means that moving a conductor through a magnetic fields produces a voltage which is directly proportional to the speed of the movement. Equation 3.2 means that during the charge or discharge of a capacitor, a magnetic field appears. Equation 3.3 means that an electric charge generates an electric field around it, and Equation 3.4 means that the divergence of a magnetic field is always equal to zero.

3.2.2 Antennas radiation patterns

An antenna is a fundamental component of telecommunications systems such as radio, cellular networks, Wi-Fi and satelite communications. It is a device that is used to transmit and receive signals in the form of electromagnetic waves. When used for transmission, it takes electrical signal and converts it into electromagnetic waves that propagate through the surrounding space. And when used for reception, it captures the electromagnetic waves from the environment and converts them back into electrical signals.

Overall, there exist three kinds of antennas based on radiation patterns. An isotropic antenna is a hypothetical one and radiates over all directions in the horizontal plane as well the vertical one, its radiation diagram resembles that of a sphere and can be seen in Fig.3.1. An omnidirectional antenna radiates over all directions in the horizontal plane as can be seen in Fig.3.2, it is commonly used in applications where signals need to be transmitted and received in multiple directions, an example of the omnidirectional antenna is the dipole antenna. Whereas a directional antenna radiates in the form of a concentrated blob over one direction as can be seen in Fig.3.3, examples of directional antennas include parabolic antennas and the phased array antenna.

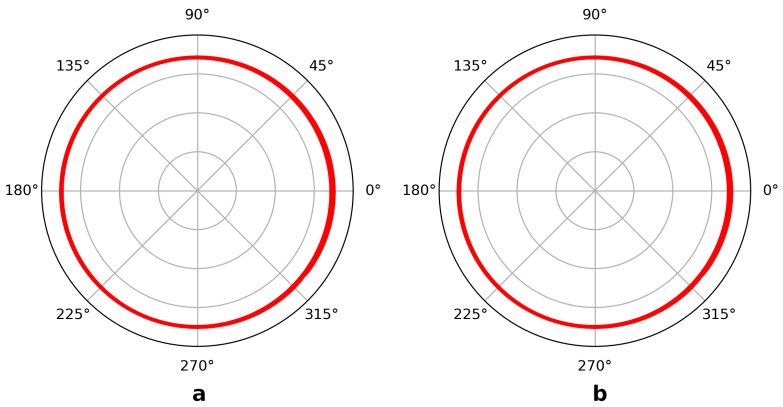

FIGURE 3.1: Radiation diagram for an isotropic antenna : a- vertical plane, b- horizontal plane.

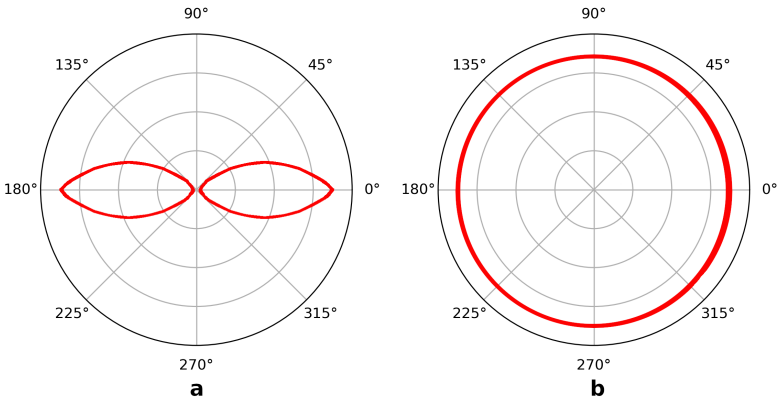

FIGURE 3.2: Radiation diagram for an omnidirectional antenna : a- vertical plane, b- horizontal plane.

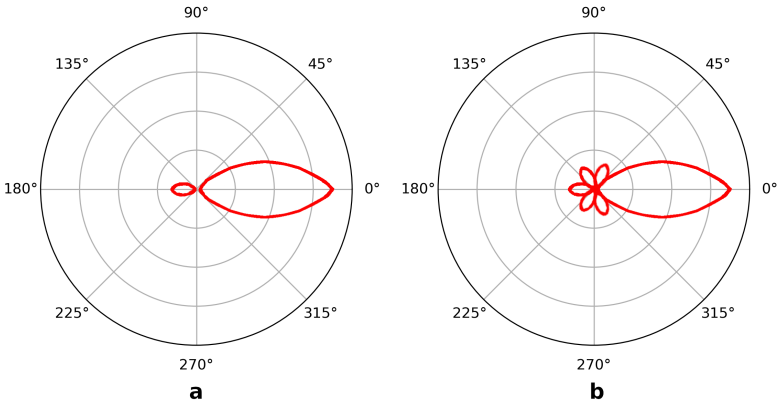

FIGURE 3.3: Radiation diagram for a directional antenna : a- vertical plane, b- horizontal plane.

The used RSUs in this study are either omni-directional or have each two directional antennas, each antenna is directed towards a side of the road and has a relatively high directivity. As long as the road is relatively parallel to the antennas lobes, the vehicle will be able to receive messages from the RSU. Our target is a specific failure which is when one of the two antennas fails, the bidirectional RSU will thus behave as a directional antenna and radiate over only one side of the road.

In order to detect failures in RSUs, we establish a method in which we compare the behaviors of functioning and failing RSUs in the next section. We consider two behavioral properties, the first one is the correlation between range and the signal intensity of communications received by the vehicle from an RSU. The second one is a particular property concerning the relation between the ID and OD.

3.3 Study environment

During the C-roads and the Indid projects (two Europen - Connecting Europe Facilities- projects), we have collected many datasets through real conditions on more than 300km highway roads in Italy, Germany and France. More than 1 Giga bytes of traces have been collected in pcap format produced by one driving vehicle and more than 50 Road Side Units, both in ITS-G5 and C-V2X. For each RSU, more than 20 Mega bytes have been collected in each experiment. This work however concerns only ITS-G5 RSUs, which have shorter ranges than their C-V2X counterparts. Whereas, this work has collected more than 10 000 messages that were sent from each of the concerned RSUs to the experimentation vehicle. Fig 3.4 shows an example of a path taken by the test vehicle in the city of Reims in France, getting in contact with four RSUs.

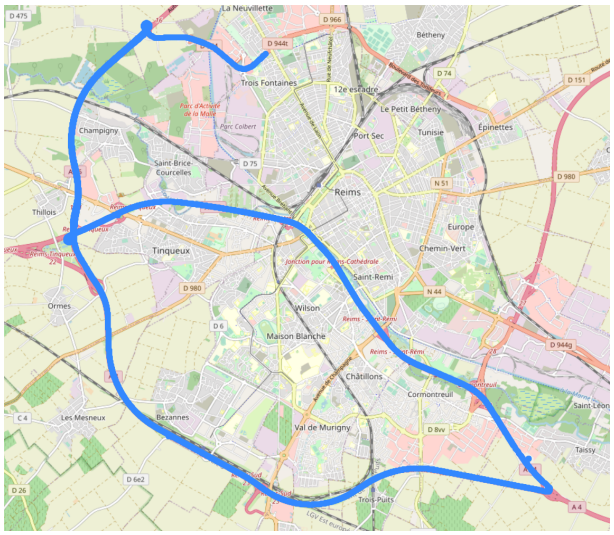

FIGURE 3.4: The path that the test vehicle traveled in the city of Reims, France

In the experiment, each captured packet comes with a radiotap information layer. Radiotap is a protocol which is the standard for the 802.11 frame injection and reception. It doesn't belong to the original ITS-G5 protocol stack, but is created at the moment of capture of a packet. The test vehicle that was used had the ability to generate the radiotap layer concerning the captured packet. The recorded information in the radiotap layer, concerns the radio transmission, such as the signal intensity at the moment of capture. If the packet was sent from the vehicle, then the transmission power (in dBm) is stored, and if the packet was received then the signal intensity at the reception moment is stored. We're interested in the latter since it gives us the signal intensity of transmissions coming from the RSU.

3.3.1 RSU packets characterization

One way to determine if the packet was either sent or received by the vehicle without looking at the source address is via radiotap. The difference between a sent and received packet is the format of the signal intensity, in the first it is denominated transmission power and concerns the signal power that was used to send the packet from the vehicle, whereas in the latter it is denominated antenna signal and concerns the transmission power at the moment of reception of the packet by the test vehicle.

We then needed to identify which packets were sent by an RSU in contact with the vehicle. Assuming that each RSU has sent at-least one CAM message we search through all CAM instances for the field stationType. The stationType of a vehicle is equal to 5 whereas that of an RSU is equal to 15. We can therefore identify all CAMs that were sent by RSUs and save their MAC addresses. We use the MAC address rather than the CAM's StationID for identification because the stationID is supposed to change each 10 minutes, whilst the MAC address remains static during the experiment.

An RSU could also be identified through its positioning. A car moves, whereas an RSU remains static, therefore if a station doesn't change its position in the ITS or the GeoNet layers in all records, it is assumed to be an RSU. Concerning the pcap data set, and so far, we verified that all static position stations have sent atleast one CAM with a stationType of 15.

3.3.2 RSU coverage

A vehicle is assumed to be under the coverage of an RSU once it starts receiving packets from it until it stops. This method may give false indications as the vehicle could leave the coverage area and enter once again, having thus two coverage periods or more and still be considered as one single coverage. This verification is however not needed since we only consider the packets sent by an RSU in the analysis and the coverage remains irrelevant for the study topic. The comparison between the ID and OD is also taken from the absolute point of view (meaning taking the possible multiple coverages as one coverage). Therefore, coverage from the point of view of the vehicle as a metric is dismissed.

To calculate the ID and OD from the pcap files, the Euclidean distance D between two points on the earth's globe is calculated as follows :

$$D = 6378 * arccos[(sin(lat_1) * sin(lat_2)) + cos(lat_1) * cos(lat_2) * cos(lon_2 - lon_1)]$$

where 6378 is the radius of earth in kilometers, lat_1 and lon_1 are the longitude and latitude of the first point, and lat_2, lon_2 are those of the second point.

3.4 RSU failure detection

Each figure from Fig. 3.5, Fig. 3.6, and Fig. 3.7 represent a set of captured packets by the vehicle from the RSU during the vehicle's travel in the area of coverage. Each packet here is represented by the duality (range, signal), where red dots represent the distance of the packets, and blue dots represent their signal intensity. In order to correlate between them, both these variables have been normalized between zero and one.

Overall, and for the ITS-G5, packets were captured from 23 RSUs from France, Germany and Italy. These RSUs are located both in urban and rural locations. Since the presence of buildings and other factors in the urban context causes signal attenuation, the RSU's range can be modified by these factors. Therefore, we chose the rural context for the comparative study of the RSUs.

When analyzing the data, we have observed several characteristics concerning the properly functioning RSUs. Mainly that the ID is usually larger than the OD, which is validated by [36]. This particular behavior can be observed in Fig.3.5 and Fig.3.6 which concern two proper functioning RSUs both in France, the first one is located in Reims, and the second one in Saint Maurice.

Whereas Fig. 3.7 represents the behavioral profile of a defective RSU located in Reims. This RSU is bidirectional but is behaving as a uni-directional RSU.

Another characteristic of a functioning RSU in this context is that range is inversely proportional to the intensity of the received signal. We verify this relation by calculating the Pearson's coefficient of all records of range and signal during the coverage period. Thus, well functioning RSUs should give a negative value below a certain threshold. The relation between distance and signal intensity of RSUs that are assumed well functioning is not perfectly linear, as the factors that contribute to the signal attenuation are environmental such as air quality and the presence of various obstacles. Thus, the inversely proportional relation does not give a Pearson's coefficient that is strictly equal to -1. We therefore chose a certain threshold to determine the RSUs that are behaving properly and those which are not. And we should note that the Pearson's coefficient condition is applicable in both contexts rural and urban. We also consider that both distances should be superior to 50 meters for rural RSUs, otherwise it is considered in failure.

Pearson's coefficient between two variables X and Y is calculated as follows :

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y}$$

Where $cov(X,Y)$ is the covariance of variables X and Y and σ_X, σ_Y their standard deviations.

Fig. 3.5 deals with the first discussed RSU in the city of Reims. We can observe here that both characteristics are present, the calculated ID is 1229 meters, which is the largest distance recorded in the set. And the OD is 671 meters. In terms of the relation between distance and signal, we can observe in the figure that the signal tends to increase when the distance becomes shorter. The Pearson's coefficient in this set is equal to -0.69.

The RSU in Fig. 3.6 has a lower range than the previous one, thus fewer packets were captured. But the pattern remains the same, the ID of coverage was calculated to be 407 meters, whereas the OD was 300 meters. The inverted relation between distance and signal is more visible here, and the Pearson's coefficient calculated here is equal to -0.83.

For the defective RSU in Fig. 3.7 however, the ID is way smaller than the OD. In fact, this RSU is located on the side of the road that the vehicle was travelling in. The first contact the vehicle had with this RSU was after it had surpassed it on the road by around 297 meters, which is the recorded ID, while the OD is at 1572 meters. The relationship between signal intensity and range is also not as in the case of previously discussed RSUs, we see in the figure that in the first 300 packets, the signal and range have a direct relationship, whereas in the last packets, the relationship becomes inverted. The Pearson's coefficient for this set is equal to 0.01. This means that the rule of correlation could not be applied in this context. This RSU is definitely in failure.

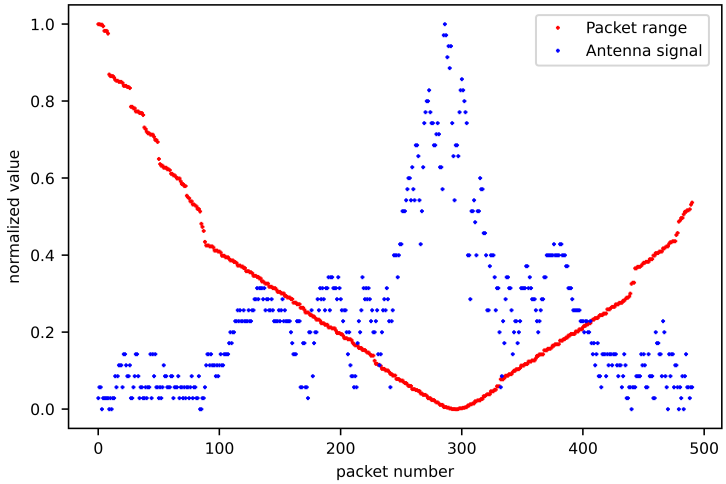

FIGURE 3.5: The scheme of the correlation between signal and distance for a normal RSU in Reims

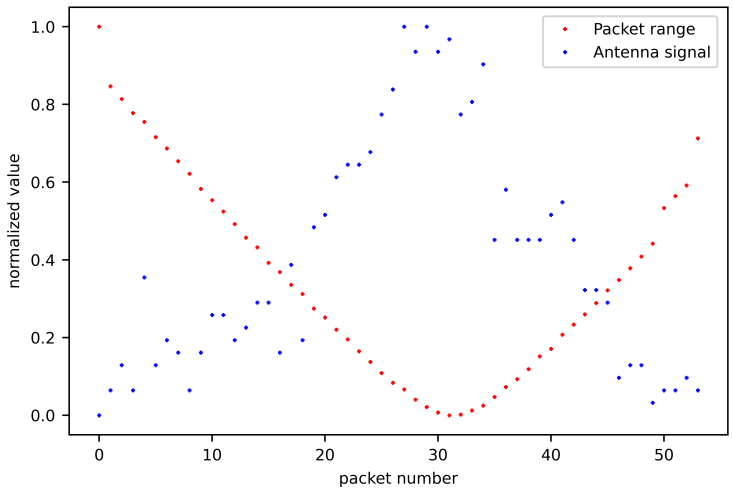

FIGURE 3.6: The scheme of the correlation between signal and distance for a normal RSU in Saint Maurice, Paris

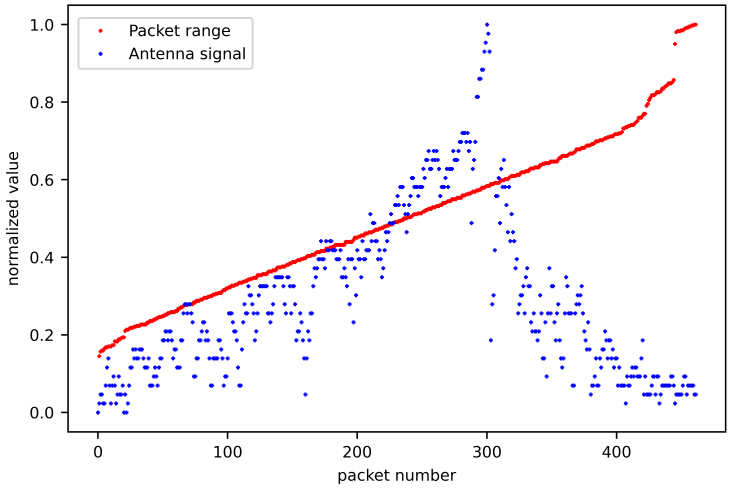

FIGURE 3.7: The scheme of the correlation between signal and distance for a failing RSU in Reims

The proposed method to determine failure in the case of omni-directional and bi-directional RSUs is to verify the properties:

- the ID is greater than the OD.
- the Pearson coefficient of the coverage data is less than a certain threshold. In our case, we choose it to be less than -0.4.
- the coverage should be larger than a threshold, and we choose the threshold to be 50 meters.

In table 3.1, is the summary of all calculations made in regards to the RSUs. For each RSU, range calculations are given with remarks on both characteristics as well as the geographical context. The RSUs in Fig.3.5, Fig.3.6 and Fig.3.7 are numbered 4, 8 and 2 respectively.

In addition to that, we have used the K-Medoïds algorithm in order to classify the behavior of RSUs using the variables : range, signal strength, vehicle speed, and the Pearson's coefficient of the signal to range correlation. The latter variable is repeated for all packets from the same RSU. The classification results are shown in Fig. 3.8, where the orange cluster regroups the packets that are transmitted from a failing RSU. The Medoïds of each cluster are represented by the cyan larger dots. This classification will be enhanced in order to give more precise classification which will be used in the future on all vehicles in order to be able to detect non proper RSUs.

TABLE 3.1: Recorded communications with the RSUs

RSU	context	max range	in-distance	out-distance	Pearson's coef	coef condition	ID > OD cond
1	Rural	979 m	979 m	393 m	-0.6048	correct	correct
2	Rural	1572 m	297 m	1572 m	0.0108	uncorrect	uncorrect
3	Rural	1546 m	1546 m	1112 m	-0.5705	correct	correct
4	Rural	1229 m	1229 m	671 m	-0.6921	correct	correct
5	Rural	2871 m	2871 m	1139 m	-0.6213	correct	correct
6	Rural	502 m	447 m	502 m	-0.6198	correct	uncorrect
7	Rural	580 m	580 m	398 m	-0.807	correct	correct
8	Rural	407 m	407 m	300 m	-0.8393	correct	correct
9	Urban	501 m	501 m	467 m	-0.1690	uncorrect	correct
10	Urban	502 m	502 m	436 m	-0.3441	uncorrect	correct
11	Urban	289 m	289 m	225 m	-0.5013	correct	correct
12	Urban	296 m	296 m	220 m	-0.7935	correct	correct
13	Urban	218 m	155 m	197 m	0.8439	uncorrect	uncorrect
14	Urban	280 m	196 m	280 m	0.1489	uncorrect	uncorrect
15	Urban	350 m	329 m	241 m	-0.7976	correct	correct
16	Urban	456 m	320 m	15 m	-0.5148	correct	correct
17	Urban	1023 m	1023 m	936 m	-0.3466	uncorrect	correct
18	Urban	537 m	537 m	503 m	-0.0266	uncorrect	correct
19	Rural	1857 m	548 m	1857 m	-0.6720	correct	uncorrect
20	Rural	410 m	410 m	100 m	-0.4571	correct	correct
21	Urban	249 m	249 m	214 m	-0.5072	correct	correct
22	Urban	1542 m	202 m	667 m	-0.5235	correct	uncorrect
23	Urban	709 m	709 m	48 m	-0.1011	uncorrect	correct

Then an alarm will be sent through a specific message defined in the next section.

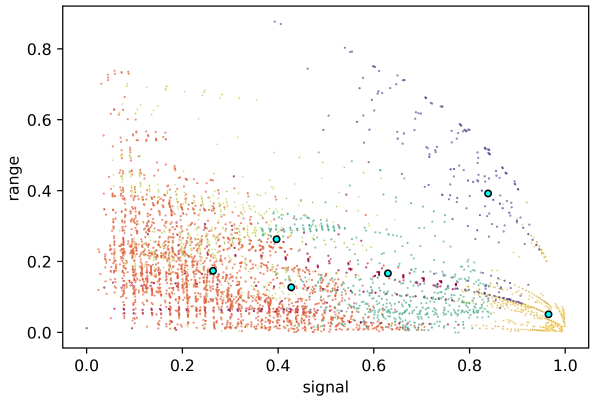

FIGURE 3.8: Clustering using range, signal, speed, and Pearson's coefficient in function of range and signal with normalized values

3.4.1 Alarm message

In this section we detail our proposed message which permits to raise an alarm about RSU failures. This message is composed of the timestamp and the location when the RSU is met the first time and the timestamp and the location when it is met the last time. The vehicle records the timestamp and location of received

messages from the RSU and filters the last one and the first one. As an option, we add the signal strength RSSI. This message has to be sent to the road operator using ITS-G5 or C-V2X channels. Contrary to usual C-ITS messages, this message will target the RSU which has been analysed. It will be sent through a uni-cast protocol. If ITS-G5 is used, that means the message will be forwarded by vehicles driving in the opposite roads. The structure of the packet is shown in Fig.3.9.

Message-type	Version	Sender-id	In-timestamp	In-location	Out-delta-time	Out-location	RSU-Id	In-RSSI	Out-RSSI
1 byte	1byte	2 bytes	5 bytes	4 bytes	2 bytes	4 bytes	2 bytes	2 bytes	2 bytes

FIGURE 3.9: The structure of the alarm packet

A vehicle builds this message each time it meets an RSU. At that moment, it collects beacons and CAM messages sent by the RSU and keeps the first message and the latest message. When a new message is received, the former one is dropped. When no message is received from the RSU after a time delay, the previous one is considered as the last one.

If an RSU overlaps another RSU, the vehicle filters CAMS and beacons sent from both RSUs in order to be able to alert each RSU with precision and efficiency. The algorithm 1 gives the instructions that a vehicle should follow to report RSUs status.

Algorithm 1 Create Alarm message

Input: $RSUs[RSU_ID$,$in_location$, $out_location$, in_RSSI, out_RSSI,$RSU_Time]$ list of all RSUs with their attributes, $timeThreshold$ timestamp, $currentTime$ timestamp, m last received C-ITS message

```
if m.origin == RSU then
  if m.RSU_ID NOT IN RSUs[RSU_ID] then
    /* First message from this RSU */
    RSU ← createRSU()
    RSU.in_location ← getVehicleLocation();
    RSU.in_RSSI ← m.RSSI
    RSU.RSU_ID ← m.RSU_ID
    RSUs.add(RSU)
  else
    /* Second message from this RSU */
    RSU ← RSUs.get(RSU_ID == m.RSU_ID)
    RSU.out_location ← getVehicleLocation();
    RSU.out_RSSI ← m.RSSI;
    RSU.RSU_Time ← m.time
  end if
end if
for RSU in RSUs do
  if RSU.RSU_Time IS NOT NULL then
    T = currentTime − RSU.RSU_Time
    if T ≥ timeThreshold then
      sendAlarmMessage(RSU);
      RSUs.remove(RSU_ID == m.RSU_ID)
    end if
  end if
end for
```

The road operator has the list of all RSUs in any area including their geographical context which could be rural or urban. Each RSU is associated with a score determining its status. If the score for an RSU is high it means that the RSU is properly functioning and if it is negative and low it means that the RSU is in failure. If the score is around zero, it means that the RSU is either urban or there were not enough alarm messages reported by end-users. The algorithm 2

gives the instructions that the road operator follows to treat the upcoming alarm messages. Upon the reception of an alarm message, the geographical context of the concerned RSU is verified, if it is rural, then the ID and OD are compared and when the ID is significantly less than the OD then we are in the case of an RSU failure such as in Fig.3.7 and the score of the concerned RSU is decremented by one, else it is incremented by one. Last, and whether the RSU is rural or urban, its overall range is verified, if it does not exceed a minimal threshold then the score is also decremented. Once an RSU's score reaches the minimal threshold, then the road operator considers that enough vehicles reported that the RSU is in failure and reports it. A time window is added so that the scores do not get saturated, at the beginning of each time window the scores are set to zero. The window is shifted over the time and its width could be fixed depending on the whole environment.

Algorithm 2 Alarm message treatment

Input: $RSUs :< RSUID, Score >$ list of RSUs with their scores, $FailureThreshold$, $rangeThreshold$, $maxScore$, $time$ $timeWindow$, dt, $distanceThreshold$

Output: $time$

```
m ← receiveAlarmMessage;
in_distance ← calculateRange(m.RSUID, m.in_location)
out_distance ← calculateRange(m.RSUID, m.out_location)
if m.RSUID is Rural then
  /* no obstacles, not urban */
  if in − distance < (out − distance + rangeThreshold) then
    /* reduce the score by one */
    if RSUs[RSUID].Score > −maxScore then
      RSUs[RSUID].Score− = 1
    end if
  else
    if RSUs[RSUID].Score < maxScore then
      /* increment the score by one */
      RSUs[RSUID].Score+ = 1
    end if
  end if
end if
if in − distance < distanceThreshold or out − distance < distanceThreshold
then
  /* failure */
  if RSUs[RSUID].Score > −maxScore then
    RSUs[RSUID].Score− = 1
  end if
end if
for R in RSUs do
  if R.Score < FailureThreshold then
    Report(R.RSUID);
  end if
end for
time ← time + dt
if time ≥ timeWindow then
  for R in RSUs do
    R.Score ← 0
  end for
  time ← 0
end if
```

From the point of view of vehicles, Fig.3.10 illustrates the steps that the vehicle takes considering the alarm message: a) At the contact of an RSU, a vehicle starts building the alarm message b) it finishes building the alarm message after quitting the zone of coverage of the RSU, c) then, upon the encounter of other vehicles on the road it sends the message in broadcast mode to inform them of the status of the RSU, the vehicles themselves will broadcast the message in multi-hop fashion, d) upon the encounter of another RSU it disseminates the message in V2I fashion to the road operator.

3.5 Conclusion

In this chapter, detection of failures on operational RSUs by end-users is investigated. This issue is critical and crucial for the C-ITS. We have shown that maintenance of Road Side Units could be achieved in a simple way without any additional investment or any equipment. Vehicles collect RSU observations during their journeys and analyse them using our failure detection mechanism. This mechanism compares the observed behavior to the known proper one, when an anomaly is detected, any vehicle raises alerts to the road operators in order to inform about the suspicious RSU. RSU behavior profiles are studied as well and a difference between urban and rural is drawn and carefully considered in the study.

We have observed that in the case of a functioning RSU, there is an acceptable inverse correlation between the range and signal intensity of its transmissions. We have also observed that in the rural context with line of sight communications, the ID tends to be larger than the OD for functioning RSU. Both these properties were used for our assessment of RSUs.

We have so far addressed one particular RSU failure, which is when the RSU radiates only over one direction rather than two directions. But there are other kinds of failures, for instance the RSU could be completely dysfunctional in both directions. One way to detect that by the vehicles is to receive an alarm message about an RSU from a single vehicle and then the informed vehicles find no RSU on the road (because it became faulty recently), by this, they can detect that there is an RSU in the designated area that went dysfunctional, thus they can report it also. Software failures are also a possibility, a malformed packet from an RSU could occur, we could extend our failure mechanism by calculating the ratio of malformed packets from an RSU, if it exceeds a certain threshold it would mean that the RSU is in failure.

The rest of this chapter is covered by the following article:

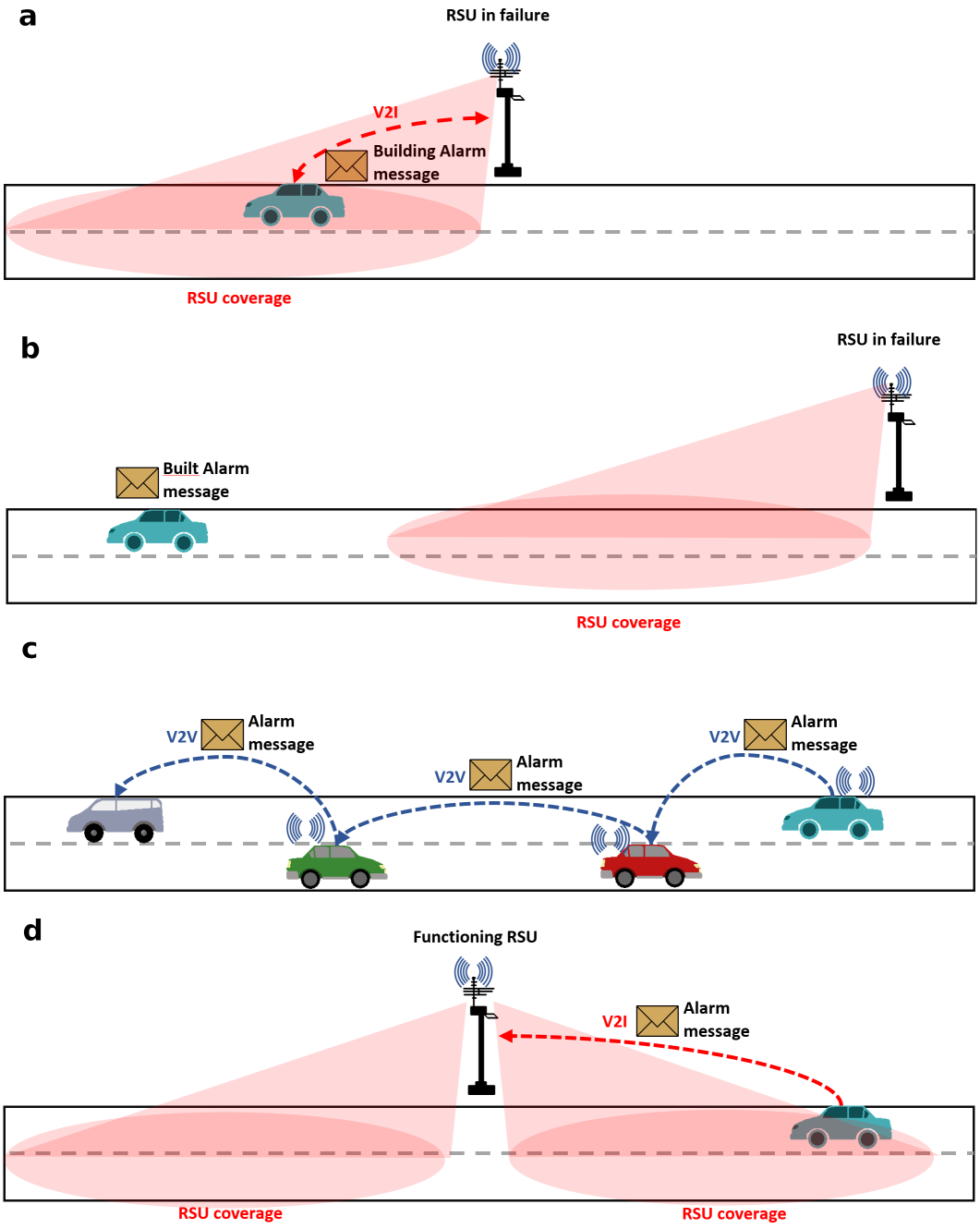

FIGURE 3.10: Treatment of the alarm message by the vehicles, a) at the coverage of an RSU, the vehicle starts building the alarm message, b) the vehicle finishes building the alarm message after leaving the coverage of an RSU, c) the message is broadcast in multi-hop fashion by the vehicles to inform each other of the status of the RSU, d) the vehicle sends the alarm message to the next RSU it encounters

Benzagouta, M.-L.; Aniss, H.; Fouchal, H.; El Faouzi, N.-E. Road-Side Unit Anomaly Detection. Vehicles 2023, 5, 1467-1481. https://doi.org/10.3390/vehicles5040080

Chapter 4

Road profiles

4.0.1 Introduction

This chapter covers a work done on the database of the CoopITS smartphone application. Road profiles are derived by observing and analyzing the information found in the CAM messages sent by the different users. Thereafter, four clustering techniques are applied to the paths of individual cars. A road profile is defined by the speed of the vehicle over time.

The first step in our process involves using road profiles to track the speed of every vehicle on a certain road segment. These route profiles provide a basis for what we will do next. We categorize trajectories by using clustering methods and arranging them according to common features and patterns. The method of clustering improves the entire analysis's understanding by facilitating a deeper comprehension of the many movement patterns included in the road data. The results of the clustering algorithms are then compared. In order to be proactive, we raise alarms when we detect abnormal trajectories. This allows us to promptly notify users and the road operator of any potential anomalies in the movement of vehicles.

4.1 Data visualisation using T-SNE

t-distributed Stochastic Neighbor Embedding (T-SNE) [73] is a machine learning nonlinear method for the reduction of dimensionality of data sets. It is a statistical method for visualizing high-dimensional data by giving each datapoint a location in a two or three-dimensional map.

The key idea behind t-SNE is to model the similarity between data points in the high-dimensional space and the low-dimensional space, preserving the pairwise

similarities as much as possible. It achieves this by defining a probability distribution over pairs of high-dimensional data points and a similar distribution over the low-dimensional counterparts. The algorithm then minimizes the divergence (Kullback-Leibler divergence) between these two distributions.

4.2 Trajectory analysis

Here, we use the CoopITS database, we are mainly interested in the CAM messages which are periodically sent by each C-ITS station. The frequency of the CAM message in C-ITS standards is of between 100ms and 1s depending on the velocity, but its frequency in the CoopITS application is either 1 s or 5 s.

A CAM message contains information about the station's state in an instant t, such as its velocity, heading and position in means of longitude and latitude. Each station has a unique pseudonym called the stationID which is attributed randomly and gets updated every 10 minutes for anonymity.

Having the set of CAM messages, we wanted to find a specific pattern, which is a trajectory that was traversed several times, and from the dataset containing a total of 1 336 573 CAMs coming from 8 667 distinct stationIDs from the region "Nouvelle-Aquitaine". At the time of the study we identified a single trajectory that was traversed 71 times (71 distinct stationIDs) in the same direction in the city of Bordeaux which contains a total of 46 570 CAM messages.

Fig. 4.1 shows the trajectory that was chosen. The trajectory starts south and goes to the north and was plotted on an OSM map in python using the folium library where each dot represents the position recorded in each CAM message from a single trip (a single stationID).

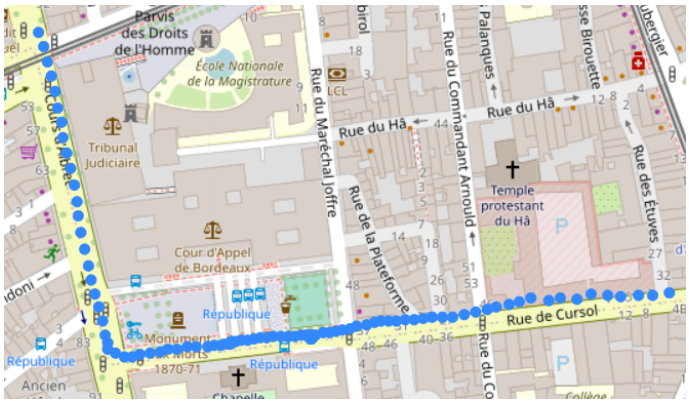

FIGURE 4.1: An example of the trajectory that was extracted from CAM messages, plotted on top of an OSM map

We then cut the trajectory into 10 elementary road segments, a segment is the portion of road between two successive road crossings. Fig. 4.2 shows the segments that were cut from the trajectory.

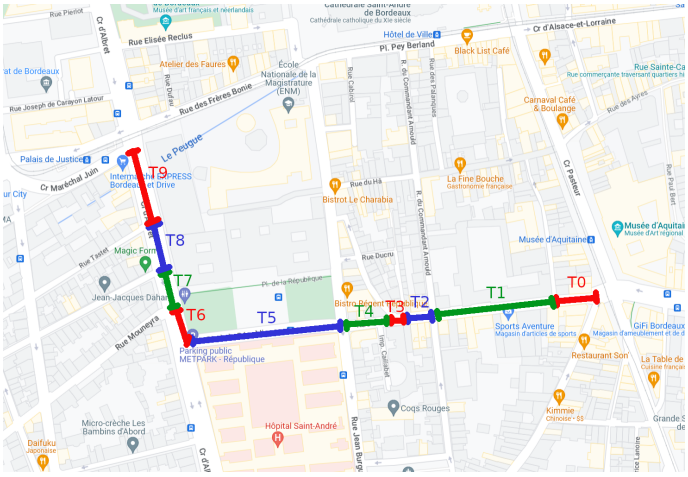

FIGURE 4.2: The determined segments, plotted on Google maps

To determine which CAM point (latitude and longitude) belong to which segment, and as in figure 4.3 we draw a rectangle out of the segment, its length is the length of the segment and its width is of 20 meters, if the point belongs to the rectangle it gets attributed to it. We chose a large width due to GPS precision issues, as a large portion of the CAM points were out of the borders of the road.

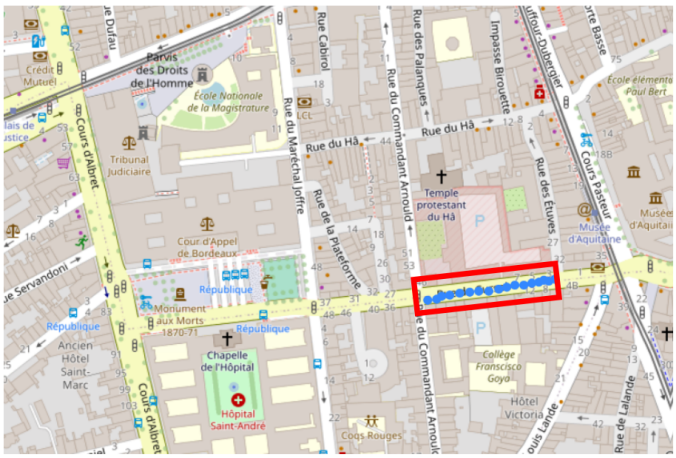

FIGURE 4.3: The rectangle over segments, plotted on an OSM map

After doing so, we ended up with 609 distinct trajectories, 71 multiplied by 10 gives 710 which should be the number of trajectories to end up with, but we had 609 and this is due to the frequency of the CAM messages, certain road segment are small and don't fit to a single CAM point.

609 different trajectories have been collected. They have used as an input for each of the four algorithms.

The obtained trajectories have different dimensions so we can't feed them into clustering algorithms which require that all observations have the same dimensions. So we converted each trajectory into a 60 dimension observation by projecting 60 points on the curve of velocity of each trajectory as can be seen in .Fig4.4.

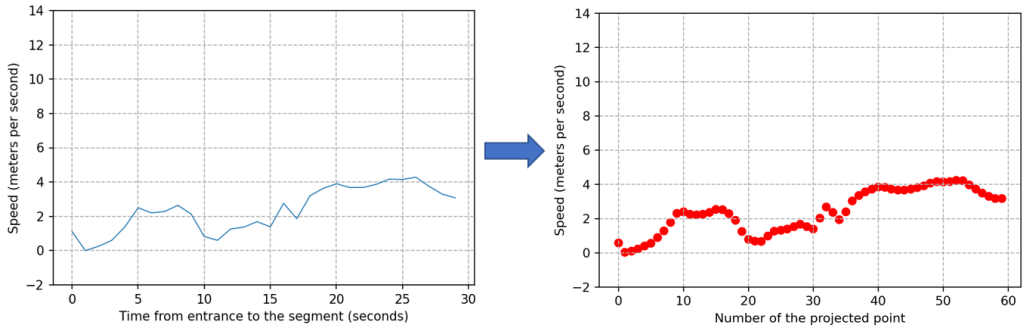

FIGURE 4.4: An example of a normalized trajectory points (60 points)

4.2.1 k-means analysis

We have run the k-means algorithm on the trajectory data. We have chosen 5 clusters, Table 4.1 shows the number of classification per cluster, the cluster 3 is the largest as it has 181 observations. The cluster 4 contains the least number of observations. In Fig. 4.5 we see a sample of trajectories in cluster 0 and it is equivalent to a long stop, this is in comparison with cluster 2 in Fig. 4.7 which contains trajectories with a short stop. We can see also that cluster 1 in Fig.4.6, cluster 3 in Fig .4.8, and cluster 4 in Fig .4.9 equivalent to a free flow without a stop, with velocities ranging from low in cluster 3, medium in cluster 1 and high with cluster 4. A total of all clusters visualized using T-SNE is in Fig. 4.10.

TABLE 4.1: Kmeans: number of trips for each cluster

cluster	nbrTrips
0	137
1	112
2	92
3	181
4	87

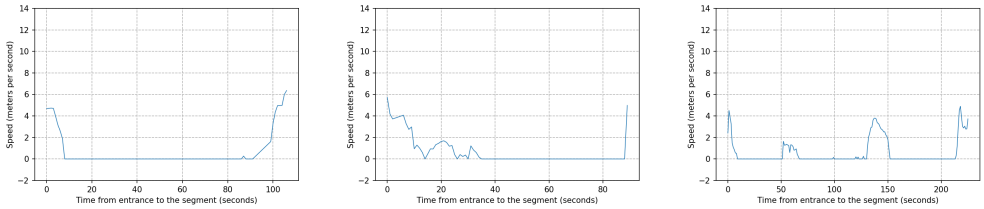

FIGURE 4.5: Samples of trajectories from cluster 0 for the kmeans method.

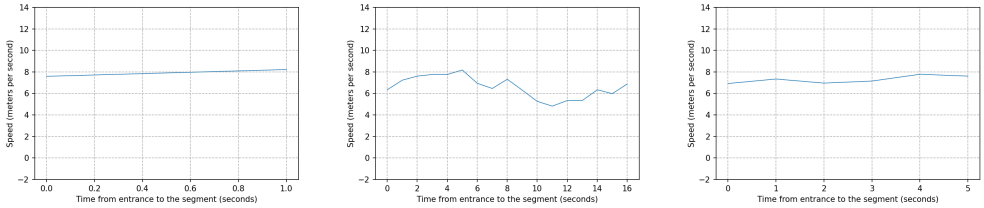

FIGURE 4.6: Samples of trajectories from cluster 1 for the kmeans method.

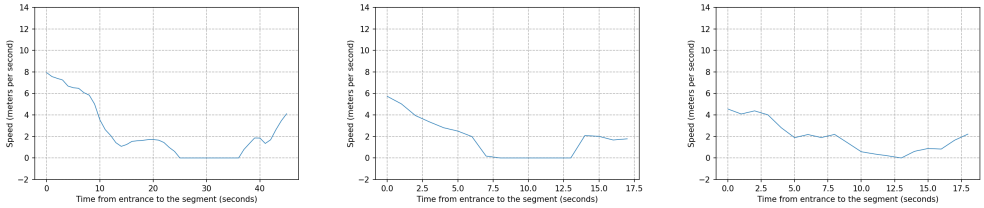

FIGURE 4.7: Samples of trajectories from cluster 2 for the kmeans method.

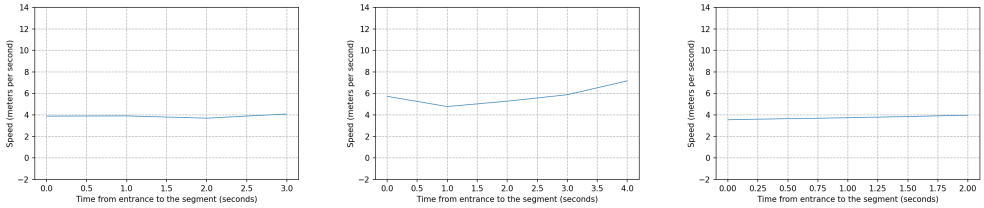

FIGURE 4.8: Samples of trajectories from cluster 3 for the kmeans method.

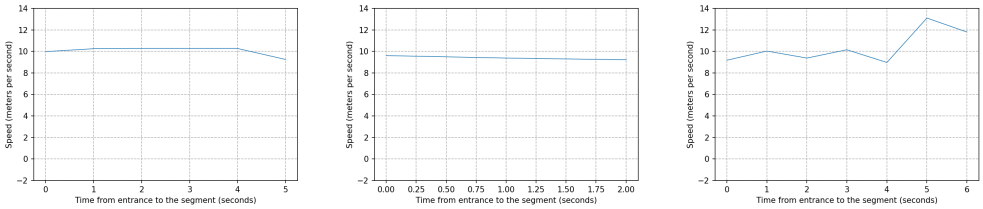

FIGURE 4.9: Samples of trajectories from cluster 4 for the kmeans method.

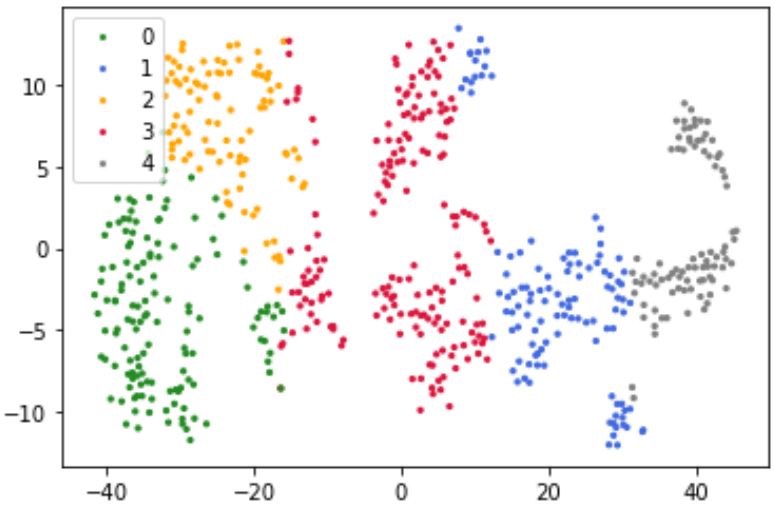

FIGURE 4.10: K-means clusters visualized using T-SNE.

4.2.2 Agglomerative clustering analysis

We have run the Agglomerative Clustering algorithm on the trajectory data and chose 5 clusters. As can be seen in Table 4.2 cluster 0 has the largest number of samples with 219 observations, and with cluster 2 having the lowest number of trajectories. As can be seen in Fig.4.13, cluster 2 contains trajectories with a short stop, while Fig.4.15 shows samples of cluster 4 which contains trajectories with a long stop. Cluster 0 in Fig. 4.11 contains trajectories with low velocity free flows or with very short stops. Cluster 1 in Fig .4.12 contains trajectories with free flow with high velocity. Cluster 3 as in Fig .4.14 contains trajectories with free flow medium velocities. A total of all clusters visualized using T-SNE is in Fig. 4.16.

TABLE 4.2: Agglomerative clustering: number of trips for each cluster

cluster	nbrTrips
0	219
1	111
2	85
3	92
4	102

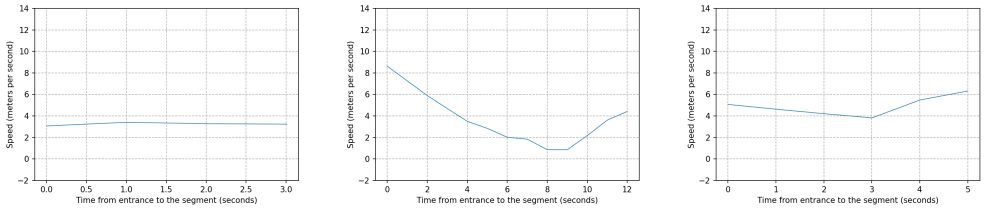

FIGURE 4.11: Samples of trajectories from cluster 0 for the Agglomerative clustering method.

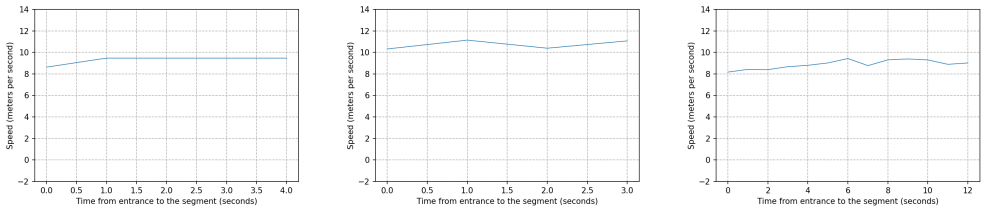

FIGURE 4.12: Samples of trajectories from cluster 1 for the Agglomerative clustering method.

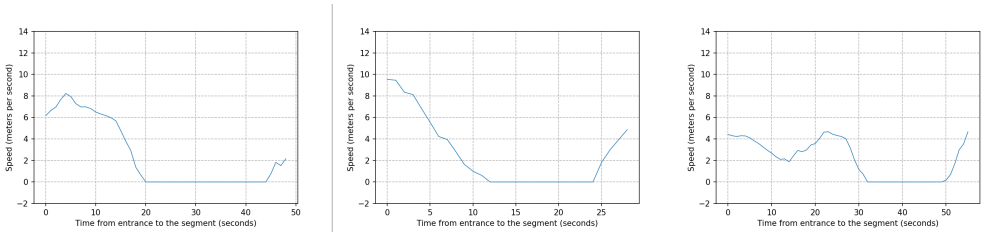

FIGURE 4.13: Samples of trajectories from cluster 2 for the Agglomerative clustering method.

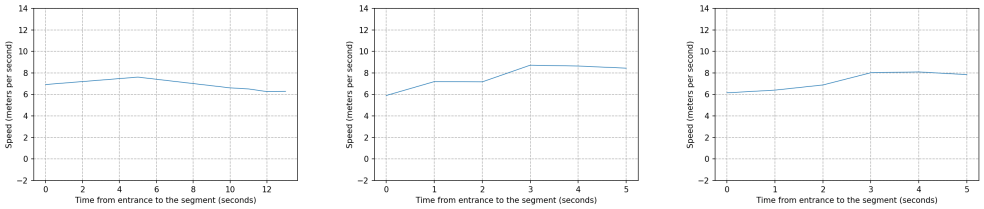

FIGURE 4.14: Samples of trajectories from cluster 3 for the Agglomerative clustering method.

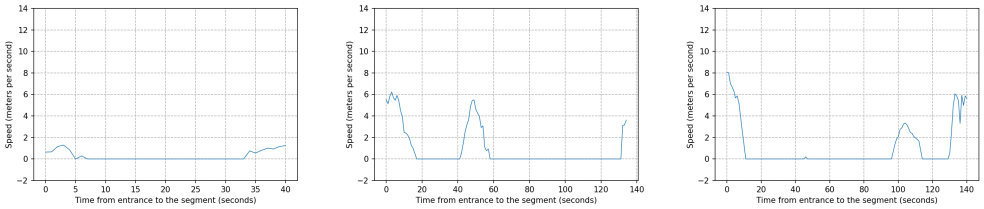

FIGURE 4.15: Samples of trajectories from cluster 4 for the Agglomerative clustering method.

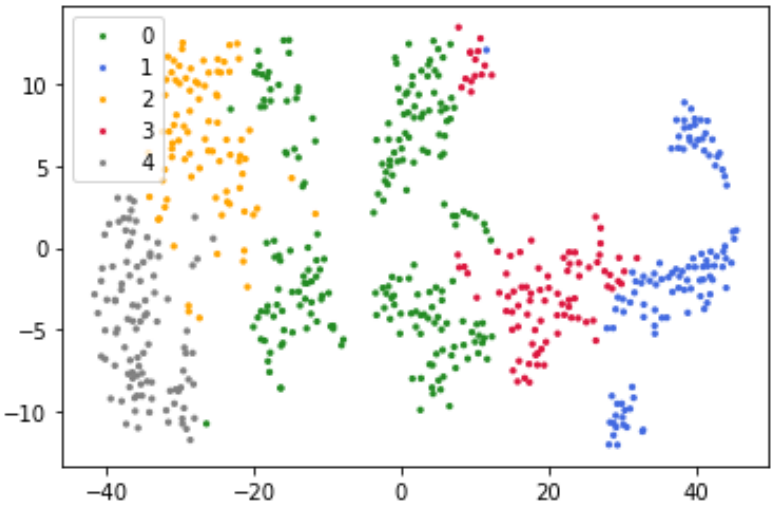

FIGURE 4.16: Agglomerative Clustering clusters visualized using T-SNE.

4.2.3 BIRCH Clustering analysis

We have run the BIRCH algorithm and chose 5 clusters. As can be seen in Table 4.3 cluster 0 has the largest number of observations and cluster 3 has the lowest number of observations. Clusters 2 and 4 in Fig. 4.19 and Fig. 4.21 respectively contain trajectories with a stop in the segment, cluster 2 has shorter stops than

cluster 4. Cluster 0 as in Fig. 4.17 contains trajectories with free flow and low velocity, whereas cluster 1 in Fig. 4.18 contains free flow trajectories with medium velocity, and cluster 3 in Fig. 4.20 contains free flow trajectories with a high velocity. A total of all clusters visualized using T-SNE is in Fig. 4.22.

TABLE 4.3: BIRCH: number of trips for each cluster

cluster	nbrTrips
0	219
1	142
2	85
3	61
4	102

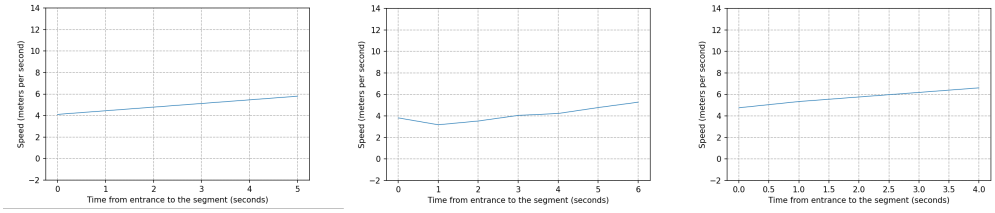

FIGURE 4.17: Samples of trajectories from cluster 0 for the BIRCH clustering method.

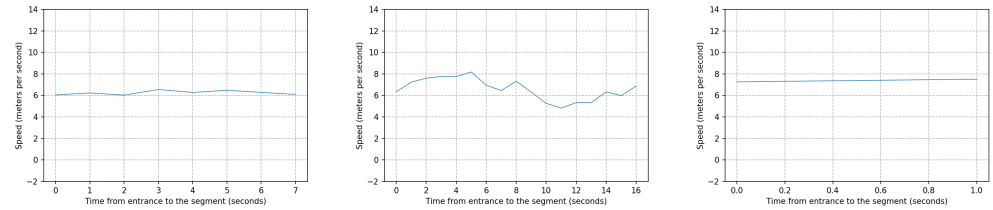

FIGURE 4.18: Samples of trajectories from cluster 1 for the BIRCH clustering method.

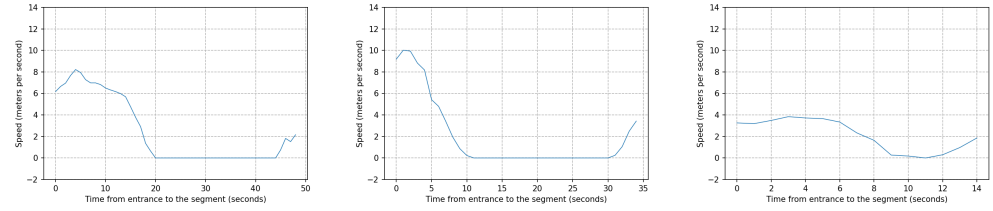

FIGURE 4.19: Samples of trajectories from cluster 2 for the BIRCH clustering method.

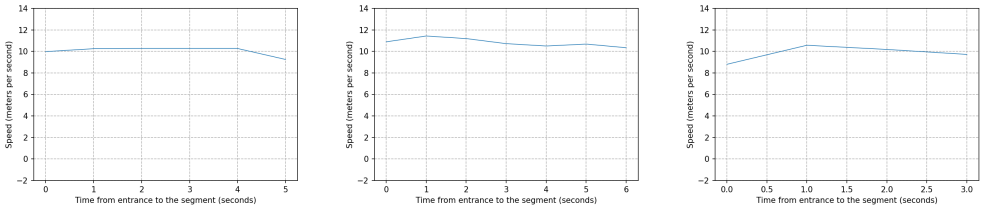

FIGURE 4.20: Samples of trajectories from cluster 3 for the BIRCH clustering method.

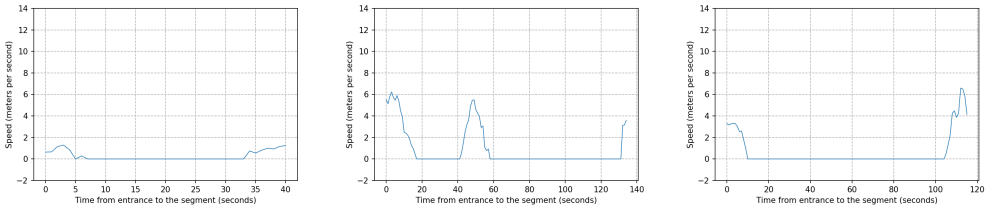

FIGURE 4.21: Samples of trajectories from cluster 4 for the BIRCH clustering method.

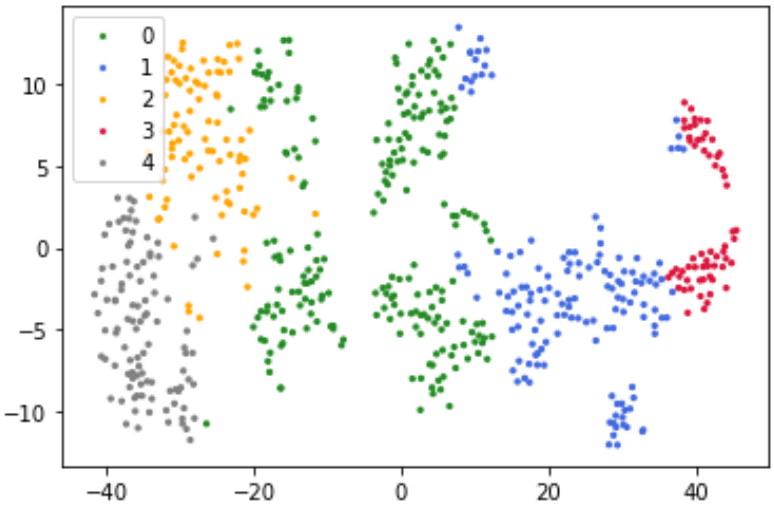

FIGURE 4.22: BIRCH clusters visualized using T-SNE.

4.2.4 DBSCAN Clustering analysis

We have run the DBSCAN algorithm and we have obtained 6 different clusters. The cluster 0 contains 81% of the observations, and 15% of the observations are considered aberrant. Cluster 1 contains 7 observations and clusters 2, 3, 4, and 5 contain each 3 observations. The reason the cluster 0 contains 81% of the

observations is that the dataset is continually dense as can be seen in the T-SNE representation of the data in Fig. 4.23.

TABLE 4.4: DBSCAN: number of trips for each cluster

cluster	nbrTrips
-1	95
0	495
1	7
2	3
3	3
4	3
5	3

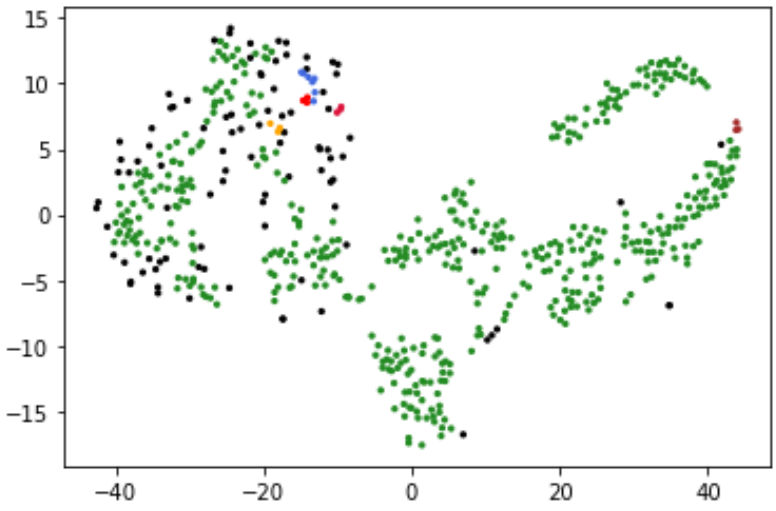

FIGURE 4.23: DBSCAN clusters visualized using T-SNE

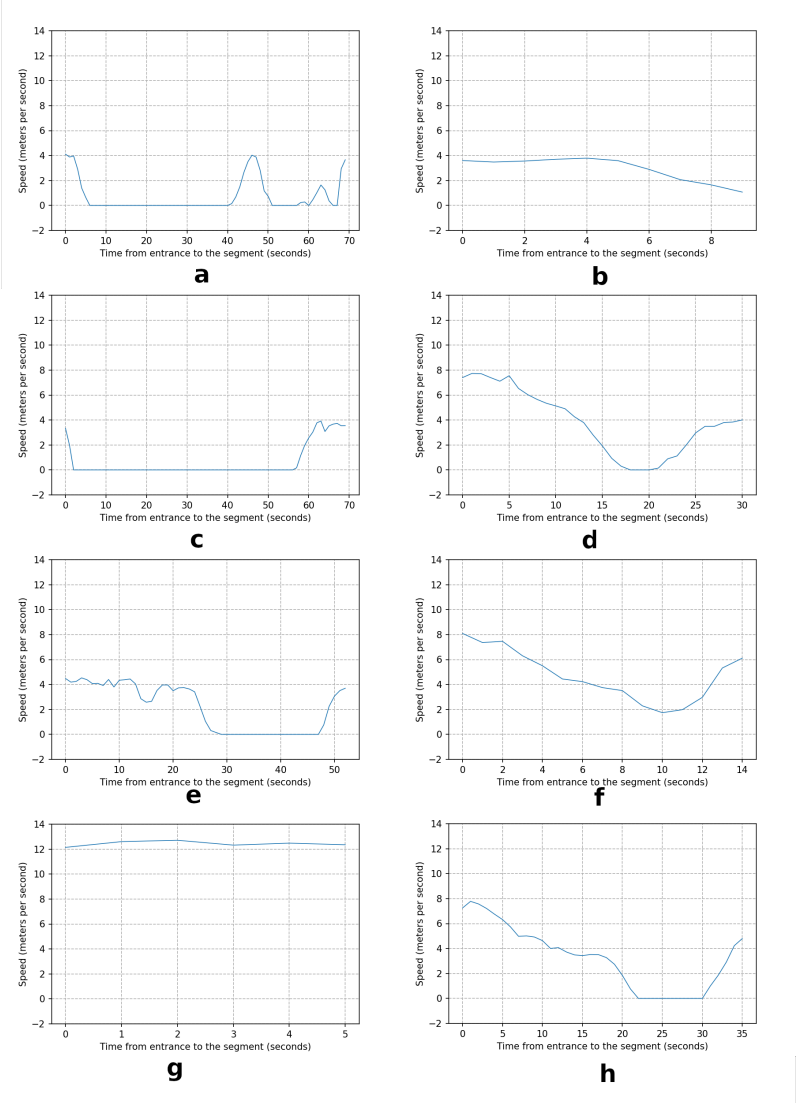

FIGURE 4.24: Samples of trajectories from clusters for the Dbscan clustering method : a) aberrant observation, b) cluster 0, c) another sample from cluster 0, d) cluster 1, e) cluster 2, f) cluster 3, g) cluster 4, h) cluster 5.

4.2.5 Combination of four algorithms experimentation

4.2.5.1 Combination of Agglomerative Clustering and K-means

Table. 4.5 shows a comparison between the AC and K-means, we notice that 175 observations are in clusters 0 and 3 in AC and K-means respectively, cluster 3 in k-means has 1 and 5 observations common with clusters 2 and 3 of the AC respectively, which allows us to safely say that clusters 0, and 3 in the AC and k-means are relatively equivalent which represent the free flow with low velocities. The same can be observed in regard to cluster 0 from the k-means, as it has 102 common observations with cluster 4 of the AC, and 16, 19 common observations

with clusters 0 and 2 of the AC respectively, thus we can assume that clusters 4 and 0 of the AC and k-means respectively relatively constitute the same cluster.

From a one sided perspective, we notice that cluster 4 of the k-means has 86 common observations with cluster 1 of the AC and has one common observation with cluster 3 of the AC. so we can assume that cluster 1 in AC mostly includes cluster 4 in K-means.

TABLE 4.5: Comparison between the AC and k-means

AC=Kmeans	number
0=3	175
0=2	27
0=0	16
0=1	1
1=1	25
1=4	86
2=2	65
2=0	19
2=3	1
3=1	86
3=4	1
3=3	5
4=0	102

4.2.5.2 Combination of BIRCH and Agglomerative Clustering

In Table. 4.6 a comparison between the BIRCH and AC algorithms. We observe that clusters 0 and 0 of the BIRCH and AC algorithms are equivalent and are in a single instance, thus it means that the clusters are the same. We observe the same thing with clusters 2 and 4 of both BIRCH and AC with 85 and 102 common observations respectively, meaning that cluster 0 is equal to 0 from the AC, cluster 2 is equal to 2 of the AC, and cluster 4 is equal to 4 from the AC. The resemblance between the results of both algorithms is because they're both hierarchical algorithms.

TABLE 4.6: Comparison between the BIRCH and AC

BIRCH=AC	number
0=0	219
1=1	50
1=3	61
2=2	85
3=1	92
4=4	102

4.2.5.3 Combination of BIRCH and Dbscan

In Table. 4.7 we see a comparison between the algorithms BIRCH and DBSCAN, and we see that cluster 0 of the DBSCAN is distributed between clusters 0, 1, 2, 3, and 4 of the BIRCH. Clusters 1, 3, and 5 of the DBSCAN are included in cluster 0, cluster 2 of DBSCAN is included in cluster 2 of the BIRCH and cluster 4 of DBSCAN is included in cluster 3 of the BIRCH.

TABLE 4.7: Comparison between the BIRCH and DBSCAN

BIRCH=DBSCAN	number
0=0	180
0=-1	26
0=5	3
0=3	3
0=1	7
1=0	138
1=-1	4
2=-1	36
2=0	46
2=2	3
3=0	57
3=4	3
3=-1	1
4=-1	28
4=0	74

4.2.5.4 Combination of BIRCH and K-means

In Table. 4.8 the comparison between BIRCH and K-means results. We observe that cluster 3 of the K-means and cluster 0 of BIRCH have 175 observations in common. Cluster 3 of k-means and cluster 1 of BIRCH have 5 common observations and also that cluster 3 of k-means and cluster 2 of BIRCH have one observation in common, which means that cluster 3 of K-means is almost included in cluster 0 of BIRCH. We also observe that cluster 1 of k-means is almost included in cluster 1 of BIRCH and have 111 common observations.

TABLE 4.8: Comparison between the BIRCH and K-means

BIRCH=K-means	number
0=3	175
0=2	27
0=0	16
0=1	1
1=1	111
1=4	26
1=3	5
2=2	65
2=0	19
2=3	1
3=4	61
4=0	102

4.2.5.5 Combination of Agglomerative Clustering and Dbscan

In Table. 4.9 a comparison between the algorithms DBSCAN and AC and we see that cluster 0 of the DBSCAN is distributed between clusters 0, 1, 2, 3, and 4 of the AC. Clusters 1, 2, 3, 4, and 5 of DBSCAN are distributed between (and included in) 0, 2, and 1 of the AC.

TABLE 4.9: Comparison between the AC and Dbscan

dbscan=AC	number
0=0	180
0=-1	26
0=5	3
0=3	3
0=1	7
1=0	105
1=-1	3
1=4	3
2=-1	36
2=0	46
2=2	3
3=0	90
3=-1	2
4=-1	28
4=0	74

4.2.5.6 Combination of Dbscan and K-means

In Table. 4.10 a comparison between DBSCAN and K-means. And we see that cluster 0 of the DBSCAN is distributed between clusters 0, 1, 2, 3, and 4 of the K-means. Clusters 1, 2, 3, and 4 of DBSCAN are distributed between clusters 0, 2, and 4 of the K-means

TABLE 4.10: Comparison between the DBSCAN and K-means

dbscan=Kmeans	number
-1=2	35
-1=0	41
-1=3	14
-1=1	4
-1=4	1
0=1	108
0=3	164
0=4	83
0=0	96
0=2	44
1=2	7
2=2	3
3=3	3
4=4	3
5=2	3

4.3 Conclusion

We have shown in this chapter that connected vehicles are an important source of data provision. This issue could be exploited positively to solve some troubles as abnormal situation detection without any additional device on roads. Indeed, in this study we have collected various trajectories on various roads at different periods. We aimed to classify all these trajectories into different clusters. Then each cluster has to be checked in order to understand what are the features of each cluster. We have applied 4 clustering algorithms in order to check the accuracy of each of them.

The main conclusion is that each algorithm has its own classification and the intersections between clusters from one algorithm to another are not empty. Afterwards, we have focused on the differences of the obtained results by each algorithm. These differences are of high interest since they express a kind of sub-classes having their own features.

The rest of this chapter is covered by the following article:

Benzagouta, Mohamed Lamine ; Aniss, Hasnaa ; Fouchal, Hacène ; El Faouzi,

Nour-Eddin. (2023). Towards Road Profiling with Cooperative Intelligent Transport Systems

Chapter 5

Impact of speed limitation on traffic

In this chapter, we are interested in intersections equipped with traffic lights. We study the speed profiles of drivers in the road segment prior to the intersection, and classify their behaviors into four repeating patterns.

5.1 GLOSA

GLOSA is a C-ITS use case. It concerns intersections equipped with a traffic light and indicates to the driver the current phase, time to the next phase and also a speed advice that allows the driver to avoid a stop in the case of a red light. It is based on two C-ITS services [1] which are the TLM and the RLT. The TLM sends safety information to vehicles in the nearby intersection, informing them of the real-time state of the traffic light and its future state, as well as the time margin between the two states. The message used to this extent is the SPATEM. It is sent periodically to all participants for as long as they are close to the intersection. The RLT sends a digital topology map of the geometry of an area to the nearby vehicles, describing lanes, crosswalks, conflict zones (intersections) and permitted maneuvers. A lane consists of several connected landmarks, and at the connection of these crossing points with a conflict zone, a set of authorized maneuvers is modeled as a connection between the two end points that connect the two lanes with the zone. The message that is used by the RLT is the MAPEM and is transmitted at the same time as a SPATEM when the vehicle approaches an intersection.

Among different joint initiative projects of European member states aiming at developing and experimenting innovative C-ITS solutions. C-Roads [74] differed from the others (such as SCOOP@F1 and SCOOP@F2 [75]) by its definition of use

cases in urban environment and its focus on V2V communications. The project developed new services such as the GLOSA use case.

5.2 Smartphone data

The CoopITS application records two types of logs in a server, the first type of logs concerns the actions that are made in the app, such as the display of events and information on the screen. In this type of logs we are mainly interested in the logs of action of the TLM which displays the current phase of nearby traffic lights, and also the time until the next phase and if the lights are red it displays the advice speed to respect in order to avoid a stop. The second type of logs concerns the C-ITS messages, such as the CAM, DENM, SPATEM, MAPEM, and IVIM, the messages are logged in their raw state meaning encoded with ASN.1 uper then decoded and inserted into a database. We are mainly interested in the CAM messages.

A CAM or Cooperative Awareness Message is a periodical message that is sent by a C-ITS station in broadcast at a frequency of either 1s and 5s in the CoopITS application. A station can be a vehicle, an RSU or a different C-ITS actor. a CAM contains information about the station's state at an instant t such as its StationID which is its identification number in the C-ITS environment, its type (car, motorcycle, RSU, etc) using the field StationType, its position by means of latitude, longitude and altitude, its heading which is in degrees and represents the angle of the vehicle from true north, its velocity in m/s, and more. Since the data is generated from a smartphone, the velocity is always positive or equal to zero even if the vehicle is moving backwards.

Each smartphone when using the CoopITS application is attributed a random StationID. The smartphone maintains its StationID for 10 minutes of operation time, then it is attributed another. And so far, during the time of the study, a total number of 1 336 573 CAM messages have been recorded in the city of Bordeaux and the region of "La Nouvelle Aquitaine" in France coming from 8667 distinct StationIDs.

We use the same trajectories that were extracted in the previous chapter, that of 71 distinct trajectories (71 distinct stationIDs) in the same direction in the city of Bordeaux, which represents a total set of 46570 CAM messages. The records belonged likely to the same driver since they all start from the same area and go likely to the same destination.

Fig. 5.1 shows the trajectory that was chosen. The trajectory starts south and goes to the north and was plotted on an OSM map in python using the folium library where each dot represents the position recorded in each CAM message from a single trip (a single StationID).



FIGURE 5.1: An example of the trajectory that was extracted from CAM messages, plotted on top of an OSM map

In the previous chapter, the trajectory was cut into road segments. A road segment is the portion of the road between two crossings. We used the same data and chose a segment where a traffic signal is installed, and where the GPS data are the most precise. The segment that was chosen is shown in Fig. 5.2. It has a length of 120 meters and the shortest recorded travel time of the segment is 6 seconds.

The TLM dataset contains all traffic light signals status notifications that were sent to the smartphones. The smartphones receive notifications about the traffic light status in nearby intersections, as well as the time to the next phase and the speed advice. The speed advice is only present in the case of red light, and it tells the vehicle the speed that it should respect to avoid a stop. Fig. 5.3 represents an example of some TLM records.

A matching between the CAMs and TLMs was then done using the stationID (the TLM user is identified by its StationID). For each CAM message, if a TLM record fits within its time frame with the next CAM message, it was attributed to it. We then observe the behavior of the vehicles along some indicators such as speed variation, redlight status and speed advice over time and distance, which we will present in the next section.

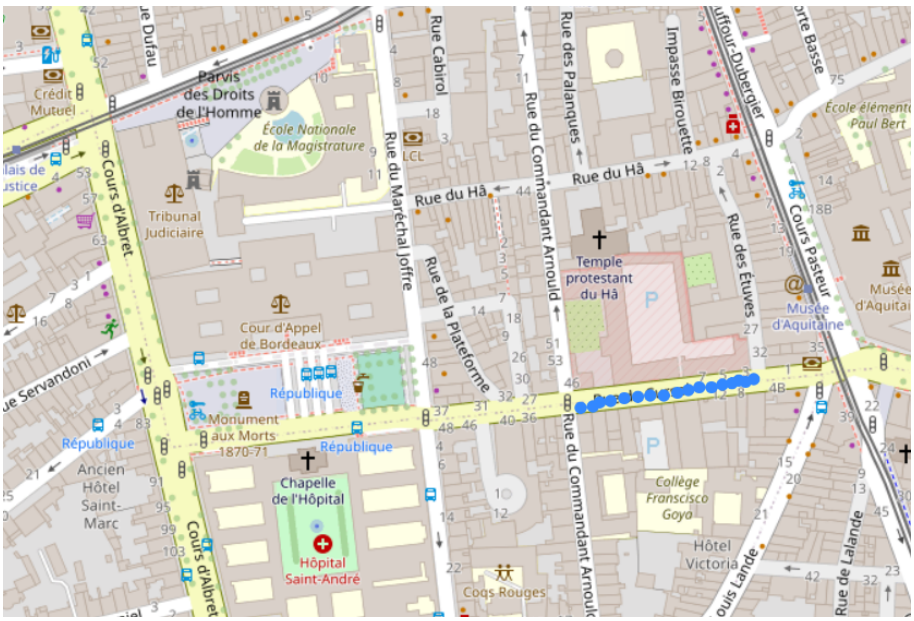

FIGURE 5.2: Trajectory data in the segment that was chosen, plotted on top of an OSM map

log_timestamp	stationID	eventid	currentphase	nextphase	advicetimetonextphase	speedadvice	intersectionid
1627538835910,00	3526965938	8	RED	GREEN	2192	22	219
1627538835912,00	3526965938	8	RED	GREEN	2192	22	219
1627538836909,00	3526965938	8	RED	GREEN	1192	27	219
1627538836910,00	3526965938	8	RED	GREEN	1192	27	219
1627538837909,00	3526965938	8	RED	GREEN	193	32	219
1627538837911,00	3526965938	8	RED	GREEN	193	32	219
1627538838910,00	3526965938	8	GREEN	YELLOW	67192	NULL	219
1627538838913,00	3526965938	8	GREEN	YELLOW	67192	NULL	219
1627538839910,00	3526965938	8	GREEN	YELLOW	66292	NULL	219
1627538839911,00	3526965938	8	GREEN	YELLOW	66292	NULL	219
1627538840910,00	3526965938	8	GREEN	YELLOW	65292	NULL	219
1627538840911,00	3526965938	8	GREEN	YELLOW	65292	NULL	219
1627538841912,00	3526965938	8	GREEN	YELLOW	64291	NULL	219

FIGURE 5.3: An example of some TLM records

5.3 Analysis

In this section we present our observations on the trajectory profiles in the chosen segment. We have observed that there exist 4 elemental patterns that are constantly repeating and are summarized as follows:

(a) A free flow driving with a green or yellow light.

(b) A stop in the segment with a green or yellow light.

(c) A free flow driving with a red light.

(d) A stop in the segment with a red light.

These four patterns are elements that constitute the vehicular behavior in the segment, a trajectory in the segment may constitute of one or more of these

elemental patterns. The first pattern, which is the free flow at a green light is due to the emptiness of the road before head of the vehicle and until the segment's end, so the vehicle travels without a stop. The second pattern which is the stop at a green light is due to the presence of a queue before the end of the segment so the vehicle has to stop. The third one which is the free flow at a red light is because the vehicle's velocity isn't high enough, it happens in the case where the driver respects the speed advice voluntarily or involuntarily. The fourth one which is the stop at a red light happens when the vehicle's velocity is higher than the speed advice, so it reaches the end of the segment or the beginning of the queue and stops.

Figures Fig. 5.4, Fig. 5.6, Fig. 5.8, and Fig. 5.10 represent the speed profiles of the vehicle in the same road segment matched with the traffic light states. Each dot symbolizes a matched CAM record represented by its velocity in function of time and distance from the start of the segment. Its color represents the state of traffic light it received from the TLM service, where it is either red, green, yellow or blue. Blue means that there is no TLM data for the CAM point. The black dots represent the speed advice the vehicle receives whenever the light is red.

Fig. 5.4 shows the pattern (a), the vehicle entered the segment with an initial velocity of 7.69 m/s which was reported by the smartphone's GPS. The velocity that was reported when leaving the segment was at 10.03 m/s. The vehicle sent all along the segment a total of 13 CAM messages, the measured average velocity from these CAMs was at 9.0 m/s and the time frame between the first and last CAM is 12 seconds. 120 meters which is the segment's length divided by 12 seconds gives 10 m/s which should be the average velocity. It is not the case due to precision issues in the smartphone's GPS. The vehicle received a green light status all along the segment, and there was no queue at the end of the segment so it traveled it without a stop.

Fig. 5.5 also shows the pattern (a), the initial velocity at the beginning of the segment is of 8.17 m/s, and at the segment's end was 9.02 m/s. The vehicle sent an overall of 13 CAM messages. The average velocity from the CAM messages is equal to 8.89 m/s and the time between the first and last CAM message in the segment is equal to 12 seconds. The average velocity should also be at 10 m/s but it is not due to GPS precision. The vehicle received a green light along all the segment and there wasn't a queue at stop before it so it didn't have to stop.

Pattern (b) can be observed in Fig. 5.6, the vehicle entered the segment with a velocity of 8.68 m/s, it reached its top velocity of 11.69 m/s at around second 4 and 58 meters from the start of the segment. It left the segment at a velocity of

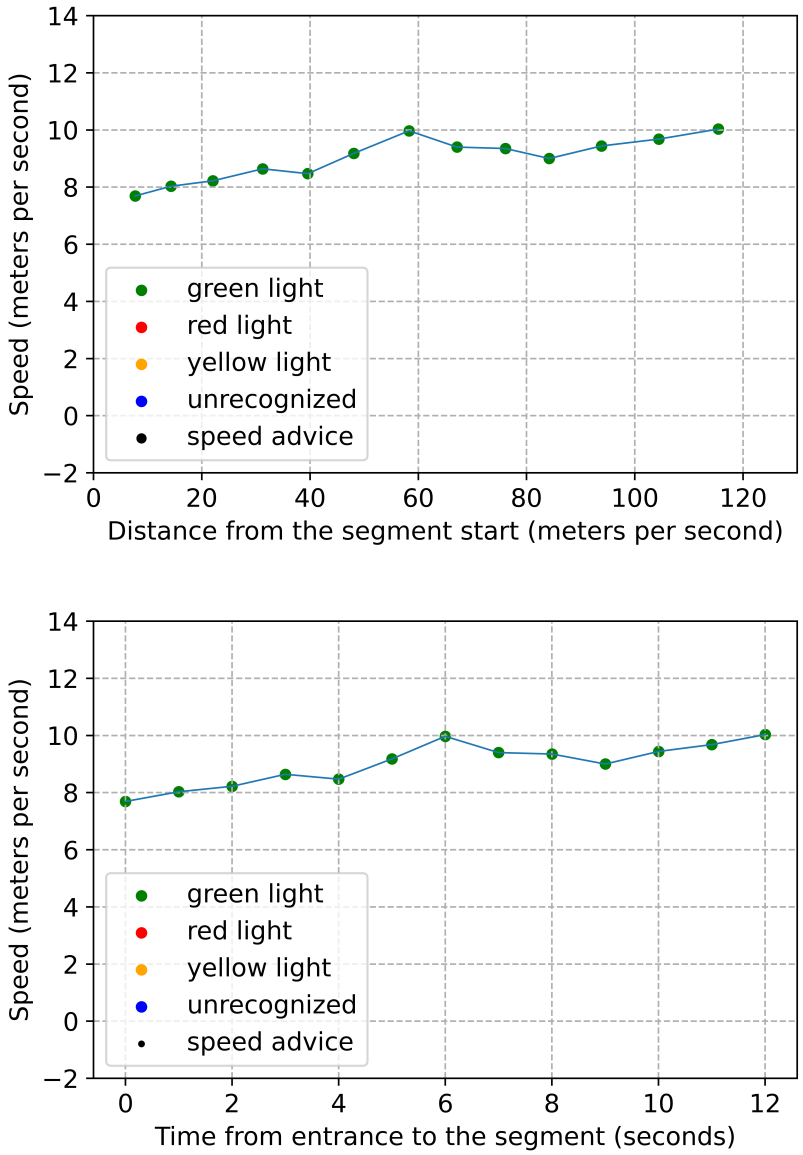

FIGURE 5.4: A trajectory with free flow at a green light (a) velocity in function of the distance (b) velocity in function of time

3.7 m/s. The vehicle arrived at a stop at 103 meters from the start of the segment and at 15 seconds. It spent 33 seconds to travel the segment and sent an overall of 32 CAM messages. The mean velocity from the CAMs is equal to 3.72 m/s which has an acceptable accuracy. The vehicle encountered a green light all along the segment, yet it stopped 17 meters from the end of the segment, this is mostly due to the presence of a queue at the end. The queue was probably generated by the previous red light phase.

Pattern (b) can also be observed in Fig.5.7 where the vehicle encountered a green light along all the segment but had to stop due to the presence of a queue.

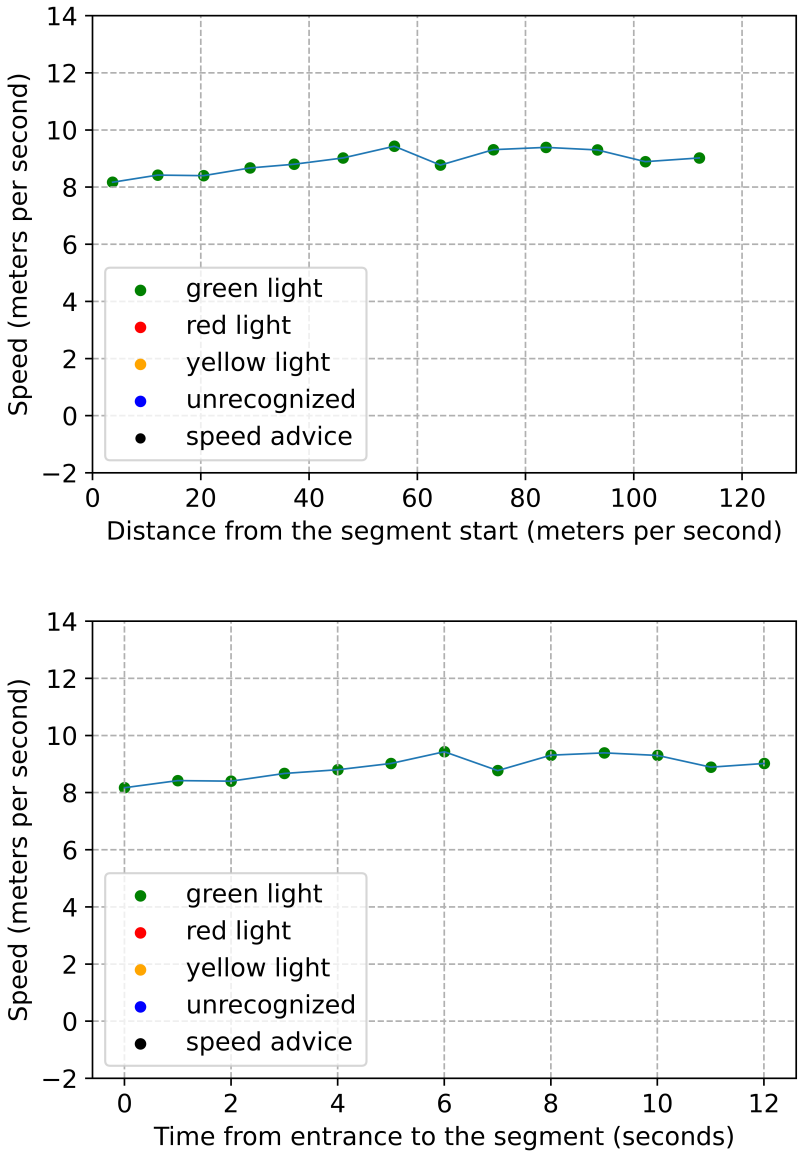

FIGURE 5.5: A trajectory with free flow at a green light (a) velocity in function of the distance (b) velocity in function of time

The vehicle sent an overall of 25 CAM messages in the segment, it entered it at a velocity of 6.11 m/s and left it at a velocity of 6.09 m/s. It spent a total of 24 seconds to traverse the segment. The average velocity from the CAMs is at 4.53 m/s whereas the calculated average velocity by dividing the segment length (120) on the time to traverse (24) is equal to 5 m/s. The vehicle stops at around 64 meters from the start of the segment, the issue of GPS precision can also be seen in the top figure where the vehicle seems to have returned backwards before it stops.

Pattern (c) can be observed in Fig. 5.8, the vehicle sent 26 CAM messages, it

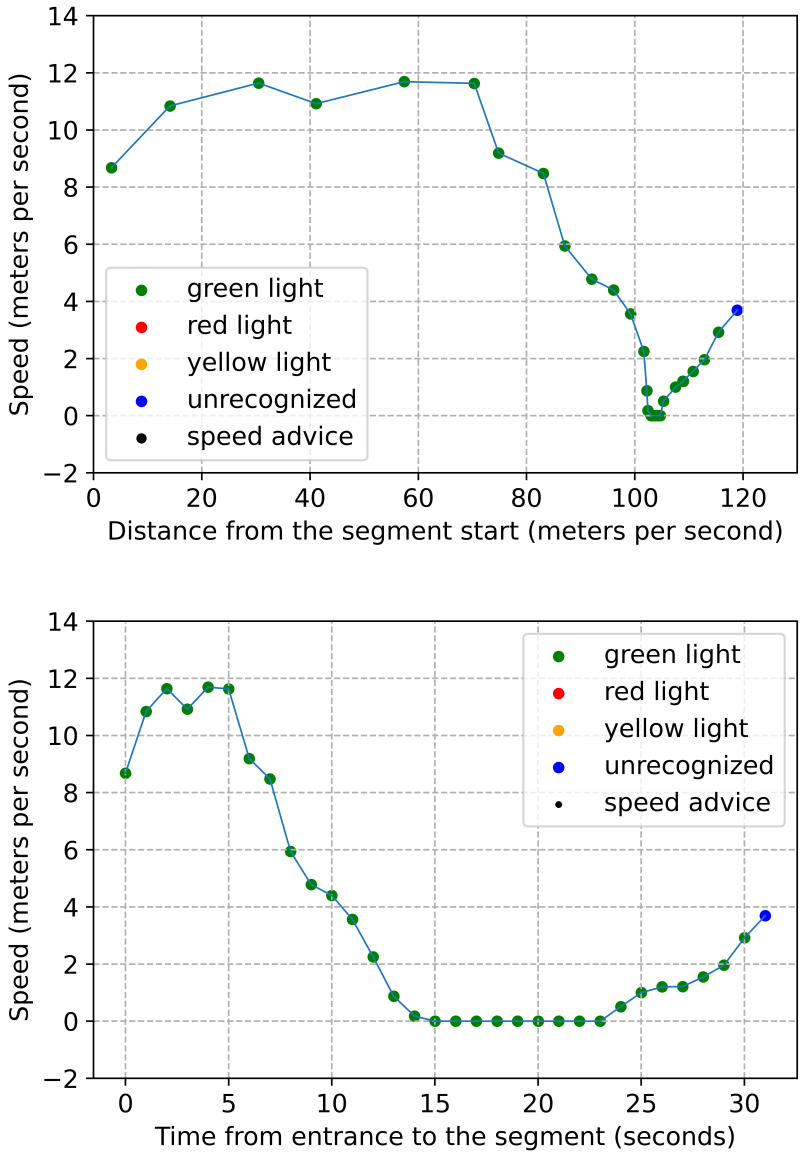

FIGURE 5.6: A trajectory with a stop at a green light (a) velocity in function of the distance (b) velocity in function of time

entered the segment with a velocity of 6.34 m/s, and left it with a velocity of 6.44 m/s. it lowered its velocity in the middle until it reached 2.22 m/s but doesn't stop. Its average velocity was at 4.57 m/s which has an acceptable accuracy since 120 divided by 25 gives 4.8. Since the light is red the vehicle received the speed advice, and overall, the vehicle's velocity was lower than the speed advice, therefore it didn't stop

Pattern (c) can be also be observed in Fig.5.9. The vehicle entered the segment at the velocity of 6.34 m/s and left it with a velocity of 6.88 m/s. It traversed the segment in 16 seconds and sent an overall of 17 CAM messages. It's average

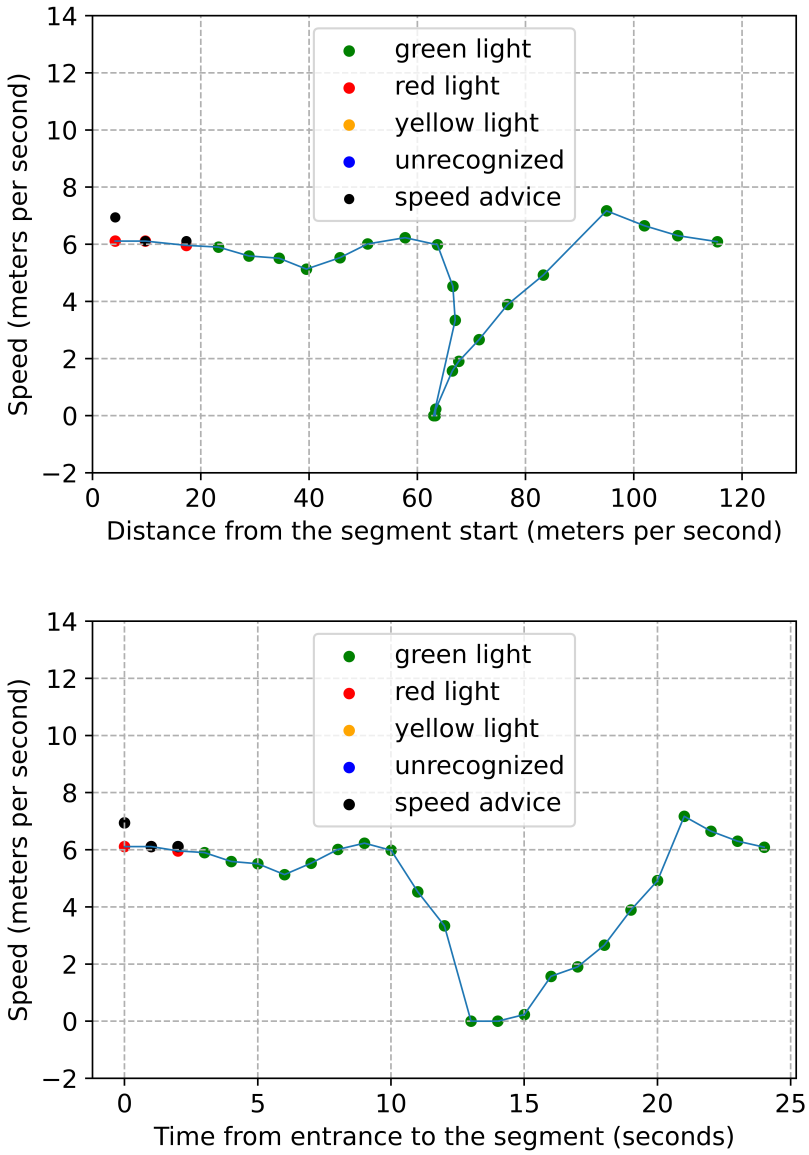

FIGURE 5.7: A trajectory with a stop at a green light (a) velocity in function of the distance (b) velocity in function of time

velocity is of 6.58 m/s whereas 120 divided by 16 gives 7.5 m/s, the difference is due to GPS precision again. At second 6 the vehicle's velocity was inferior to the given speed advice, therefore it avoided a stop.

In Fig. 5.10 we can observe the last pattern, which is when the vehicle stops at a red light. The vehicle entered the segment at a velocity of 5.36 m/s, and left it at the velocity of 5.26 m/s. It's average velocity was at 3 m/s and it took 34 seconds to traverse the segment which gives it a calculated average velocity of 3.52 m/s. As with the first case, this is due to the GPS precision. The issue of precision can also be seen at the stop in Fig. 5.10 (a), which is when the position

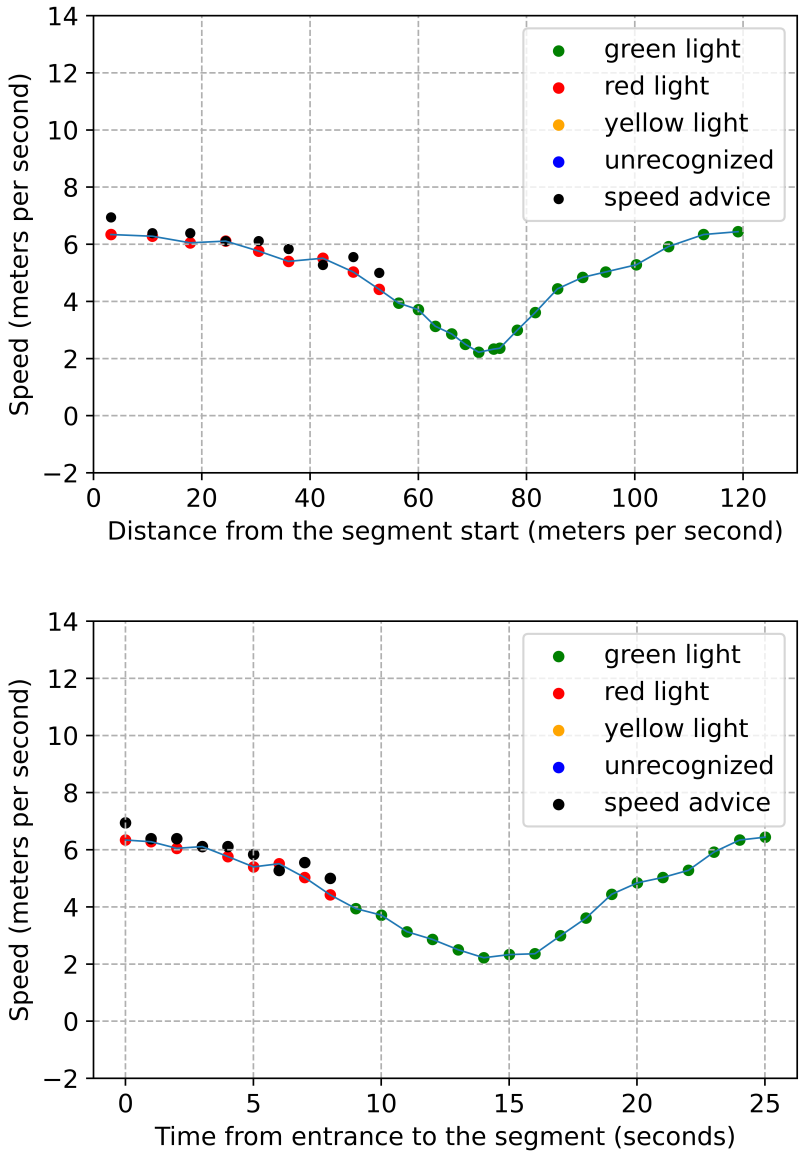

FIGURE 5.8: A trajectory with free flow at a red light (a) velocity in function of the distance (b) velocity in function of time

slightly returns backwards although the velocity was at zero.

Pattern (d) can aslo be observed in Fig.5.11. The vehicle entered the segment at a velocity of 4.48 m/s and left it with a velocity of 3.7 m/s. The vehicle sent 53 CAM messages and spent 52 seconds to traverse the segment. It's average velocity recorded by CAMs is of 2.14 m/s whereas the calculated average velocity by dividing the segment length (120) by the traversal time (52) is equal to 2.30 which we consider acceptable. The vehicle started receiving the speed advice at the encounter of a red light is around second 20, but it didn't respect it so it had to stop at around second 28 at around 18 meters from the end of the segment.

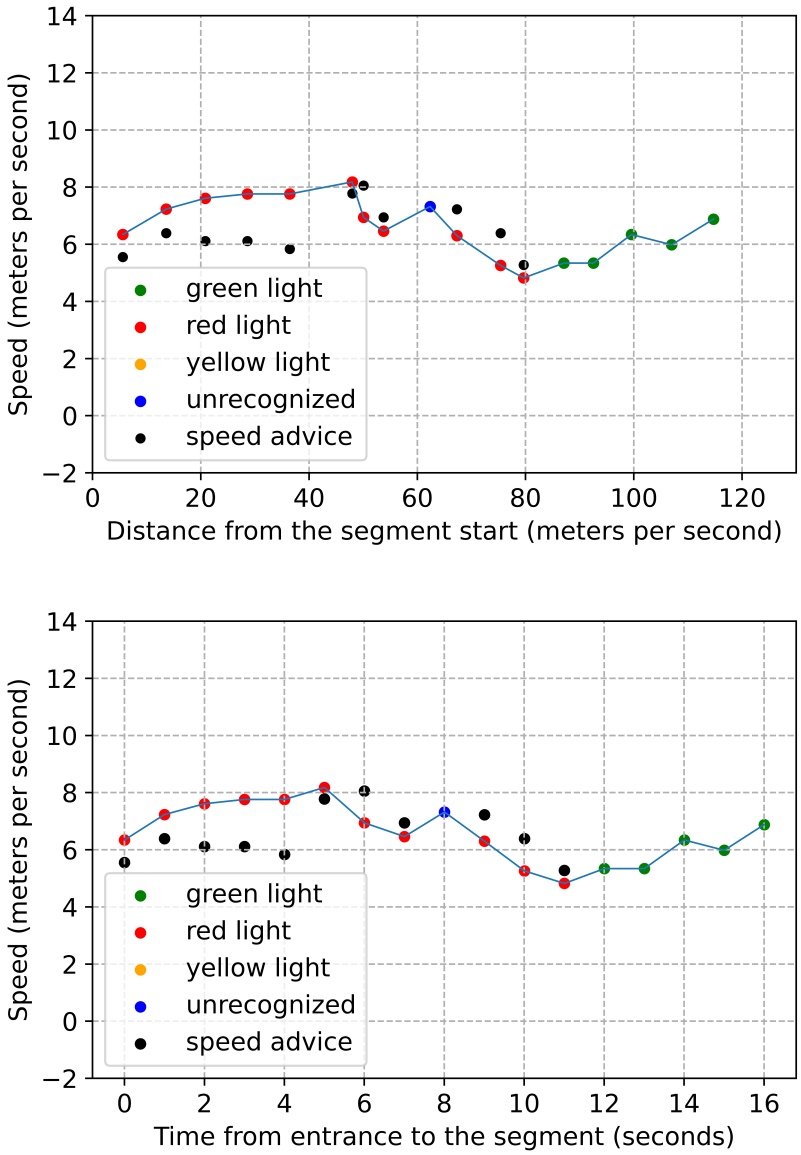

FIGURE 5.9: A trajectory with free flow at a red light (a) velocity in function of the distance (b) velocity in function of time

In Fig. 5.12 where the vehicle sent 84 CAM messages, we see a combination of two patterns (d) followed by (b), the vehicle entered the segment at a velocity of 7.84 m/s and stopped at around 46 meters from the start of the segment at red light mainly due to the disrespect of the speed advice. After the light turned green it remained still for 20 seconds then started moving only to stop again at a queue. The queue is probably due to an operation happening at the road crossing. It left the segment at a velocity of 4.51 m/s. The same remark can be done, the vehicle changing its position while at stop and with a null velocity is mainly due to GPS precision.

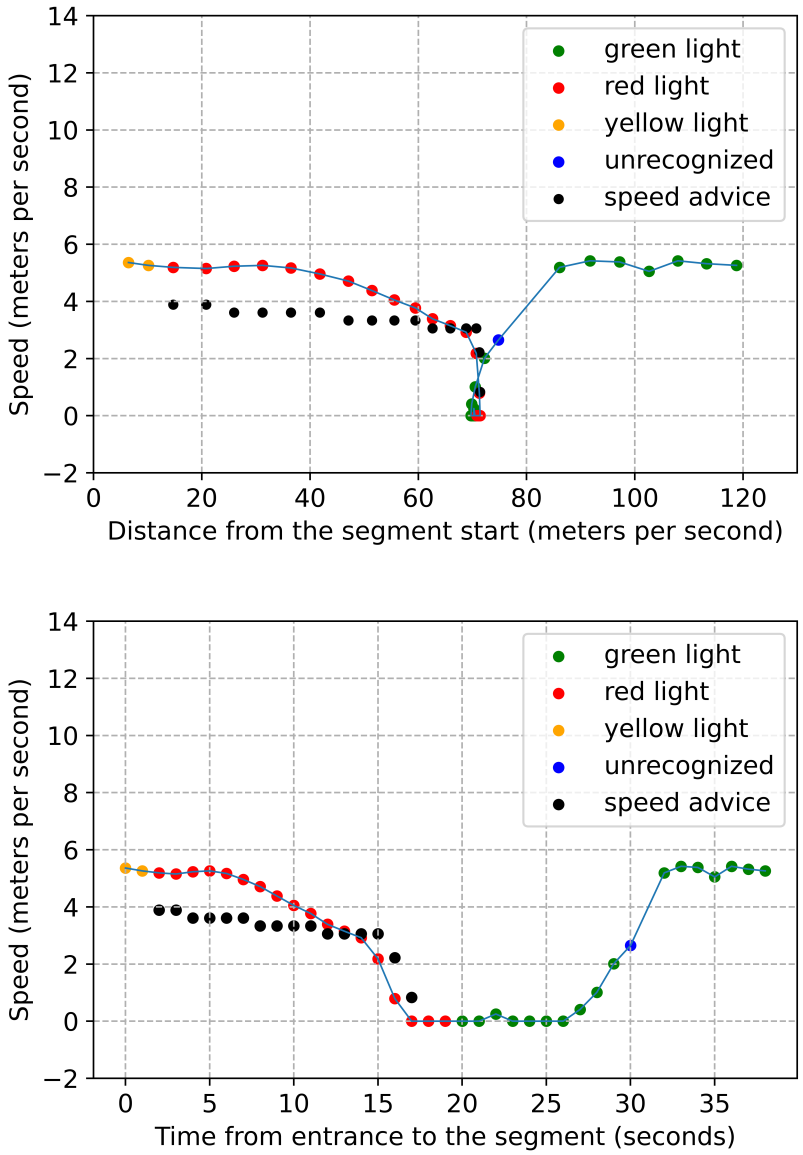

FIGURE 5.10: A trajectory with a stop at a red light (a) velocity in function of the distance (b) velocity in function of time

5.4 Traffic jam prediction using a single vehicle in a stop

We previously have seen four different patterns, the stops are hypothesized to be due to a queue. In patterns a and c we can deduce that no significant queues were present, otherwise the vehicle would have stopped. But in patterns b and d, the vehicle stops. For pattern b, there is a green light and it is hypothesized that the stop was due to a queue. If the segment was empty in pattern d, the vehicle would stop at the end of the segment, and if it does stop in the middle, it is also

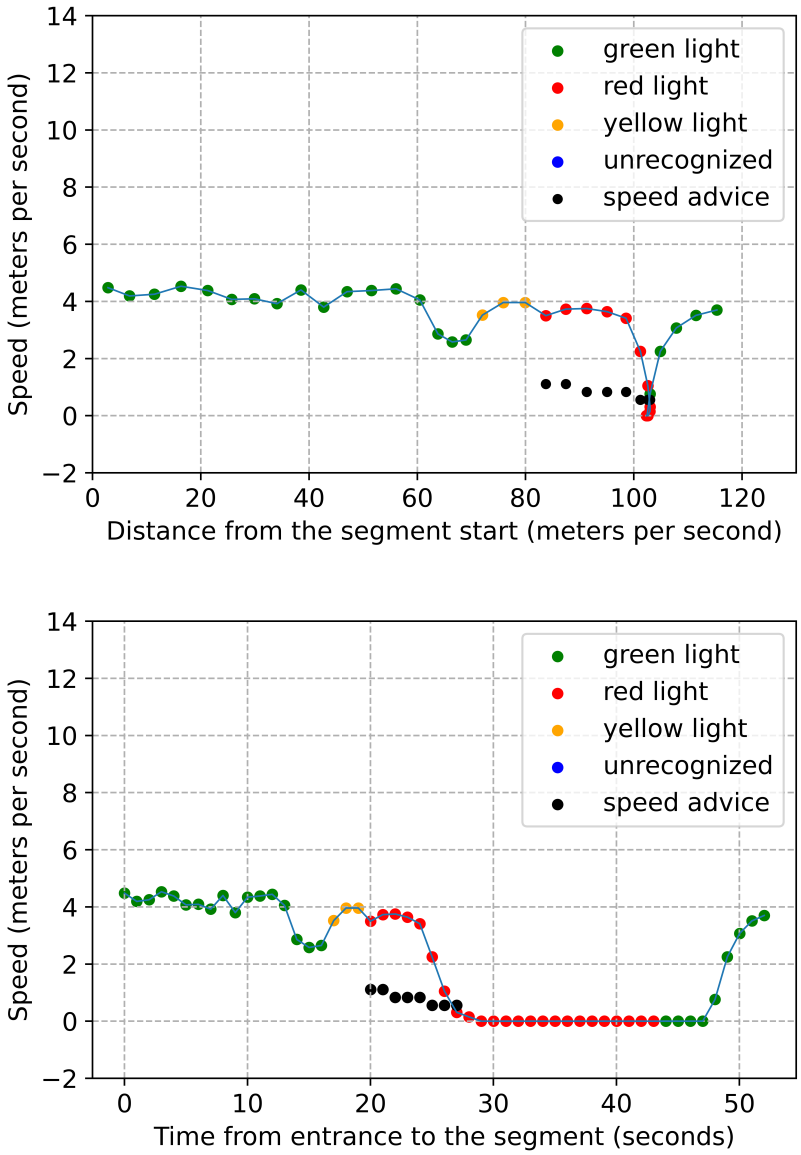

FIGURE 5.11: A trajectory with a stop at a red light (a) velocity in function of the distance (b) velocity in function of time

hypothesized to be due to queue.

Therefore, we consider the patterns b and d for the detection of a potential queue on the road at stops. Assuming all vehicles have the same length l and are all at stop, the number of vehicles in the queue in relation to the controlled vehicle would be equal to :

$$N = \frac{L - x_i}{l + d_0} \tag{5.1}$$

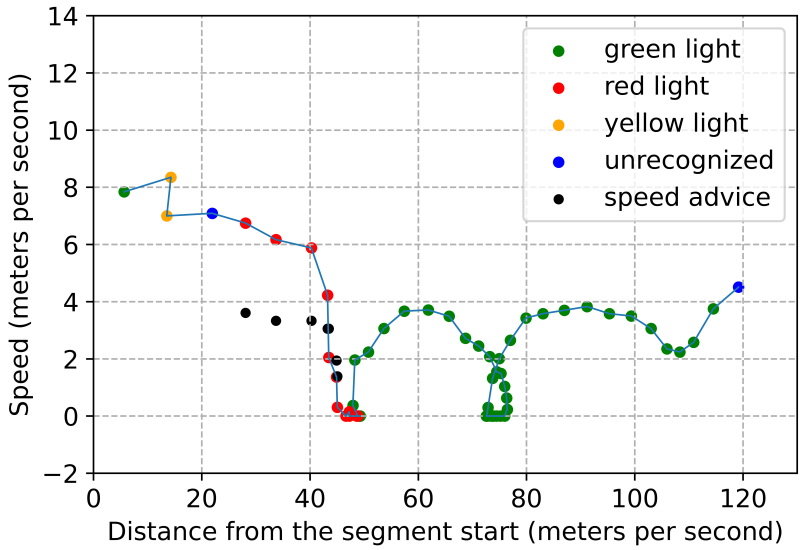

FIGURE 5.12: A trajectory with combined pattern, velocity in function of the distance

Where L is the length of the segment, x_i is the relative position of the vehicle in the segment and d_0 the standstill distance between the vehicles.

Assuming different lengths of vehicles, where l_{min} is the minimum size of vehicles and l_{max} is the maximum size that a vehicle can have (a semi-trailer for example). the number of vehicles in the queue would be :

$$\frac{L - x_i}{l_{max} + d_0} < N < \frac{L - x_i}{l_{min} + d_0}$$

Therefore, if given $l_{min} = 3m$, $l_{max} = 16m$ and $d_0 = 1m$ the number of vehicles in the queue at the moment of the stop of the vehicle in Fig5.10 would be between 2.94 and 12.5 vehicles. And the number of vehicles in Fig5.6 between 1 and 4.25 vehicles.

Assuming an $l = 4m$ and $d_0 = 1m$, and given that the length of the segment $L = 120m$ the predicted queues according to equation 5.1 are given in Table.5.1.

5.4.1 Traffic jam prediction using a moving vehicle

In car-following models it is assumed that the velocity of a vehicle is relative to the velocity of its predecessor. The goal of a vehicle is to follow its predecessor and keep a certain distance from it called the headway. one distancing strategy called the Constant Time Headway [Constant Spacing Strategies For Platooning

TABLE 5.1: Recorded communications with the RSUs

StationID	stop relative position	queue length
140859343	107 m	2,6
1097025897	100 m	4
1125743700	97 m	4,6
1228133956	91 m	5,8
1520594289	89 m	6,2
1693752982	100 m	4
1966236983	108 m	2,4
2053268129	110 m	2
2582698734	103 m	3,4
2589910337	48 m	14,4
2725142206	65 m	11
2976041898	100 m	4
3171950913	110 m	2
3311968502	107 m	2,6
3405657847	106 m	2,8
3432719340	93 m	5,4
3544466934	86 m	6,8
3688497722	103 m	3,4
4114888848	104 m	3,2

In Automated Highway Systems] denotes that each of the vehicles keeps a constant time headway λ, the distance (which is relative to the velocity) is then given by :

$$s_i^* = x_{i-1} - x_i = v_i\lambda + d_0 \tag{5.2}$$

Where x_i is the position of the ego vehicle and x_{i-1} the position of its predecessor, v_i is the velocity of the ego vehicle, λ is the time headway and d_0 is the standstill distance.

Deriving equation 5.2 gives the following, which the acceleration that the vehicle should follow to maintain the distance s_i^* :

$$a_i = \frac{v_{i-1} - v_i}{\lambda}$$

Where v_{i-1} is the velocity of the predecessor.

Substituting d_0 with 5.2 in equation 5.1 and assuming all the vehicles have the same velocity meaning a state of equilibrium, gives the following :

$$N = \frac{L - x_i}{l + v_i\lambda + d_0}$$

Thus, for example in Fig 5.4 and at the distance of 40m from the start of the segment, where it had a velocity of 8.4 m/s. And assuming all vehicles have the same velocity at that instant, and a length of 4m, that the standstill distance is equal to 1m and that the time headway λ is equal to 1.5 seconds, then the queue inside the segment would have 3.4 vehicles

5.5 Conclusion

In this chapter we used data from the CoopITS application. CoopITS is a C-ITS simulation application deployed in France, where a smartphone plays the role of a C-ITS station. Each action is logged by the application including sending and receiving C-ITS messages. We were concerned with two specific recorded actions by the application, the sending of a CAM message, and the TLM actions. A matching between the CAMs and TLM records has been done and the results were used to investigate the behavior of vehicles in signalized intersections. We chose a segment where a traffic light intersection lies at its end, and we have analyzed the speed profiles of vehicles where we observed and classified four patterns : a free flow with a green light, a stop with a green light due to a queue, a freeflow with a red light due to the respect of the speed advice given by the TLM and last a stop with a red light due to the disrespect of the speed advice.

The rest of this chapter is covered by the following article:

Benzagouta, M. L., Aniss, H., Fouchal, H., El-Faouzi, N. E. (2023, September). Impact of Speed Limitation on Urban Road Traffic. In International Conference on Innovations for Community Services (pp. 89-102). Cham: Springer Nature Switzerland.

Chapter 6

Conclusion

In this thesis, we delve in the state of the art into the varied domain of C-ITS, exploring the interplay between communication standards, anomaly detection, clustering algorithms, driver profiles, trajectory clustering, and traffic prediction methodologies. In Chapter 1, a comprehensive introduction to vehicular applications sets the foundation for our research. Two communication standards for V2X are introduced: C-V2X, exploiting cellular networks, and ITS-G5, utilizing WiFi adhoc connectivity. The following chapters provide a coherent investigation that integrates theoretical frameworks with empirical findings.

Chapter 2 delves into an extensive state-of-the-art review of C-ITS, offering an understanding of existing works. This chapter also explores anomaly detection methodologies, laying the groundwork for our focused investigation in Chapter 3. Our research methodology involves intricate data analysis using two distinct datasets: PCAP files from a test vehicle's travels in France, Germany, and Italy, and data from the CoopITS phone application simulating a C-ITS-equipped vehicle in the "Nouvelle Aquitaine" region.

Chapter 3 concerns our work on anomaly detection for Roadside Units (RSUs), where end-user vehicles play a pivotal role. Two RSU profiles are defined: a functioning RSU and one in failure. Criteria for the functioning RSU include an ID larger than the OD and a Pearson's coefficient of correlation between range and signal power less than -0.4. A failing RSU fails to meet one or both conditions, providing a robust framework for end-user vehicles to discern anomalies themselves in the C-ITS environment.

Building upon an extensive review presented in Chapter 2, which covered clustering algorithms and driver profiles, Chapter 3 focuses on trajectory clustering as a means to detect anomalies and predict traffic patterns.

The trajectory clustering process started by selecting a set of trajectories along

the same path traversed by multiple or a single driver. These trajectories are then segmented into 10 distinct stretches, where each stretch or segment represents the road between two consecutive crossings. A total of 609 trajectories, are subjected to clustering using four algorithms: K-means, Agglomerative clustering, BIRCH, and DBSCAN. The results of each algorithm are meticulously compared to ascertain their effectiveness in identifying anomalies within the trajectory data.

Chapter 2 also laid the foundation for an overview on traffic prediction methodologies concerning road intersections. In Chapter 3, utilizing the same segmented data, a specific segment with a traffic light is chosen. All trajectories within this segment share a uniform direction. Leveraging Traffic Light Management (TLM) data and Cooperative Awareness Message (CAM) records, a robust association between speed advice and trajectories is established. This innovative approach reveals four distinct driving patterns in intersections: 1) Free Flow with Green or Yellow Light: Vehicles move seamlessly through the intersection with a favorable signal. 2) Stop at a Green Light Due to Queue: Vehicles come to a halt at a green light due to traffic congestion. 3) Free Flow with a Red Light Due to GLOSA: Vehicles continue at a free flow despite a red light, adhering to the speed advice provided by the Green Light Optimal Speed Advisory (GLOSA) system. and 4) Stop at a Red Light Due to Non-Compliance: Vehicles come to a stop at a red light due to non-compliance with the recommended speed advice.

Bibliography

[1] Etsi ts 103 301; intelligent transport systems (its); vehicular communications; basic set of applications; facilities layer protocols and communication requirements for infrastructure services. 02 2020.

[2] Kevin Ashton. That 'internet of things' thing. 1999. URL https://api.semanticscholar.org/CorpusID:109032799.

[3] Hacène Fouchal, Geoffrey Wilhelm, Emilien Bourdy, Geoffrey Wilhelm, and Marwane Ayaida. A testing framework for intelligent transport systems. In *2016 IEEE Symposium on Computers and Communication (ISCC)*, pages 180–184, 2016. doi: 10.1109/ISCC.2016.7543736.

[4] Etsi en 302 637-2; intelligent transport systems (its); vehicular communications; basic set of applications; part 2: Specification of cooperative awareness basic service.

[5] Etsi en 302 637-3; intelligent transport systems (its); vehicular communications; basic set of application; part 3: Specifications of decentralized environmental notification basic service.

[6] Etsi ts 136 101; lte; evolved universal terrestrial radio access (e-utra); user equipment (ue) radio transmission and reception (3gpp ts 36.101 version 16.7.0 release 16). 02 2020.

[7] Etsi ts 302 663; intelligent transport systems (its); its-g5 access layer specification for intelligent transport systems operating in the 5 ghz frequency band. 02 2020.

[8] Etsi en 302 636-4-1; intelligent transport systems (its); vehicular communications; geonetworking; part 4: Geographical addressing and forwarding for point-to-point and point-to-multipoint communications; sub-part 1: Media-independent functionality.

[9] Robayet Nasim and Andreas Kassler. Distributed architectures for intelligent transport systems: A survey. In *2012 Second Symposium on Network Cloud Computing and Applications*, pages 130–136, 2012. doi: 10.1109/NCCA.2012.15.

[10] Juan Antonio Guerrero-ibanez, Sherali Zeadally, and Juan Contreras-Castillo. Integration challenges of intelligent transportation systems with connected vehicle, cloud computing, and internet of things technologies. *IEEE Wireless Communications*, 22(6):122–128, 2015. doi: 10.1109/MWC.2015.7368833.

[11] Wenjie Chen, Lifeng Chen, Zhanglong Chen, and Shiliang Tu. Wits: A wireless sensor network for intelligent transportation system. In *First International Multi-Symposiums on Computer and Computational Sciences (IMSCCS'06)*, volume 2, pages 635–641, 2006. doi: 10.1109/IMSCCS.2006.286.

[12] Bret Hull, Vladimir Bychkovsky, Yang Zhang, Kevin Chen, Michel Goraczko, Allen Miu, Eugene Shih, Hari Balakrishnan, and Samuel Madden. Cartel: A distributed mobile sensor computing system. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, SenSys '06, page 125–138, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933433. doi: 10.1145/1182807.1182821. URL https://doi-org.univ-eiffel.idm.oclc.org/10.1145/1182807.1182821.

[13] Zhou changyu, Duan jianyong, and Xin zheng. Beijing traffic data center based on soa technology. In *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*, volume 15, pages V15–232–V15–235, 2010. doi: 10.1109/ICCASM.2010.5622598.

[14] Ali Gohar and Gianfranco Nencioni. The role of 5g technologies in a smart city: The case for intelligent transportation system. *Sustainability*, 13:5188, 2021. URL https://api.semanticscholar.org/CorpusID:235557032.

[15] Fernando Camacho Ochoa, Cesar Cardenas, and David Muñoz. Emerging technologies and research challenges for intelligent transportation systems: 5g, hetnets, and sdn. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 12:1–9, 02 2018. doi: 10.1007/s12008-017-0391-2.

[16] Soufiene Djahel, Nafaâ Jabeur, Robert Barrett, and John Murphy. Toward v2i communication technology-based solution for reducing road traffic congestion in smart cities. 05 2015. doi: 10.1109/ISNCC.2015.7238584.

[17] Maanak Gupta, James Benson, Farhan Patwa, and Ravi Sandhu. Secure v2v and v2i communication in intelligent transportation using cloudlets. *IEEE Transactions on Services Computing*, 15(4):1912–1925, 2022. doi: 10.1109/TSC.2020.3025993.

[18] András Wippelhauser, Tamás Attila Tomaschek, Máté Verdes, and László Bokor. Real-life traffic data based its-g5 channel load simulations of a major hungarian c-its deployment site. *Applied Sciences*, 13(14), 2023. ISSN 2076-3417. doi: 10.3390/app13148419. URL https://www.mdpi.com/2076-3417/13/14/8419.

[19] Marco Guerrieri, Raffaele Mauro, Andrea Pompigna, and Natalia Isaenko. Road design criteria and capacity estimation based on autonomous vehicles performances. first results from the european c-roads platform and a22 motorway. *Transport and Telecommunication Journal*, 22:230–243, 04 2021. doi: 10.2478/ttj-2021-0018.

[20] Maximilian Bauder, Andreas Festag, Tibor Kubjatko, and Hans-Georg Schweiger. Data accuracy in vehicle-to-x cooperative awareness messages: An experimental study for the first commercial deployment of c-its in europe, 01 2023.

[21] Boubakeur Moussaoui, Noureddine Chikouche, and Hacène Fouchal. An efficient privacy scheme for c-its stations. *Computers and Electrical Engineering*, 107:108613, 2023. ISSN 0045-7906. doi: https://doi.org/10.1016/j.compeleceng.2023.108613. URL https://www.sciencedirect.com/science/article/pii/S0045790623000381.

[22] Thomas Rosenstatter and Kateryna Melnyk. Towards synthetic data generation of vanet attacks for efficient testing. In *2023 IEEE Intelligent Vehicles Symposium (IV)*, pages 1–7, 2023. doi: 10.1109/IV55152.2023.10186685.

[23] Klaus Weinbauer, Tobias Renzler, Michael Stolz, and Daniel Watzenig. Taking standardized c-its communication to the next level - lightweight and extensible vehicle-to-everything library. In *2023 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6, 2023. doi: 10.1109/WCNC55385.2023.10118982.

[24] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Birch: An efficient data clustering method for very large databases. *SIGMOD Rec.*, 25(2):103–114,

jun 1996. ISSN 0163-5808. doi: 10.1145/235968.233324. URL https://doi.org/10.1145/235968.233324.

[25] R. Sibson. Slink: An optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, 16(1):30–34, 1973. doi: 10.1093/comjnl/16.1.30. URL http://dx.doi.org/10.1093/comjnl/16.1.30.

[26] J. MacQueen. Some methods for classification and analysis of multivariate observations. 1967. URL https://api.semanticscholar.org/CorpusID:6278891.

[27] L Kaufman and Peter Rousseeuw. Clustering by means of medoids. In *Statistical Data Analysis Based on the L1 Norm and Related Methods*, pages 405–416. North-Holland; Amsterdam, 1987. ISBN 0444702733.

[28] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[29] Alexander Hinneburg and Daniel A. Keim. An efficient approach to clustering in large multimedia databases with noise. KDD'98, page 58–65. AAAI Press, 1998.

[30] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: Ordering points to identify the clustering structure. In *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, SIGMOD '99, page 49–60, New York, NY, USA, 1999. Association for Computing Machinery. ISBN 1581130848. doi: 10.1145/304182.304187. URL https://doi.org/10.1145/304182.304187.

[31] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3), jul 2009. ISSN 0360-0300. doi: 10.1145/1541880.1541882. URL https://doi.org/10.1145/1541880.1541882.

[32] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422, 2008. doi: 10.1109/ICDM.2008.17.

[33] Stewart Worrall, Gabriel Agamennoni, James Ward, and Eduardo Nebot. Fault detection for vehicular ad-hoc wireless networks. In *2013 IEEE Intelligent Vehicles Symposium (IV)*, pages 298–303, 2013. doi: 10.1109/IVS.2013.6629485.

[34] Jiaxi Liu, Fei Ding, and Dengyin Zhang. A hierarchical failure detector based on architecture in vanets. *IEEE Access*, 7:152813–152820, 2019. doi: 10.1109/ACCESS.2019.2948599.

[35] Mikhail Lytaev, Eugene Borisov, and Andrei Vladyko. V2i propagation loss predictions in simplified urban environment: A two-way parabolic equation approach. *Electronics*, 9(12), 2020. ISSN 2079-9292. URL https://www.mdpi.com/2079-9292/9/12/2011.

[36] Sébastien Demmel, Alain Lambert, Dominique Gruyer, Gregoire Larue, and A. Rakotonirainy. Ieee 802.11p empirical performance model from evaluations on test tracks. *Journal of Networks*, 9, 06 2014. doi: 10.4304/jnw.9.6.1485-1495.

[37] Rreze Halili, Noori BniLam, Marwan Yusuf, Emmeric Tanghe, Wout Joseph, Maarten Weyn, and Raf Berkvens. Vehicle localization using doppler shift and time of arrival measurements in a tunnel environment. *Sensors*, 22:847, 01 2022. doi: 10.3390/s22030847.

[38] S. Ammoun and Fawzi Nashashibi. Design and efficiency measurement of cooperative driver assistance system based on wireless communication devices. *Transportation Research Part C: Emerging Technologies*, 18:408–428, 06 2010. doi: 10.1016/j.trc.2010.02.004.

[39] Tejasvi Alladi, Bhavya Gera, Ayush Agrawal, Vinay Chamola, and Fei Yu. Deepadv: A deep neural network framework for anomaly detection in vanets. *IEEE Transactions on Vehicular Technology*, PP, 09 2021. doi: 10.1109/TVT.2021.3113807.

[40] Julia Silva Weber, Tiago Ferreto, and Nur Zincir-Heywood. Exploring anomaly detection techniques for enhancing vanet availability. In *2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring)*, pages 1–7, 2023. doi: 10.1109/VTC2023-Spring57618.2023.10200360.

[41] Fei Guo, Zichang Wang, Suguo Du, Huaxin Li, Haojin Zhu, Qingqi Pei, Zhenfu Cao, and Jianhong Zhao. Detecting vehicle anomaly in the edge via sensor consistency and frequency characteristic. *IEEE Transactions on Vehicular Technology*, 68(6):5618–5628, 2019. doi: 10.1109/TVT.2019.2907692.

[42] Laisen Nie, Yongkang Li, and Xiangjie Kong. Spatio-temporal network traffic estimation and anomaly detection based on convolutional neural network in

vehicular ad-hoc networks. *IEEE Access*, 6:40168–40176, 2018. doi: 10.1109/ACCESS.2018.2854842.

[43] Laisen Nie, Huizhi Wang, Shimin Gong, Zhaolong Ning, Mohammad S. Obaidat, and Kuei-Fang Hsiao. Anomaly detection based on spatio-temporal and sparse features of network traffic in vanets. In *2019 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, 2019. doi: 10.1109/GLOBECOM38437.2019.9013915.

[44] Yixuan Wu, Wang Huizhi, and Li Yongkang. Anomaly detection based on spatio-temporal and sparse features of network traffic in vanets. *IEEE Access*, 7:177954–177964, 01 2019. doi: 10.1109/ACCESS.2019.2958068.

[45] Edivaldo Pastori Valentini, Geraldo Pereira Rocha Filho, Robson Eduardo De Grande, Caetano Mazzoni Ranieri, Lourenço Alves Pereira Júnior, and Rodolfo Ipolito Meneguette. A novel mechanism for misbehavior detection in vehicular networks. *IEEE Access*, 11:68113–68126, 2023. doi: 10.1109/ACCESS.2023.3292055.

[46] Haowen Tan, Ziyuan Gui, and Ilyong Chung. A secure and efficient certificateless authentication scheme with unsupervised anomaly detection in vanets. *IEEE Access*, 6:74260–74276, 2018. doi: 10.1109/ACCESS.2018.2883426.

[47] Emilien Bourdy, Kandaraj Piamrat, Michel Herbin, and Hacène Fouchal. New method for selecting exemplars application to roadway experimentation. In *International Conference on Innovations for Community Services*, pages 75–84. Springer, 2018.

[48] Juliet Chebet Moso, Stéphane Cormier, Cyril de Runz, Hacène Fouchal, and John Mwangi Wandeto. Anomaly detection on data streams for smart agriculture. *Agriculture*, 11(11), 2021. ISSN 2077-0472. doi: 10.3390/agriculture11111083. URL https://www.mdpi.com/2077-0472/11/11/1083.

[49] Evelyne Akopyan, Angelo Furno, Nour-Eddin El Faouzi, and Eric Gaume. Unsupervised real-time anomaly detection for multivariate mobile phone traffic series. pages 469–474, 01 2021. doi: 10.14428/esann/2021.ES2021-113.

[50] Jianfeng Zheng and Henry X. Liu. Estimating traffic volumes for signalized intersections using connected vehicle data. *Transportation Research Part C: Emerging Technologies*, 79:347–362, 2017. ISSN 0968-090X. doi: https://

doi.org/10.1016/j.trc.2017.03.007. URL https://www.sciencedirect.com/science/article/pii/S0968090X17300827.

[51] John Krumm and Eric Horvitz. Predestination: Inferring destinations from partial trajectories. In Paul Dourish and Adrian Friday, editors, *UbiComp 2006: Ubiquitous Computing*, pages 243–260, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-39635-2.

[52] Walid Fourati, Anysia Mayerhofer, and Bernhard Friedrich. Continuous compilation of directional counts at intersections fusing floating car data and section counts. *Transportation Research Procedia*, 37:235–241, 2019. ISSN 2352-1465. doi: https://doi.org/10.1016/j.trpro.2018.12.188. URL https://www.sciencedirect.com/science/article/pii/S2352146518306033. 21st EURO Working Group on Transportation Meeting, EWGT 2018, 17th – 19th September 2018, Braunschweig, Germany.

[53] Gurcan Comert and Mecit Cetin. Queue length estimation from probe vehicle location and the impacts of sample size. *European Journal of Operational Research*, 197(1):196–202, 2009. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2008.06.024. URL https://www.sciencedirect.com/science/article/pii/S0377221708004918.

[54] Peng Hao, Zhanbo Sun, Xuegang (Jeff) Ban, Dong Guo, and Qiang Ji. Vehicle index estimation for signalized intersections using sample travel times. *Transportation Research Part C: Emerging Technologies*, 36:513–529, 2013. ISSN 0968-090X. doi: https://doi.org/10.1016/j.trc.2013.06.018. URL https://www.sciencedirect.com/science/article/pii/S0968090X13001411.

[55] John Krumm. Where will they turn: Predicting turn proportions at intersections. *Personal and Ubiquitous Computing*, 14:591–599, 10 2010. doi: 10.1007/s00779-009-0248-1.

[56] John Krumm. A markov model for driver turn prediction. 04 2008. doi: 10.4271/2008-01-0195.

[57] Axel Wolfermann, Babak Mehran, and Masao Kuwahara. Data fusion for traffic flow estimation at intersections. 09 2011.

[58] Yi Lu Murphey, Robert Milton, and Leonidas Kiliaris. Driver's style classification using jerk analysis. In *2009 IEEE Workshop on Computational*

Intelligence in Vehicles and Vehicular Systems, pages 23–28, 2009. doi: 10.1109/CIVVS.2009.4938719.

[59] Adrian B. Ellison, Stephen P. Greaves, and Rhonda Daniels. Profiling drivers' risky behaviour towards all road users. 2012. URL https://api.semanticscholar.org/CorpusID:127502314.

[60] Rizwana Kalsoom and Zahid Halim. Clustering the driving features based on data streams. In *INMIC*, pages 89–94, 2013. doi: 10.1109/INMIC.2013.6731330.

[61] Tomislav Erdelić, Silvija Vrbančić, and Lovro Rošić. A model of speed profiles for urban road networks using g-means clustering. In *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1081–1086, 2015. doi: 10.1109/MIPRO.2015.7160436.

[62] Arya R Anil and Anudev J. Driver behavior analysis using k-means algorithm. In *2022 Third International Conference on Intelligent Computing Instrumentation and Control Technologies (ICICICT)*, pages 1555–1559, 2022. doi: 10.1109/ICICICT54557.2022.9917899.

[63] Chalermpol Saiprasert, Suttipong Thajchayapong, Thunyasit Pholprasit, and Chularat Tanprasert. Driver behaviour profiling using smartphone sensory data in a v2i environment. In *2014 International Conference on Connected Vehicles and Expo (ICCVE)*, pages 552–557, 2014. doi: 10.1109/ICCVE.2014.7297609.

[64] Minglin Wu, Sheng Zhang, and Yuhan Dong. A novel model-based driving behavior recognition system using motion sensors. *Sensors*, 16:1746, 10 2016. doi: 10.3390/s16101746.

[65] Jair Ferreira, Eduardo Carvalho, Bruno Ferreira, Cleidson Souza, Yoshi Suhara, Alex Pentland, and Gustavo Pessin. Driver behavior profiling: An investigation with different smartphone sensors and machine learning. *PLOS ONE*, 12:1–16, 04 2017. doi: 10.1371/journal.pone.0174959.

[66] Brice Leblanc, Hacene Fouchal, and Cyril de Runz. Driver profile detection using points of interest neighbourhood. In *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, pages 1–4, 2019. doi: 10.1109/VTCFall.2019.8891118.

[67] Yan Mao, Weihan Zhang, Meng Wang, and Da Guo. Driving simulator data based driver behavior ethogram establishment. In *2019 1st International Conference on Industrial Artificial Intelligence (IAI)*, pages 1–5, 2019. doi: 10.1109/ICIAI.2019.8850803.

[68] Brice Leblanc, Secil Ercan, and Cyril de Runz. C-its data completion to improve unsupervised driving profile detection. In *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, pages 1–5, 2020. doi: 10.1109/VTC2020-Spring48590.2020.9128731.

[69] Juliet Chebet Moso, Stéphane Cormier, Hacène Fouchal, Cyril de Runz, John M. Wandeto, and Hasnaâ Aniss. Road speed signatures from c-its messages. In *ICC 2021 - IEEE International Conference on Communications*, pages 1–6, 2021. doi: 10.1109/ICC42927.2021.9500949.

[70] H. Eren, S. Makinist, E. Akin, and A. Yilmaz. Estimating driving behavior by a smartphone. In *2012 IEEE Intelligent Vehicles Symposium*, pages 234–239, 2012. doi: 10.1109/IVS.2012.6232298.

[71] Amine Kchiche and Farouk Kamoun. Centrality-based access-points deployment for vehicular networks. pages 700 – 706, 05 2010. doi: 10.1109/ICTEL.2010.5478800.

[72] Yi Huang and Kevin Boyle. Antennas: From theory to practice. *Antennas: From Theory to Practice*, pages 1–363, 08 2008. doi: 10.1002/9780470772911.

[73] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL http://jmlr.org/papers/v9/vandermaaten08a.html.

[74] Zdenek Lokaj, Martin Srotyr, Miroslav Vanis, and Jiri Broz. Technical part of evaluation solution for cooperative vehicles within c-roads cz project. In *2020 Smart City Symposium Prague (SCSP)*, pages 1–5, 2020. doi: 10.1109/SCSP49987.2020.9133885.

[75] Hasnaâ Aniss. Overview of an its project: Scoop@f. In Jaizki Mendizabal, Marion Berbineau, Alexey Vinel, Stephan Pfletschinger, Hervé Bonneville, Alain Pirovano, Simon Plass, Riccardo Scopigno, and Hasnaa Aniss, editors, *Communication Technologies for Vehicles*, pages 131–135, Cham, 2016. Springer International Publishing. ISBN 978-3-319-38921-9.

Données des réseaux véhiculaires pour la gestion du trafic

Résumé : Alors que les communications véhiculaires continuent de se développer et d'évoluer ...

Mots clés : ....

Vehicular networks data for traffic managment

Abstract: .....

Keywords : ....

Discipline : MÉCANIQUE

Université Lyon
COSYS-ERENA,
Université Gustave Eiffel