Continuous Learning and Cooperative Prediction Based on Adaptive Multi-Agent System Applied for Traffic Dynamics Prediction

Ha-Nhi Ngo

RÉPUBLIQUE FRANÇAISE
Liberté Égalité Fraternité

Université de Toulouse

Doctorat de l'Université de Toulouse

préparé à l'Université Toulouse III - Paul Sabatier

Apprentissage continu et prédiction coopérative basés sur les systèmes de multi-agents adaptatifs appliqués à la prévision de la dynamique du trafic

---

Thèse présentée et soutenue, le 28 février 2024 par

Ha-Nhi NGO

École doctorale
EDMITT - Ecole Doctorale Mathématiques, Informatique et Télécommunications de Toulouse

Spécialité
Informatique et Télécommunications

Unité de recherche
IRIT : Institut de Recherche en Informatique de Toulouse

Thèse dirigée par
Marie-Pierre GLEIZES et Elsy KADDOUM

Composition du jury
M. Flavien BALBO, Rapporteur, Ecole Nationale Supérieure des Mines de Saint-Etienne
Mme Ann NOWE, Rapporteure, Vrije Universiteit Brussel
M. Wilco BURGHOUT, Examinateur, KTH Royal Institute of Technology
Mme Marie-Pierre GLEIZES, Directrice de thèse, Université Toulouse III - Paul Sabatier
Mme Elsy KADDOUM, Co-directrice de thèse, Université Toulouse II - Jean-Jaurès

Membres invités
Mme Anaïs Goursolle, Continental Digital Services France
M. Jonathan Bonnet, Continental Digital Services France

RÉPUBLIQUE FRANÇAISE
Liberté Égalité Fraternité

Université de Toulouse

Doctorat de l'Université de Toulouse

préparé à l'Université Toulouse III - Paul Sabatier

Continuous Learning and Cooperative Prediction Based on Adaptive Multi-Agent Systems Applied for Traffic Dynamic Predictions

---

Thèse présentée et soutenue, le 28 février 2024 par

Ha-Nhi NGO

École doctorale
EDMITT - Ecole Doctorale Mathématiques, Informatique et Télécommunications de Toulouse

Spécialité
Informatique et Télécommunications

Unité de recherche
IRIT : Institut de Recherche en Informatique de Toulouse

Thèse dirigée par
Marie-Pierre GLEIZES et Elsy KADDOUM

Composition du jury
M. Flavien BALBO, Rapporteur, Ecole Nationale Supérieure des Mines de Saint-Etienne
Mme Ann NOWE, Rapporteure, Vrije Universiteit Brussel
M. Wilco BURGHOUT, Examinateur, KTH Royal Institute of Technology
Mme Marie-Pierre GLEIZES, Directrice de thèse, Université Toulouse III - Paul Sabatier
Mme Elsy KADDOUM, Co-directrice de thèse, Université Toulouse II - Jean-Jaurès

Membres invités
Mme Anaïs Goursolle, Co-encadrante, Continental Digital Services France
M. Jonathan Bonnet, Co-encadrant, Continental Digital Services France

Ha Nhi Ngo

Continuous Learning and Cooperative Prediction Based on Adaptive Multi-Agent Systems Applied for Traffic Dynamic Predictions

Thesis Supervisors **Marie Pierre Gleizes**, Professor, Université Toulouse III Paul Sabatier
Elsy Kaddoum, Associate Professor, Université Toulouse II Jean-Jaurès
Jonathan Bonnet, PhD, Continental Digital Services France
Anais Goursolle, Data Scientist, Continental Digital Services France

Rapid advances in hardware, software, and communication technologies of transportation systems have come up with both promising opportunities and significant challenges for human society. Along with improving transportation quality, the increase in vehicles has led to frequent traffic congestion, especially in big cities at peak hours. Traffic congestion leads to numerous consequences for the economic cost, environment, human mental health, and traffic safety. Thus, predicting traffic dynamics towards anticipating the appearance of traffic congestion is an important task, helping to prevent and mitigate disrupted traffic situations, as well as dangerous collisions at the end of a jam's queue.

Nowadays, advances in Intelligent Transportation Systems (ITS) technologies have brought diverse and large-scale traffic data sets that are continuously collected and transferred between devices as real-time data streams. Consequently, many Intelligent Transportation Systems (ITS) services have been developed based on Big Data Analytics, including traffic prediction. However, traffic contains many varied and unpredictable factors that make modeling, analyzing, and learning historical traffic evolution challenging. Thus, our proposed system aims to fulfill five following components for a traffic prediction system: **temporal analysis, spatial analysis, interpretability, streaming analysis, and multiple data scale adaptability** to capture historical traffic patterns from traffic data streams, provide the explicit explanation of input-output causality and enable different applications with various scenarios.

To achieve the mentioned objectives, we propose an agent model based on dynamic clustering and adaptive multi-agent system theory to provide continuous learning and cooperative prediction mechanisms. The proposed agent model comprises two interdependent processes functioning in parallel: **local continuous learning** and **cooperative prediction**. The learning process aims to detect, at the agent level, different representative states from the received data streams. Based on dynamic clustering, this process enables continuous updates of the learned database of the representative states adapting to new data.

Simultaneously, the prediction process leverages the learned database from the learning process, aiming at estimating the future potential states that can be observed. This process takes into account

the spatial dependency analysis by integrating the cooperation between the agents and their neighborhood. The interactions between agents are designed based on the Adaptive Multi-Agent System (AMAS) theory with the set of self-adaptation mechanisms including **self-organization**, **self-correction** and **self-evolution**, allowing the system to avoid perturbations, manage the prediction quality and take into account the newly learned information in the prediction calculation.

The conducted experiments in the context of traffic dynamics prediction evaluate the system on both generated and real-world data sets in various data scales and scenarios. The obtained results have shown the outperformance of our proposal compared to the baselines when traffic data expresses high variations. Moreover, the same conclusions drawn from different study cases enhance the system's ability for multi-scale applications.

MODÈLE AGENT POUR L'APPRENTISSAGE CONTINU ET LA PRÉDICTION COOPÉRATIVE BASÉ SUR LES SYSTÈMES MULTI-AGENTS AUTO-ADAPTATIFS APPLIQUÉ À LA PRÉVISION DES DYNAMIQUES DE TRAFIC

Le développement rapide des technologies matérielles, logicielles et de communication des systèmes de transport ont apporté des opportunités prometteuses mais aussi des défis importants pour la société humaine. Parallèlement à l'amélioration de la qualité des transports, l'augmentation du nombre de véhicules a entraîné de fréquents embouteillages, en particulier dans les grandes villes aux heures de pointe. Les embouteillages ont de nombreuses conséquences sur le coût économique, l'environnement, la santé mentale des conducteurs et la sécurité routière. Il est donc important de prévoir la dynamique du trafic et d'anticiper l'apparition des embouteillages, afin de prévenir et d'atténuer les situations de trafic perturbé, ainsi que les collisions dangereuses à la fin d'un embouteillage.

De nos jours, les technologies innovantes des systèmes de transport intelligents ont apporté des ensembles de données diverses et à grande échelle sur le trafic qui sont continuellement collectées et transférées entre les dispositifs sous forme de flux de données en temps réel. Par conséquent, de nombreux services de systèmes de transport intelligents ont été développés basé sur l'analyse de données massives, y compris la prévision du trafic. Cependant, le trafic contient de nombreux facteurs variés et imprévisibles qui rendent la modélisation, l'analyse et l'apprentissage de l'évolution historique du trafic difficiles. Le système que nous proposons vise donc à remplir les cinq composantes suivantes d'un système de prévision du trafic : **analyse temporelle, analyse spatiale, interprétabilité, analyse de flux et adaptabilité à plusieurs échelles de données** pour capturer les patterns historiques de trafic à partir des flux de données, fournir une explication explicite de la causalité entrée-sortie et permettre différentes applications avec divers scénarios.

Pour atteindre les objectifs mentionnés, nous proposons un modèle d'agent basé sur le clustering dynamique et la théorie des systèmes multi-agents adaptatifs afin de fournir des mécanismes d'apprentissage continu et de prédiction coopérative. Le modèle d'agent proposé comprend deux processus interdépendants fonctionnant en parallèle : **apprentissage local continu** et **prédiction coopérative**. Le processus d'apprentissage vise à détecter, au niveau de l'agent, différents états représentatifs à partir des flux de données reçus. Basé sur le clustering dynamique, ce processus permet la mise à jour continue de la base de données d'apprentissage en s'adaptant aux nouvelles données. Simultanément, le processus de prédiction exploite la base de données apprise, dans le but d'estimer les futurs états potentiels pouvant être observés. Ce processus prend en compte l'analyse de la dépendance spatiale en intégrant la coopération entre les agents et leur voisinage. Les interactions entre les agents sont conçues sur la base de la théorie AMAS avec un ensemble de mécanismes d'auto-adaptation comprenant **l'auto-organisation**, **l'autocorrection** et **l'auto-évolution**, permettant au système d'éviter les perturbations, de gérer la qualité de la prédiction et de prendre en compte

les nouvelles informations apprises dans le calcul de la prédiction.

Les expériences menées dans le contexte de la prévision de la dynamique du trafic évaluent le système sur des ensembles de données générées et réelles à différentes échelles et dans différents scénarios. Les résultats obtenus ont montré des meilleures performances de notre proposition par rapport aux méthodes existantes lorsque les données de trafic expriment de fortes variations. En outre, les mêmes performances obtenues en considérant différents cas d'étude renforcent la capacité du système à s'adapter à des applications multi-échelles.

Acknowledgment

Three years of my doctorate journey have passed with challenges and rewards. I would like to take this opportunity to express my deep gratitude to everyone who has supported and contributed to the successful completion of my doctoral research, directly or indirectly.

Firstly, I would like to thank Pr.Flavien Balbo and Pr.Ann Nowé for accepting to be reviewers of my thesis. Your evaluations and comments help me a lot improving my thesis and preparing my defense.

I would like to extend my gratitude to Wilco Burghout for accepting my invitation to join my jury and for the collaboration between our laboratories, which began during my three-month visit to your team at KTH Royal Institute of Technology. Special thanks go to Erik, Matej, and all members of the Transport Planning team for their warm welcome and support during my stay. This collaboration has been a valuable experience, significantly contributing to the completion of my thesis.

Next, I would to express my sincere thanks to my academic supervisors, Pr.Marie-Pierre Gleizes, Pierre Glize and Elsy Kaddoum for their supervisions for my thesis and for being always by my side whenever I need you. I really appreciate all of their advises, personally or professionally, and the opportunities that they gave me. I have learned a lot from them.

I would like to thank to my industrial supervisors Jonathan Bonnet and Anaïs Goursolle for their supports and cooperation during my doctoral research. I would like to extend my appreciation to Continental Digital Services France. The company's resources and facilities have contributed in technical and financial supports in conducting the experiment and validating the practical applications of my work.

During those three years, I have always felt welcomed at IRIT laboratory and especially SMAC team. I would like to thank every member of IRIT for their helps and services. For the SMAC team, I'm grateful being your colleague and being a part of your team. Your encouragements, advises and all the joyful moments we shared during coffee breaks, after-work, *etc.* have been an important part for the successful completion of my thesis.

Aside from professional support, I'm grateful to have friends who are always by my side to share with me all up and down moments during my doctorate journey.

Although I have moved and lived far from my family for seven years, my family has never made me feel alone. I would like to thank to my dad, my mom and my brothers for their supports. They are always by my side to encourage and motivate me to be confident to continue my journey. I'm proud being a part of our family.

Lastly, I would like to thank you, Alix for everything you have done for me. Your supports and your encouragements in those last months mean a lot to me. The last part of my doctorate journey must have been more difficult to finish without you being on my side.

Ha Nhi Ngo

Contents

Chapter 1

Résumé de thèse

1.1 Introduction

De nos jours, l'amélioration de la qualité de vie entraîne l'innovation dans divers secteurs pour répondre aux besoins croissants des consommateurs, y compris le transport. Par conséquent, le volume croissant de véhicules circulant sur les routes contribue à la formation d'importants embouteillages, surtout aux heures de pointe du matin et du soir. Ces embouteillages engendrent de graves conséquences sur le trafic notamment dans les grandes villes. Nous notons tout particulièrement l'augmentation des coûts et du temps de déplacement, les problèmes environnementaux, les impacts négatifs sur la santé mentale des conducteurs et notamment la dégradation de la sécurité du trafic causée par les accidents. En effet, le risque de collisions notamment en fin d'embouteillage est considérablement élevé, surtout lorsque les conducteurs ont un temps de décélération limité. Ainsi, anticiper l'existence d'un embouteillage nécessitant une réaction rapide des conducteurs est important pour éviter ou atténuer les conséquences des accidents.

Le système d'alerte de détection d'embouteillage est actuellement une solution prometteuse pour résoudre ce défi. En exploitant la technologie V2X (communication de véhicule-à-tout), ce système est capable de transmettre des informations sur le trafic entre les véhicules pour faciliter à la prise de décision et éviter des situations dangereuses en fin d'embouteillage. Cependant, la capacité d'anticipation de ce système est limitée par la qualité de la communication entre les véhicules, telle que la distance faible de communication, la disponibilité du réseau et le débit des transmissions. Ces limitations réduisent l'efficacité des prévisions fournies, se traduisant par des temps de réaction insuffisants, une fiabilité réduite des anticipations dans les zones sans réseau et un manque d'informations anticipées.

Face à ces défis, Continental Digital Services France (CDSF), entreprise du secteur automobile et société filiale de Continental AG, a proposé un sujet de recherche pour ce doctorat. Ce sujet porte sur la modélisation et l'anticipation de fin d'embouteillages nécessitant une prévision du trafic précise et en temps réel. Dans le contexte de la prévision des embouteillages, la prévision du trafic permet

non seulement de prédire la localisation des embouteillages, mais aussi d'anticiper la longueur de la file d'attente et les caractéristiques de propagation. En outre, l'obtention d'informations sur l'évolution des dynamiques futures du trafic permet de prévenir et d'atténuer de manière proactive la formation d'embouteillages avant qu'ils ne se produisent. En plus, ces prévisions peuvent apporter un soutien aux centres de contrôle du trafic dans leurs efforts de gestion du réseau routier et d'allocation efficace des ressources routières.

Dans ce contexte, ma thèse de doctorat approche le défi posé en proposant un système de prévision du trafic nommé ADRIP (Adaptive multi-agent system for DRIving behavior Prediction), basé sur les systèmes multi-agents adaptatifs (AMAS : Adaptive Multi-Agent Systems). ADRIP est capable d'apprendre en continu à partir des flux de données du trafic et d'estimer la prévision de la dynamique future en temps réel, offrant les propriétés suivantes :

- **Dynamique :** Le système de prévision est capable d'évoluer en s'adaptant aux changements de l'environnement de conduite.
- **Ouverture :** Le système permet aux entités participantes d'entrer ou de sortir sans perturber son fonctionnement.
- **Explicabilité :** Le système est capable de fournir une explication entre les données d'entrée et la prévision estimée.
- **Capacité d'application à plusieurs niveaux de trafic :** Le système développé peut être utilisé pour différentes applications à plusieurs niveaux de trafic afin d'analyser l'influence du trafic dans différentes zones urbaines.
- **Sécurité :** Il s'agit de la capacité du système à éviter la diffusion de données à caractère personnel.

1.2 Systèmes de Transport Intelligents et Analyse des Données Massives

Cette partie résume le chapitre 3 *Intelligent Transportation Systems and Big Data Analytics*.

1.2.1 Systèmes de Transport Intelligents

Le transport est fondamental pour la société humaine, car il permet la circulation des personnes et des biens à travers le monde. Avec l'évolution rapide de la société, le transport ainsi que les infrastructures ont continuellement changé pour s'adapter et résoudre des problèmes causés par l'augmentation du volume et de la densité de véhicules, tels que les collisions, les embouteillages, etc.

Récemment, les innovations dans le domaine des systèmes embarqués, de l'information et des réseaux de communication ont entraîné le développement et l'installation d'équipements connectés,

ouvrant la voie à de nouvelles méthodes, appelées les systèmes de transport intelligents (STI). Les STI apportent des solutions innovantes, d'une part, en fournissant une assistance intelligente aux conducteurs et à la gestion, et d'autre part, en observant et en analysant les données de trafic pour améliorer le service. En conséquence, une base de données massive, diversifiée et à grande échelle est établie. Pour traiter ces données, le concept d'*Analyse des données massives* pour le domaine du transport est proposé.

1.2.2 Analyse des Données Massives dans STI

1.2.2.1 Motivations

À l'ère numérique, les systèmes d'information sont devenus omniprésents, entraînant une augmentation des données collectées et partagées à partir de diverses sources. Cette évolution demande de nouvelles méthodes de traitement et d'analyse de ces données, car la capacité des ordinateurs n'est plus suffisante pour traiter les données massives en utilisant les méthodes conventionnelles. Ainsi, depuis plusieurs années, l'analyse de données massives est devenue un sujet de recherche important dans le milieu académique et industriel, dans tous les domaines de la vie humaine, y compris la santé, l'éducation, les affaires, l'environnement, l'énergie, les transports, $etc.$ Au vu du succès de l'analyse de données massives dans de nombreux domaines, leurs applications pour les STI semblent être prometteuses.

L'adoption de l'analyse de données massives dans les STI a été motivée par l'augmentation des données de trafic collectées grâce à des technologies innovantes d'exploration des données et à l'avancement dans la recherche des méthodes de traitement des données. Pour le premier aspect, les données du trafic sont collectées à partir de différentes sources :

- Données des cartes à puce : Les systèmes de collection automatisée des titres de transport (AFC) ont été largement déployés dans les systèmes ferroviaires urbains et les réseaux de transport public, permettant aux STI d'étudier les itinéraires de déplacement des passagers [22], [118]. En effet, les cartes à puce peuvent capturer les détails essentiels du déplacement et les informations sur les passagers, telles que les informations personnelles anonymes, l'heure d'embarquement, la localisation, les informations sur l'origine et la destination, $etc.$

- Données des capteurs de perception environnementale des véhicules intelligents : De nombreux capteurs internes et externes ont été installés, permettant aux véhicules d'obtenir les données embarquées et de percevoir les informations relatives à l'environnement de conduite via des équipements tels que : GPS (le système mondial de positionnement), radars, la télédétection par laser ou LIDAR, les capteurs de vision, le bus de données CAN.

- Données des communications véhiculaires : La technologie V2X permet aux véhicules connectés d'échanger leurs informations. De plus, le réseau ad hoc de véhicules (VANet) consiste en des groupes de véhicules et d'infrastructures connectés par des réseaux sans fil, dans

le but d'étendre la zone de communication et d'améliorer la capacité des communications véhiculaires.

- Données des capteurs fixes : Les capteurs de position fixe sont regroupés dans les types suivants : boucle inductive, capteurs magnétiques, processeurs d'images vidéo, capteurs radar à micro-ondes, capteurs infrarouges, capteurs radar laser et capteurs audio [101]. Ils mesurent des données telles que le nombre de véhicules ($i.e.$ volume), la vitesse ou la densité du trafic.

- Données d'autres sources : Des données de trafic sont également collectées à partir des réseaux sociaux tels que Facebook, LinkedIn, Twitter, etc., via le partage des utilisateurs pour échanger des informations mises à jour, des annonces et interagir avec les fournisseurs.

Pour traiter ces données, plusieurs modèles d'intelligence artificielle (IA) ont été développés, notamment basés sur l'apprentissage automatique, catégorisés en 4 types d'apprentissage : l'apprentissage non supervisé, l'apprentissage supervisé, l'apprentissage par renforcement et l'apprentissage semi-supervisé. Ces méthodes permettent aux STI d'extraire des informations utiles à partir des données de trafic et d'estimer des recommandations ou des prévisions de trafic.

1.2.2.2 Avantages et Défis

Ainsi, de nos jours, la méthode d'analyse des données massives est largement appliquée dans les STI, adressant différents aspects, y compris : **l'analyse et prévision des accidents du trafic, la prévision du trafic, la planification du transport en commun, la planification d'itinéraires personnels, la gestion et contrôle des transports ferroviaires et la maintenance des biens**. Cette méthode a montré ses avantages par rapport aux méthodes classiques pour les STI notamment :

- La capacité à modéliser des scénarios de trafic complexe et dynamique et à capturer des schémas imprévisibles sans hypothèses ;

- L'adaptabilité en temps réel aux changements de l'environnement de conduite en étudiant les flux de données ;

- Le passage à grande échelle avec des volumes importants de données, la variété et la haute vitesse de transmission des données ;

- La capacité à obtenir des informations directement à partir des données, offrant ainsi une meilleure connaissance des patterns de trafic irréguliers et imprévisibles.

Malgré les avantages acquis, la mise en place d'une méthode d'analyse de données massives dans les STI, notamment la méthode de prévision du trafic basée sur des données, est un véritable défi. Afin d'obtenir une performance adéquate, cette méthode doit être capable de répondre à plusieurs éléments. Premièrement, **l'analyse de la dépendance temporelle et spatiale à long terme**

doit être intégrée dans les méthodes de prévision du trafic afin de capturer les propriétés temporelles telles que la saisonnalité, la dépendance non linéaire, *etc.*, ainsi que l'interdépendance entre des segments de route dans le réseau routier. Deuxièmement, **l'adoption d'une analyse continue** pour le flux de données est importante pour intégrer les nouvelles informations du trafic au modèle et fournir des prévisions à jour. Cet élément pose également un nouveau défi concernant l'équilibre entre la complexité du modèle pour assurer une analyse complète des dépendances et la flexibilité du modèle pour les mises à jour dynamiques. Troisièmement, **la prise en compte de l'explicabilité** du modèle est également nécessaire pour fournir une explication explicite de la causalité entre les données en entrées et la sortie du système. Enfin, **un modèle prédictif du trafic qui fonctionne bien à différentes échelles de données** peut être intéressant pour étudier les influences du trafic entre les différentes zones urbaines.

1.3 État de l'art

Cette partie résume le chapitre 4 *State of the Art of Traffic Prediction*.

De nombreuses études existantes se sont concentrées sur la prévision de la dynamique du trafic en estimant la vitesse moyenne, le volume ou la densité du trafic pour différents horizons de prévision. Selon le contexte considéré et le cadre étudié dans cette thèse, nous nous concentrons uniquement sur l'analyse des approches de prévision du trafic basées sur l'analyse des données massives. Nous classons ces modèles en trois groupes principaux en fonction des techniques utilisées : **modèles basés sur les séries temporelles, modèles basés sur le clustering et modèles basés sur les réseaux neuronaux**.

1.3.1 Modèles basés sur les séries temporelles

Les modèles basés sur les séries temporelles expriment les prévisions via une formule mathématique proposant une combinaison linéaire de différents termes tels que les données précédentes, le bruit aléatoire, la propriété saisonnière, la moyenne mobile, *etc.* Parmi ces modèles, les modèles paramétriques et notamment la famille des modèles ARIMA sont les plus connus pour la prévision du trafic. Ils montrent la dépendance linéaire des valeurs futures par rapport aux valeurs précédentes (Auto-Regressive - AR) et aux séries de bruit aléatoire (Moving Average - MA) avec l'hypothèse de stationnarité. ARIMA, une version étendue de ARMA appliquée à [134], [26] et [110], peut traiter des données non stationnaires. Dans [188], la propriété de saisonnalité est incluse dans le modèle de prévision, appelé SARIMA, ce qui permet d'améliorer la performance de la prévision quand le trafic contient des patterns répétés. Certains modèles plus récents peuvent traiter les relations spatiales, comme les modèles multivariés appelés VARMA (Vector Auto-Regressive Moving Average) et STARIMA (Space-Time Autoregressive Integrated Moving Average). Les résultats obtenus dans [95] et [130] ont démontré l'amélioration des performances de prévision par des modèles multivariés appliqués à de grands réseaux avec un grand nombre de capteurs.

Ces modèles sont très explicites et clairs pour comprendre les relations entre les paramètres et les sorties. La mise en œuvre et l'exécution de ces modèles ne nécessite pas de grandes capacités de calcul. Ils peuvent atteindre de bonnes performances dans des trafics réguliers. Cependant, ils ne peuvent résoudre que des problèmes linéaires avec des fortes hypothèses imposées sur les données de trafic, ce qui n'est pas adapté aux applications complexes et aux données de trafic irrégulier.

1.3.2 Modèles basés sur le clustering

L'approche basée sur le clustering est l'une des méthodes non supervisées la plus connue, appliquée à divers aspects de l'intelligence artificielle, notamment la classification, la détection de modèles, la prévision, *etc.*, permettant de découvrir la structure des ensembles de données analysés. Elle répartit les données d'entrée dans différents groupes en fonction de leur similarité. L'interprétation est donc facilitée par la compréhension du critère de regroupement et de la définition de la distance de similarité.

Les modèles de clustering ont été appliqués à la prévision du trafic pour les aspects temporels et spatiaux. Pour l'application temporelle, le modèle K-Nearest Neighbor (KNN) a été étudié dans [203] et [30] pour regrouper des séquences de flux de trafic similaires au cours d'une fenêtre temporelle fixe. La prévision est ensuite estimée en calculant la moyenne des états suivants à partir des observations historiques appartenant au même groupe. De plus, DBSCAN (Density-Based Spatial Clustering of Applications with Noise) a été utilisé dans [157] et [161] pour détecter et prévoir les embouteillages ou les situations de trafic anormales. Le GMM (Gaussian Mixture Model) [149] a été appliqué en hybridation avec le réseau bayésien pour la prévision du trafic à l'échelle du réseau, montrant ses avantages en termes d'explicabilité, de généralisation et d'efficacité. Le clustering agglomératif a été appliqué dans [15] pour une méthode d'estimation et de prévision de l'état du trafic local basée sur des données dynamiques. Cette méthode montre la flexibilité de cette approche dans l'incorporation de variables explicatives supplémentaires.

En outre, pour le **clustering spatial**, les modèles de clustering sont appliqués pour le partitionnement des routes. En effet, de nombreuses études ont démontré les avantages des relations spatiales entre les routes voisines pour améliorer les performances de prévision pour étudier la propagation du trafic ou pour adapter le modèle aux caractéristiques des routes. Dans [91], [156], les auteurs appliquent l'algorithme K-Means pour la localisation des routes afin de regrouper les réseaux urbains. Pour cela, ils définissent la nouvelle matrice de similarité entre les observations et appliquent l'algorithme N-cut pour la décision de regroupement. Un autre travail présenté dans [36] applique différentes méthodes de partitionnement pour diviser le réseau en groupes et utilise un modèle de prévision adaptatif pour chaque groupe.

Récemment, de nombreux travaux ont visé à construire des modèles de clustering pour l'aspect temporel et spatial. La régularité du trafic a été étudiée à l'aide d'une carte 3D, qui consiste en une partition conjointe de l'espace (liens du réseau routier) et du temps (observations de séries temporelles) appliqués à la détection des embouteillages présentée dans [119] et à la prévision du

trafic dans [35] et [37].

Les modèles basés sur le clustering ont élargi leur utilisation pour les problèmes de prévision du trafic en raison **de leur explicabilité et de leur capacité à s'appliquer sans hypothèses spécifiques sur les données**. Cependant, la performance des modèles basés sur le clustering dépend fortement de la sélection de paramètres tels que le seuil de similarité, le nombre de clusters, le nombre minimum de points de données requis pour définir un cluster, $etc.$ Cet inconvénient impacte la reproductibilité et la robustesse de ces modèles.

1.3.3 Modèles basés sur les réseaux neuronaux

Des recherches récentes dans le domaine de la prévision du trafic ont mis en évidence les promesses des modèles basés sur les réseaux neuronaux (NN) dans le traitement et l'analyse des données massives à grande dimension. Grâce aux progrès au niveau des puissances de calcul et des traitements graphiques, les ordinateurs peuvent exécuter des modèles d'apprentissage utilisant des données de trafic plus complexes et capturer des niveaux de dépendance plus élevés.

Le réseau de neurones à action directe (FFNN), un réseau neuronal simple, a été utilisé dans [170] pour estimer plusieurs étapes des prochains flux de trafic sur plusieurs segments de route. De plus, cette méthode est utilisée dans [106] et [145] avec succès pour estimer la prévision du trafic dans des scénarios complexes. Les modèles basés sur les RNN (réseaux neuronaux récurrents), tels que le réseau récurrent à mémoire courte et long terme (LSTM) ou le réseau récurrent à portes (GRU), sont spécialement conçus pour traiter les problèmes de dépendances à long terme. Leurs applications sont présentées dans [123], [174], [197] et [201], soulignant leur efficacité dans la prévision du trafic. Des approches récentes, notamment les réseaux neuronaux convolutifs (CNN) et les réseaux convolutifs graphiques (GCN), visent à intégrer la modélisation de la corrélation spatiale dans les modèles précédents. Les travaux dans [124], [41], [206] et [89] démontrent une amélioration significative de la précision de la prévision avec l'intégration de la dépendance spatiale dans le modèle.

Malgré le progrès significatif des modèles basés sur les réseaux neuronaux dans l'amélioration de la qualité de la prévision du trafic, ces modèles rencontrent des limitations liées à la complexité du modèle et à leur caractère de "boîte noire". En effet, ces méthodes demandent une capacité de calcul très élevée et ne peuvent pas fournir une explication explicite de la fonctionnalité du modèle.

1.3.4 Discussion

Approche	Modèle	Modélisation temporelle	Modélisation spatiale	Analyse continue	Explicabilité	Flexibilité
Modèles basés sur les séries temporelles	ARIMA	+	- -	- -	++	- -
	SARIMA	+	- -	- -	++	- -

	Modèles multivariés	+	+	- -	++	- -
Modèles basés sur le clustering	Clustering temporel	+	-	- -	++	+
	Clustering spatial	+	+	- -	++	+
Modèles basés sur les réseaux neuronaux	FFNN	+	-	- -	-	-
	RNN	++	-	- -	- -	-
	CNN	++	+	- -	- -	-
	GNN	++	++	- -	- -	+

Table 1.1: Liste comparative des modèles de prévision du trafic

Après avoir effectué une étude sur l'état de l'art des méthodes de prévision du trafic, la Table 1.1 fournit une évaluation par rapport aux cinq critères que nous avons identifiés pour obtenir une performance adéquate du modèle prédictif. Selon cette table, aucune des méthodes existantes ne peut répondre à tous ces critères. Dans les sections suivantes, nous présentons **le clustering dynamique et le paradigme multi-agent**. Le clustering dynamique offre une méthode d'analyse efficace pour les flux de données, tandis que les systèmes multi-agent sont bien connus pour la résolution de problèmes complexes. Ils sont largement utilisés dans divers domaines pour résoudre des défis similaires à ceux rencontrés dans la prévision du trafic et ont obtenu des résultats prometteurs. Grâce à leurs propriétés, nous basons notre proposition sur ces techniques afin de répondre aux cinq critères mentionnés.

1.4 Clustering dynamique

Cette partie résume le chapitre 5 *Dynamic Clustering*.

Dans le monde numérique, les données sont transférées, collectées et stockées automatiquement, à grande échelle et en flux continu. Par conséquent, nous observons des séquences massives et illimitées de données transférées chaque seconde, appelées flux de données. Le flux de données est défini comme une séquence de données qui arrivent continuellement dans le système, avec une taille infinie, une distribution inconnue et non stationnaire. Ces caractéristiques exigent une nouvelle approche conçue pour résoudre les défis posés par l'évolution des flux de données. Parmi les nombreuses méthodes de traitement du flux de données, le clustering dynamique est reconnu pour sa capacité à s'adapter aux changements structurels par des comportements tels que la création de nouveaux clusters, la fusion de clusters existants, la division d'un cluster existant en plusieurs clusters, la suppression d'un cluster et l'ajustement des clusters.

Grâce à ces propriétés intéressantes, de nombreux algorithmes du clustering dynamique sont développés tel que le Gaz Neuronal, les méthodes hiérarchiques, les méthodes de partitionnement,

les méthodes basées sur la densité ainsi que les méthodes basées sur les agents.

Ces algorithmes sont appliqués pour résoudre divers problèmes dans différents domaines : Internet des objets (IoT) [97], [98], [18], [65], Géochimie [25], [44], Protection des données personnelles [167], Détection des intrusions dans les réseaux informatiques [88], [103], Graphe du Web [104], Système de transport intelligent [9], Médical [193], Segmentation de la clientèle [34], [122] et Analyse des marchés financiers [79], [21].

L'application du clustering dynamique est nouvelle pour les STI. Cependant, dans le cadre de l'exploration et de l'analyse des données des STI, le clustering pour la détection des modèles de trafic est un sujet important car il permet d'améliorer les stratégies de gestion du trafic afin d'améliorer l'efficacité et la sécurité des transports urbains. En effets, certaines caractéristiques spécifiques des données du trafic exigent un nouvel algorithme, intégrant les propriétés suivantes : (1) des changements structurels dynamiques et flexibles pour prendre en compte les nouveaux comportements des données, (2) un traitement en ligne pour s'adapter à une fréquence élevée d'arrivée des données, (3) des critères de regroupement dynamiques, (4) un traitement de données hétérogènes et (5) un stockage efficace des données pour un volume massif de données. Basés sur les avancés du clustering dynamique et des défis posés par le contexte des STI, nous proposons un nouvel algorithme de clustering dynamique et l'intégrons dans notre système de prévision.

1.5 Systèmes multi-agents

Cette partie résume le chapitre 6 *Multi-Agent System*.

Face aux problèmes complexes et réels, les méthodes de modélisation conventionnelles ne sont plus adaptées en raison du grand nombre de composants, de l'environnement ouvert et dynamique, ainsi que des diverses interactions entre eux, conduisant à de nombreuses situations imprévisibles. Une nouvelle approche, appelée système multi-agent (SMA), est reconnue comme une solution efficace pour résoudre des problèmes complexes grâce au calcul distribué et au contrôle décentralisé au niveau de l'agent.

Dans les SMA, les agents sont des entités autonomes qui possèdent leurs propres compétences, règles de décision et objectifs. Ils évoluent dans l'environnement du système tout en ayant une connaissance partielle de celui-ci. Ils sont capables d'interagir les uns avec les autres et avec leur environnement. Chaque agent effectue un cycle de vie continu comprenant trois étapes : la perception, la décision et l'action. Durant la première étape, les agents perçoivent de nouvelles informations à partir de leur environnement. Lors de l'étape de décision, les agents choisissent une action en fonction de leurs perceptions de l'environnement. Dans la dernière étape, les agents exécutent l'action sélectionnée à l'étape précédente et modifient localement leur environnement.

Un SMA est ainsi composé d'un ensemble d'agents interactifs, qui peuvent être de même type ou de types différents. Chaque agent possède un objectif et une fonction locale définie en fonction de son objectif. Par leurs interactions locales, la fonction globale du système émerge, nous parlons ainsi

de phénomène d'émergence. Dans le cadre de la théorie AMAS, les interactions locales des agents sont guidées par la coopération. Le comportement coopératif des agents leur permet d'organiser leur relation d'une manière autonome et de s'auto-adapter à des évènements dynamiques. Ainsi, déterminer les cas de non-coopération et proposer des solutions sont des tâches importantes dans la conception d'un AMAS.

La capacité de calcul distribué et de contrôle décentralisé des AMAS est adaptée pour résoudre des problèmes complexes dans les STI, qui impliquent de nombreux éléments interactifs et le concept de service centré sur l'utilisateur. En effet, les AMAS ont été utilisé pour diverses applications dans le domaine des STI, notamment la simulation, le contrôle et la gestion des STI [126], [74], [165], et [49]. Ces applications ont démontré l'efficacité des AMAS pour les STI, renforçant l'optimisme quant à leur utilisation dans la résolution des problèmes de prévision du trafic.

1.6 La prévision du trafic routier à différentes échelles

Cette partie résume la définition du problème de la prévision du trafic routier à différentes échelles qui est détaillée dans le chapitre 7 *ADRIP - Adaptive multi-agent system for DRiving behaviors Prediction*, section 7.1.

La gestion du trafic routier est devenue de plus en plus difficile et complexe dans le contexte actuel du réseau routier et de son infrastructure. Cette complexité est due au nombre croissant de véhicules, à l'intégration d'équipements de circulation intelligents et à la nature complexe de la topologie du réseau routier. Ces facteurs entraînent de nombreuses interactions possibles entre des entités hétérogènes, notamment physiques et logiques, dans un environnement dynamique. Par conséquent, l'adoption d'une approche multi-niveau, intelligente et distribuée est précieuse pour la gestion du trafic, tel qu'une architecture à multi-niveau utilisant l'approche multi-agents.

1.6.1 Architecture du réseau routier

Nous proposons une architecture adaptée à notre étude, présentée dans la figure 1.1 (à droite), basée sur l'architecture hiérarchique du réseau routier multi-agents proposée par [96] (à gauche). La définition des entités du réseau routier est présentée du niveau microscopique au niveau macroscopique :

- **Véhicule (niveau 1) :** les entités au niveau le plus bas de l'architecture du réseau routier. Ils collectent les *données de véhicule* à haute fréquence le long de leurs itinéraires et partagent ces données avec d'autres entités de l'architecture du réseau routier grâce à la connectivité V2X.

- **Capteur fixe (niveau 2) :** ce niveau concerne les capteurs installés sur le segment de route permettant la collecte de données agrégées à partir des véhicules qui se croisent, telles que la vitesse moyenne, le nombre de véhicules, $etc.$

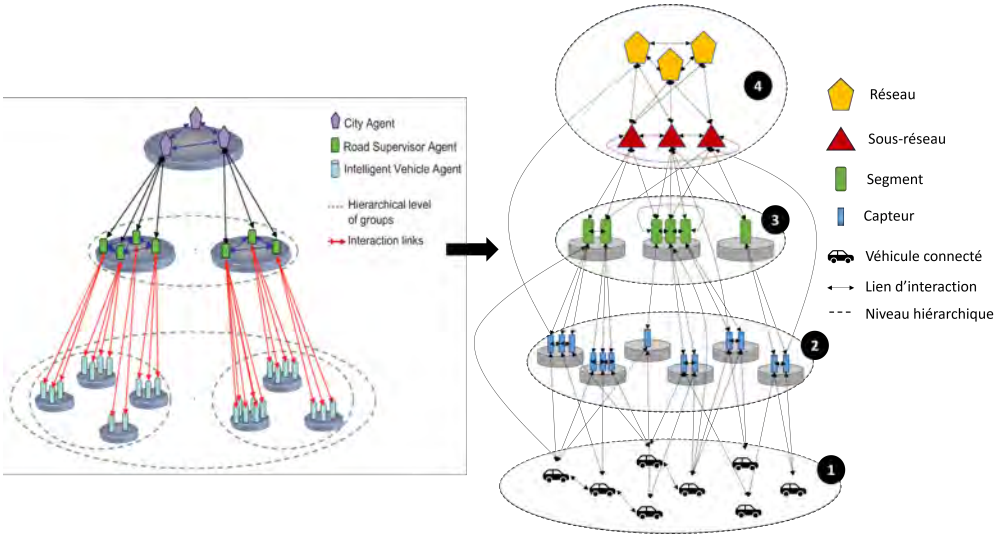


Figure 1.1: Architecture hiérarchique du réseau routier multi-agents proposée par [96] (à gauche) et son extension pour cette étude (à droite).

- **Segment (niveau 3) :** ce niveau concerne les entités logiques associées aux segments de route. La division de l'infrastructure routière en segment suit la définition proposée par Open Street Map (OSM).
- **Sous-réseau (niveau 4) :** un sous réseau est représenté pour une entité logique gérant la dynamique du trafic au niveau d'un groupe donné de segments ($e.g.$ autoroutes, quartiers, villes, etc).
- **Réseau (niveau 4) :** Le niveau le plus macroscopique de l'infrastructure, il contient l'entité logique liée au réseau routier géographique étudié.

Dans l'architecture proposée, les entités communiquent avec d'autres entités du même niveau ou à des niveaux différents. Dans cette communication flexible, chaque entité du réseau routier peut jouer deux rôles : **l'entité fournissant des données**, qui collecte des données sur le trafic et partage ses observations avec des entités à d'autres niveaux, et **l'entité de traitement**, qui perçoit des données à partir des autres et estime la prévision du trafic. L'entité fournisseur de données se situe toujours à un niveau plus bas que l'entité de traitement.

1.6.2 Problème de prévision du trafic multi-échelles : Définition

Nous définissons ainsi le problème de prévision du trafic multi-échelle considéré dans notre étude par :

Étant donné **un flux de données de trafic** $DS = \{DP_{Ts_1}, \ldots, DP_{Ts_t}, \ldots, DP_{Ts_N}\}$ constitué d'une séquence de N points de données (DP) arrivant pendant les timestamps $Ts_1, \ldots, Ts_t, \ldots, Ts_N$ et **envoyés par des entités fournissant des données, les entités de traitement analysent et apprennent,**

à chaque Ts_i, les DP_{Ts_i} perçus et calculent les prévisions de trafic pour les timestamps futures (jusqu'à l'horizon de prévision requis).

1.7 ADRIP - Adaptive multi-agent system for DRIving behaviors Prediction

Cette partie résume la présentation d'ADRIP détaillée dans le chapitre 7 *ADRIP - Adaptive multi-agent system for DRiving behaviors Prediction*.

Pour résoudre le problème de prévision du trafic à différentes échelles, nous proposons le système ADRIP (Adaptive multi-agent system for DRIving behaviors Prediction) possédant les caractéristiques suivantes :

- **apprentissage continu :** L'adaptation du comportement d'ADRIP aux changements de l'environnement.
- **apprentissage local :** processus de distribution de l'apprentissage du modèle au niveau de l'agent pour renforcer l'ouverture du système, décentraliser le stockage et la collecte des données et réduire le **temps de calcul**, ce qui permet un traitement en temps réel des flux de données de trafic.
- **explicabilité du modèle :** la capacité d''ADRIP à fournir **l'explication de la causalité entrée-sortie**.
- **prévision coopérative :** processus de collecte des données nécessaires à une prévision précise basé sur l'**interaction coopérative entre les entités**.
- **forte flexibilité du modèle :** la capacité d'instancier ADRIP pour la résolution de diverses applications.

1.7.1 ADRIP : Architecture générale

L'architecture générale d'ADRIP est présentée dans la figure 1.2. ADRIP se compose de deux processus principaux : **processus d'apprentissage local (L-ADRIP)** et **processus de prévision coopérative (P-ADRIP)**.

Le **processus d'apprentissage** est composé d'un **système de clustering adaptatif et dynamique basé sur un SMA** pour détecter localement différentes dynamiques de trafic à partir du flux de données transmis des entités fournissant les données aux entités la traitant. Le processus consiste à regrouper les données dans différents clusters, **chacun représentant une dynamique de trafic**. L'ensemble des clusters construits constitue une **base de données locale apprise** par une entité de traitement. Pour effectuer le clustering dynamique, L-ADRIP définit deux types d'agents : **Agent**

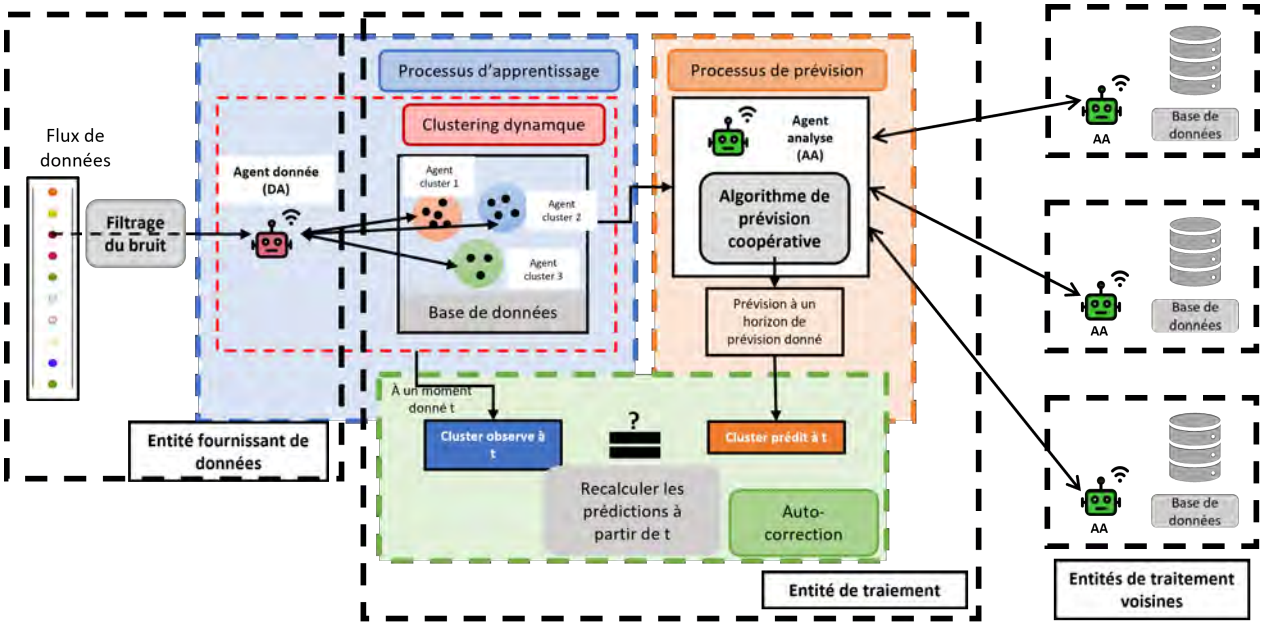


Figure 1.2: ADRIP : Architecture générale

donnée représentant une donnée (une dynamique du trafic) et **Agent cluster** représentant un cluster. Un nouvel agent donnée interagit avec des agents cluster existants pour intégrer la dynamique qu'il présente dans le système. Leurs interactions sont illustrées dans la Figure 1.3. Le nouvel agent donnée recherche une liste d'agents cluster similaires en envoyant une demande à tous les agents cluster existants. Ces derniers communiquent en évaluant leur degré de similarité. Trois cas sont à distinguer :

- Si aucun agent cluster existant est similaire, l'agent donnée crée un nouveau cluster pour sa donnée.
- S'il existe plusieurs agents cluster similaires, ces agents s'auto-évaluent pour décider s'ils doivent fusionner.
- S'il existe un agent cluster similaire, cet agent est ajusté pour prendre en compte la nouvelle donnée.

Simultanément au processus d'apprentissage, le **processus de prévision P-ADRIP** exécute une **méthode de prévision coopérative** qui fournit des estimations de prévision du trafic jusqu'à l'horizon de prévision requis. Pour effectuer le calcul des prévisions, P-ADRIP définit un **agent analyse**.

- **Agent analyse (AA)** : un AA est associé à chaque entité de traitement de données, il est responsable du calcul de la prévision.

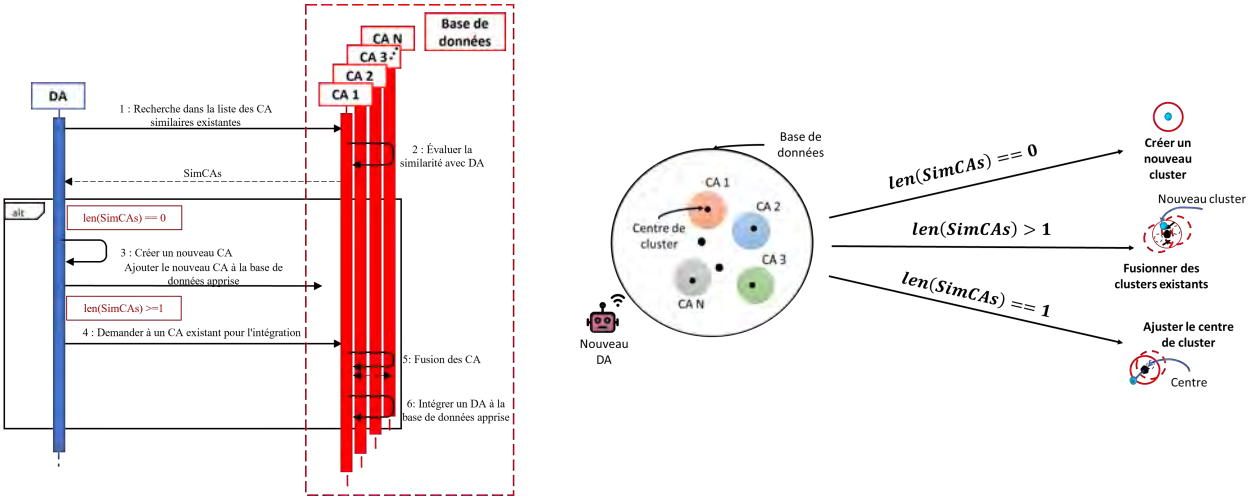
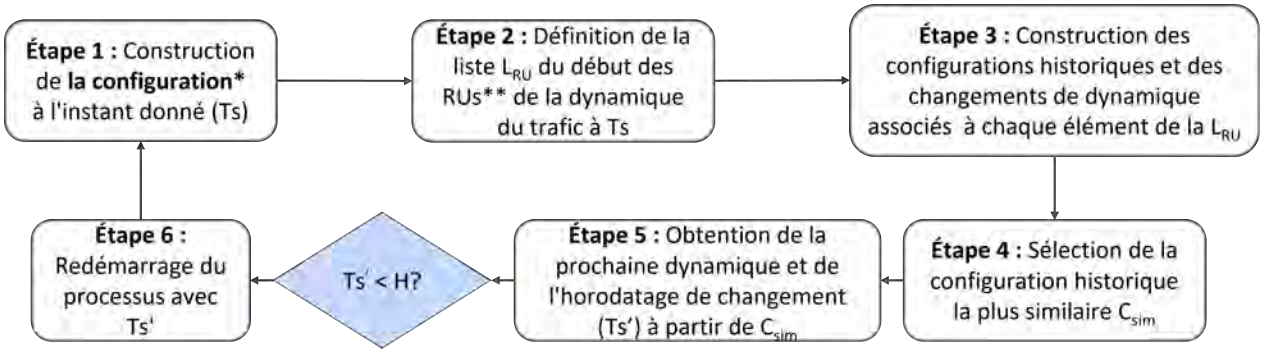


Figure 1.3: Diagramme d'interaction et règle de décision dans le L-ADRIP

L'AA est capable d'analyser la base de données locale issue du processus d'apprentissage au sein de l'entité de traitement et de coopérer avec les AA des entités de traitement voisines afin de rassembler les informations nécessaires au calcul de la prévision. La définition du voisinage d'une entité du réseau routier dépend des applications qui visent à étudier la propagation de la dynamique du trafic. Le processus de prévision prend donc en compte les dépendances spatiales. Le comportement de l'AA suit les étapes illustrées dans le schéma 1.4. Le principe de ce comportement est, étant donné une configuration du trafic constitué localement sur un ensemble de segments voisins à un instant donné Ts, de déterminer la configuration la plus similaire dans le passé, appelée la configuration historique. La prédiction est alors définie comme la dynamique du trafic qui a été observé après cette configuration historique similaire. Cette prédiction dure aussi longtemps que la dynamique du trafic sélectionnée a duré dans le passé. Si l'horizon de prévision n'atteint pas l'horizon demandé, l'AA lance à nouveau l'algorithme de prévision en utilisant comme entrées la dernière dynamique calculée et son timestamp. Ce processus est répété jusqu'à ce que l'AA atteigne au moins l'horizon de prévision demandé.

L-ADRIP et P-ADRIP fonctionnent en parallèle pour garantir que les prévisions fournies sont à jour et de haute précision. Cependant, il est difficile de maintenir une bonne performance de prévision à long terme. Pour pallier à cela, chaque AA intègre **un mécanisme d'auto-correction** permettant de détecter et corriger les mauvaises prévisions. Concrètement, l'AA compare la dynamique du trafic prévue par P-ADRIP et celle observée par L-ADRIP. Si la différence entre les deux dépasse un seuil de similarité prédéfini, l'AA recalcule la prévision en utilisant la configuration actuelle. En effet, ce mécanisme permet de réduire la dégradation des performances de prédiction à long terme.



**Configuration* : *la configuration à l'instant T du point de vue de l'AA est l'ensemble des dynamique du trafic et de leurs RU correspondants à T des AAs voisins*
***RU* : *plage d'utilisation d'une dynamique du trafic*

Figure 1.4: Les étapes du processus de prévision dans le P-ADRIP

1.7.2 Généricité du comportement d'ADRIP

ADRIP est présenté comme **une solution générique** capable de résoudre les problèmes de prévision du trafic à différents niveaux de l'architecture du réseau routier étudié. Ses comportements sont décrits par l'échange de données sur le trafic entre les entités fournissant les données et les entités de traitement et par l'estimation de la prévision au niveau des entités de traitement. Ainsi, appliquer ADRIP nécessite de spécifier deux éléments principaux qui s'adaptent aux scénarios envisagés. Le premier élément concerne **les caractéristiques des flux de données de trafic** utilisées comme entrées et sorties. La représentation de la dynamique du trafic à différents niveaux du réseau routier peut être variée. Par exemple, les données de trafic collectées à partir des véhicules offrent des informations microscopiques telles que les profils de conduite individuels. Parallèlement, les données agrégées collectées par des capteurs peuvent fournir des informations macroscopiques sur le trafic, telles que le flux ou la densité du trafic. Deuxièmement, la diversité des données sur le trafic nécessite la prise en compte de **mesures de similarité et seuils** distincts. Le choix de ces éléments doit garantir la représentativité de différents états du trafic.

1.8 Instantiation et Évaluation

Cette partie résume l'application d'ADRIP et son évaluation présentées dans le chapitre 8 *ADRIP Instantiations and Evaluations*.

Dans le cadre de cette thèse, nous avons instancié ADRIP sur deux cas : **prévision du trafic pour le niveau d'information microscopique (véhicule-segment)** et **prévision du trafic pour le niveau d'information macroscopique(capteur-segment)**. Pour chaque cas, nous détaillons l'instanciation d'ADRIP, y compris les entités du réseau routier qui jouent le rôle de fournisseurs de données et d'entités de traitement, la définition de la dynamique du trafic, la définition de la similarité et le

seuil de similarité adapté.

- **Prévision du trafic pour le niveau d'information microscopique (véhicule-segment)**

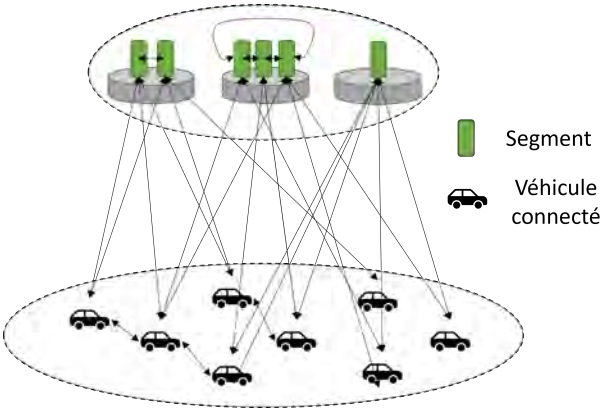


Figure 1.5: Scénario de prévision du trafic pour l'instanciation du niveau d'information microscopique

Entités fournissant des données : un ensemble de véhicules $V = v_1; v_2; ...; v_n$. Chaque véhicule suit un itinéraire I segmenté en une séquence de segments de route notés $I = \{rds_1, \ldots, rds_d\}$.

Entités de traitement : un ensemble de segments de route déterminé en fonction du réseau routier dans Open Street Map (OSM), leurs points de départ et d'arrivée étant localisés par des dispositifs GPS.

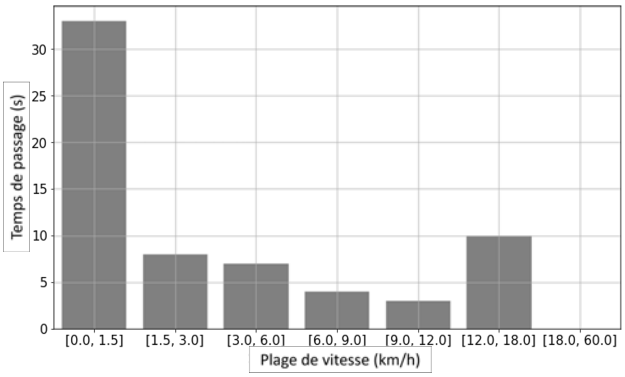


Figure 1.6: Illustration de MP

L'information du trafic étudiée dans ce scénario est représentée par le **Profil de Mobilité (MP)** (l'illustration graphique est montrée dans la figure 1.6). Le profil de mobilité est défini comme la distribution du temps de passage sur le segment par rapport à différentes plages de vitesse. Avec un MP, il est possible de communiquer aux conducteurs des informations précises telles que la durée totale du trajet, la vitesse moyenne ou la variation de la vitesse. En outre, par

rapport à la série temporelle des vitesses, un MP est plus succinct et s'adapte aux restrictions de mémoire et de temps de calcul pour l'apprentissage continu et la prévision en temps réel.

- **Prévision du trafic pour le niveau d'information macroscopique (capteur-segment)**

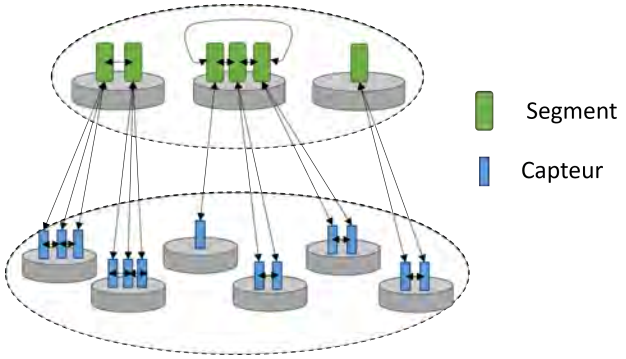


Figure 1.7: Scénario de prévision du trafic pour l'instanciation du niveau d'information macroscopique

Entités fournissant des données : un ensemble de capteurs fixes installés sur les segments de route dans le réseau considéré. Des capteurs peuvent collecter des données agrégées sur le trafic provenant des véhicules qui se croisent pendant une fenêtre temporelle, telles que la vitesse du trafic, le flux du trafic, etc. Plusieurs capteurs peuvent être déployés sur le même segment de route, ce qui permet d'observer différents états du trafic à différents endroits de la route.

Entité de traitement : un ensemble de segments de route déterminé en fonction du réseau routier dans Open Street Map (OSM), leurs points de départ et d'arrivée étant localisés par des dispositifs GPS.

L'information du trafic étudiée dans ce scénario est représentée par le **Vecteur d'observation**. Un vecteur d'observation est désigné par $O_{Ts_t} = \{o^1_{Ts_t}, \ldots, o^i_{Ts_t}, \ldots, o^I_{Ts_t}\}$, où $o^i_{Ts_t}$ représente les données de trafic provenant du capteur i et I le nombre de capteurs sur un segment de route donné. Les vecteurs d'observation sur différents segments de route n'ont pas la même taille, en fonction du nombre de capteurs installés sur le segment de route considéré. Par exemple, les segments de route de grande longueur ou à voies multiples peuvent être équipés de plusieurs capteurs afin de capturer des données sur le trafic à différents endroits.

L'évaluation d'ADRIP se fait en deux étapes : **l'évaluation du processus d'apprentissage continu et du processus de prévision coopératif**. Le processus d'apprentissage est évalué à l'aide d'un ensemble de données simulé et réel. Dans le premier ensemble de données, ADRIP a obtenu de meilleures performances que CluStream en termes de métriques de silhouette. Pour le second ensemble de données, une évaluation prédictive basée sur le clustering a été réalisée. ADRIP a démontré son efficacité, en particulier dans les scénarios où les données de trafic présentent une

grande variation, en surpassant le clustering agglomératif, K-Means et le clustering spectral à l'aide de la métrique MAPE. Dans les deux cas, **la méthode d'apprentissage continu d'ADRIP a montré son avantage par rapport aux méthodes statiques lorsqu'il s'agit de données à forte variation**.

L'évaluation du processus de prévision coopératif comprend **la comparaison des valeurs prédictives avec les observations réelles du trafic, l'évaluation de la précision de la prévision, l'évaluation du mécanisme d'auto-correction et le stockage des données**. Trois points essentiels ressortent de cette évaluation. Premièrement, le mécanisme d'auto-correction s'avère efficace pour corriger les prévisions inexactes et limiter la dégradation de la qualité des prévisions. Deuxièmement, le nombre d'instances d'activation du mécanisme d'auto-correction reste raisonnable, ce qui garantit la faible exigence en termes de temps et de coûts de calcul d'ADRIP. Troisièmement, ADRIP permet un stockage efficace des données tout en maintenant des performances de prévision adéquates.

En résumé, ADRIP améliore considérablement la prévision du trafic, en particulier dans les scénarios caractérisés par une forte variation du trafic, en surpassant les méthodes actuelles de pointe. Il assure également la mise à jour en temps voulu des informations prédites afin de garantir la précision des prévisions. En outre, ADRIP fait preuve d'efficacité dans le stockage des données, soulignant sa capacité à gérer et à stocker efficacement les informations tout en garantissant des prévisions robustes et précises.

1.9 Conclusions et Perspectives

Cette partie résume le chapitre 9 *Conclusion and Perspectives*. Dans cette thèse, nous avons présenté ADRIP, une solution générique pour les prédictions dynamiques du trafic à plusieurs niveaux, basée sur l'approche du clustering dynamique et les systèmes multi-agents.

L'architecture d'ADRIP permet de traiter efficacement les défis du problème de prédiction du trafic à plusieurs niveaux : **L'analyse temporelle** est incluse dans le processus d'apprentissage en étudiant la durée des dynamiques du trafic et la transition entre eux, **l'analyse spatiale** est traitée grâce à la coopération entre les agents analyse dans le processus de prédiction, **l'analyse du flux** est intégrée dans l'algorithme de clustering dynamique, **l'explicabilité claire du modèle** est obtenue en comprenant la décision d'affectation de l'algorithme de clustering et finalement **la forte flexibilité du modèle** est obtenue grâce à l'ouverture des systèmes multi-agents.

Dans les expériences menées, ADRIP est instancié sur deux cas : **prédictions de trafic pour des données microscopiques et macroscopiques**. Dans les deux cas, ADRIP montre une meilleure performance pour la prédiction du trafic lorsque les données du trafic expriment de grandes variations. La performance robuste d'ADRIP pour différents scénarios de test démontre **sa capacité à gérer des échelles de données multiples**.

Pour les travaux futurs, de nombreuses possibilités sont intéressantes à explorer pour améliorer ADRIP. Tout d'abord, nous visons à intégrer le clustering dynamique pour regrouper les segments

de route présentant des caractéristiques similaires. Cela permettra d'améliorer la qualité de la prédiction, en particulier en cas de manque de données. Deuxièmement, les critères de comparaison entre les configurations pourront également prendre en compte la saisonnalité du trafic afin d'aider ADRIP à détecter les schémas de trafic répétés. Ensuite, une expérience de prédiction du trafic avec un pourcentage variable de voitures connectées partageant leurs données sera intéressante à étudier car elle permettra à ADRIP d'approcher les applications du monde réel et d'évaluer ses performances lorsque les données sont manquantes. En outre, un cas de test appliquant ADRIP pour les prédictions de trafic à différents niveaux de trafic sera intéressant pour analyser la corrélation entre les prédictions à plusieurs niveaux de données de trafic.

Chapter 2

Introduction

2.1 Background

Nowadays, the improvement of our life quality has motivated innovations across various aspects, especially in transportation and human mobility. To answer the increasing demands of consumers, significant advances have been developed within the transportation system. As a result, the volume of vehicles in daily traffic shows a rapid increase on our roadways. However, despite the simultaneous efforts to enhance transportation infrastructure in line with this growth of mobility, numerous adverse effects have persisted, with the most noticeable being the increase in traffic congestion during the morning and evening peak hours.

Traffic congestion has evolved into a significant challenge in modern cities due to the severe inconveniences, including substantial travel expenses, environmental issues, detrimental effects on human mental health, and especially, degradation of traffic safety. According to [55], congested traffic leads to increased transportation costs for both individual travelers and businesses due to prolonged travel times. The increased costs in long traffic congestion mainly arise from higher fuel consumption due to frequent idling, acceleration, and braking, as well as the extended periods of non-productive activity. Besides, traffic congestion substantially contributes to the increase of CO_2 emissions. This consequence is primarily due to the stop-and-go traffic conditions associated with the congestion and the increase in the acceleration and deceleration events experienced by vehicles [73]. The implications of this effect on human life are evident as CO_2 and other greenhouse gas emissions from traffic are the main contributors to global warming. Global warming, leading to severe climate changes, is getting more and more attention from governments, authorities, and societies worldwide. Nowadays, people have already experienced severe events such as intense floods, wildfires, heat waves, stronger hurricanes, storms, $etc.$ which are consequences of such climate changes. Additionally, traffic congestion also impacts the human mental health of drivers during and post-congestion driving. A study with the North York business sector and the York University [80] showed that driver's stress is more significant in high congestion conditions.

Stressed and frustrated behaviors can lead to road rage and reduce driver's health. In addition, the study in [111] demonstrated that drivers tend to exhibit more aggressive behaviors with degraded situation awareness and declined speed perception after driving in long congested traffic conditions. This finding underlines the long-term negative effect of traffic congestion on human cognitive awareness.

The most critical consequence of traffic congestion that is getting the attention of ITS researchers is the decrease in traffic safety and reliability due to the accidents, collisions, $etc.$ that are very likely to happen at the End of Queue (EoQ) where the jam is not totally formed and where the speed quickly decreases. Indeed, more collisions occur at near locations where the downstream traffic condition is congested and upstream traffic is free-flow [194], [189]. The risk of rear-end collision is maximum when the downstream is congested while the upstream reaches its maximum capacity [114]. Indeed, when the upstream occupancy becomes higher than maximum capacity, if the approaching vehicles do not intend to gradually alter their velocity before entering the queue, the collision risk thus increases. Furthermore, the estimates of collision likelihood can be described following the joint normal distributions, which show the increase in accidents when both temporal and spatial proximity toward jam queue [115].

The danger at the end of the jam queue is obvious. Thus, anticipating the existence of traffic congestion leading to a change in the driving behaviors of approaching drivers is necessary to avoid and mitigate the severity of traffic collisions. Vehicle crash is less severe if the entry vehicle velocity is low. Meanwhile, the sudden deceleration can cause dangerous rear-end collisions [150], [164], [136]. Indeed, the lateness of recognition of ahead traffic queue and driving behavior change can lead to grow the collision risk [115]. Thus, a warning system is necessary to help drivers recognize and start decelerating far from the end of queue.

From the addressed inconveniences resulting from traffic congestion, it is essential to anticipate the existence of congestion in advance and predict the position of EoQ. This proactive information can offer effective solutions to improve the overall quality of mobility, mitigate the negative effects on the economy and environment, and enhance traffic safety.

2.2 Motivation

Queue Warning System (QWS) nowadays is an interesting topic, especially in connected traffic environments where the vehicles can communicate with other vehicles and traffic infrastructures through V2X connectivity (Vehicle-to-Everything).

QWS leverages the Connected Vehicle (CV) technologies to enable vehicles within a traffic jam queue to automatically broadcast their queue status to the upcoming vehicles and to traffic infrastructures. The exchanged messages can contain information such as lane location, traffic density, rapid required deceleration, $etc.$ This information is transmitted to upcoming vehicles aiming at preventing rear-end collisions at EoQ. Additionally, the appropriate warnings communicated by

QWS can help approaching vehicles to decrease their speeds safely, make lane change decisions, or select alternative routes if necessary. The advantages of QWS are significant in case of limited visibility (e.g. road bends, bad weather conditions, $etc.$) to extend the vehicle's horizon or in case of heavy vehicles to give them sufficient reaction time because of their longer stopping distances. Especially in the context of Autonomous Vehicles (AV), QWS helps drivers have a better driving experience with the automated ADAS (Advanced Driver Assistance Systems) deployed in AVs. Indeed, besides their obvious advantages in improving driving safety, these systems usually bring many inconvenient driving experiences due to the conflicts of take-over control between drivers and automated functions, especially in traffic congestion situations [29]. Therefore, the proactive information provided by QWS allows drivers to manually activate/deactivate the automated braking systems if necessary or allows AV manufacturers to improve automated systems by adapting their functions to traffic states. As the global view of traffic management, QWS helps mitigate traffic perturbations, makes better itineraries for route choices, and reduces the risk of formation of prolonged traffic congestion.

Numerous queue warning algorithms and communication protocols have been developed to optimize communications, extract pertinent and succinct information before sending, and provide long-distance warning capabilities by leveraging the V2X/V2E technologies. Despite the improvements in studied methods, they still have several limitations. Firstly, they are constrained by the availability of vehicle connections, making them unsuitable for use in *dead zone* situations where no network connection exists or bandwidth is limited for the massive amount of data transfers. Second, the transferred traffic data are based on real-time observation from leading vehicles. Given the highly dynamic nature of traffic, the conditions observed by leading vehicles may differ from those encountered by the following vehicles when they arrive at the following road segments. Third, the warning horizon is limited by the communication range of V2X connectivity, which may not always provide a sufficient reaction time for vehicles, particularly in cases involving heavy vehicles or rapidly propagating traffic queues.

Motivated by these challenges, Continental Digital Services France (CDSF), an automotive company and affiliated company of Continental AG, proposes the research topic for this doctoral thesis for on-board predicting dynamics of traffic jams towards modeling and anticipating the End of Queue based on the traffic prediction approach. This topic can contribute to the main activities of CDSF that aim to develop digital services for connected vehicles, from in-vehicle data to applications via the cloud.

The traffic prediction approach can offer several advantages across different aspects. In the context of congestion prediction, the computed traffic prediction holds the potential to not only predict the location of traffic congestion but also anticipate its queue length and propagation characteristics. Moreover, gaining insights into the evolution of future traffic conditions empowers the ability to proactively prevent and mitigate the formation of traffic congestion before it occurs. Furthermore, these predictions can provide invaluable support to traffic control centers in the

efforts to systematically manage the road network and allocate resources effectively, including lane management (opening and closing lanes as needed), dynamic pricing for parking facilities, and the implementation of adaptive traffic signal systems, all facilitated with a high degree of automation.

2.3 Research contribution

This thesis introduces ADRIP (Adaptive multi-agent system for DRIving behavior Prediction), which is based on an Adaptive Multi-Agent System (AMAS). ADRIP is capable of continuously learning from traffic data streams and predicting future traffic dynamics in real time. An adaptive MAS is composed of multiple autonomous and interacting entities called *agents*, which can partially perceive information from the dynamic environment, make decisions based on their perceptions and knowledge, and cooperate to achieve their local goals. Thanks to the distribution of global tasks among agents and the decentralization of the system's functioning, MAS has been proven to be well-suited for tackling complex problems [33]. Given that the traffic prediction problem involves numerous connected and interacting vehicles and traffic infrastructure elements evolving within a dynamic driving environment, ADRIP emerges as a promising solution for addressing traffic prediction challenges.

The novelty of ADRIP lies in developing two processes:

- **L-ADRIP - continuous learning process** of ADRIP: utilizing a dynamic clustering method, aiming at learning historical traffic dynamics from communicated data streams. L-ADRIP is capable of continuously updating the learned model and database, adapting to new data, and taking into account the instant changes in the driving environment in the calculation of traffic prediction. In ADRIP, the learning process is performed at the agent's level, allowing for localized learning and adaptation.
- **P-ADRIP - cooperative prediction process** of ADRIP: designed based on the cooperative interaction in MAS, aiming at estimating the future traffic dynamics in real-time by leveraging the learned database from the learning process. P-ADRIP consists of a local prediction algorithm of each agent, a set of cooperative interactions among related agents, and the self-adaptation functions as self-organization, self-correction allowing ADRIP to overcome the potential conflicts that may arise during the exchange of information and enhance its prediction capabilities.

ADRIP proposes a novel solution for traffic prediction issues whose benefits are shown through the following characteristics:

- **Dynamic**: In today's real-world road network context, traffic exhibits a high degree of dynamism. Traffic entities and the driving environment undergo rapid and continuous changes. Therefore, the traffic prediction method taking place in this condition must evolve

alongside these dynamic conditions to ensure its relevance and long-term sustainability. With the continuous learning process, ADRIP allows the integration of the new changing states of the surrounding environment into its predictive model.

- **Openness**: The openness of a system refers to its ability to allow participating entities to enter or leave without disrupting its operation. In ADRIP, the learning and predictive tasks are distributed at the agent's level. Thus, the entry or exit of entities does not prevent the system from continuing to work.
- **Interpretability**: Traffic prediction is a topic related directly to human safety. Thus, it is important to be able to explicitly explain the causality between inputs and outputs of predictive models. Some of existing methods lack this characteristic due to their black-box nature. The dynamic clustering method in ADRIP can provide a clear understanding of input-output relations.
- **Ability for multi-traffic level application**: Traffic data are also interesting to study at different scales as this analysis enables us to analyze the traffic influences among different urban areas. A traffic prediction model that performs well across different data scales can be beneficial for multi-level traffic management applications since it allows a better understanding of traffic patterns at various scales.
- **Privacy friendly**: It refers to the ability of the system to avoid the diffusion of personal data. In the traffic domain, driver's locations, particularly GPS data, are among the most sensitive information to share. Predictive models are required to respect this privacy by avoiding the exchange of precise location data among processing entities. In ADRIP, a privacy-conscious approach is adopted. Data are locally learned to generate representative information. Only this synthesized data is shared within the system. Besides, ADRIP does not require any personal data of drivers or vehicles.

These five characteristics are the primary keys that enable ADRIP to adapt to the current traffic context. The dynamism of ADRIP enables the data stream processing within the context of extensive and continuous traffic data transfers. Furthermore, its openness aims to address the dynamic changes occurring in the driving environment, such as road closures, maintenance activities, speed limit adjustments, and the installation or removal of traffic lights, $etc.$ This openness characteristic ensures uninterrupted functionality without the need for re-initialization. Moreover, the last characteristics collectively contribute to enhancing the reliability, flexibility, and safety of predictive models within ADRIP.

2.4 Manuscript Organization

This manuscript is organized as follows:

- **Chapter 2**: This chapter introduces a comprehensive overview of Intelligent Transportation Systems (ITS), highlighting some well-known services within the ITS framework. Then, we discuss the significant role of traffic prediction within ITS services, examining the challenges and opportunities posed by modern transportation prediction systems. Furthermore, we explore the motivation for using Big Data Analysis in ITS, showing existing applications of this paradigm to enhance its potential for traffic prediction problems. Finally, the chapter engages in a thorough discussion of the prospects and challenges associated with the application of Big Data Analysis for traffic prediction systems.

- **Chapter 3**: This chapter describes the well-known methods employed for the traffic prediction problem. We discuss their advantages and limitations according to five components: temporal modeling, spatial modeling, stream analysis, model interpretability, and model versatility. These components are defined as the goals of our proposal.

- **Chapter 4**: This chapter provides an introduction to the main characteristics of the data stream and dynamic clustering as a well-known approach for stream analysis. The existing methods are described and analyzed according to their ability to address the properties of data streams. Then, their ability to address the stream analysis is presented through some existing applications in various domains motivating their usage for our study.

- **Chapter 5**: This chapter brings an introduction to multi-agent systems and adaptive multi-agent systems as a distributed and decentralized solution for complex problems. The discussion lies in the properties of self-organization and cooperation of agents to achieve the objectives of the system from the emergence of local functionalities. Then, some applications of ADRIP are presented to motivate their usage for our study.

- **Chapter 6**: This chapter presents the contribution of this thesis. ADRIP - a multi-agent system for continuous learning and cooperative prediction is introduced. This system aims to provide an effective solution for multi-level traffic prediction by fulfilling five components defined in the discussion of Chapter 4. In this chapter, we present a generic description of ADRIP by clarifying its principles, agent behaviors, and main functions.

- **Chapter 7**: This chapter demonstrates the performance of ADRIP through two applications. We describe the generated and real data sets used for the experiment and detail ADRIP for each application. Then, we compare the prediction quality of ADRIP with well-known baselines of the traffic prediction problem and discuss the obtained results.

- **Chapter 8**: This chapter presents the conclusion and the perspectives of our study, both on scientific and applicative points of view.

Chapter 3

Intelligent Transportation Systems and Big Data Analytics

This chapter presents:

- An overview of Intelligent Transportation Systems (ITS) as well as some well-known services in ITS.
- The importance of traffic prediction in ITS's services, the challenges and opportunities of traffic prediction systems in modern transportation.
- The motivation towards Big Data in ITS and its existing applications.
- A discussion about the opportunities and challenges when applying Big Data Analytics for traffic prediction systems.

3.1 Intelligent Transportation Systems

3.1.1 Motivations

Transportation is fundamental for human society, allowing the movement of people and goods through the world. Along with the evolution of human society, transportation, and infrastructures have continuously changed and significantly impacted our society and environment.

In modern transportation, the significant increase in volume and density of vehicles in road traffic has raised many issues, such as traffic congestion and accidents that negatively impact the economy, environment, and life quality. This situation required the evolution of transportation systems when the current rules, regulations, infrastructures, and traffic control methods became

obsolete and insufficient to face these new challenges. The expansion and construction of new roads and infrastructures are considered as quick solutions to attenuate the pressure of crowded traffic. However, they are insufficient and unsustainable for addressing the underlying problems due to their expensive requirements of economic investments and lands.

Nowadays, advances in computer science and communication network technologies have paved for new methods and applications of transportation systems, generally called **Intelligent Transportation Systems (ITS)**. ITS brings innovative solutions based on the analysis of data gathered by sensors and equipment of vehicles and road infrastructures, aiming to improve the quality of current transportation systems, making them safer, more efficient, sustainable, and environmentally friendly.

3.1.2 Intelligent Services in ITS

Numerous intelligent services of ITS have been developed and deployed in the real-world marketplace for different transportation modes and across various aspects. Among these innovations, the improvement of traffic safety in high-density traffic is the main focus of ITS's research. In the next, we present a quick overview of three systems of ITS that provide direct or indirect solutions for traffic congestion detection, avoidance, and anticipation, leading to mitigate vehicle collisions.

- **Advanced Traffic Management System (ATMS)**: ATMS provides the management perspective from the macroscopic view that primarily integrates technologies to manage and improve the traffic flow. According to the National ITS Architecture [2], the goals of ATMS include increasing transportation system efficiency, enhancing mobility, improving safety, reducing fuel consumption and environmental cost, increasing economic productivity, and creating an environment for an ITS market. ATMS contains several services such as real-time traffic monitoring, automated warning systems, travel demand management, traffic signal monitoring and control, route guidance, $etc.$

- **Advanced Driver-Assistance System (ADAS)**: ADAS consists of technologies developed to assist driver safety and better driving, enabling autonomous driving at different levels. ADAS uses the automated technologies installed inside the cars to detect nearby obstacles or driving errors and respond adaptively. Features provided by ADAS can be categorized into three groups: *safety, adaptive, and assistance features*. Safety features such as collision avoidance systems aim to detect danger risks, alert the driver to problems such as unsafe lane changes or hide obstacles, implement safeguards, and take control of the vehicle if necessary. Adaptive features provide suggestions to help vehicles safely adapt to their driving environment, such as automated lighting, adaptive control cruise, intelligent speed adaptation, $etc.$ Assistance features include technologies helping drivers to have better visibility of the surrounding environment, such as automotive night vision, automotive head-up display, traffic sign recognition, $etc.$

- **Advanced Traveler Information System (ATIS)**: ATIS refers to systems that assist travelers in trip planning from their current position to destination with optimal itinerary and comfortable driving experience. ATIS can gather both micro and macro traffic information from intelligent vehicles ($e.g.$ ADAS) and traffic management ($e.g.$ ATMS). Some relevant information may include locations of incidents, weather and road conditions, optimal routes, recommended speeds, lane restrictions, traffic conditions, $etc.$ ATIS transfers traffic information to travelers by text messages using in-vehicle communication or variable-message signs, $etc.$

3.1.3 Traffic Prediction in Intelligent Transportation Systems

In ITS, traffic prediction constitutes a fundamental component of the effectiveness of numerous services. In fact, the success of many intelligent transportation systems depend on the availability of timely and accurate estimates of future traffic conditions [43].

In ATMS, the estimations of future traffic flow can assist management authorities in implementing adaptive strategies for different areas and periods. One of the relevant applications of traffic prediction in ATMS is **dynamic traffic signal control and modeling** [183]. In dense areas, traffic signal control can be automatically adjusted based on the anticipated traffic conditions. This adjustment involves dynamically adapting signal durations and phases to accommodate the flow of vehicles, effectively mitigating traffic disruptions. Moreover, transportation authorities can allocate resources and infrastructures effectively based on traffic predictions [40]. For instance, they can adjust the deployment of control or maintenance crews to expand or close roads to address specific traffic challenges. Other applications of traffic prediction in ATMS are lane management, route guidance, $etc.$

Traffic prediction is necessary to **convey reliable information about the future states of traffic** in ATIS [148]. The drivers can be provided accurate and real-time traffic predictions, as well as suggestions for alternative routes, allowing them to make informed decisions, avoid traffic congestion, and have shorter travel times, especially during peak hours. Furthermore, traffic prediction in ATIS can help **public transportation agencies adjust schedules, itineraries, and resources in real-time** to respond to traffic demands. These assistance services of ATIS lead to smoother traffic flow, efficient and reliable public transportation services, and better driving experience.

Additionally, traffic prediction in ADAS aims to **provide the anticipations at the microscopic level**, such as providing individual vehicles available and updated predictions following the vehicle's itinerary [132]. It aids in preventing unexpected traffic conditions, allowing ADAS to warn drivers about the appearance of potentially dangerous situations in advance. **Collision avoidance system** is among the most advantageous features in ADAS from accurate and long-term traffic prediction, mainly when the driver's visibility is limited by road topology or weather conditions. To extend the observation horizon, collision avoidance systems use specific cameras such as LIDAR that can function under extreme conditions or gather information from other vehicles via V2X

communication. However, these methods often provide only a short reaction time due to the limitation of camera resolution and communication range. Consequently, drivers may not have sufficient time to respond to critical situations such as traffic jam queues, increasing the risk of sudden braking incidents. Long-term traffic prediction aims to address this challenge by providing drivers the anticipation with necessary reaction times to safely decelerate, avoid collisions, or reduce their impact.

From the above advantages, traffic prediction has become one of the most interesting research topics in ITS. The detailed literature review of this topic is presented in chapter 4.

3.1.4 Discussion

The advantages of ITS have encouraged researchers in both academia and industry to deploy new hardware technologies and develop novel software to improve transportation services. Thus, the development of ITS goes along two interdependent directions: **infrastructure installation** and **software development** by modeling transportation networks or analyzing traffic data, $etc.$

Nowadays, numerous intelligent transportation devices have been widely installed. With the innovation of digital technology, ITS technologies have developed significant advances, including electronic sensors, data transmission, and intelligent control technologies [146]. These advances, on the one hand, provide intelligent assistance, and on the other hand, observe the information, store and analyze it to improve the services. As a result, complex, diverse, and massive data sets have been generated in ITS. To enable analyzing these data, the need for *"Big Data Analytics"* concept in ITS was born.

In the next section, we focus on the overview of Big Data in ITS, including the introduction, motivations, and existing applications.

3.2 Big Data Analytics in ITS

In the digital age, information devices have become ubiquitous - a cell phone in every pocket, a laptop in every backpack, and information workstations in every office. During the early stage of this era, people were interested in the novelty of technologies, such as how they functioned, evolved, and contributed to human life. However, the notice about information itself is lacked [128]. It is understandable since the initial generation of information systems was not as widely accessible and practical as today (small size, versatile applications, intelligent features, $etc.$). Consequently, the volume of collected data was not noteworthy. Half-century computers entered our society, the pace of information growth accelerated significantly, leading to the growth of collected data in both quantitative and qualitative aspects. Around the 2000s, the term *"Big Data"* was first coined to describe this phenomenon. The era of *"letting the data speak"* has begun.

The inception of big data emerged from the exponential growth of massive and complex data sets from diverse digital sources. These large data sets necessitated the development of new processing

tools because they no longer fit into the computer's memory with conventional processing methods. Thus, along with the widespread of big data, a new generation of data processing techniques was explored, including data mining, management, processing, analysis, visualization, $etc.$ The big data technologies marked the beginning of significant transformations, unlocking novel solutions that were previously impossible to achieve at smaller ones. Moreover, these technologies empowered us to extract new insights and make predictions about future values. These transformations have changed the ways of human interactions with the world, altered our daily routines, facilitated social relationships, and more. Nevertheless, they also come with numerous unprecedented challenges, especially concerning the reliability and security of information that big data utilizes and provides.

Despite the challenges, the benefits of big data are bigger. Over the past few decades, big data has become a hot research topic in academia and industry across all fields of human life, including healthcare, education, business, environment, energy, transportation, $etc.$ From the success of big data in many domains, its applications for ITS hold a big promising opportunity.

3.2.1 Motivation of Big Data Analytics in ITS

The adoption of Big Data Analytics in ITS was motivated by the increase of collected traffic data through innovative **data mining technologies** and the advancement of **data processing methods**. These novel data mining technologies facilitate the collection of diverse and large-scale data, with the ability to transmit it at high speeds and low latency, thereby enabling the extraction of a wide range of insights. To effectively analyze these data, the evolution of data processing methods has played an important role. Numerous innovative techniques have been developed to leverage the full potential of collected traffic data, extracting valuable and helpful information to drive transportation improvements. In the next, we provide an overview of data mining and processing technologies in ITS. This overview will show the details of traffic data associated with each mining technology and highlight the effective processing method adapting to different types of available data and research objectives.

3.2.1.1 Data Mining Technologies

Advances in ITS technologies have led to an increase in the amount, complexity, and diversity of collected data. Different categories of big data in ITS are presented in Table 3.1 inspired by the study in [210].

- **Data from Smart Cards**

  Automatic Fare Collection (AFC) systems have been widely deployed in urban rail systems and public transportation networks, enabling ITS to investigate the passengers' movement patterns [118], [22]. The primary purpose of AFC is revenue collection by requiring passengers to validate their cards when using transportation services. Electronic readers capture the essential trip details and passenger information such as anonymous personal information,

Table 3.1: Categories of big data in ITS

Source	Tools	Data
Smart cards	Smart cards	OD flows, travel time, trip details
IVs perception sensors	GPS, radars, LIDAR, vision sensors, CAN	Vehicle position, vehicle speed, vehicle density, coordinate, speed, acceleration, $etc.$
Vehicular communications	V2V, V2I, V2X, VANET, digital maps	Surrounding vehicle and environment information
Fixed sensors	Induction loops, road tubes, microwave radar, LIDAR/Infrared acoustic, video camera, toll plazas	Vehicle position, vehicle speed, vehicle density, vehicle classification
Other sources	Social media, mobile phone data, smart grid, smart meters, cellular services, dedicated tests	Travel time, OD flows, electric and energy consumption, location, channel data

boarding time, location, origin-destination (OD) information, $etc.$ This functionality of AFC opens the potential use of smart cards as a valuable transport data source for analyzing travel behaviors, helping service providers improve customer's experience and facilitate customer relationship management.

In recent years, substantial works have explored the potential powers of smart card data for various ITS services. These data have been employed for short-term prediction of passenger volume, leading to improving passenger flow management and shuttle scheduling, especially in large networks with multi-destination transit hubs [192]. Furthermore, [52], [204] have leveraged smart card data to study travel behaviors and activity patterns. These longitudinal data sets from smart cards offer insights beyond the limitations of survey data. Additionally, the values of data from smart cards have been proven in studying passenger behaviors during specific situations such as the COVID-19 pandemic to understand the change in travel patterns better and demonstrate the influence of socioeconomic status on mobility behaviors [11].

- **Data from Environmental Perception Sensors of Intelligent Vehicles**

  Numerous internal and external sensors have been installed, allowing vehicles to obtain the in-car data and perceive the driving environmental information. According to [209], intelligent vehicles acquire information through the following installed devices:

  – Global Positioning System (GPS): is the most popular tool for location tracking. Integrating with geographic information system (GIS) and map visualization technologies, GPS provides time and location information of objects. However, GPS can introduce measurement errors, necessitating the use of techniques such as map matching to align GPS

data with road network positions, particularly in applications such as vehicle itinerary identification.

- Radars: is an object detection system based on the signal of radio waves.
- LIDAR: is extensively applied for obstacle detection systems and can provide the distance to objects using laser light.
- Vision sensors: can be camera, low-light level night vision, infrared night vision, and stereo vision. They are well suited for intelligent vehicles since they provide diverse and rich data sets at different conditions. The raw information is the light intensity.
- Controller Area Network (CAN): this tool can link the sensors, engines, and controllers to provide detailed vehicle data such as acceleration, inertial measures, *etc.*

- **Data from Vehicular Communications**

  Vehicular-infrastructure communications such as Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), and Vehicle-to-Everything (V2X) technologies give communication protocols to vehicles for different communication contexts. This connectivity allows intelligent vehicles to communicate with others and with transport infrastructures to exchange information. Several solutions have benefited from these technologies [17] such as anti-collision systems, intelligent navigation systems, driver notification systems, assisted parking systems, and autonomous driving.

  Vehicle Ad Hoc Network (VANET) consists of groups of vehicles and infrastructures connected by wireless networks. VANET was developed to extend the coverage area and enhance the capacity of vehicular communications. Thanks to the benefit of cooperation between numerous connected entities, VANETs can generate large amounts of data helping to warn and detect traffic congestion [90], accidents [208], traffic prediction [4], *etc.* However, the security of VANET architecture and protocol is the most important issue that needs a serious study before deploying it in the real-world [77].

  Additionally, positioning and digital maps are also data sources for intelligent vehicles. Indeed, data collected from intelligent vehicles are sent, stocked, and processed at the data logical centers to assist the upcoming vehicles. Indeed, upcoming vehicles download from the cloud a local map enriched with important information (danger, accidents, road hazards, weather information, *etc*.) that could perturb the driver's journey. This information provides the global traffic view at a given area, help vehicles avoid dangerous situations, and extend their horizon to prepare for upcoming traffic conditions.

- **Data from Fixed Sensors**

  The fixed position sensor technologies are summarized into the following types: *inductive loop, magnetic sensors, video image processors, microwave radar sensors, infrared sensors, laser radar*

sensors, and audio sensors [101]. *Inductive loops* are deployed at a fixed position at the roadside or under the road surface ($i.e.$ intrusive sensors) to detect vehicles based on the changes of electromagnetic field caused by the vehicle's ferrous body. *Magnetic sensors* detect vehicles by measuring the change in the Earth's magnetic field caused by the presence of a vehicle near the sensor. *Video image processors* analyze the video image of the roadway from surveillance cameras and provide traffic flow data across several lanes. *Microwave radar sensors* emit microwaves and then detect the objects' reflections. *Infrared sensors* have two types: active and passive. Active infrared sensors emit low-level infrared energy into a specific zone to detect vehicles by observing energy interruptions when vehicles appear. Passive infrared sensors detect energy emitted from vehicles and other objects nearby. *Laser radar sensors* are active sensors that transmit scanning infrared beams in the near-infrared spectrum over one or more lanes. *Audio sensors* are passive sensors and use different audio signal processing techniques to calculate traffic density or volume.

Fixed sensors measure data such as traffic count ($i.e.$ volume), speed, or density. The collected data at a fixed sensor can be described as an ordered sequence of measurement ($i.e.$ time series of measurement) or temporal data. The most important advantage of fixed sensors compared to mobile sensors relates to the high quality of collected data. In addition, an installed sensor can capture all crossing vehicles. Thus, the aggregate statistics can be estimated with high precision. However, the expensive costs of the deployment and the maintenance of the large number of sensors are its main inconveniences, leading to the limitations for large-scale applications. Moreover, the fixed sensor cannot observe the data for all the paths of vehicles. Thus, it is hard to observe the relations between the sequence of road segments (upstream and downstream segments).

- **Data from Other Sources**

  In addition to the data gathered from sensors and techniques designed specifically for ITS applications, Big Data in ITS can be obtained from other sources. The rapid development of mobile technologies has led to the collection of massive amounts of data, including mobility data. While these data are not intentionally generated for ITS purposes, they nonetheless provide valuable insights for analyzing human mobility, travel behavior, transportation planning, and passenger preference.

  Moreover, social media network is also a rich passive data source for big data in ITS. With the widespread of Facebook, LinkedIn, Twitter, $etc.$, people feel more comfortable to create and share information and ideas on these networks. Thus, they become the practical way for communications, announcements, and interactions between providers and customers. Through data collected via social media, traffic managers and providers can study mobility patterns from different social groups [10], passengers' attitudes to transit disruption [140], and more.

Then, other data, such as energy and electric consumption, can be collected via smart grids to enable optimizing performance vehicles. In special cases, some dedicated tests can be conducted to generate data on-demand.

3.2.1.2 Data Processing Methods

Numerous Artificial Intelligence (AI) models and algorithms have been adopted thanks to the diversity of data sets in ITS. Machine learning-based models are well-known for data processing in ITS [27]. In this section, we analyze the existing models employed for Big Data Analytics in ITS, structured according to four main learning types in AI: *unsupervised learning*, *supervised learning*, *reinforcement learning* [198] and *semi-supervised learning* as the hybrid version of unsupervised and supervised learning. This categorization addresses the different methodologies that AI models acquire knowledge data, construct their underlying model, and make estimations or predictions. The usage and development of AI models depend on the characteristics and completeness of data sets.

- **Unsupervised Learning**

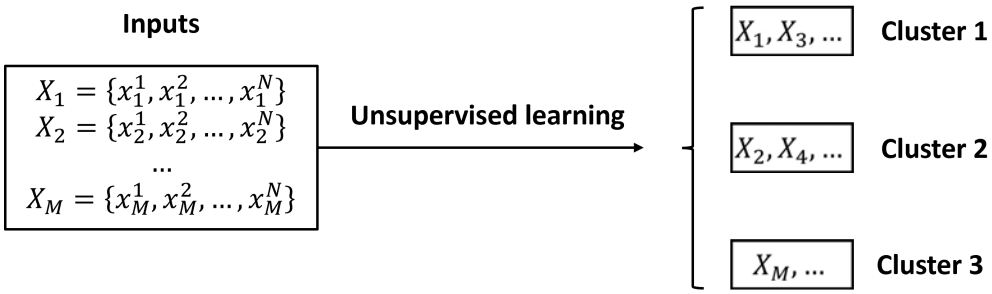


Figure 3.1: Unsupervised learning scenario

Unsupervised Learning (UL) [129] is well-known for dealing with data sets without labeled variables and involving high dimensionality. UL aims to understand and discover the hidden structure of input data by reducing its dimensionality and representing data points under distinguished clusters (Figure 3.1). This method is mainly applied for pattern detection, recommendation systems, segmentation, prediction, *etc.* Some well-known unsupervised learning methods include K-Means, KNN (K-Nearest Neighbors), PCA (Principal Component Analysis).

- **Supervised Learning**

Supervised Learning (SL) [129] models the relationships and dependencies between input variables and label variables in labeled data sets (Figure 3.2) There are two phases in supervised learning: the training phase and the testing phase. The training data sets provide the inputs and the associated labels of each data point. A supervised learning model learns from

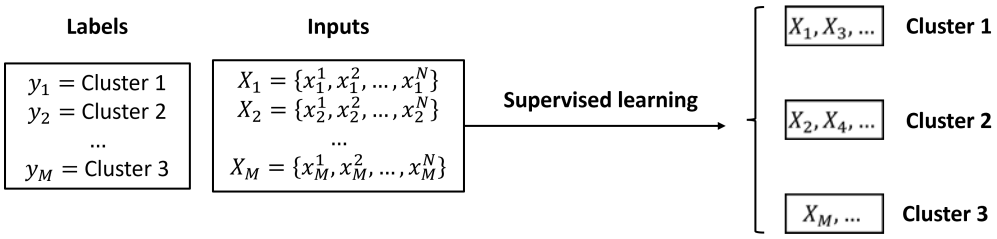


Figure 3.2: Supervised learning scenario

training data to build the mapping functions between inputs and labels by estimating the parameters in its model. Then, it applies the built model to the inputs of testing data sets to estimate the corresponding labels. In supervised learning, the testing data sets also provide the label so that we can evaluate the model's performance by comparing the real and estimated labels. Supervised learning is mainly used in two major problems: *classification* and *regression*. Classification models such as support vector machine (SVM), tree decision, random forest, *etc.* are used to group data points into different categories. Meanwhile, regression models are applied for problems whose output is real or continuous values. The most well-known regression model is linear regression, which tries to fit data with the linear model through the points of training data.

- **Semi-Supervised Learning**

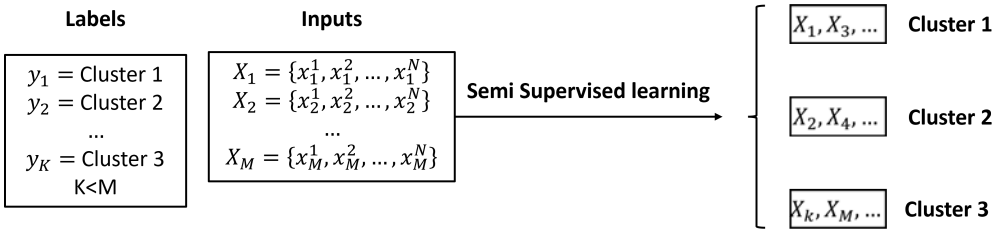


Figure 3.3: Semi-supervised learning scenario

Semi-Supervised Learning (SSL) [39] is characterized by using the data set combining a small amount of labeled data (following the supervised learning paradigm) and a large amount of unlabeled data (following the unsupervised learning paradigm) (Figure 3.3). Semi-supervised learning represents a valuable approach for tackling the remaining challenges of both unsupervised and supervised learning. Compared to unsupervised learning, supervised learning provides labeled outputs that give more informative insights for real-world applications. However, obtaining labeled data is often a challenging, expensive, and time-consuming task demanding significant human resources. One application that highlights the advantages of SSL is noise detection. The noises can manifest in various forms, but it is not necessary to categorize and label them. SSL is trained by a small data set with only correct data, learning the underlying nature of these data by, for example, constructing a probabilistic distribution

function. Then, during the testing phase, data points that deviate from this learned function can be considered as noise.

- **Reinforcement Learning**

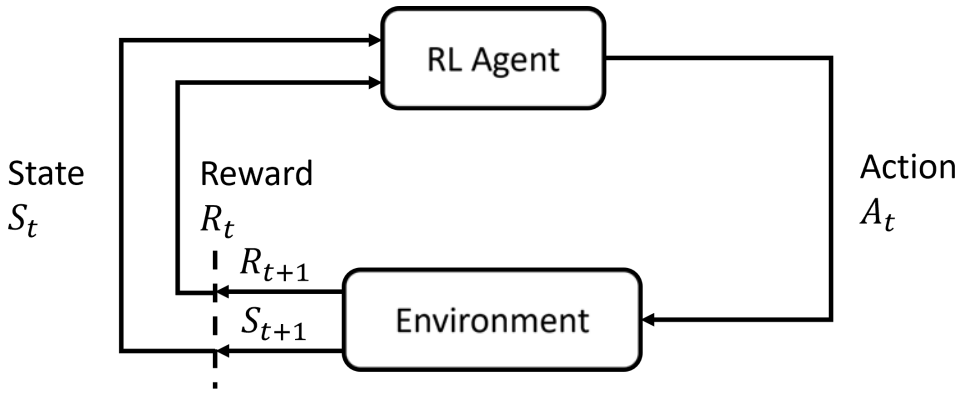


Figure 3.4: Reinforcement learning scenario

Unlike previous approaches that only learn from training data, Reinforcement Learning (RL) [129] reaches the optimal solution by taking into account the interaction between models and the environment (Figure 3.4). In RL, models estimate the optimal policy from experimental data and explore other opportunities in the interacting environment to maximize the long-term accumulative rewards. The balance between exploitation and exploration phases allows RL to adapt to the dynamic environment and overcome the over-fitting issue in AI. In an RL scenario, multiple agents can coexist and interact within a shared environment, known as Multi-Agent Reinforcement Learning (MARL). Agents in MARL operate independently to optimize their individual rewards and impact on others, either positively if they have the same interest or negatively if their interest is opposed. The goal of MARL is to develop collaborative decision-making algorithms that optimize the collective rewards of a group of interacting agents. Therefore, this field is closely related to the concept of game theory [135] and multi-agent systems [186]. Thanks to their underlying principle, single RL and MARL are highly relevant to control and optimization problems, AI gaming, skill acquisition, robot navigation, and real-time decisions.

3.2.2 Applications of Big Data Analytics in ITS

Big Data Analytics in ITS has quickly grown and become an attractive research topic in academia and industry. The expansion of big data research is demonstrated through two measures: **publication numbers** and **average citation per year**. The study in [94] conducted a bibliometric analysis of existing works for the period of 1997–September 2019 to provide a deep insight into applications of Big Data algorithms in ITS. Both indicators have shown a smooth increasing trend with a remarkable growth rate throughout the year. Moreover, Big Data Analytics not only stays in

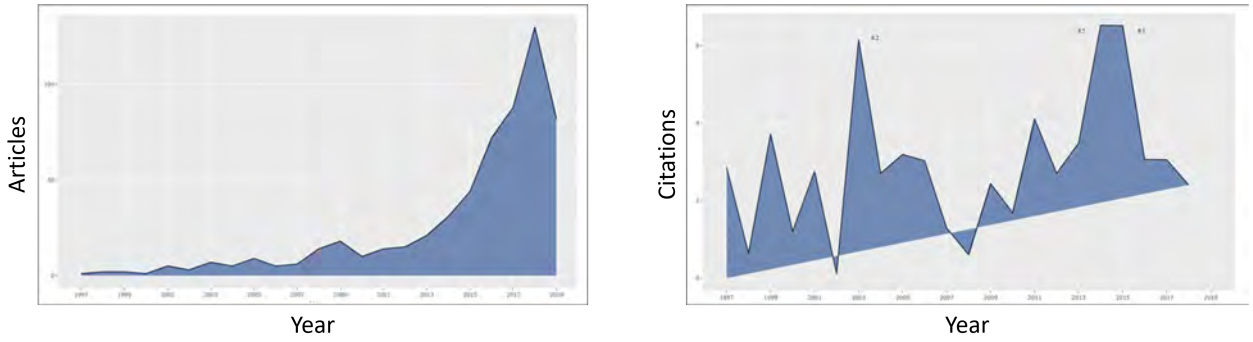


Figure 3.5: Number of publications and citations per year

academic research. Numerous services have been deployed in ITS thanks to the outcomes of these analyses for Advanced Driver-Assistance System (ADAS), Advanced Traveler Information System (ATIS), *etc.*

3.2.2.1 Objective of Big Data Analytics in ITS

The advancements in data mining and processing technologies have motivated numerous applications of Big Data Analytics in ITS. The objective of Big Data Analytics in this domain is to exploit the massive and complex datasets gathered within the diverse sources of transportation system to extract meaningful insights from these data. By applying suitable processing methods such as machine learning techniques, Big Data Analytics aims to uncover patterns, trends, and correlations within traffic data. Depending on the purpose of each application, researchers can leverage different data types and employ adequate analytical methods to reach their goals. In the following, we present **main applications of Big Data Analytics in ITS** that have been implemented successfully, illustrating the expansion of this field to highlights the growth of Big Data Analytics in ITS.

3.2.2.2 Existing Applications

Wide applications of Big Data Analytics in ITS have been studied.

- **Traffic Accident Analysis and Forecast**

  Road traffic accidents are one of the most significant causes of injuries and death worldwide. According to the global status report on road safety of WHO [137], over 1.2 million people die each year on the world's roads (18/100000 in global proportion), and between 20 and 50 million suffer non-fatal injuries. Therefore, analyzing traffic accidents to provide a method to prevent, warn, and mitigate them has become an essential field of research in ITS.

  Using Big Data Analytics, researchers conduct an in-depth analysis of historical traffic accidents and construct prediction models to prevent their existence in advance. The advanced

technologies in data mining provide data sets incorporated from diverse sources of information such as government data sets and open data, onboard equipment, measurement technologies, and social media. Data sets for traffic accident analysis mainly contain demographic information, level of detail of road conditions and environmental features, vehicle's technical information and their geographical positions, degree of accident severity, $etc.$ Moreover, additional information about drivers, passengers, and pedestrians, their habits, and daily activity, as well as the information about events and incidents on neighboring roads, are helpful for an accurate traffic accident prediction.

By using data analysis methods, researchers aim to determine the variables describing the characteristics of road accidents. These variables allow us to discover hidden patterns, identify repeated behaviors, determine the impacts of driver's behaviors and environmental features on the existence of road accidents, and construct inferences. Study in [171] employed a decision tree classifier, multi-layer perceptron, and Naive Bayes to determine the most important variables impacting the severity of a traffic accident. In [166], the prior violation and crash records of road users and roadway were demonstrated as important risk indicators of crash severity. Moreover, a Bayesian network approach was developed to explore their impacts on crash severity and their interactions. Another study in [144] presented an experiment using machine learning methods to explain driver-injury severity in run-off-roadway and rollover types of accidents. The obtained results highlighted the importance of using safety belts, psycho-physical conditions, and injury localization on the severity of accidents.

From the understanding of risk indicators impacting accident severity, several studies have focused on developing prediction methods to assess the likelihood of accidents, prevent them, and suggest solutions for mitigating their severity. Authors in [202] conducted a comparative study to evaluate the prediction performance of crash and injury severity methods. Four well-known machine learning methods, K-Nearest Neighbor, Decision Tree, Random Forest, and Support Vector Machine, were compared with the highest prediction accuracy obtained by Random Forest. Some recent works presented novel prediction methods for traffic accident severity. In [191], a hybrid model integrating random forest and Bayesian optimization was developed. The results highlighted the high prediction accuracy and model interpretability of the proposed model, leading to providing insights to mitigate the severity of traffic accident consequences and contribute to the sustainable development of transportation. Another work in [207] introduced an accident duration prediction model based on heterogeneous ensemble learning. This study gave a comprehensive interpretation of the importance degree of factors and features influencing traffic accidents, such as the time, location, weather, and historical statistics of the accident on accident duration.

- **Traffic Prediction**

The diversity of big data in ITS enables various predictive information of traffic dynamics

such as the mean of vehicle's speed, traffic flow, traffic density, travel time, *etc.* at different ranges of prediction horizon. Figure 3.6 shows an example of a typical traffic flow prediction framework in ITS presented in [210]. The original ITS data is first preprocessed to get the adequate data set by eliminating noises and re-formalizing the data format. Then, using the adaptive data analysis method, the traffic model is established by learning from data. The traffic model gives decision support to the traffic management department and gets feedback from the environment to update the model. The deep review of traffic dynamic predictions will be in Chapter 4.

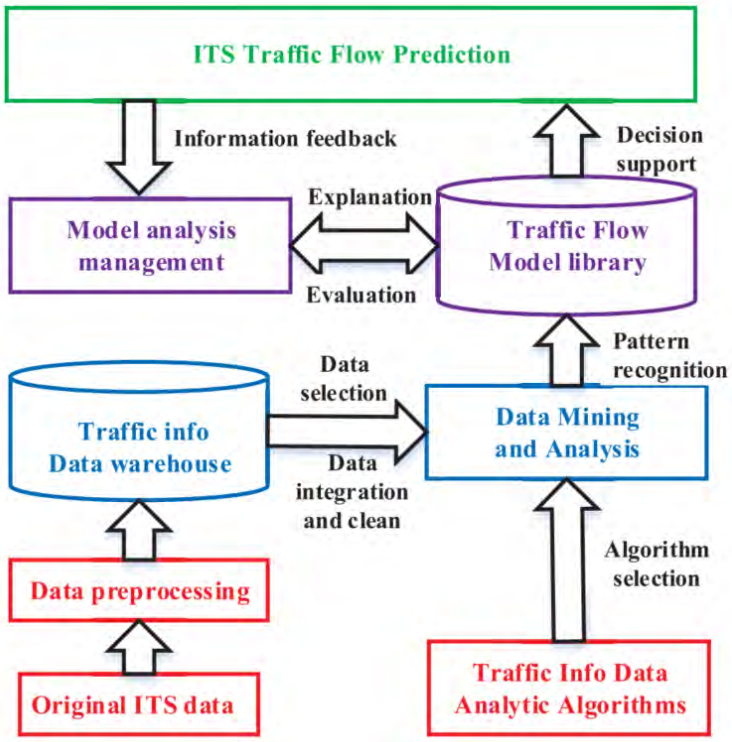


Figure 3.6: An example of typical traffic flow prediction framework in ITS [210]

- **Public Transportation**

  Public transportation systems bring substantial solutions to reduce traffic congestion and improve the efficiency of urban traffic. The application of Big Data Analytics in public transportation facilitates a comprehensive understanding of individual behaviors and the functioning of a large public transit system, thereby informing decisions to enhance service quality. The sources of big data used in public transportation applications are diverse, mainly GPS points and traces, Automated Passenger Counts (APC), Automated Fare Control (AFC), Automated Vehicle Location (AVL), sensor data, mobile phone data, web data, and social media data. The utilization of big data are highlighted in various aspects of public transportation.

- Service/performance improvement: By using Big Data Analytics, agencies can evaluate their services, discern customer's needs and identify potential improvements across different aspects. This approach proposes several **web tools/applications** such as: *Strategway* [70] to build an optimal route network based on residents' preferences while minimizing total cost of ownership, *BusViz* [16] to help bus service operators and transit regulators in monitoring and visualizing the performance of bus fleet with large streams of data and *ESTRI - an optimized and efficient spatio-temporal trajectory data retrieval method based on the Cloudera Impala query engine* to improve the efficiency of massive data sharing. Moreover, for **public transit performance analysis**, it can provide comprehensive passenger OD data across the city to identify potential passenger service improvements such as rerouting, stop changes, and route optimization [76], generate bus routes with increasing service efficiency and decreasing costs [168], and optimize public transit adapting to route changes [24]. Big Data Analytics also addresses **delay identification** [184], **schedule optimization** [46] and real-time transit timetables [84]. In addition, the service for **transit passengers with disabilities** is important in public transportation and can be improved by analyzing traffic data [100].
- Transit user's behavior analysis: Understanding passenger behaviors is important for designing and scheduling a public transit system [99]. Transit user behaviors analysis can address several interesting aspects, such as analyzing relationships between transit user's behaviors and weather conditions [133], [185], evaluating transit passengers' travel patterns, identifying the groups of passengers with similar behaviors [138], [57], and determining changes in passenger behavior and abnormal situations [86].
- Travel demand: Studying travel demands in public transportation allows providers to adapt their services to the number of demands. Big Data Analytics in this application focuses on the definition of OD matrices to evaluate passenger flow in normal situations [116] or in special events/emergencies [113], estimate the mode split between different transport modes [147], analyze the impact of a new transit opening [61], *etc.*

- **Personal Itinerary Planning**

  The concept of MaaS (Mobility as a Service) has recently gained widespread recognition. This notion refers to a type of service that offers travelers personalized solutions for multi-modal trip planning, such as travel recommendation systems for route choice and itineraries for public transportation with optimal time and cost. MaaS has a great vision in the future ITS, especially in the era of Big data. Massive sharing data from cellphones, personal computers, *etc.* allows ITS to understand the needs of travelers and learn their preferences, thereby providing adapted suggestions.

  Several technologies have been exploited for personal travel planning. Wireless sensor networks allow the enhance of data collection and transmission from services to users, leading to

providing better context awareness, real-time recommendations, opportunities to redesign the route during the trip, and adapting to changed circumstances [63], [158]. Additionally, to understand the historical user's preferences, several AI models are applied for data processing, such as Bayesian network [83], fuzzy logic [121], case-based reasoning [12] and genetic algorithm [117]. In personal itinerary planning, most applied AI models are based on context-aware decision-making principles aiming to propose suggestions adapting to gathered environment information. Besides, ontology and semantic web technologies [151] play an important role in sharing processed information under various formats to enhance the accessibility of services. Recently, agent-based technologies have been widely applied for personal itinerary planning due to their ability to decentralized and distributed modeling [78], [13], [93]. The agent-based method is capable of personalizing the decision-making algorithm at the user's level and dynamically adapting to the user's characteristics.

- **Rail Transportation Management and Control**

  Rail transportation is one of the primary beneficiaries of Big Data Analytics in ITS. Indeed, the collection of trip data in rail transportation guarantees the completeness of data sets with large amounts and high frequency since this domain does not encounter the personal privacy issue as in individual mobility. Moreover, due to the strict control, data on rail transportation is precise and detailed. For example, in public transportation, it is difficult to determine the destination of travelers. Meanwhile, in rail transportation, detailed trip information is contained in the ticket. Big data in rail transportation includes real-time train speed and position, train departure and arrival time from and to certain stations, and passenger OD information. Big Data Analytics can help to improve the system operation efficiency and operate better control and management for rail transportation. Therefore, this technique is attractive for both industry and academia.

  In industry, Bay Area Rapid Transit (BART) is a well-known rail transportation system that maintains supervision over all necessary phases of train operations, including train operations, passenger services, power delivery, and wayside facilities. Big Data Analytics is used in this system to monitor and analyze train arrivals and departures, monitor the flow of passengers based on ticket information, and reach the accurate projection of service schedules [20].

  In academia, Big Data Analytics is explored in substantial works to improve the operation of rail transportation systems. First, by analyzing the passenger OD information, work in [92] proposed a method to evaluate the efficiency of train timetables. Then, an intelligent train operation method is introduced in [195] that provides better solutions for energy consumption, riding comfort, punctuality, and parking accuracy than conventional methods. Moreover, several works investigated the accuracy of train stops. Work in [82] presented three train stop control algorithms based on initial braking position and braking force. An online learning control algorithm for automatic train stop control was proposed in [42].

- **Asset Maintenance**

  In public transportation systems, especially in railways, the two main goals of an efficient system are reducing operational costs and increasing performance in financial assets and safety. To achieve these objectives, the Computerized Maintenance Management System (CMMS) has been employed to follow the asset's status and assist service managers in making informed decisions (repairing cost estimation, preventive maintenance, $etc.$. To properly control the maintenance of a facility, CMMS needs information from Big Data Analytics.

  Data used in maintenance systems can be grouped into three types [173]. First, infrastructure and vehicle state data such as temperature, humidity, surface condition, $etc.$ can be processed with data-driven methods to obtain the condition indicators. Second, symbolic description and experience-based information are collected under text format and processed to extract the keywords. Third, the physics of failure describes the pavement degradation, track geometry, ballast aiding, $etc.$ By analyzing these data, maintenance systems can provide insights into asset condition and determine the remaining useful life of assets to give recommendations for maintenance activities.

  A typical application of Big Data Analytics in asset maintenance is accurately modeling the diagnostic asset condition. Works in [173] and [177] prove the usefulness of Big Data Analytics in informing the maintenance decision-making. Another application is the track defect position that allows a significant reduction in the repairing time of the rail track [200]. Then, to estimate the likelihood of system failure, several machine learning-based methods in [112] aim to learn regulations and build failure estimation models.

3.2.2.3 Benefits of Big Data Analytics in ITS

From the above discussion, we can draw some benefits of Big Data Analytics for ITS applications:

- **Handling Complexity:** Transportation systems are inherently complex with dynamic and interconnected variables. Big Data Analytics can approach more to the complexity of real-world traffic scenarios and capture unpredictable patterns that can not be addressed by traditional approaches, such as physical modeling, due to their strong assumptions.
- **Real-Time Adaptability:** Big Data Analytics leverages real-time data streams, enabling immediate responses to changes in the environment. Unlike static models, which may be fixed for regular situations, Big Data Analytics can adapt dynamically, providing more real-time insights.
- **Scalability:** As transportation systems grow and evolve, Big Data Analytics offers scalability. It can handle massive data volume, variety, and transmission velocity.
- **Data-Driven Insights:** Big Data Analytics derives insights directly from data, thus offering more knowledge about irregular and unpredictable traffic patterns. Meanwhile, traditional

models relying on oversimplified or inaccurate representations may result in incorrect understanding.

3.2.2.4 Open Challenges

Although Big Data Analytics has brought significant benefits to ITS, there are still important challenges that have not been fully studied.

- **Data Privacy:** In the era of Big Data, privacy emerges as the most challenging and concerning issue. While data collection and transfer are essential for processing, the potential leakage of personal information during transmission, storage, and usage becomes a significant challenge. In ITS, this concern is particularly raised by the collection of the location of individuals and vehicles. Even if these data are necessary for several services, such as route recommendations for users, the owner of data can be harmful if they are used for malicious purposes. Therefore, one solution for privacy protection in ITS involves limiting data leakage during transmission through local processing. By implementing processing mechanisms at the local level, sensitive information can be safeguarded.

- **Data Processing:** The applications of complex processing methods for Big Data Analytics lead to high time and computational costs. The diverse formats of traffic data from various sources necessitate a flexible approach for effective processing. Additionally, the high-speed transfer of traffic data streams imposes specific time constraints. Therefore, ensuring timely calculations with large and fast data streams becomes a significant challenge in traffic data processing.

- **Data Storage:** Collected data in ITS has massively increased, surpassing the capacity of traditional data storage solutions. Consequently, managing the storage of big data has become a significant challenge in ITS. Nowadays, some cloud storage providers such as Google or Amazon have offered services for cloud-based data storage that is emerging as an efficient solution for handling large data sets. However, alongside adopting cloud-based storage, it remains important to select relevant data for retention while eliminating redundant or unnecessary information in data processing.

- **Openness:** Openness in the context of Big Data Analytics relates to the dynamics of the road network and driving environment. The capacity to maintain smooth functionality across diverse scenarios characterized by dynamic and heterogeneous inputs is important for enhancing the effectiveness of Big Data Analytics in this domain.

3.3 Discussion

In this chapter, we discussed the importance of ITS's services in our society, the role of traffic prediction, and the opportunities to apply Big Data Analytics in ITS. This motivates us to approach a traffic prediction method based on Big Data Analytics.

However, developing a data-driven traffic prediction method is challenging since traffic implies many factors that vary rapidly and unpredictably. These factors include weather conditions, special events, traffic accidents, peak hours, vacations, $etc.$ Besides, traffic is influenced by many variables, such as number of vehicles, speed, density, driver's behaviors, $etc.$ Previous studies have pointed out two indispensable elements: **long-term temporal and spatial dependency analysis** that must be included in traffic prediction methods to capture the influences of these factors on traffic. Temporal dependencies include strong seasonality ($e.g.$ holidays impact on traffic dynamics) and complex properties such as non-stationarity and non-linearity, making theoretical assumptions challenging to adapt in real-time. Besides, spatial dependencies due to the road network topology require prediction models to analyze multivariate observations and evaluate the correlations between neighboring roads. By analyzing the evolution of traffic data in space and time, we can gain insight into historical traffic patterns and traffic propagation between road segments.

Moreover, traffic data are collected at a massive scale through time and continuously transmitted with high speed and low latency between devices as **data streams**. The growing demand for analyzing such data encourages researchers to adopt an approach known as **streaming analysis** [62] for traffic data. This approach is able to continuously capture temporal dependencies over upcoming data while also providing the required outputs in real time. However, it also comes up with a new challenge regarding the balance between **model complexity** to ensure **complete dependency analysis**, and **model flexibility for dynamic updates**.

Additionally, traffic prediction models need to provide an explicit explanation of input-output causality to manage the quality of predictive information. Thus, the consideration of model **interpretability** is also necessary.

Finally, traffic data are also interesting to study at **different data scales and levels** as it enables analyzing the traffic influences among different urban areas. A traffic prediction model that performs well across different data scales can be beneficial for applications in multi-level traffic management since it allows us to gain a better understanding of traffic patterns at various scales [181].

The next chapter discusses the fundamental principles, advantages, and limitations of existing methods for traffic prediction according to the context presented in this chapter. These solutions are analyzed based on **their abilities of spatio-temporal dependency analysis, stream analysis, interpretability, and model versatility**. These components are defined as the key advantages of our proposal.

Chapter 4

State of the Art of Traffic Prediction

Objectives of this chapter:

- Discussing the traffic prediction problem
- Discussing the essential components of traffic prediction approaches
- Introducing main AI approaches that have been exploited for traffic prediction problem
- Discussing their advantages and remaining challenges

4.1 Traffic Prediction

Accurate and real-time traffic forecasting is essential for urban traffic control, safety, and guidance functions of the ITS. It is widely applied for various transportation services to make better travel recommendations, reduce the consumed energy, improve traffic efficiency, and especially alleviate traffic congestion and dangerous collisions at jam queues. Indeed, traffic prediction provides advanced warnings about the appearance of traffic congestion, enabling drivers to decelerate their vehicle's speed before approaching the queue. Furthermore, by estimating future traffic dynamics on the road network, we can anticipate the approximate location of the traffic jams, their queue, and their propagation characteristics.

Traffic prediction refers to the task of estimating the future traffic dynamics of a studied road network based on analyzing the historical data. The predictive information can be classified into three categories: *microscopic data:* the detailed information at the individual vehicle level such as vehicle's speed, acceleration, and travel time; *mesoscopic data:* the lane-level information, providing a slightly higher level of aggregation compared to microscopic data and *macroscopic data:* higher-level traffic characteristics such as mean traffic speed, traffic flow, traffic density. Furthermore, the

prediction horizon has various ranges adapting to different applications: *short and medium term (5-60 min)*: employed for controlling and optimizing vehicle behaviors towards applying energy management strategy or for route planing problems, *long-term (over 60 min)*: mainly applied for urban traffic control and management [107].

Due to the promising benefits of traffic prediction, numerous methods have been developed adapting to various studied scenarios, available data sets, and applications. According to the considered context and the scope of this research discussed in Chapter 3, we only focus on analyzing the traffic prediction approaches based on Big Data Analysis. We categorize these models into three main groups based on their underlying technique: **time series-based models, clustering-based models, and neural network-based models**, as illustrated in the following schema 4.1. Time series and neural network-based models are widely applied for addressing the traffic prediction problem, while the clustering-based model operates a similar principle to our learning method developed further.

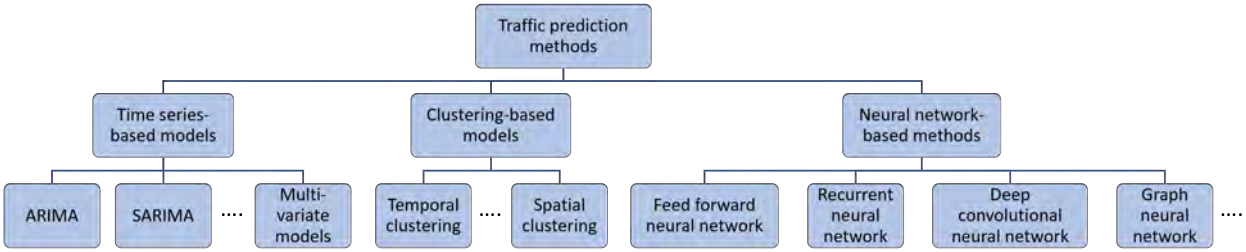


Figure 4.1: Traffic prediction methods. The presented algorithms categorized according to the nature of their underlying techniques

According to the challenges mentioned in the discussion of Chapter 4, the selected methods are analyzed according the 5 following components: **temporal analysis, spatial analysis, interpretability, streaming analysis, and multiple data scale adaptability**.

4.2 Time Series-Based Models

Time series-based models are well-known as parametric models based on the mathematical formula expressing the next values of time series from the linear combination of different terms such as the previous data, the random noise, the seasonal property, moving average, $etc.$ There are two groups of time series-based models:

- Univariate models: applied for time series consisting of scalar observations (speed, flow) at each timestamp.
- Multivariate models: applied for time series consisting of observation containing many features (an observation vector) at each timestamp. In the traffic problem, multivariate time

series-based models allow the integration of spatio-temporal relations with neighboring road segments to predict future values of considered location.

In order to apply time series-based models, the behaviors of data need to satisfy some assumptions such as linearity ($i.e.$ data points have a linear dependency on each other), stationarity ($i.e.$ the distribution of data does not change over time (constant mean and variance)). These assumptions aim to guarantee the reasoning of mathematical calculation in the underlying models.

In the following sections, we present ARIMA, SARIMA, and two multivariate models known as VARMA and STARIMA to analyze. These models can highlight the progression of time series-based models by expanding domain applications across different data types and reducing the assumption requirements [28].

4.2.1 ARIMA model

4.2.1.1 Principle

The basic model assumes that the future values of traffic information depend **linearly** only on the previous values and the random noise. Auto-Regressive (AR) and Moving Average (MA) are the main components to model time series. AR estimates the next value of the time series at time t X_t using the linear combination of previous value $X_{t-1}, X_{t-2}, \ldots, X_{t-p}$ and the noise term σ_t at time t. An auto-regressive model of order p, noted $AR(p)$ is expressed as:

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-p} + \sigma_t = \sum_{i=1}^{p} \phi_i X_{t-i} + \sigma_t \tag{4.1}$$

where X_t is required to be stationary, σ_t is the noise at t generated following a normal distribution and $\phi_1, \ldots, \phi_p$ are model parameters. The hyper-parameter p is called the model order representing the number of previous values that are used to compute X_t.

However, models using only AR component ignore the correlated noise structure in the time series. Thus, noise terms in previous steps $\sigma_{t-1}, \sigma_{t-2}, \ldots, \sigma_{t-q}$ must be integrated into the model. They are called the MA component. A moving average of order q or $MA(q)$ model is defined as follows:

$$X_t = \sigma_t + \theta_1 \sigma_{t-1} + \theta_2 \sigma_{t-2} + \cdots + \theta_q \sigma_{t-q} = \sigma_t + \sum_{j=1}^{q} \theta_j \sigma_{t-j} \tag{4.2}$$

where $\sigma_{t-1}, \sigma_{t-2}, \ldots, \sigma_{t-q}$ are the noise following a normal distribution and $\theta_1, \ldots, \theta_p$ are model parameters.

Auto-regressive and moving average models can be combined together forming ARMA models. An $ARMA(p, q)$ model is described as:

$$X_t = \sigma_t + \sum_{i=1}^{p} \phi_i X_{t-i} + \sum_{j=1}^{q} \theta_j \sigma_{t-j} \tag{4.3}$$

In most situations, especially for traffic data, time series have a trend and seasonality, which impact future values. Thus, they are not stationary. An extension of the ARMA model called ARIMA (Auto-regressive Integrated Moving Average) can deal with non-stationary property by considering the differentiation (the differences between consecutive observations). The expression of an $ARIMA(p, d, q)$ model can be written as follows:

$$\phi(B)(1 - B)^d X_t = \theta(B)\sigma_t \tag{4.4}$$

where:

- B is the backshift operator: $BX_t = X_{t-1}$
- $\phi(B)$ is the auto-regressive operator: $\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p = 1 - \sum_{i=1}^{p} \phi_i B^i$
- $\theta(B)$ is the moving average operator: $\theta(B) = 1 + \theta_1 B + \theta_2 B^2 + \cdots + \theta_q B^q = 1 + \sum_{j=1}^{q} \theta_q B^q$
- $(1 - B)^d = \nabla^d X_t$ is the differencing operator of order d. The first order of differencing operation is defined as: $\nabla X_t = X_t - X_{t-1} = (1 - B)X_t$

4.2.1.2 Application

The first application of the ARIMA model for traffic flow prediction was studied in [134]. This study fits the ARIMA model as well as a multiplicative model to traffic data by using the Box and Jenkins (BJ) technique for model identification [28]. Since ARIMA models have numerous parameters and hyper-parameters, the BJ technique proposes an iterative three-stage approach to identify an adequate ARIMA model and estimate its parameters. The first stage chooses the suitable ARIMA model by checking the stationarity and seasonality and estimating the order of differencing. The second stage computes the parameters that give the best fit to the selected ARIMA model using *the maximum likelihood estimation or non-linear least squares estimation*. The last stage relates to the model checking; if the test is failed, the process goes back to the first stage. The advantages of the BJ technique relate to improve the accuracy and flexibility of ARIMA models.

In [26], authors focused on using the ARIMA model to study the arterial travel time prediction problem. This study shows the promising effectiveness of applying ARIMA models for travel time prediction.

In [110], authors applied a subset ARIMA model, which is represented with only a few nonzero coefficients, for short-term traffic volume forecasting. The typical ARIMA contains large numbers of parameters associated with the orders of auto-regression and average moving. The BJ technique is used to determine the hyper-parameters (p, d, q), and then the Akaike Information Criterion (AIC)

[8] was applied to estimate the parameters of the selected model. According to the conclusion, the subset ARIMA model obtains more stable and more accurate results than other time series models, especially the full ARIMA model.

4.2.1.3 Analysis

The most relevant limitation of the ARMA model is that it can deal only with the stationary time series. The stationary property refers to the independence of statistic values of data on time. To address non-stationary time series, ARIMA was developed and has shown its effectiveness for traffic prediction problems. However, traffic prediction models require the integration of other relevant properties of traffic data to obtain more accurate estimations: (1) the seasonality, characterized by definite periodic cycles of traffic data, and (2) the spatio-temporal relationships between neighboring road segments. Some extended models, such as SARIMA, were developed to overcome this limitation.

4.2.2 SARIMA Model

4.2.2.1 Principle

To deal with the lack of seasonality of the ARIMA model, an extended version of ARIMA called Seasonal Auto-regressive Integrated Moving Average (SARIMA) is proposed. A seasonal ARIMA model is defined by integrating the seasonal terms in the ARIMA model, denoted as:

$$ARIMA(p,d,q)(P,D,Q)_m \tag{4.5}$$

$i.e.$

$$\phi(B^m)\phi(B)(1-B^m)^D(1-B)^d X_t = \theta(B^m)\theta(B)\sigma_t \tag{4.6}$$

where m represents the number of observations per seasonal period, (p,d,q) is the orders of non-seasonal components and (P,D,Q) is the orders of seasonal components. The seasonal components of the model consist of terms that are similar to the non-seasonal components of the ARIMA model but also involve the backshifts of the seasonal period.

4.2.2.2 Application

In [188], the theoretical basis for predicting univariate traffic condition data streams using SARIMA processes was presented. The empirical experiment shows that the seasonal difference of a one-week lag induces stationarity. This study proves that the seasonal difference significantly impacts results, and it is the key to the proper application of ARIMA modeling to time-indexed traffic volumes.

The SARIMA model was also studied to forecast the traffic flow in [187] by combining it with an exponential smoothing model. The auto-correlation function and the partial auto-correlation

function were used to determine the seasonal lag for each data set, thus contributing to model identification.

4.2.2.3 Analysis

The above studies have shown that **the integration of seasonal components in time series models allows for improvement in the model performance** for traffic prediction applications. However, these models can **only apply and test for univariate time series**. Meanwhile, traffic conditions on a road segment are highly dependent on its neighbors. Thus, spatio-temporal relations between road segments need to be considered in traffic prediction models. The multivariate extensions of the ARIMA model were developed to deal with this challenge.

4.2.3 Multi-Variate Models

4.2.3.1 Principle

The multivariate models describe the traffic flow in various locations through the state-space formulation. The consideration of the spatio-temporal relations allows to quantify the effects of the traffic conditions occurring at a road segment on its neighbors. Two well-known multivariate models are the Vector Autoregressive Moving Average (VARMA) and Space-Time Autoregressive Integrated Moving Average (STARIMA) [143], which are, respectively, the multivariate extensions of standard ARMA and ARIMA.

The VARMA model can estimate the interdependent interactions between multiple time series. As the ARMA model, the VARMA model consists of two components: the p auto-regressive order and the q moving average order and can be written as:

$$X_t = \epsilon_t + \sum_{i=1}^{p} \Phi_i X_{t-i} + \sum_{j=1}^{q} \Theta_j \epsilon_{t-j} \tag{4.7}$$

where $X_t = (x_t^1, \ldots, x_t^k)$ is the vector of measurements at time t of the k time series, Φ_i and Θ_j are respectively the matrices of unknown coefficients of auto-regression and moving average components which need to be estimated and $\epsilon_t = (\epsilon_t^1, \ldots, \epsilon_t^k)$ is the vector of white noise process.

The consideration of spatio-temporal relationships in multivariate models leads to the increase of parameters, resulting in demanding high computational capacities. Fortunately, a multivariate time series model called STARIMA can be refined by systematically expressing the spatial correlation between road segments. STARIMA expresses the value at time t of i^{th} road segment as the linear combination of auto-regressive, moving average components and the spatial lag. The spatial lag operator of order l $L^{(l)}$ is described as follow:

$$L^{(0)} x_t^i = x_t^i; \quad L^{(l)} x_t^i = \sum_{j=1}^{k} w_{ij}^{(l)} x_t^j \tag{4.8}$$

where $w_{ij}^{(l)}$ is the dependence weight between the locations i and j with $\sum_{j=1}^{k} w_{ij}^{(l)} = 1$ and $w_{ij}^{(l)}$ is nonzero if locations i and j are the l^{th}-order neighbors. The configuration of road segments can reflect the specification of weight value, and it may depend on the distance or the ease of accessibility between road segments.

4.2.3.2 Applications

The study in [95] proposes two applications of VARMA and STARIMA models for traffic prediction and compares them to the univariate models. The results show a significant improvement in prediction performance when using multivariate time series models in cases of large road networks with a high number of installed loop detectors.

The paper [130] introduces a refined version of VARMA to reduce the number of parameters. In this study, the authors integrated the spatial correlation matrices into the standard VARMA model. These matrices are noted $S^{ri} \in \{0,1\}^{N \times N}$ with r being the index of the current road and i being the number of intermediate steps that the other road can reach the current road following the considered road network. Thus, the values of matrices S^{ri} are nonzero if other roads can reach r^{th} road in i steps. Based on the mentioned principle, this study achieves a highly accurate prediction and fine granularity while guaranteeing reasonable computation.

4.2.3.3 Analysis

Multivariate models overcome the limitation of previous time series-based models by addressing the spatial correlations within the road network. These advancements have led to enhanced prediction accuracy compared to univariate models. This capability allows **the handling of larger datasets with complex temporal and spatial dependencies in traffic data**. Nevertheless, these models currently lack a method to evaluate **how other road segments influence the ones being considered** and **how large the neighborhood optimizes prediction performance**.

Restricting the analysis to a small neighborhood might result in nearby roads demonstrating similar traffic dynamics, potentially lacking in expressing beneficial traffic propagation characteristics. In contrast, an extensive neighborhood analysis risks including redundant information; roads too distant from the considered segment may not significantly impact or correlate with its traffic dynamics.

Hence, evaluating neighboring influence is essential in determining variables integrated into the prediction model and reducing the complexity and calculation resources.

4.2.4 Conclusion

This section presented the well-known time series-based models for traffic prediction problems. The presentation has shown their adaptation to deal with specific properties of traffic data, such as seasonality and multi-variate interdependence. However, despite these enhancements, time

series models rely on the linear nature of data for optimal performance and thus cannot deal with non-linear problems. They achieve good performances when traffic exhibits regular variations. However, the prediction error becomes significant when irregular situations occur.

In the following sections, clustering-based and neural network-based models are introduced to address the non-linear modeling for traffic data, known as **non-parametric model**s. Non-parametric models do not fix the dependencies between variables; the set of parameters and the model structure can change by adapting to arrival data. This property brings high flexibility for models but requires training them on large and diverse data sets to calibrate parameters. Thus, they are well-suited for analyzing complex and large data sets.

4.3 Clustering-Based Models

4.3.1 Principle

Clustering-based approach is one of the most well-known unsupervised methods applied for various aspects of AI, including classification, pattern detection, prediction, $etc.$ by discovering the structure of the analyzed data sets. They assign input data to different groups based on the distance between them such that the data belonging to the same group are more similar to each other than to those in other groups. In traffic prediction problem applications, clustering-based models find the adequate cluster to assign the current observation of the traffic state and predict the future states by considering the successive states of historical observations assigned to this cluster. Clustering-based models do not require assumptions concerning the data. They are able to model the non-linear, non-stationary behavior and are highly expansible. The process of applying these models for traffic prediction includes three steps: **Build data**, **Apply clustering models** and **Predict**.

In the data construction phase, traffic data is preprocessed before its usage as inputs of the clustering model. This process relates to clean noises, check up the duplicates, and standardize data. Then, an appropriated vector space is also defined to describe the traffic state. Each vector is a sliding window with size q noted as: $X_t = \{x_t, x_{t-1}, \ldots, x_{t-q}\}$. X_t indicates the number of previous measures relevant to qualify the traffic state at time t. The value q must be reasonably chosen since the small window can produce excessively similar values when comparing the traffic state. Meanwhile, large windows may contain redundant values to describe the traffic state, thus affecting model performance.

Then, depending on the data nature as well as the needs of applications, different clustering models are applied for traffic prediction problems. Work in [190] presented a complete list of clustering algorithm categories based on the technique used to group data into clusters. However, in the next, we highlight four groups that are widely applied to the traffic prediction problem:

- Centroid-based clustering: These models aim to determine the centroids of data and group data points together based on the proximity of data points to the centroid (cluster center).

K-Nearest Neighbors (KNN) [59] and K-Means [125] are the most well-known models of this group.

In KNN, the pairwise distances between all data points are computed using a distance measure such as Euclidean distance. Then, the k nearest neighbors of each data point are selected to build a cluster. The parameter k is a user-defined constant and needs to be delicately chosen to achieve the optimal clustering structure.

Unlike KNN, K-Means is designed to separate data points into k distinct and non-overlapping clusters where each data point belongs to only one cluster. To do that, K-Means performs iterative updates of the centers of clusters (centroids) until the criteria for convergence are reached. The convergence criteria are defined to create clusters in which data points within each cluster are more similar to each other than data points in other clusters. For example, K-Means is considered to obtain the optimal clustering structure if the sum of squared distances between the data points and their cluster's centroid is minimized. The main steps of the K-Means algorithm can be summarized as follows:

1. Specify the number of clusters k
2. Randomly select k data points as the k initial centroids
3. Assign data points to the closest centroid
4. Recompute the centroids of clusters by taking the average of all data points assigned to each cluster
5. Evaluate the convergence criteria
6. Repeat steps 3 to 5 until reaching the convergence condition

- Density-based clustering: The density-based models define clusters as the areas with high density in data space. DBSCAN [54] is the most well-known algorithm of density-based clustering algorithms. DBSCAN groups the close data points together to construct a cluster. Only the data points that have enough neighbors can be assigned to a cluster; otherwise, they are considered as noise. Due to this characteristic, DBSCAN can be applied for noise detection issues. Before deploying DBSCAN, two parameters need to be defined: the radius of the neighborhood ϵ and the minimum number of points in a neighborhood $minPts$. The main steps of DBSCAN can be resumed as follows:

  1. For every point, find the neighboring points that are inside the neighborhood circle.
  2. Identify the points having more than $minPts$ neighbors, called as *core points*.
  3. Find the connected components of *core points* on the neighbor graphs.
  4. Assign each non-core point to a nearby cluster represented by a *core point*; otherwise, assign it to noise.

- Distribution-based clustering: These models assume that data follow a single or mixture distribution such as the Gaussian distribution. The data points belonging to the same probability distribution will be grouped into the same cluster. GMM (Gaussian Mixture Model) [149] is a typical algorithm in this group. GMM algorithm aims to fit the original data set into the mixture models combined from different Gaussian distributions. Data points following the same independent Gaussian distribution are considered to belong to the same cluster. To apply GMM, the number of Gaussian components and the initial center of these components must be defined.

- Hierarchical clustering: The hierarchical algorithms evaluate the distance between data points to create the tree of clusters. They start by supposing that each data point stands for an individual cluster in the beginning, and then, the two closest neighboring clusters are merged into a new cluster until there is only one cluster left. Agglomerative algorithm [50] is a typical example of this group, with the main steps resumed as follows:

  1. Compute the dissimilarity matrix for every pair of data points.
  2. Search for the closest pairs.
  3. Group this pair into a new cluster and replace the original data points with the new cluster.
  4. Repeat the above steps until reaching one cluster.

In the final phase, the prediction will be computed. Once the new data is assigned to a cluster, many strategies can be applied to estimate the prediction. The most used strategy is taking the average/weighted average of the next traffic states of all historically assigned data in the same cluster.

4.3.2 Applications

The applications of clustering models in traffic prediction problems relate to both **temporal and spatial aspects**.

For **temporal clustering**, clustering-based models aim to determine the similar historical situations to the current one and observe the following traffic progression to estimate the prediction. In [203], KNN was first applied for traffic flow prediction. The used data set shows **the uncertain, non-linear, and complex characteristics**. The high accuracy achieved according to the chosen evaluation metrics shows the feasibility of the proposed method for short-term traffic flow prediction **without data constraints**. An improved version of the KNN model for traffic prediction was introduced in [30]. The spatiotemporal correlation is highlighted, where the nearest neighbors are selected according to the Gaussian Weighted Euclidean Distance to estimate the forecasted value at the final step of KNN. Besides, DBSCAN is mainly applied for detecting and predicting traffic

congestion or abnormal traffic situations due to its principle. Paper [157] presents a model for traffic pattern identification under **both normal and abnormal conditions** based on DBSCAN. The obtained prediction accuracy shows that this model outperforms KNN, ARIMA, and Support Vector Regression (SVR) algorithms. [161] introduces a short-term traffic congestion prediction method based on DBSCAN and Random Forest (RF) algorithms. DBSCAN was applied to identify the different levels of traffic congestion. The combination of these models achieves good performances with 94.36% of prediction accuracy. GMM was applied in [211] combined with the Bayesian network for network-wide traffic prediction. The proposed model has shown its advantages in terms of the **interpretability, generalization, and efficiency** compared to both traditional and advanced deep neural work-based models. Agglomerative clustering was applied in [15] for a dynamic data-driven local traffic state estimation and prediction method. This method shows the advantages for its **flexibility in incorporating additional explanatory variables**.

Besides, for **spatial clustering**, clustering models are applied for road partitioning. The roads in traffic networks can be grouped according to GPS position, similar dynamics of traffic evolution, etc. This partitioning allows us to determine the roads with high impacts on a given road. Indeed, many studies have demonstrated the advantages of spatial relationships over neighboring roads to improve prediction performance. However, **only neighboring roads that have significant effects or similar traffic dynamics to the considered road are beneficial for the prediction**. Considering insignificant neighbors risks including useless and redundant variables for the learning model. Moreover, grouping road segments with similar traffic properties enables the **adaptation of predictive models** for each road group. In [91], [156], authors apply the K-Means algorithm for road locations to cluster urban networks, define the new similarity matrix between observations, and apply the N-cut algorithm for the assignment decision. Another work presented in [36] applies different partitioning methods to divide the network into groups and uses an adaptive prediction model for each group.

Recently, many works have aimed at constructing clustering models for both time and space. Traffic regularity was studied using a 3D map, which consists of a joint partition of space (road network links) and time (time series observations) into homogeneous clusters applied for traffic congestion detection presented in [119] and for traffic prediction in [35], [37].

4.3.3 Analysis

Clustering-based models have broadened their utilization for traffic prediction problems due to their **ability to work without specific data assumptions**. This ability allows them to deal with non-linear and non-stationary data sets. Additionally, their functions, including the assignment rule and cluster identification, can be explicitly explained. Thus, they gain the advantages in terms of model **interpretability and explainability**, helping users to understand the effects of input variables on the output predictions and retain the control of intellectual functions.

Nonetheless, the most important inconvenience of clustering-based models is about **the selection**

of pre-defined parameters such as similarity threshold, number of clusters, minimum required data points to define a cluster, $etc.$ The experiments in the studied papers demonstrate that the values of these parameters significantly influence the model's performance. Moreover, the optimal parameter values vary for the different used data sets. This inconvenience **decreases the capacity of re-productivity and robustness** of clustering-based models. Additionally, most clustering-based models necessitate the computation of distance between data points, which raises two issues. First, the chosen distance function must adapt to the characteristics of the data. Second, the computational cost can substantially scale up when dealing with large data sets.

4.4 Neural Network-Based Models

Recent research in the field of traffic prediction has highlighted the promise of Neural Network-based (NN) models in handling the analysis of big data and high-dimensional features. With advancements in computational power and graphics processing, computers can efficiently execute complex NNs, facilitating the analysis of large amounts of data within reasonable time frames.

These enhancements empower the models to learn from more complex traffic patterns and capture higher dependence levels in traffic data. In the following, we provide an overview of various types of NNs that are widely employed to solve traffic prediction problems.

4.4.1 Feed Forward Neural Network

4.4.1.1 Principle

Feed Forward Neural Network (FFNN) [152] is a simple type of neural network in which the data flow in one direction from the input layer to the output layer, called forward direction. FFNN consists of an input layer, single or multiple hidden layers, and an output layer. Each layer contains a set of nodes or neurons (Multi-layer perceptron). The neurons are connected by weighted links to transfer the information from one layer to the subsequent one. FFNN is illustrated in the figure 4.2.

The output of FFNN y' is the value of the function f of the input $x = (x_1, \ldots, x_d)$ (d is the dimension of input) which is weighted by a vector of connection weights $w = (w_1, \ldots, w_d)$, completed by a bias b , and approximate non-linear function with an adaptive activation function ϕ:

$$y' = f(x) = \phi(< w, x > +b) \tag{4.9}$$

Back-propagation is the most used common technique to train the FFNN. The objective of this algorithm is to adjust the weighted parameter and the bias step by step to reach the minimum value of the loss function. The loss function is defined as:

$$L = \sum_{i=1}^{N} (y_i - y_i')^2 \tag{4.10}$$

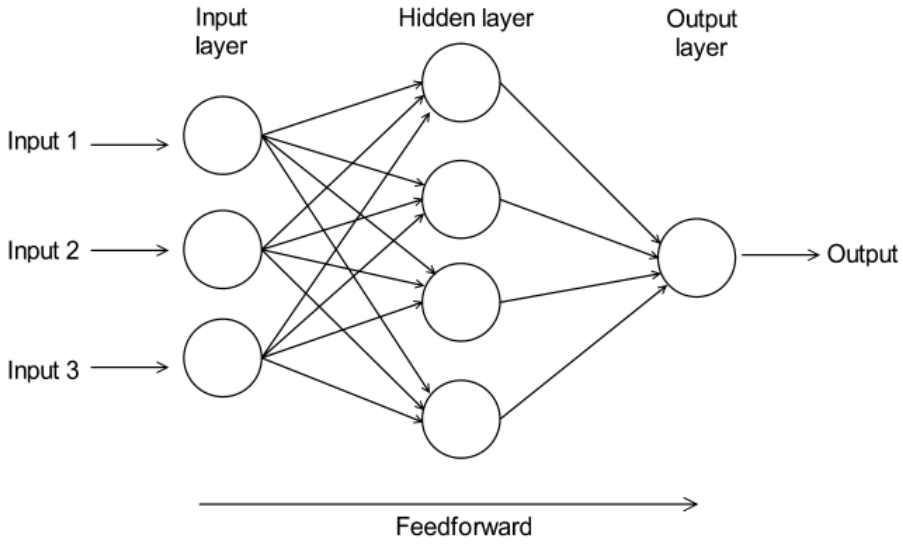


Figure 4.2: Feed Forward Neural Network with one hidden layer

where y_i is the real value of output and y_i' is the estimated output of FFNN and N is the size of training data set.

4.4.1.2 Application

Many recent researches illustrate the capacity of FFNN in traffic prediction. In [170], several FFNNs with one hidden layer were introduced with different structures to estimate the traffic flow in one or multiple next steps for one or multiple road links. In these models, the future traffic flow is estimated from five historical previous flows. This paper also employs Graphical Lasso for relevant feature selection. Similar to [170], [106] applies FFNN with one hidden layer to predict the traffic flow for the next 5 minutes. However, instead of only using the previous traffic flow as input variables, this model uses 19 input variables, which are the traffic density, the number, and the average speed of different types of vehicles. This study reinforces the stable performance of FFNN for short-term traffic prediction by confirming its ability in heterogeneous traffic conditions.

With the development of deep learning architecture, many researchers have proven that the multiple hidden layers are more capable of capturing the non-linear dependencies between input variables and outputs than a single hidden layer. In [145], authors use an FFNN with a stack of hidden layers to predict the traffic speed from the 40-previous speed measures of a set of sensors. Since only a few sensors have a significant influence on the target one, a regularized linear vector auto-regressive model is used to identify the spatial-temporal correlation between the sensors to simplify the model. The obtained results show that the MSE (Mean Squared Error) decreases 14% compared to the traditional FFNN with one hidden layer.

4.4.1.3 Analysis

FFNN with single or multiple hidden layers is a powerful approach for short-term traffic prediction thanks to **the ability of non-linear modeling**. FFNN is flexible for integrating the environment variables, other traffic information, or the traffic parameters of neighboring roads/sensors in the model.

However, capturing the complex and long-term dependencies of traffic time series data is a big challenge for FFNN due to **the increased model complexity**. Indeed, the more previous measures are taken into account, leading to the higher dimension problem in which FFNN is not an appropriate solution. Because of this drawback, few researchers used FFNN to analyze the spatiotemporal relations because the number of variables increases exponentially with the number of considered neighboring roads.

4.4.2 Recurrent Neural Network

4.4.2.1 Principle

The standard Recurrent Neural Networks (RNNs) [53] and the variations such as Long Short Term Memory (LSTM) [81] and Gated Recurrent Units (GRU) [45] have shown their success in time series forecasting by their ability to capture more complex and non-linear temporal dependencies than conventional neural networks. Their advantages are very important in network traffic prediction since the time series of traffic data have demonstrated complex and long-term dependencies and high degrees of non-linearity. **Unlike other feed-forward neural networks, RNN can use internal memory cells to handle timing inputs of any length.**

One of the most critical challenges for estimating the prediction of sequential data is the representation of serial order and the high-level interaction between the input and output sequences. For example, in natural language processing, to predict the next word in a sentence, the representation of previous parts of the sentence, including the word order, the meaning of words, or the grammar of language, must be considered. Unlike conventional neural networks, RNNs are capable of scanning the variable-length input sequences ($i.e.$ the time series of previous data) to create powerful representations from these contexts. The representations express what RNNs see in the past and capture all high-level dependencies of previous data to produce the subsequent data. Thus, RNNs are dynamic processing systems that are responsive to temporal sequences [53]. To do that, RNNs must provide the **network memory**.

In this part, three well-known variations of RNNs will be considered:

- Elman recurrent networks (standard RNNs)
- Long Short-Term Memory (LSTM)
- Gated Recurrent Units (GRU)

The standard recurrent unit was introduced in [53]. Figure 4.3 describes the structure of an Elman recurrent unit. Compared to the conventional FFNN, RNN includes the *hidden cell* or *context cell* in its architecture. In Figure 4.3, the hidden state at $t-1$ and at t are respectively noted $h_{t-1} \in \mathbb{R}^d$ and $h_t \in \mathbb{R}$ (where d is the cell dimension), $X_t \in \mathbb{R}^m$ is the input at t with size m and $Y_t \in \mathbb{R}^d$ is the input at t. The relationships between the hidden state, input, and output are expressed as:

$$h_t = \sigma(W_h.h_{t-1} + W_X.X_t + b_h) \tag{4.11}$$

$$Y_t = tanh(W_Y.h_t + b_Y) \tag{4.12}$$

Where $W_h \in \mathbb{R}^{d\times d}, W_X \in \mathbb{R}^{d\times m}, W_Y \in \mathbb{R}^{d\times d}$ are respectively the weight matrices of hidden state, input and output cells; $b_h \in \mathbb{R}^d, b_Y \in \mathbb{R}^d$ are respectively the bias vector of hidden and output cells. Note that RNNs use only one set of weight matrices and bias through all time steps. In Equation 4.11, σ is the activation function of the hidden state (ex. sigmoid function), the hyperbolic tangent function (indicated by tanh in Equation 4.12) as the activation of the output. In Equation 4.11, the current hidden state h_t depends on the current input X_t and the hidden state of the previous time step h_{t-1}. Then, the hidden state, which captures all information of previous contexts contained in the previous hidden state and the current context, is used to calculate the output. This recurrent connection is the key of RNN, which explains the network memory principle and guarantees the consideration of past information when updating the output.

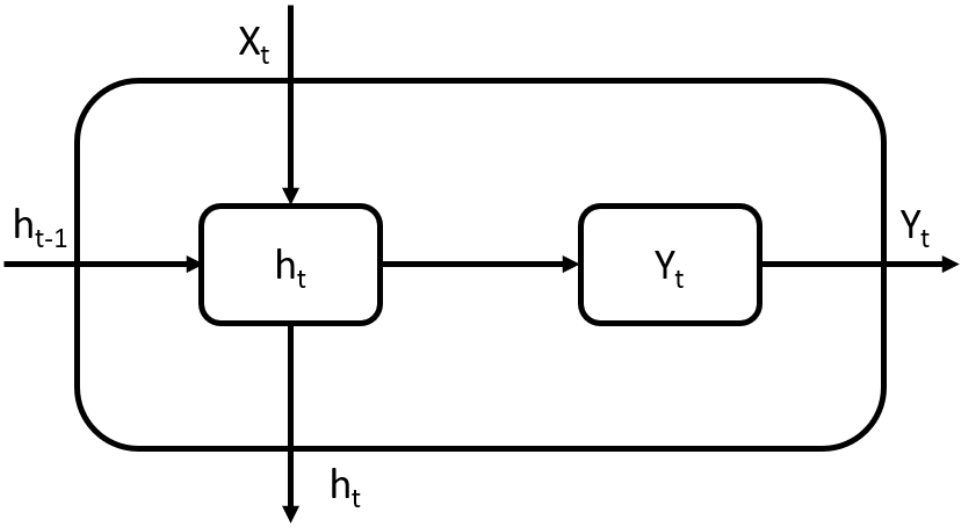


Figure 4.3: Elman recurrent unit [53]

However, the standard RNN suffers the vanishing and exploding gradient problem when learning long-term dependencies [23]. This problem degrades the update of weight matrices during back-propagation algorithms for long-term learning. RNNs train the model by back-propagating the gradient through time. If the effect of the previous layer on the current layer is small, then

the gradient value will be small, and the back-propagated gradients tend to shrink exponentially. Consequently, the model does not consider the previous inputs. In contrast, when the gradients are large, the product of derivatives can explode over numerous time steps, resulting in an unstable model. The mentioned problem implies that the standard RNNs are inefficient for capturing long-term dependencies.

To overcome the addressed problem of standard RNNs, Long Short-Term Memory (LSTM) was proposed in [81]. LSTM unit uses multiple gate mechanisms to control the gradient to avoid vanishing/exploding issues. LSTM cell has four gates: input gate, output gate, update gate, and forget gate. Besides the hidden state, LSTM adds a state component called the internal cell state into each cell. The structure of the LSTM unit is described in Figure 4.4.

$$\text{Concatenating vector} : X = [X_t, h_{t-1}] \tag{4.13a}$$
$$\text{Forget gate} : f_t = \sigma(W_f.X + b_f) \tag{4.13b}$$
$$\text{Update gate} : u_t = \sigma(W_u.X + b_u) \tag{4.13c}$$
$$\text{Output gate} : o_t = \sigma(W_o.X + b_o) \tag{4.13d}$$
$$\text{Input gate} : i_t = tanh(W_i.X + b_i) \tag{4.13e}$$
$$\text{New cell state} : c_t = c_{t-1} * f_t + i_t * u \tag{4.13f}$$
$$\text{New hidden state} : h_t = o_t * tanh(c_t) \tag{4.13g}$$

In the above equations, the description of the mentioned terms are as follows:

- $X \in \mathbb{R}^{d+m}$ is the concatenation of input and previous hidden state vectors where d is the cell dimension and m is the input dimension.
- $i_t \in \mathbb{R}^d$ is the input gate, $W_i \in \mathbb{R}^{d\times(d+m)}$ and $b_i \in \mathbb{R}^d$ are the weight matrix and the bias associating to the input gate.
- $c_t \in \mathbb{R}^d$ is the cell state which corresponds to the long-term memory component.
- h_t is the hidden state which corresponds to the short-term memory component.
- $f_t \in \mathbb{R}^d$ is the forget gate, $W_f \in \mathbb{R}^{d\times(d+m)}$ and $b_f \in \mathbb{R}^d$ are the weight matrix and the bias associating to the forget gate.
- $u_t \in \mathbb{R}^d$ is the update gate, $W_u \in \mathbb{R}^{d\times(d+m)}$ and $b_u \in \mathbb{R}^d$ are the weight matrix and the bias associating to the update gate.
- $o_t \in \mathbb{R}^d$ is the output gate, $W_o \in \mathbb{R}^{d\times(d+m)}$ and $b_o \in \mathbb{R}^d$ are the weight matrix and the bias associating to the output gate.

These three gates learn the behaviors of inputs to identify the unnecessary parts to forget (f_t), the important information that will persist through many time steps (u_t), and the part participating in updating outputs (o_t). The applied activation function for these three gates is the sigmoid function, thus their values are between 0 and 1. The value close to zero kills most of the input signal, and the value close to one lets the signal pass through nearly unchanged, adapting to the purpose of each gate.

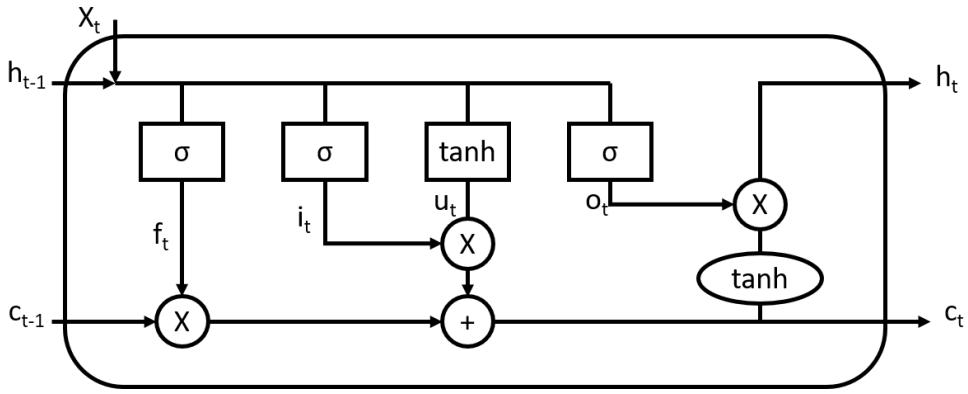


Figure 4.4: LSTM unit

The calibration of parameters in LSTM requires a high computational complexity since it has four sets of weight matrices and bias parameters that need to be estimated. Another variant of RNNs called GRU (Gated Recurrent Unit) [45] is well known as a simpler and more efficient manner to deal with long sequences. Indeed, GRU considers only one component of the state, $i.e.$, the hidden state, instead of two components in the case of LSTM. Moreover, GRU has only two gates: the update gate and the reset gate, whereas the update gate plays the combined role of the forget gate and input gate in LSTM. GRU can yield similar performance as LSTM while requiring less computational cost. Figure 4.4 shows the structure of a GRU cell. The relations between the elements are expressed as follows:

$$\text{Concatenating vector} : X = [X_t, h_{t-1}] \tag{4.14a}$$

$$\text{Update gate} : u_t = \sigma(W_u.X + b_u) \tag{4.14b}$$

$$\text{Reset gate} : r_t = \sigma(W_r.X + b_r) \tag{4.14c}$$

$$\text{Candidate hidden state} : \tilde{h_t} = tanh(W_h * [X_t, r_t * h_{t-1}] + b_h) \tag{4.14d}$$

$$\text{New hidden state} : h_t = u_t * \tilde{h_t} + (1 - u_t) * h_{t-1} \tag{4.14e}$$

The weight matrices and bias follow the same notation and have the corresponding dimensions with their associated gates. The reset gate decides the part of the previous hidden state that contributes to the candidate state of the current step. Since there is no forget gate in GRU, (1 - u_t)

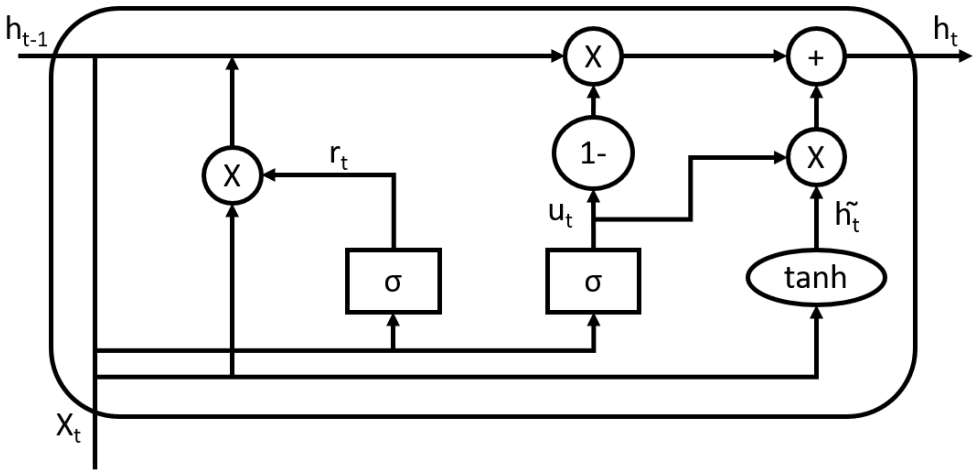


Figure 4.5: GRU unit

is an alternative. Indeed, since u_t indicates the important part of the input signal that contributes to the hidden state, (1 - u_t) can refer to the irrelevant part that can be ignored from the previous hidden state.

4.4.2.2 Applications

One of the first applications of RNN for traffic prediction was introduced in [123]. This paper enables long-term traffic forecasting by using the Bayesian normalized combined with the Elman neural network to overcome the influence of the hidden layer nodes on prediction accuracy. The **reliability and stability** of this model have been demonstrated by the comparative analysis with different classical NNs.

In [174], LSTM was applied for traffic flow prediction, aiming at determining the optimal time lags dynamically without requiring the predefined and static length of the input historical data. Moreover, the conducted experiment shows the **efficiency and generalization for difference prediction horizons** of the proposed model. Additionally, to address the **traffic prediction problem in extreme conditions**, the study in [197] builds a deep neural network based on long short-term memory (LSTM) units. This model was applied to predict peak-hour traffic and identify unique characteristics of the traffic data. Moreover, it was also improved for **post-accident forecasting** to model regular traffic conditions and the pattern of accidents jointly. In [47], a deep stacked bidirectional and unidirectional LSTM neural network architecture was proposed for network-wide traffic speed prediction, which can **capture spatial features and bidirectional temporal dependencies** from historical data. This model is scalable for both freeway and complex urban traffic networks. In [131], a temporal information-enhancing LSTM (T-LSTM) was proposed for traffic flow prediction. This model can integrate the view of the similar characteristics of traffic flows at the same time each day to capture **the correlation between traffic flow and temporal**

information. A combined prediction method for short-term traffic flow based on the ARIMA model and LSTM neural network was proposed in [120]. In this study, **both linear and non-linear features of traffic data are captured** respectively by the rolling regression ARIMA and LSTM. This combined model has shown its outperformance compared to the single ones.

Recently, GRU has been widely applied to tackle traffic prediction problems. [201] proposes an urban traffic flow prediction considering weather conditions using GRU. In [48], GRU was applied combined with a spatio-temporal analysis for short-term traffic flow prediction. The novelty of this model relates to **the ability to define the optimal input time interval and spatial data volume** by spatio-temporal analysis. In the experiment, this model outperforms the single GRU and CNN in accuracy and stability. Authors in [169] aim at increasing model efficiency and reducing the computing cost of traffic prediction models applied for complex road networks. To do that, the Selected Stacked Gated Recurrent Units model (SSGRU) was proposed, enabling to **limit the number of potential hyper-parameters for a reasonable model complexity**. The two most recent applications of GRU are presented in [199] and [160]. Paper [199] enables **the integration of heterogeneous data sources** within a preprocessing data pipeline, resulting in hybrid feature space. This research applies a variety of deep learning algorithms such as LSTM, GRU, CNN, and their combinations to test their performance on these hybrid data. The hybrid LSTM–GRU model outperforms the rest with a Root Mean Squared Error (RMSE) of 4.5 and Mean Absolute Percentage Error (MAPE) of 6.67%. In [160], a Random Forest- Gated Recurrent Unit- Network Traffic Prediction algorithm (RF-GRU-NTP) was developed for **the network traffic flow prediction in the context of Vehicular Ad hoc Networks (VANETs)**, Vehicle-to-Vehicle (V2V) and Vehicle-to-Road Side Units (V2R) communications. To deal with the combined data from different sources, a hybrid model was proposed to select the important features. The obtained results show that the proposed RF-GRU-NTP model has better performance in execution time and prediction errors than other well-known algorithms for network traffic prediction.

4.4.2.3 Analysis

RNN-based models (LSTM and GRU) have been widely applied for traffic prediction recently thanks to their capacity to analyze long-term dependencies. Their design with sequences of layers enables processing sequences of inputs, making them **well suited with the sub-sequences of time series inputs**. Through the existing works, RNNs have demonstrated their efficiency for traffic prediction problems in which the analysis of **long-term temporal dependencies is necessary**. Moreover, the applications of RNN-based models for traffic prediction is diverse. LSTM and GRU have been successfully applied to process different data sets with various constraints in various scenarios of road networks and driving environments. Besides, they are combined with other advanced techniques to improve prediction performance.

Nonetheless, most of the works using RNN-based models have not addressed the spatial relationships in traffic networks. Indeed, RNN-based models are **not well-suited with multi-**

variate analysis. The high dimensionality of observation leads to **significantly increasing the model's complexity**. Therefore, when dealing with time-space traffic data, they need to combine with other methods to improve prediction performance.

4.4.3 Deep Convolutional Neural Network

4.4.3.1 Principle

Convolutional Neural Network (CNN) [109] is a class of Neural Network which is successfully applied in computer vision domain such as image classification and recognition. The most significant advantage of CNN compared to the classical NN is the consideration of the locality of features. Indeed, the analysis of small zones in an image allows to capture the local spatial dependency between the pixels and their neighbors. The capacity of CNN can respond to the important challenge in traffic prediction, which is the integration of both temporal evolution and spatial dependencies of collected data of road networks. In this application, traffic network data are transformed as an image which extracts the spatiotemporal traffic features. Then, CNN is employed to predict traffic parameters using the traffic images.

Apart from the input and output layers, a CNN architecture includes different types of hidden layers, such as the convolutional layers, pooling layers, and fully connected layers. A convolutional layer convolves the input features with different kernels, which are the set of weights learned over the training process. The kernel convolves across the width and height of the input volume and computes the dot product between them and the pixels covered by the size of the kernel at each move. As a result, the convolutional layer detects some specific local connections at some zones on the image. A pooling layer is applied after a convolutional layer to downsample and aggregate data. One of the most popular types of pooling layers is *maxpooling*, which applies the max operation on the adjacent values in the specific zone to extract only max values from them. The pooling layers guarantee extracting the invariant and prominent features of CNN. These two types of hidden layers allow the extraction of the relevant features from inputs. After that, the matrices are flattered before passing through the fully connected layers for prediction. The CNN model is illustrated in figure 4.6

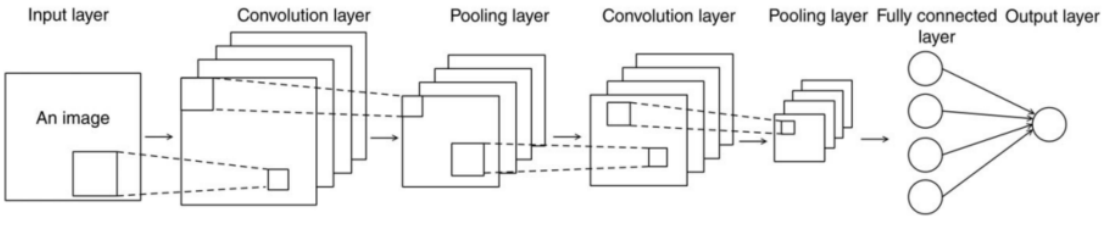


Figure 4.6: CNN model illustration

4.4.3.2 Applications

[124] proposes a convolutional neural network (CNN)-based method that **learns traffic as images and predicts large-scale, network-wide traffic speed with a high accuracy**. Spatio-temporal traffic dynamics are converted to images to describe the time and space relations of traffic flow via a two-dimensional time-space matrix. In [196], a spatiotemporal recurrent convolutional network (SRCN) was proposed based on DCNNs and LSTM. This study introduces **a network grid representation method** that can present network-wide traffic speeds as a series of static images. Thus, the spatial dependencies of network-wide traffic can be captured by DCNNs via image analysis, and LSTMs can learn the temporal dynamics via time series analysis. Paper [75] proposes a novel end-to-end deep learning model, called ST-3DNet, for traffic raster data prediction. ST-3DNet introduces 3D convolutions to automatically capture the correlations of traffic data in both spatial and temporal dimensions. This model can **explicitly quantify the difference of the contributions of the correlations in space**. The study in [41] aimed at jointly capturing multiple spatio-temporal dependencies. A multiple gated spatio-temporal CNNs (MGSTC) was developed with **a novel gated scheme to control the spatio-temporal features**, enabling extracting spatio-temporal features simultaneously from low-level to high-level layers and dynamically combining these features with external factors.

4.4.3.3 Analysis

The applications of CNN-based models for traffic prediction problems have enabled **the extraction of spatio-temporal traffic features** in large road networks. The conversion of traffic data as the time series of images has leveraged the strong ability of CNN-based models to extract representative features from high-dimensional data. Over numerous studies, CNN has shown a significant improvement in prediction accuracy when taking into account the complex spatial dependencies of the road network for traffic prediction. Moreover, the combination of CNN and RNN-based models is able to complete the analysis of long-term temporal dependencies realized by RNNs and spatial dependencies done by CNN. These combined approaches have shown their outperformance compared to the single ones.

Nevertheless, CNN is essential for addressing problems in Euclidean space, such as images and grids. Meanwhile, **the spatial dependence in traffic networks is highly complex due to road topological structure**. Thus, CNN may not provide the most appreciated solution for spatial modeling in traffic-related challenges.

4.4.4 Graph Neural Network

4.4.4.1 Principle

Spatio-temporal dependencies in traffic data are complex due to the strong temporal dynamics of traffic conditions and the complex relationships of road networks. Thus, modeling these de-

pendencies is a challenging task for traffic prediction models. NN-based models such as CNN tackle this problem by transforming the time-space data into images (grid-structured data). However, this data format is insufficient to express the flexibility and complexity of spatio-temporal dependencies in traffic data. Indeed, it is necessary to describe how the roads impact each other (downstream/upstream dependencies, opposite/parallel direction, $etc.$

Recently, graphs have become well-known as a powerful data representation tool due to their ability to express the complex relationships and interactions between entities. However, processing graph data requires specific characteristics of the learned model. In NN-based models, Graph Neural Networks (GNN) [159] are designed to deal with this data format.

To apply GNN, the road network is first transformed into graphs. In [206], an unweighted graph $G = (V, E)$ is used to describe the topological structure of the road network. In this graph, each road is associated with a node; thus, $V = \{v_1, v_2, \ldots, v_N\}$ is a set of N road nodes, and E is a set of edges. The adjacency matrix A is used to represent the connection between roads, $A \in \mathcal{R}^{NXN}$. The adjacency matrix contains only elements of 0 and 1. The element is 0 if there is no link between roads, and 1 denotes there is a link. A feature matrix $X_t \in \mathcal{R}^{NXP}$ is defined to describe the traffic data at t, where P is the number of features for each node equal to the length of the historical time series. $X_t \in \mathcal{R}^N$ is an observation on road network at each timestamp t. Thus, the goal of GNN is to estimate a mapping function f on the premise of road network topology G and feature matrix X to compute the traffic information in the next T time intervals based on the n historical observations.

$$[X_{t+1}, \ldots, X_{t+T}] = f(G; (X_{t-n}, \ldots, X_{t-1}, X_t)) \tag{4.15}$$

4.4.4.2 Applications

Work in [206] proposes the **temporal graph convolutional network** (T-GCN) model, which is a combination of the Graph Convolutional Network (GCN) and gated recurrent unit (GRU). Specifically, the temporal dependence is captured by GRU, while the spatial dependence is learned by the complex topological structures of GCN. The experiments conducted on real-world data sets demonstrated the outperformance of T-GCN compared to state-of-the-art baselines. In [182], another spatial-temporal graph neural network for traffic flow prediction was proposed. This method is able to **dynamically model the influence of the traffic on one road to others** instead of using static modeling. [89] investigates **a data-driven, long-term, high-granularity traffic speed prediction approach** based on graph deep learning techniques. To reach these objectives, this study adopts a graph convolution-based multi-scale latent information extraction mechanism to construct the predictive model. The experiments on real-world data sets show consistent improvements in their performance compared to the baselines. In [141], a long-term traffic flow prediction method based on dynamic graphs was developed. The novelty of this model relates to the policy network based on **reinforcement learning to generate dynamic graphs to deal with data sparsity issue**. The test demonstrated that this model can achieve stable and effective long-term predictions of traffic

flow and can reduce the impact of data defects on prediction results.

4.4.4.3 Analysis

Recently, GNN-based models have been widely exploited for traffic prediction problems due to the **efficient spatio-temporal modeling**. Using this technique, the topological structure of the road network is represented through the set of nodes and edges of the graph to keep the original spatial dependence between traffic data. Moreover, the flexible structure of the graph allows it to adapt to different road network scenarios, leading to the high versatility of the predictive model. This point has been demonstrated by the impressive results achieved by recent research applying GNN-based models for traffic predictions. Their outperformance has crossed diverse experiments with different prediction horizons, traffic information, road scenarios, $etc.$

However, like other NN-based models, GNN exhibits a black-box nature that cannot provide an explicit explanation of the model's functionality and limits the interpretability of the model. Furthermore, the prediction calculation is complex and costly due to the addition of a graph transformation step that is not appreciated to deal with traffic data streams arriving at high frequency.

4.5 Discussion

Approach	Model	Temporal modeling	Spatial modeling	Stream analysis	Model interpretability	Model versatility
Time series-based	ARIMA	+	- -	- -	++	- -
	SARIMA	+	- -	- -	++	- -
	Multivariate models	+	+	- -	++	- -
Clustering-based	Temporal clustering	+	-	- -	++	+
	Spatial clustering	+	+	- -	++	+
Neural network-based models	FFNN	+	-	- -	-	-
	RNN	++	-	- -	- -	-
	CNN	++	+	- -	- -	-
	GNN	++	++	- -	- -	+

Table 4.1: A comparative list of traffic prediction models

Table 4.1 summarizes the advantages and limitations of the above methods according to five

characteristics of the traffic prediction model: **temporal modeling, spatial modeling, stream analysis, model interpretability, and model versatility**. The evaluation indicators are placed following their abilities addressing to these components: (++) methods can fully address a component and provide the appreciated solution to deal with related challenges, (+) methods address a component, but the following methods have proven that this component can be tackled by other approaches, (-) methods are able to deal with a component but they are not the appreciate one, (- -) methods cannot deal with a component due to their characteristics or the existing researches have never mentioned about this ability.

According to this table, all methods address the **temporal modeling** component especially recent methods like RNN, CNN, and GNN that presented their strong ability to capture historical temporal traffic patterns.

Concerning the **spatial modeling**, its handling was limited to some methods. The time series-based models include the spatial modeling by analyzing the inter-dependency between time series that are observed on multiple road segments by multi-variate models. The clustering-based models address this component by regrouping road segments having similar characteristics and adapting learning models to each cluster. Among the solutions for the spatial modeling, GNN has proven to provide an appropriate solution for this challenge. However, the strong modeling ability comes with a trade-off of higher model complexity. Indeed, RNN, CNN, and GNN exhibit their **black-box natures**, restricting the explanation of the input-output causalities. From the **interpretability** point of view, the time series-based models gain the clear interpretation thanks to their explicit equation to express the dependencies between parameters and the clustering-based models are explained by understanding the assignment rules. In addition, the **versatility** of methods is limited when they rely on many data assumptions (in case of time series-based models) and lack the interpretability (in case of neural network-based models). Finally, all well-known methods did not explore the stream analysis component to evolve and adapt themselves in real-time when data arrive continuously. From the limitations of previous methods, we aim to develop a traffic prediction system composed of **continuous learning and cooperative prediction**, which can effectively address and provide the appropriate solutions for the mentioned challenges.

4.6 Conclusion

This chapter presented a short state of the art of traffic prediction problem. The existing methods were described and discussed according to five characteristics: temporal modeling, spatial modeling, stream analysis, model interpretability, and model versatility. These characteristics enable addressing the challenges of traffic data stream analysis and enhance the accuracy and reliability of traffic predictions under different test scenarios. While the discussed methods have addressed the main issues of traffic prediction problems, and their solutions significantly improve prediction performance, none of them can fulfill these characteristics.

In the following chapters, we introduce the **dynamic clustering and multi-agent paradigm**. We explore their fundamental principle, key features, and applications. Dynamic clustering offers an effective analyzing method for data streams, while a multi-agent system is well-known for complex problem-solving. They are widely employed across various domains to address challenges similar to those encountered in traffic prediction and have achieved promising results. Therefore, we base our proposal on these techniques to fulfill the aforementioned five characteristics.

Chapter 5

Dynamic Clustering

Objectives of this chapter:

- Introducing main properties of data streams
- Introducing key characteristics of dynamic clustering
- Discussing the existing methods for data stream clustering
- Discussing their deployed applications.

5.1 Introduction

Nowadays, every simple transaction by credit cards, phones, web browsers, and transport cards has led to automated data transfer and storage. Indeed, recent advances in all hardware, software, and connection technologies have allowed data acquisition at large scale and continuous flows. As a result, for the last decades, we have seen the massive and unbounded sequences of data transferred every second, the so-called *data stream* [6]. The data stream is defined by the following characteristics [162].

- Data arrive continuously.
- The size of a stream is potentially infinite.
- Data generation process is unknown and possibly non-stationary ($i.e.$ its probability distribution may change over time)

The characteristics of the data stream lead to several challenges for the conventional data processing methods in terms of storing, querying, analyzing, extracting information, and processing

to compute predictions. First, with **the increase of data volume**, storing all data requires an expensive cost and, in some cases, becomes impractical. Thus, random access to the entire data set is not feasible. Second, **the processing of large data volume** needs high computational capacity of computers. Third, as **data may evolve over time**, the straightforward adaptation of one-pass mining algorithms may not be an efficient solution for this task. Indeed, this property of conventional methods can lead to the over-fitting problem. Thus, the challenges require a novel approach designed by focusing on the evolution of data streams.

Numerous methods for data stream mining have been developed, including clustering, classification, frequent pattern mining, change detection in data streams, and stream cube analysis of multi-dimensional streams [6]. Among them, clustering is widely studied as an essential domain of data stream mining. In order to analyze the structure of the data stream, the dynamic clustering technique is based on its ability to structural changes referring to the following behaviors [14]:

- **Formation of new cluster:** When new data cannot be assigned to existing clusters since its behaviors differ from the learned ones, a new cluster is created to represent this data.
- **Merging of clusters:** Two or more clusters can be merged into one new cluster if they are sufficiently close to each other.
- **Splitting of clusters:** If the similarity zone of a cluster is too big, it becomes dispersed (low density) and appears as two or more clusters when it absorbs historical data. In this case, that cluster is split into several smaller clusters.
- **Destruction of clusters:** Clusters may be removed if there is no new data being assigned or their representative behavior is obsolete due to changes in the environment.
- **Adjustment of clusters:** When new data is assigned to a cluster, that cluster gradually moves towards the new data by adjusting its centroid.

Thanks to these interesting properties, various approaches for data stream clustering have been widely investigated in diverse domains such as network intrusion detection, financial transactions, phone recording, web click stream processing, $etc.$ [67].

5.2 Data Stream Clustering Methods

In this section, we discussed different existing data stream clustering approaches following their category grouped according to their underlying principle.

5.2.1 GNG Based Algorithms

Growing Neural Gas (GNG) is an incremental network model inspired by the family of topological maps such as Self-Organizing Maps (SOM) [102], Neural Gas (NG) [127]. GNG is an unsupervised

learning algorithm able to learn the topological structures of a given input data set distribution and construct a graph of nodes and edges representing clusters and connected links (edges) between them. Each node in the graph has the **position** and an **accumulated error variable**. Two nodes can be connected by an edge that is characterized by an **age**. The main steps of GNG are described by the following steps [60]:

- Start with two nodes a and b at random positions w_a and w_b respectively.
- Generate the new input data ξ.
- Among the existing nodes, determine the nearest node s_1 and the second-nearest node s_2 of ξ.
- Increment the age of all edges from s_1
- Compute the Euclidean distance from ξ with the position of s_1 and add it to the accumulated error of s_1.
- Slightly move s_1 and all the nodes connected with s_1 towards ξ.
- If s_1 and s_2 are connected by an edge, set the age of this edge to zero. Otherwise, create it.
- Check all the existing edges nodes. The edges having an age larger than age_{max} and the nodes having no connection with others will be removed.
- If the number of data points passed is an integer multiple of a parameter β, insert a new node.
- Decrease errors of all units.

From the basic algorithm, various variations of GNG have been proposed to adapt to different applications and challenges. GNGC (Growing Neural Gas for Temporal Clustering) introduced in [163] modifies the original version of GNG by detecting incrementally emerging cluster structures. To do that, GNGC creates a new node whenever the new data is significantly far from all existing nodes. Then, the studied method in [64] addresses the ability of a self-organizing network model with time constraints for real-time applications. Indeed, this study conducted an experiment to estimate the optimal parameters in GNG to achieve good pattern detection under a given time limit. Another variation of GNG named AING (Adaptive Incremental Clustering Method Based on the Growing Neural Gas Algorithm) aims to limit the large incorrect nodes in the graph that can decrease the computational efficiency over time. In AIGN, when the number of nodes reaches a given *upper bound*, nodes are merged. A recent version of GNG is presented in [66] that performs an exponential fading function, weighted edge management, and reservoir management.

5.2.2 Hierarchical Stream Methods

A hierarchical clustering method analyzes the hierarchical structure of a given data set by constructing a tree of clusters. Two strategies are used to build this tree:

- *Agglomerative* or *bottom-up* approach where each observation starts in its own cluster, then the merging step combines pairs of clusters to move up the hierarchy;
- *Divisive* or *top-down* approach, where all observations belong to one cluster, is recursively split to move down the hierarchy.

The obtained hierarchical tree is presented in a *dendrogram* shown in figure 5.1. In the dendrogram, the ordinate axis presents the distance or the dissimilarity measure between clusters (*leaves*). This measure is determined using a cluster linkage criterion such as maximum linkage, minimum linkage, average linkage, *etc.* When applying this approach to clustering problems, partitioning clusters at a desired level of precision is achieved by selecting a cutting line along the tree. Typically, the cutting location can be decided visually at the ordinate with a significant cluster distance or at the minimum cluster distance required for the considered problem.

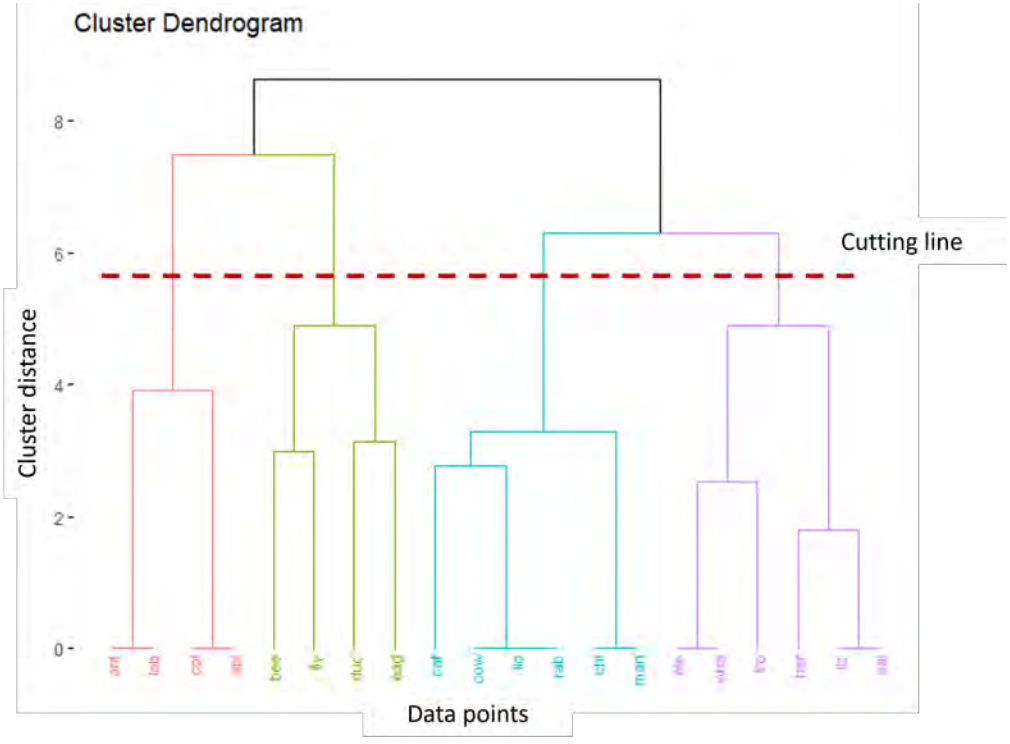


Figure 5.1: Illustration of a dendrogram

For data stream clustering problems, several adaptations have been introduced. **Balanced Iterative Reducing and Clustering using Hierarchies** (BIRCH) [205] is well-known as the first method based on hierarchical clustering for data streams. BIRCH introduces a new data structure

called **clustering feature (CF)**. The CF is considered a concise summary of each cluster with three pieces of information: the number of data points N, the linear sum LS, and the squared sum SS of N data points. When a new data point arrives, it traverses down from the root until it finds the appreciated leaf with the closest CF to assign. If the closest cannot absorb this data, a new CF is created. If the size of the tree does not allow the creation of a new leaf, the parent node is split. Note that BIRCH is not primarily designed for data stream problems. Thus, it cannot address the concept of drift problems, such as the self-evolution of clusters. Another method called **E-Stream** (Evolution-based stream clustering) [178] addresses the five categories of data evolution: **appearance, disappearance, self-evolution, merge and split**. E-Stream highlights the split step with Fading Cluster Structure (FCS) representing each cluster. FCS uses **a α-bin histogram to present each feature of the data set**. Cluster split is based on the distribution of feature values summarized by the cluster histogram. If a statistically significant valley is found between two peaks in histogram values along any dimensions, the cluster is split. ClusTree - **a parameter-free method** based on hierarchical approach is introduced in [105]. ClusTree is capable of processing the stream in a single pass with limited memory usage. This advance comes from the usage of the *micro-cluster structure* as a compact representation of the data distribution. However, ClusTree needs an offline phase using k-means or DBSCAN to determine the global partitioning structure of a given data set.

5.2.3 Partitioning Stream Methods

A partitioning-based clustering algorithm aims to group the data points in a given set into partitions (or clusters). The clusters are determined based on a *distance function* such as k-means or k-nearest neighbors algorithms. This approach is also known as the distance-based clustering approach to distinguish with *density-based approach* presented in the section 5.2.4. Two well-known methods of partitioning-based clustering approach are **CluStream** and **StreamKM++**.

CluStream [7] is able to **divide the clustering process into an online component** and **extend the algorithm over different portions of streams**. The process of CluStream is divided into two phases: **online micro-clustering and offline macro-clustering**. In the online micro-cluster phase, the arrival of new data from a stream can either be appended to an existing micro-cluster or create a new one. The decision is based on the comparison of the distance between the new data and the existing micro-clusters and the maximum boundary defined for each cluster. Thus, if the new data does not lie within the boundary of its nearest cluster, a new cluster is created. Then, the offline macro-cluster phase regroups the micro-clusters obtained from the online process to determine the final set of clusters based on the K-Means algorithm [87] where K is the defined number of clusters. The obtained clusters are specific for the data stream perceived during a given time horizon. The drawback of CluStream is that **the number of clusters is fixed** and provided by the user, which limits the adaptability of the algorithm.

StreamKM++ [5] proposes a novel method to compute small weighted samples, called the *coreset* of the data stream. The coresets in StreamKM++ are constructed following the tree structure that

aims to significantly speed up the time necessary for sampling non-uniformly during the coreset construction. After the coreset tree is extracted from the data stream, the K-Means++ algorithm [19] is applied to detect the final set of clusters that presents the structure of the original data stream.

5.2.4 Density-Based Stream Methods

The density-based stream methods are based on the underlying principle of density-based clustering described in Section 4.3. However, the process of density evaluation in these algorithms can be intensively costly. Therefore, in the application for data stream scenarios, the main challenge is constructing density-based algorithms that can efficiently process a single pass of data streams within a reasonable time frame. Some well-known density-based data stream clustering methods are DenStream, SOStream, and SVStream.

DenStream [31] is an improving algorithm which **does not require the assumption on the number of clusters and enables to deal with evolving data stream with noise**. The principle of determining the micro-cluster is similar to the CluStream one. Its main improvement is the mechanism for **detecting the outlier micro-clusters (*i.e.* noise)** based on data density. The criterion used to distinguish the normal micro-cluster and noise is due to the different constraints on weight, $w > threshold$ and $w \leq threshold$, respectively. As the weights of micro-clusters that do not match new data decay gradually, micro-clusters that do not satisfy the condition on weight are considered as noises. On the other hand, noise clusters can turn to micro-clusters if new data matches with them. Finally, to generate meaningful clusters for a time horizon, DenStream applies the DBSCAN algorithm [54] on the set of micro-clusters in the online phase.

SOStream [85] is a density-based clustering algorithm inspired by both the principle of the DBSCAN algorithm to identify the set of neighbors and self-organizing maps (SOM) to describe the influence of the winner cluster on its immediate neighborhood. SOStream also starts with a set of micro-clusters with a predefined radius. However, clusters are self-adapted through the upcoming data using the novel online cluster update and merging strategies to establish the final representative clusters for the data stream. When the closest cluster adjusts its centroid when assigning new data, the clusters in its neighborhood determined based on DBSCAN are also updated to move closer to new data. Additionally, the online merging strategy allows the identification and correction of the overlapped clusters. Thus, the final representative clusters become less sensitive to outliers.

SVStream (Support Vector based Stream clustering) [180] is a data stream clustering algorithm based on **support vector domain description and support vector clustering**. In this algorithm, data points are projected into a kernel space such as a Gaussian kernel. The support vectors are used as the summary information of original data points to establish the cluster boundaries. The mapped data that are enclosed by a sphere belong to the same cluster. To adapt to both dramatic and gradual changes, multiple spheres are dynamically maintained, each describing the corresponding data domain presented in the data stream.

5.2.5 Agent-Based Methods

The above data stream clustering models were tested on applications with homogeneous clusters. That means clusters have the same assignment decision and the same data natures/characteristics. To enable the local decision at the cluster level, a clustering system needs to distribute the assignment decision at the cluster level and decentralize the control and data storage. This requirement can be satisfied by the Multi-Agent System (MAS) paradigm. The aims of MAS are to decentralize the modeling and distribute the global task at the agent's levels. An agent is defined as an autonomous entity pursuing its own local goal, having partial knowledge of the system environment as well as a local processing mechanism consisting of a three-step cycle (perception, decision, and reaction). Agents have to cooperate together to achieve the global goal of the system that cannot be reached individually. During their interactions, agents can autonomously organize and evolve to adapt to different situations and solve potential conflicts.

Work in [71] introduces **a self-organization mechanism** to form the clusters with **continuous adaptive decisions**. Agent's actions are evaluated, and they are validated if these actions allow an increase in the overall system performance ($e.g.$, density of cluster).

[38] presents **a multi-agent clustering system** whereby each cluster is agentified. The decisions of cluster agents are based on K-Means and KNN strategies. The K-Means-based strategy aims to minimize the sum of the distance from all data points to their nearest cluster. Meanwhile, the objective of the KNN-based strategy is to ensure that each data point is assigned to its closest cluster. However, the existing MAS-based clustering methods need to memorize all the historical data to evaluate the agent's actions. That is why, for the data stream, this requirement can lead to a high demand for storing capacity and costly computational capacity due to the huge amount of arrival data.

5.3 Applications

Data stream clustering methods have been widely applied across various domains to address diverse problems. In the next, we introduce an overview of relevant applications of data stream clustering that are developed recently.

Application	Paper	Proposition
Internet of things (IoT) management	[97]	Pattern recognition and occurrence identification in IoT data streams, towards improving the efficiency of storing, transmitting and computing of large data streams and detecting anomalies
	[98]	Weighted reservoir sampling algorithmic framework based on K-Means for stream event identification, aiming at revealing the *black swans* referring to rarely occurring events deserving special handling

	[18]	Streaming sliding window local outlier factor coreset clustering algorithms for real-time anomaly detection from IoT data
	[65]	Distributed denial of service for attack detection in IoT-based networks, involving the role of machine learning and deep learning in cyberattack identification from smart sensing devices
Geochemistry	[25]	SOM and K-means clustering to reveal the regional geochemical patterns of regularly sampled stream sediment data, allowing to detect the anomalous zones helping for *"delineating ore-related geochemical anomalies"*
	[44]	SOM for the recognition of geochemical patterns, element associations and anomalies related to mineralization, providing guidance for further mineral exploration
Data privacy protection	[167]	Privacy-preserving location data stream clustering method with effective integration of edge computing, cloud computing and differential privacy for location-based data clustering, providing an essential element for service recommendations
Network intrusion detection	[88]	A K-Means clustering and SVM based hybrid concept drift detection technique for network anomaly detection, helping to detect the intrusion and protect the network
	[103]	Real-time event detection in social media streams using semantic representation and Semantic Histogram-based Incremental Clustering based on semantic relatedness, aiming at improving analysis of noisy terms, representation/embedding of contents, and summarization of event clusters in social media streams
Web graphs	[104]	Clustering-based partitioning for large web graphs, including three-steps pipelined techniques consisting of streaming clustering, cluster partitioning, and partition transformation for web graph construction
Intelligent transportation system	[9]	Cauchy density-based model for VANETs clustering in 3D road environments, ensuring the reliability and stability of VANETs communications
Medical	[193]	Clustering algorithm based on information entropy to detect outliers in the high-dimensional medical data stream, helping for disease prevention and source analysis
Customer segmentation	[34]	A customer segmentation method based on stream clustering for transactional data adapting to evolving and dynamic context of e-commerce, helping to improve marketing strategies

	[122]	Hierarchical fragmentation-coagulation processes for dynamic customer segmentation, allowing to understand better customer's behaviors, towards developing efficient marketing strategies or tailoring social programs adapting to public values
Financial market analysis	[79]	Real-time cluster configurations of streaming asynchronous features for online state descriptor of financial markets, enabling capturing salient features of limit order book and efficient near-real-time use
	[21]	Tracking method for cluster's evolution of financial time series adapting to fluctuated and dynamic stock market

Table 5.1: Applications of data clustering

This short overview demonstrates the effectiveness of data stream clustering methods for solving diverse problems in various domains. The aforementioned applications commonly aim to **detect the different patterns or behaviors of data in the context of large and high-speed data transmission**, which are similar to the current ITS context considered in our study. Therefore, we consider the data stream clustering-based method as an appreciated solution to traffic prediction problems in the current ITS context. However, due to the specific characteristics of traffic data and the objectives of this study, we aim to develop **a novel data stream clustering-based method that fully expresses the main properties of the data stream analysis approach while adapting to our study case**. The following discussion will detail the potential of data stream clustering-based methods in traffic prediction solving and the components that we aim to consider in our proposal.

5.4 Discussion

In data mining and analysis of ITS, **clustering tasks for traffic pattern detection is an important topic** since it helps to enhance traffic management strategies, both in terms of timing and location, towards improving the efficiency and safety of urban transportation. Indeed, understanding the real-time traffic conditions across diverse locations and time frames helps predict the occurrence time and the locations of traffic bottlenecks in a city [175]. Additionally, it provides data on the available capacity of roads at different times, enabling assessments of the network's robustness. Or, by integrating historical and live data, irregular states in traffic patterns, such as accidents, can be detected.

Besides, in the modern ITS, **traffic data is continuously gathered in real-time from diverse sources**, including sensors, cameras, GPS devices, and other monitoring technologies, *etc.* to provide dynamic and updated observations of traffic conditions, vehicle movements, and infrastructure status. These data are nowadays essential to improve the quality of transportation networks. However, they also come up with challenges in terms of **efficiently storing, processing, and extracting meaningful insights from the constant data streams**. First, data streams may contain

new and unpredictable behaviors, requiring processing methods to possess a **dynamic and flexible structure** capable of self-evolving and adapting to novel situations. Second, the huge volume of traffic data makes **storing all of them impractical** due to the high cost associated with storage capacity, organization, and maintenance. Third, with the rapid transfer of traffic data streams facilitated by high-speed, low-latency 4G connectivity, processing methods need to deliver results within a reasonable calculation time, necessitating a **completely online processing mechanism**. Finally, traffic data streams may involve data from various sources, resulting in heterogeneous data types. To extract valuable insights that a single data type cannot provide, a **heterogeneous data processing method with a dynamic decision-making algorithm** becomes an essential solution.

Considering the two arguments presented above, we highlight four components of a streaming clustering method for traffic data: **dynamic and flexible structural changes, online processing, dynamic assignment decisions, heterogeneous data processing, and efficient data storage.**

Methods	Dynamic and flexible structural changes	Online processing	Dynamic assignment decisions and heterogeneous data processing	Efficient data storage
ClusTree	+	-	-	-
CluStream	+	-	-	-
DenStream	+	-	-	-
GNG	+	+	-	-
MAS-based	+	+	+	-

Table 5.2: Evaluation of existing methods for traffic data stream clustering

Table 5.2 presents the evaluation of relevant data stream clustering methods within each category according to four adequate components for the traffic prediction application. ClusTree, CluStream, and DenStream enable the online construction of clusters from the arrival data stream. However, the final clustering structure is still determined by the offline algorithms, which are inadequate for the continuous and infinite data stream. GNG, on the other hand, can capture data structure by finding topological structures that closely reflect the structure of the input distribution. However, it performs with homogeneous clusters that have the same assignment decision and data natures/characteristics. Finally, existing MAS-based clustering methods address three characteristics and perform a completely online processing by evaluating, integrating each arriving data and adjusting existing clusters adapting to this data. Thanks to their distributed and decentralized function at agent's level, these methods can deal with heterogeneous data, for instance processing different data types using different types of agents, and adapt the choice of the assignment decisions by taking into account the characteristics of agents. However, it is mandatory in MAS-based clustering methods to memorize all historical data for evaluating agent actions. Consequently, they lack an efficient data storage component.

5.5 Conclusion

In this chapter, data streams are introduced as the current trend of data acquisition and transmission due to the advances in hardware, software, and connection technologies. This type of data requires specific processing methods known as dynamic clustering. The key characteristics of dynamic clustering were highlighted with the ability of dynamic structural changes to adapt to new arrival data. Moreover, the applications of dynamic clustering for data stream analysis have been widely deployed with different approaches in various domains.

The above discussion has justified the opportunity of using a dynamic clustering approach for traffic prediction problems in this thesis.

Chapter 6

Multi-Agent System

Objectives of this chapter:

- Introducing multi-agent systems
- Presenting the self-organization and cooperation properties of agent-based methods towards adaptive multi-agent systems
- Discussing the existing applications of multi-agent systems in ITS

6.1 Multi-Agent System

Nowadays, real-world applications become more and more complex, characterized by numerous components, the open and dynamic environment, and the diverse interactions between them, leading to a large amount of unpredictable situations. Therefore, traditional information modeling that aims at centrally controlling and studying all potential situations of systems is no longer suitable for complex problems. To tackle the complex system, the agent-based systems known as **Multi-Agent Systems (MAS)** are the most representative [186] among artificial tools. The MAS paradigm offers an efficient solution for complex problems through distributed computation and decentralized control at the agent's level. Agents can dynamically change their behaviors, adapting to a local dynamic environment and allowing them to address unpredictable events. Then, they can interact with each other to reach the global goal. The MAS-based systems are composed of three main elements: agents, environment, and their interactions.

6.1.1 Agents

A well-known definition of *agent* is introduced in [186]: *An agent is a computer system that is situated in some environment and that is capable of autonomous action in this environment to meet its design objectives.*

Agents are highlighted by the following characteristics:

- **Autonomous:** Agents can decide their behaviors by themselves (no external intervention).
- **Located in an environment:** Agents are situated in an environment. Their behaviors affect this environment and vice versa.
- **Having partial representation of environment:** Agents perceive information from the part of the environment they interact with.
- **Having local decision functions/skills and goals:** Each agent pursues an individual goal achieved by its local skills, allowing agents to react to different perceptions, evolve, and adapt to environmental states.
- **Able to interact with the environment and other agents:** The interactions with other entities allow the exchange of information inside the system, enabling the emergence of the global goal of MAS from a set of local goals of agents.

An agent performs a continuous life cycle including three steps: **perception, decision, and action**. In the first step, agents perceive new information coming from their environment. In the decision step, agents choose the action depending on their perceptions of the environment. In the last step, agents perform the selected action in the previous step and modify the local part of their environment where they are situated.

Depending on the relevant properties in an agent, three main classes of agents exist: *reactive, cognitive and hybrid agents*.

6.1.2 Multi-Agent Systems

A MAS is composed of a set of interacting agents. Each of them has its individual objective, knowledge, and skills. They interact with each other to reach their own goals, thus resulting in the global goal of MAS. The MAS is highlighted by the following properties:

- Autonomous: As MAS is composed of autonomous agents, MAS can autonomously function, and there is no external global control.
- Distributed and decentralized: Knowledge is distributed at the agent's level since agents own a partial representation of the environment. Moreover, the control is distributed, and no supervisor controls the global system.
- Asynchronous and parallel: These properties refer to the execution ways of agents.
- Open/close: A close MAS performs with the fixed set of agents. Meanwhile, agents in an open MAS can be created or removed during the system's execution.

- Heterogeneous/Homogeneous: MAS can contain the same type of agents (homogeneous MAS) with the same patterns and models or different agents (heterogeneous)

6.1.3 Environment

The environment of the MAS is defined as all external entities of the world that influence MAS. As mentioned, agents in MAS are situated in it, interact, and modify it. Thus, depending on the characteristics of the environment, the decision-making process of agents has different levels of complexity. According to [155], the environment is mainly characterized by the following properties:

- Accessible / Inaccessible: In an accessible environment, agents can access the environment information at any time. Otherwise, only partial information is available for agents. The more accessible the environment, the simpler operation of agents is performed.
- Discrete / Continuous: The environment is discrete if its state, the set of percepts, and the actions of agents are clearly defined and distinct. In the continuous environment, they are infinite.
- Deterministic / Stochastic: In the deterministic environment, the next states are completely determined based on the current state and the action of agents. Contrastingly, the effect of agent's actions can be diverse in different contexts.
- Dynamic / Static: The dynamic environment can evolve independently of the actions of agents. Otherwise, the static environment cannot evolve without actions from the system.
- Episodic / Sequential: In the episodic environment, at each episode, agents receive and perform a single perception and action. The next episode does not depend on the results of actions from previous episodes. Otherwise, in a sequential environment, the current decision can affect all future episodes.

6.1.4 Self-organization

One of the most important properties that make MAS suitable for solving complex problems is the *bottom-up* strategy in its design. In this strategy, the system is built from the base elements or agents towards the high levels until forming a complete top–level of the system. Thus, global behaviors emerge from every individual behavior and the interactions between agents at micro levels. However, agents at low levels only have a partial view of the environment and limited skills. Therefore, to guide agents' behaviors and their dynamic interaction leading to the emergent phenomenon, MAS performs the *self-organization* mechanism.

Self-organization is generally defined as a set of dynamic interactions that raise the appearance of global structures of a system from interactions among its lower-level components. In MAS, the

embedded self-organization mechanism helps the agents to self-organize and to achieve, at the macro level, the objective of the system the designer expects.

However, ensuring the emergence that leads to desired behaviors at the global system level is a challenging task. On the one hand, emergent behaviors cannot be controlled by any global supervisors. Thus, their appearance can be under different manners. On the other hand, software designers aim to build systems and ensure they achieve the desired objective while considering constraints from environments. To tackle this antinomic situation, the solution is to understand relations between micro and macro levels of the system and implement the codes containing the functions allowing **self-organization and self-adaptation** of the system during interactions with environmental dynamics.

6.2 Adaptive Multi-Agent System

In the previous section, multi-agent systems have been introduced with the definitions of agent, environment, and self-organization. MAS highlighted the cooperation between agents to emerge the global system's goal from individual goals. However, leading the emergence towards the expected objective or **adequate function of system** is still challenging since there is no external entity evaluating the system's performance. Thus, to self-organize, the system needs to evaluate itself. In other words, agents must assess their actions and behaviors with local knowledge while ensuring adequate global behavior.

To tackle this challenge, the AMAS approach has been introduced in [33]. AMAS proposes the cooperation between agents as the answer, enabling the design of the system to perform complex tasks.

6.2.1 Fundamental theory

The AMAS approach guarantees the convergence towards the desired global function by following the theorem of **functional adequacy**: *For any functionally adequate system, there exists at least one cooperative medium system that fulfills an equivalent function in the same environment*, which is demonstrated in [69]. The functional adequate system implies that the system has no antinomic interaction with its environment. This theorem aims to indicate that the cooperative medium system is functionally adequate.

A cooperative internal medium system is a system composed of parts that always interact cooperatively. Thus, a MAS becomes a cooperative internal medium and has an adequate function if all agents in this system are in a cooperative state. To conclude, all agents in this MAS need to interact with each other and with the environment cooperatively to guarantee the emergence of an adequate global function (shown in Figure 6.1).

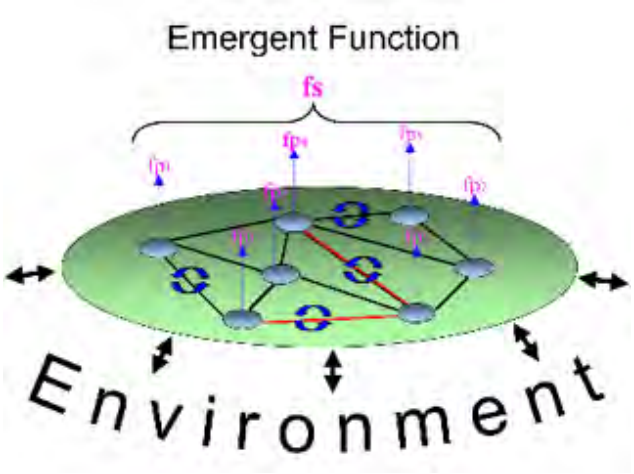


Figure 6.1: Emergent function

6.2.2 Non-cooperation situations

Previous sections showed that cooperation is the key mechanism of AMAS to implement solutions for complex problems. The functional adequacy theorem ensures that for a calculable problem, a MAS whose agents are in cooperative states can find the desired solution. To reach cooperative states, the obvious way is that agents need to avoid all non-cooperation situations. Indeed, the non-cooperation interactions within the system result in the non-adequacy of the system.

An AMAS needs to identify the perturbing situations and define the associated solutions to help agents overcome them when they encounter the perturbations. We call these situations as *Non-Cooperation Situations (NCS)*. There are seven types of NCS that an agent can encounter [68]. They can happen at all three steps of the agent's life-cycle.

- **Perception:**
  - *Incomprehension:* the agent cannot understand received information
  - *Ambiguity:* the perception can be understood in different ways, and agents are not sure which one is correct
- **Decision:**
  - *Incompetency:* the agent has no skill to process the received information
  - *Unproductivity:* the agent cannot propose any useful action
- **Action:**
  - *Concurrency:* the chosen action puts the agent in concurrence with others since their goals are the same
  - *Conflict:* the performed action of the agent is incompatible with the actions of other agents

  - *Uselessness:* the agent proposes an action that does not change the state of the common environment or that is not interesting for other agents

To solve the NCS, agents in AMAS define the set of associated *cooperative actions*. Three types of cooperative actions of agents have been identified [32]:

- **Tuning:** the agent adjusts its internal parameters to modify its behavior.
- **Reorganization:** the agent modifies the way it interacts with its neighborhood
- **Evolution:** the agent can create other agents or delete itself when its functionality is useless.

6.3 Application of AMAS in ITS

Nowadays, the evolution of ITS technologies is directing our focus towards the concept of user-centric services. This concept highlights the importance of personalized services that prioritize the consideration of user's needs in the development of digital mobility solutions. Moreover, the evolution of a large number of intelligent vehicles, connected infrastructures, *etc.* and numerous interactions between them make the current transport problems complex. Therefore, traditional modeling approaches that address the macro-level transport are no longer suited to the current context. Meanwhile, the concept of MAS oriented to micro-level agents holds promise in addressing this challenge due to its characteristics discussed in previous sections.

MAS has been employed for various applications in ITS, including the simulation, controlling, and management of ITS. A Ph.D thesis presented in [126] tackles the problem of communication in the fleet of autonomous and connected vehicles using the AMAS approach. Indeed, connected vehicles highlight the advances of V2X technologies in exchanging data between vehicles. However, since the amounts of exchanged information and the sources are large, vehicles need to know what information to transmit, receive, and how. This work proposes two AMAS-based modules allowing vehicles to extract useful information adapting to their reference and optimize the information exchanges within a fleet. The good performance of the AMAS approach in optimizing information exchanges also enhances its ability to offer solutions for data confidentiality preservation and efficient data utilization. [74] proposes a constructivist approach based on MAS for a self-adaptive decision-making system of road traffic control. The proposal leveraged the concept of MAS and reinforcement learning to tackle unknown and changing complex environments where the construction of a complete representation of the system is not feasible. Another Ph.D thesis in [165] applied MAS for the conception of cooperative and intelligent transportation. This work focuses on the behaviors of autonomous vehicles in the decision-making protocols for ITS. Work in [49] introduces a multi-agent model for resource allocation and scheduling in vehicle fleets to improve on-demand transport (ODT) systems. In this study, autonomous vehicles can communicate via peer-to-peer radio channels to address passenger needs and satisfy trip requests in an online ODT

system. The proposed multi-agent model demonstrates proficiency by handling diverse constraints and allowing different approaches to find solutions and coordinate vehicles.

6.4 Discussion

The management of modern transportation has become challenging nowadays due to the increase of intelligent and interactive entities and numerous interactions between them. Thus, a distributed and decentralized process would be an effective solution for the following reasons:

- **Fault tolerance:** Distributed and decentralized systems help mitigate the impact of faults or failures in one part of the system on the rest, minimizing disruptions to the functioning of the transportation system.
- **Openness:** Transportation systems must change and update dynamically, adapting to varying needs and demands. A decentralized management allows to easily scale up, add, or remove some entities from the network.
- **Reduced latency:** A distributed system can delegate the decision at the local entity level. This distribution enables the adaptation of transportation services according to specific conditions. Moreover, it can reduce the processing time, allowing the system to react quickly to urgent events.
- **Enhanced privacy and security:** Distributed systems improve privacy and security compared to centralized alternatives. Personal data and sensitive information can be stored and locally processed, reducing the risk of data breaches or unauthorized access.
- **Efficient communication and exchange:** Distributed systems can be designed to be more interoperable with diverse components and technologies. This is particularly important in ITS, where various vehicles, sensors, and infrastructure elements need to communicate effectively. That helps to avoid the bottleneck, especially during peak usage, and ensures smoother and more efficient operation by distributing the operation across multiple nodes.

The AMAS approach deals with complex problems in a dynamic environment through the bottom-up design of multiple cooperative agents, where the decision is decentralized, and the global task is distributed at the agent's level. Thus, the AMAS approach is a potentially adequate solution for the management tasks of modern transportation, including traffic prediction.

6.5 Conclusion

This chapter introduced the **MAS approach** as an effective solution for complex problems in an open and dynamic environment by the bottom-up design of agents. The AMAS approach deals

with complex problems by distributing the global objective into the local task of agents and enabling the cooperation between agents to solve global tasks that cannot be pursued individually. This approach allows for addressing problems that cannot be achieved by centralized modeling since the number of participating entities and their interactions are numerous.

Autonomous agents in the AMAS approach are known for their two main functionalities: **local decision and cooperation**. The local decision of agents consists of the set of skills allowing them to pursue their individual goals by interacting with a part of the environment. The cooperative behaviors allow agents to interact with other agents, enabling the emergent phenomenon. This mechanism, known as **self-organization of agents**, leads the local tasks to achieve the global objective of the system.

It is challenging to design a self-organization mechanism that can lead the system to the expected behaviors without global controls. Thus, to guarantee the convergence towards the desired functionality, **the AMAS-based systems**, at design time, define the cooperative behaviors of agents. According to the functional adequacy theory, these behaviors enable the emergence of a system's functionality that would solve unexpected and disturbing situations or **Non-Cooperation Situation (NCS)** towards adequate performance.

Seven general NCSs are defined that can happen during the interaction of agents. They are generalized to cover unpredictable situations encountered in a large variety of complex problems. The definition of NCSs allows developers to identify them and build associated cooperative behaviors.

Finally, various applications of the MAS and AMAS approach in ITS were presented, demonstrating the potential of employing MAS to address transportation-related challenges. These findings enhance optimism regarding their efficacy in solving traffic prediction problems.

Chapter 7

ADRIP - Adaptive multi-agent system for DRIving behaviors Prediction

Objectives of this chapter:

- Introducing the multi-level traffic prediction problem
- Presenting ADRIP - a continuous learning and cooperative prediction system that addresses this problem
- Describing ADRIP's architecture
- Introducing the definition of agents and their behaviors to learn and predict traffic dynamics
- Describing the self-adaptation mechanisms of agents

The two previous chapters provided an overview of the dynamic clustering method for data stream processing, the MAS paradigm, and the AMAS approach for complex problem-solving. Dynamic clustering addresses continuous learning, allowing the system to evolve by integrating new data. Meanwhile, the AMAS approach emphasizes the ability of self-organization among cooperating agents to jointly solve problems that cannot be tackled individually. These approaches present significant potential for addressing the complexities of traffic prediction problems.

This chapter defines the multi-level traffic prediction problem studied in this thesis. The objectives of the proposed solution to address the challenges of this problem are determined. Then, we introduce ADRIP - an agent-based model for continuous learning and cooperative prediction applied to traffic dynamics prediction. The description of ADRIP justifies the adoption of the dynamics clustering and AMAS approach to fulfill the objectives of the system. Lastly, we describe the role and generic function of the proposed system when dealing with multi-level traffic prediction problems.

7.1 Multi-Level Traffic Prediction Problem

Road traffic management has become increasingly challenging and intricate in the context of the current road network and transportation infrastructure. This complexity is due to the growing number of vehicles, the integration of intelligent traffic equipment, and the complex nature of road network topology. These factors result in many possible interactions between heterogeneous entities, including physical and logical ones, in a dynamic environment. Therefore, adopting a **multi-level, intelligent, and distributed approach** is valuable in traffic management.

In this way, the promising approach involves a **multi-level architecture using the multi-agent approach.** This approach focuses on understanding the system's behavior at the agent level, assigns agents to different levels based on the road network's architecture, and considers their interactions to observe the emergence of global system behaviors. On the one hand, it facilitates understanding the influences of each road network entity on the others through the analysis of their interactions.

On the other hand, decentralizing road network modeling enhances **the system's openness** for diverse applications and studies, mainly when the interest or the available data addresses only specific entities within the road network rather than the entire network.

Lastly, the multi-level road network architecture allows to demonstrate **the versatility of the prediction system**. It accomplishes this by acknowledging that observations from different road network levels exhibit distinct characteristics regarding traffic features, data variability, data scales, etc. Achieving consistent and reliable performance across these different network levels strengthens the prediction system's robustness.

7.1.1 Multi-Agent Road Network Architecture

The hierarchical organizational architecture of road networks is based on the AGRE (Agent-Group-Role-Environment) meta-model suggested by [58]. The selection of agents and the organization of multi-agent architecture are selected based on the studied context. The main elements of the AGRE model are determined as follows:

- *Set of agents:* vehicle, sensor, road segment, sub-network, and network
- *Groups:* groups of the same types of agents that are located at the same level in road network architecture
- *Role:* a representation of the function of a given agent, its service or identification
- *Environment:* including the physical areas and surrounding entities such as other roads, Geographic Information System (GIS), GPS, vehicles, drivers, servers, communication equipment

Based on the AGRE model, [96] introduced a hierarchical organizational architecture of road network, shown in Figure 7.1 (left). This architecture involves three types of agents: city, road

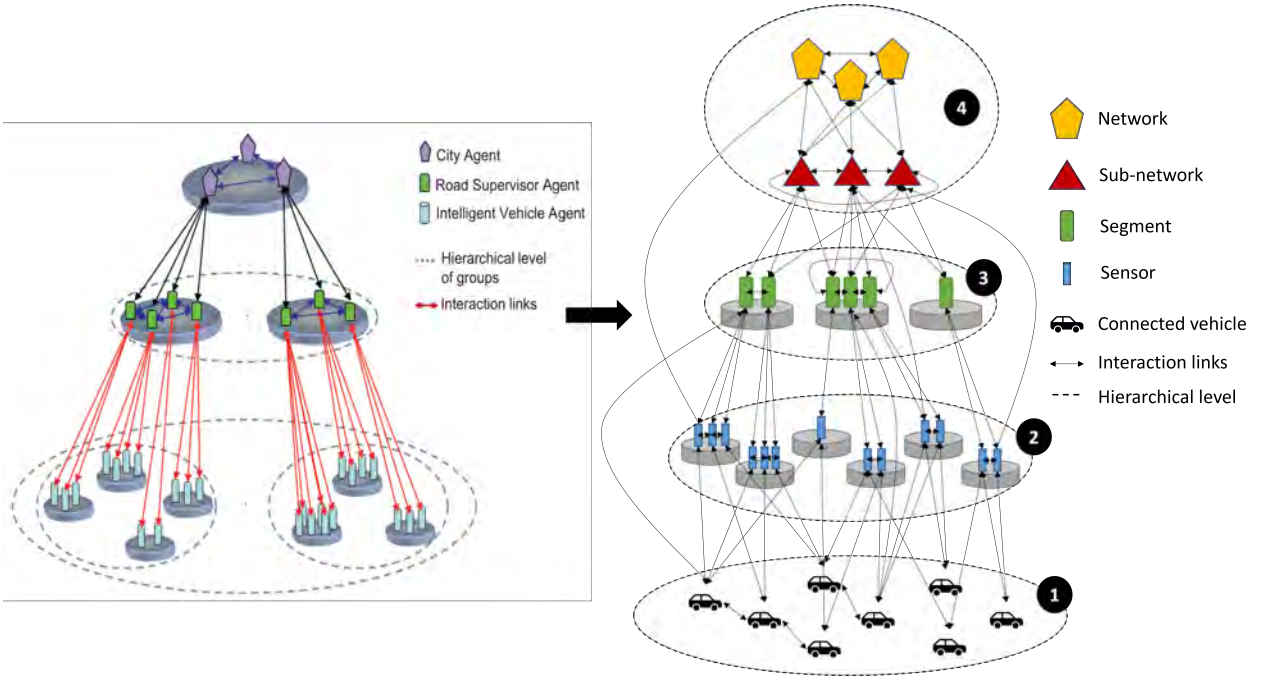


Figure 7.1: Hierarchical multi-agent road network architecture proposed by [96] (left) and its extension for this study (right)

supervisor, and intelligent vehicle agents. Agents at lower levels are grouped together if they are situated within the supervision of the same agent at high levels. The communication between levels is restricted at the adjacent levels, meaning that agents must pass the intermediate levels to communicate with the following ones. This property does not correctly reflect the V2X connectivity in the dynamic environment of ITS. Therefore, we propose an architecture adapted for our study shown in Figure 7.1 (right).

Our architecture presents four key enhancements. Firstly, entities within our architecture can communicate flexibly with others at any level without passing intermediate levels. This adaptation aligns with the real communications in ITS with V2X connectivity. Secondly, it includes fixed sensor entities in this architecture since we are interested in the macroscopic traffic data they collect. Thirdly, sub-network entities are introduced to address road segments with similar traffic characteristics or specific road types. In the last level, the network entity is defined as the highest level, referring to city, region, country, *etc.*

The definition of road network entities is presented from the microscopic to macroscopic levels as follows:

- **Vehicle (level 1):** the entities at the lowest level of the road network architecture. They are equipped with onboard units and V2X connectivity, allowing them to collect the *Floating Car Data (FCD)* with high frequency along their itineraries and share these data with other entities of the road network architecture. Connected vehicles can perform direct communication

between them to exchange data and traffic observation through V2X connectivity. Data collected from connected vehicles nowadays play a significant role in the development of ITS services.

- **Fixed sensor (level 2):** the sensors installed on the road segment enabling the collection of aggregated data from crossing vehicles such as mean speed, number of vehicles, $etc.$ These sensors can refer to physical inductive-loop traffic detector or virtual software in data collection servers that can process data from different sources and extract traffic information. The data collected from them can provide collective traffic information for a higher scale of traffic management than the individual data from vehicles.

- **Segment (level 3):** the logical entity associated with a road segment as defined by Open Street Map (OSM). Segment entities manage the data communicated from vehicles or sensors located on them. Analyzing traffic data at the segment level allows the local understanding of the evolution of traffic dynamics and thus enhances traffic information provided for future vehicles.

- **Sub-network (level 4):** the logical entity that manages traffic dynamics at the level of a given group of segments ($e.g.$ motorways, districts, cities, etc). Studying the traffic dynamics by the type of road segments allows us to understand the inter-impacts between them as well as the propagation of traffic events through the different road types.

- **Network (level 4):** the logical entity linked to the studied geographical road network. Network entities aim to understand traffic dynamics at the network level, which helps with high-level traffic management, for example the dependence of traffic between cities, the impact of a given urban area, $etc.$

The entities presented in road network architecture can communicate with each other to exchange data. Two types of communication can be established at each level. The first type is **inter-level communication** from lower to upper levels. The entities, able to directly collect the traffic data as vehicles or sensors, aim to share them with entities at higher levels to enable multi-scale traffic management. The second is the **intra-level communication**. Indeed, the entities can decide to share the raw observations or process these data before sending them. They can communicate with other entities at the same level to enhance their knowledge and decide together the adequate information to share with entities at the higher level.

7.1.2 Roles of Road Network Entities

With flexible established communication, each road network entity participating in the road network architecture can play two roles:

- **Data provider entity:** the road network entity that collects traffic data and shares their observations with entities at other levels. The recent communication technologies allow road network entities to establish a stable connection, enabling real-time and high-speed data exchange. Therefore, traffic data are transferred as **a data stream**. The nature of the shared traffic data varies depending on the type of data providers. Vehicles and sensors mainly play the role of data provider entities since they directly collect traffic data. Nevertheless, other road network entities can act as data providers when the traffic prediction is estimated at higher levels than theirs.

- **Processing entity:** the road network entity aiming to estimate the traffic predictions at its level. The processing entity interacts with data provider entities to learn historical traffic observations and cooperate to compute the prediction jointly. Road segments, sub-networks, and networks mainly play the role of processing sensors.

Data provider entities can interact with each other to enhance their observations and with processing entities to exchange data to other levels. They are always located at a lower level than processing entities. Each data provider entity communicates with only one processing entity at a time. The established communication can be fixed or dynamically changed. The following scenarios illustrate two communication types between data provider and processing entities.

- **Scenario 1:** connected vehicles as data provider entities and road segments as processing entities: in this scenario, connected vehicles share their FCD with the road segment they are crossing using GPS locations. Consequently, when they move to new road segments, their communication change to the road segment entities associated with those segments. Road segment entities process data from communicating vehicles to compute the prediction of driving behaviors. This scenario exemplifies a dynamic communication between a data provider and processing entities.

- **Scenario 2:** fixed sensors as data provider entities and road segments as processing entities: in this scenario, sensor entities communicate with road segment entities where they are installed. Since sensors are stationary and attached to a specific road segment, the communication between them is static. Road segment entities perceive data from fixed sensor entities and compute the prediction of aggregated traffic information.

7.1.3 Definition of Multi-Level Traffic Prediction Problem

Traffic prediction based on data analytics is defined as the task of estimating the future traffic dynamics of a studied road network based on analyzing historical traffic data. Therefore, to scope our studied traffic prediction problem, the following elements need to be clarified: **input, expected output, characteristics of used traffic data, and environment**.

Linking the context of multi-level traffic defined previously in the two previous sections, we define the multi-level traffic prediction problem considered in our study as follows:

Given **a traffic data stream** $DS = \{DP_{Ts_1}, \ldots, DP_{Ts_t}, \ldots, DP_{Ts_N}\}$ consisting of a sequence of N Data Points (DP) arriving during timestamps $Ts_1, \ldots, Ts_t, \ldots, Ts_N$ **sent from data provider entities**, **processing entities analyze and learn**, at each Ts_i, the perceived DP_{Ts_i} and compute the traffic predictions for future timestamps (up to the required prediction horizon). The provided definition of traffic prediction applies to each combination of data provider and processing entity in the road network architecture.

This problem exhibits the following characteristics:

- Traffic data **continuously** arrive over time and are processed sequentially by processing entities.
- A processing entity **does not have global knowledge** of the entire environment. Indeed, they only perceive the traffic information that is exchanged with them by the corresponding data provider entities, which is essential for their prediction computations. Consequently, traffic data are locally stored and processed. Additionally, the actions of a road network entity only influence the local part of the environment, comprising entities within its neighborhood.
- Local learned information of processing entities can be insufficient for prediction calculation due to the complex traffic propagation in the road network. Therefore, the **cooperation between processing entities** is appropriate for obtaining adequate predictions.
- Due to the diverse types of data provider and processing entities, the interacting environment exhibits a high level of dynamics and openness. Moreover, road network entities can be either physical or logical, having strong evolution over time, such as entering or quitting the system, modifying their behaviors, and adapting to the environment's states. Therefore, **the prediction method must have solid versatility to perform robustly in diverse scenarios**.

7.2 System objectives

The previous section defined the multi-level traffic prediction problem that is studied in this thesis, as well as detailing its significant characteristics. To deal with this problem, we propose **ADRIP (Adaptive multi-agent system for DRIving behaviors Prediction)** with the following expected characteristics:

- **continuous learning** from traffic data streams to evolve the learning and predictive algorithms, databases, and decision-making strategies. This self-evolution allows ADRIP to adapt their behaviors to the changes of the environment.

- **local learning** process to distribute the model training at the agent's level. This distribution enhances the **openness of the system**, where its entities can be added or removed without requiring the re-installation of the entire system. Moreover, the local training process allows the system to decentralize the data storage and collection. In fact, agents in the system only process a part of the studied data set that directly affects them. This property can answer the **data privacy issue** since data do not need to be shared with all entities of the system. Finally, the local learning process can reduce the **calculation time**, leading to real-time processing for traffic data streams.

- **explicit model interpretability** of ADRIP to provide **the explanation of input-output causality**. To overcome the black-box nature of learning models, ADRIP is based on a clustering algorithm that can facilitate the understanding of the dependency of the outputs on the input features or parameters of the models. That leads to the efficient management of the system's performance.

- **cooperative prediction** process to gather the necessary data for the accurate prediction. The local learned database of the system's processing entity lacks features expressing the dependencies of each entity on each other. This cooperation can fulfill the limitations of the local learning process by enabling the **interaction between entities**, allowing them to share processed information to reinforce the quality of prediction estimation.

- **strong model versatility** for various applications. As ADRIP aims to address the multi-level traffic prediction problem with the diversity of road network scenarios, interacting entities, $etc.$, the system must show **strong robustness** to guarantee its efficient performance for different applications.

To reach these objectives, ADRIP must rely on the following characteristics:

- **Self-evolution:** agents in ADRIP need to perform the dynamic and adaptive behaviors to adapt to the environment and perceived data. This characteristic is essential for the continuous learning to update the learned database, the definition of neighborhood, the choice of parameter, $etc.$

- **Self-organization:** within cooperation mechanism, agents in ADRIP must organize themselves to avoid conflicts and make the cooperation operate smoothly. For example, participating entities are willing to put on sleeping mode to give priority to other entities. By guaranteeing this characteristic of agents, ADRIP can obtain the benefits of local and distributed learning process while ensuring the adequate data sharing in the cooperative prediction process.

- **Self-correction:** agents in ADRIP are capable of self-detecting prediction errors and self-correcting them. This property is essential to ensure a robust performance when the environment shows its new behaviors. In such situations, ADRIP integrates these new behaviors

thanks to its continuous learning process. However, the prediction may not be reliable due to the lack of historical information. Thus, agents need to launch the auto-correction mechanism to ensure the quality of the provided prediction.

In the context of ITS, the expected objectives and required characteristics of ADRIP are essential to provide sustainable solutions that are feasible to deploy and apply for real-world traffic issues. As the traffic environment is highly dynamic, the decentralization of ADRIP is expected to enhance its openness, enabling reducing the installation costs when the environment changes in large-scale applications. Additionally, the distributed calculation in ADRIP can offer fast and updated predictions, guaranteeing their availability for drivers.

7.3 System architecture

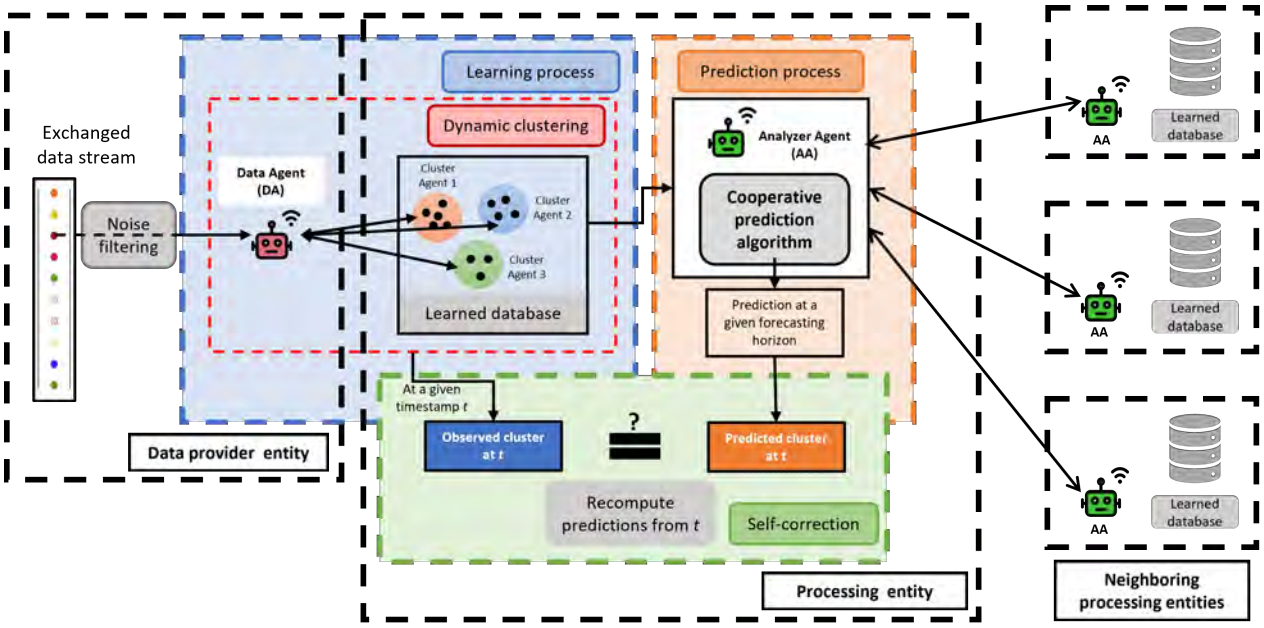


Figure 7.2: ADRIP's architecture

The architecture of ADRIP is shown in Figure 7.2. ADRIP consists of two main processes: **local learning process (L-ADRIP)** and **cooperative prediction process (P-ADRIP)**.

The **learning process** is composed of **a MAS-based adaptive and dynamic clustering system** to locally detect different traffic dynamics from the perceived data stream of a given processing entity. Traffic data are grouped into different clusters, each representing a traffic dynamic. The set of constructed clusters forms a **local learned database** of a processing entity. To perform the dynamic clustering, L-ADRIP defines two types of agents **Data Agent** and **Cluster Agent**.

- **Data Agent (DA)**: a DA represents a Data Point (DP) arriving at timestamp Ts from the communicated data stream DS, where $DS = \{DP_{Ts_1}, \ldots, DP_{Ts_t}, \ldots, DP_{Ts_N}\}$. Each DA is

created when a DP, in the data stream communicated by data provider entity, interacts with the processing entity.

The traffic information contained in DAs can be different depending on the data shared by the data provider entity, for example, the mobility profile if traffic data comes from the vehicles or the mean speed of crossing vehicles during a given time interval if data is sent from fixed sensors. DA also contains the communicating timestamp (Ts).

Each DA actively participates in the learning process of ADRIP, is integrated in the learned database of the processing entity and is subsequently removed from the system.

- **Cluster Agent (CA)**: each CA is a logical entity associated with a cluster representing a traffic dynamic extracted from the communicated data stream by the dynamic clustering method. Each CA is described by a **centroid DP** that is the combination of all historical DPs and a list of **Ranges of Use (RUs)** indicating the moments when the DAs are assigned to that CA. The ranges of use of a CA provide information about historical moments when the traffic dynamics represented by this CA occurred and how long they lasted. Memorizing these ranges allows for the integration of temporal analysis in ADRIP.

L-ADRIP describes the interactions between DAs and existing CAs to **continuously learn traffic data and build the learned database** of the processing entity. The learned database contains a set of CAs. The clustering decisions and learned database are updated at each DA arrival. That enables the system to locally self-adapt to traffic dynamics evolution and study the **temporal dependencies** by observing the transitions between clusters.

Simultaneously with the learning process, the **prediction process P-ADRIP** performs a **cooperative prediction method** that provides traffic prediction estimations until the required prediction horizon. To operate the computation of predictions, P-ADRIP defines the **Analyzer Agent**.

- **Analyzer Agent (AA)**: an AA is associated with a logical entity that is responsible for computing the prediction. AA is capable of analyzing the local learned database from the learning process within the processing entity and cooperating with the AAs of neighboring processing entities to gather the necessary information for prediction computation.

The AA in the processing entity uses the learned database from the dynamic clustering and cooperates with the AA of neighboring road entities to compute the predictions. The definition of the neighborhood of a road network entity depends on applications that aims to study the propagation of traffic dynamics. Given that, the prediction process takes into account the spatial dependencies. The interactions between AAs are implemented based on the principles of self-adaptive MAS with cooperation mechanisms to overcome the potential non-cooperation situations that can happen during the exchange of information.

Additionally, ADRIP also contains the mechanisms: **noise detection** and **self-correction**. **The noise detection mechanism** aims to detect and filter the abnormal DPs from the data stream.

The abnormal DPs are ignored and not considered in the learning and prediction process. This mechanism is applied before L-ADRIP performs the dynamic clustering method. **The self-correction mechanism** aims to manage the quality of computed prediction. Therefore, it observes both the real traffic dynamics from L-ADRIP and the predicted ones from P-ADRIP and interferes when detecting significant prediction errors.

7.3.1 Noise detection mechanism

Before diving into the main steps of the learning process, this section introduces the noise detection method in ADRIP that is developed to eliminate insignificant data. When a new DP arrives, ADRIP first verifies if it contains traffic information describing singular behaviors that do not correctly reflect traffic conditions on the road network. The singular behaviors can be due to several reasons:

- Malfunctions of sensors that can provide unreliable traffic data.
- Environmental factors such as bad weather conditions, construction activities, $etc.$
- Connectivity errors between road network entities
- Vehicles with special privileges ($e.g.$ ambulance, police cars) may exhibit distinct behaviors, causing their Floating Car Data (FCD) to deviate from typical traffic conditions
- Vehicles moving with particular behaviors of drivers

Depending on the application, one or several types of noise must be detected and eliminated. Addressing and mitigating these sources of noise is essential for improving the quality of traffic data and ensuring that the analyses and predictions based on the data are reliable and accurate. Therefore, we construct the noise detection mechanism of ADRIP as **a multi-criteria method** to detect singular behaviors according to each established application. The detection method can be diverse, mainly depending on the expert knowledge of the application scenario. For example, the noise detection mechanism can use the vehicle's information to identify privileged vehicles or compare data with the previous and following ones to detect singular driving behaviors.

If the new data is considered as noise, it is rejected. Otherwise, it is represented by a DA interact with the existing CAs in the learned database.

7.3.2 Learning process

This section introduces the behaviors of DAs and CAs in the dynamic clustering system to detect the clusters of different traffic dynamics and ensure the continuous updates of the learning database of a road network entity at each new DP. All entities perform the same learning behaviors to cluster the perceived DPs. Nonetheless, depending on the stream of perceived DPs, each road network entity possesses a different cluster structure and learned database. This distribution of clustering

mechanism enables several benefits. Firstly, road network entities function simultaneously and independently from each other to **facilitate system control**, **mitigate the potential damage** of a centralized processing node, and **reduce the calculation time** by enabling parallel processing on multiple processors. Secondly, it enhances the **openness and flexibility** of the system since road network entities can be easily added or removed when there are some changes in the network. Thirdly, it **self-adapts to the continuous update** when traffic dynamics change on the road network. For example, when the correlations between road segments evolve, the system only needs to update the relationships between related road segments rather than recomputing for the entire road network.

7.3.2.1 DA's behaviors

The DA interacts with existing CAs in the learned database of the processing entity. It performs three behaviors, including:

1. Searching the list of similar existing CAs $SimCAs$ by asking them for the similarity evaluation
2. Creating a new CA
3. Asking an existing CA for the assignment

Algorithm 1 DataAgent: searching the list of similar existing CAs

1: {—*perceive*—}
2: DP: data point in the data stream
3: Ts: communicating timestamp
4: {—*decide and act*—} *// DA interacts with all existing CAs to find the list of similar CAs* $SimCAs$
5: **for all** CAs in **Learned Database do**
6: askForSimilarityEvaluation(CA,DP)
7: **end for**
8: $SimCAs \longleftarrow$ list of similar CAs containing their identification and the associated similarity measure

Algorithm 1 describes the first behavior of DA for **searching the list of existing similar CAs** $SimCAS$ in the learned database. A new DA enters the system upon the arrival of a data point (DP) in the communicating data stream. DA perceives the traffic data point DP and the communicating timestamp Ts from the data provider entity (*Algo.1, lines 2-3*). Then, DA interacts with existing CAs in the learned database to ask for the evaluation of the similarity between its DP and CAs' centroid (*Algo.1, line 6*). The decision of similarity is made by CAs based on their local decision. Only similar CAs return the similarity measure between their centroid and the DP of this DA, forming the list $SimCAs$.

Algorithm 2 DataAgent ($SimCAs$): creating a new CA

1: {—*decide and act*—} *// DA evaluates* $SimCAs$
2: **if** len($SimCAs$) == 0 *// no similar CA has found* **then**
3: NewCA = createCA(DP, Ts) *// create a new CA*
4: Add NewCA to the learned database
5: **end if**

Algorithm 2 describes the second behavior of DA. Following its interaction with the existing CAs, DA determines $SimCAs$ containing the identification of similar existing CAs and the associated similarity measure between the centroids of these CAs and its DP. Then, DA proceeds with an evaluation of $SimCAs$. If $SimCAs$ is an empty list, signifying **the absence of similar existing CAs**, DA creates a new cluster agent (*Algo.2, line 3*). The centroid of the new CA is the new perceived DP, and the first range of use is the communicating timestamp Ts. Then, the created CA is appended to the learned database of the processing entity.

Algorithm 3 DataAgent: asking an existing CA for the assignment

1: {—*decide and act*—} *// DA evaluates* $SimCAs$
2: **if** len($SimCAs$) ≥ 1 *// At least one similar CA has found* **then**
3: *// DA chooses the most adequate similar CA to assign*
4: $OrderedSimCAs \leftarrow SimCAs$ ordered increasingly based on the similarity measure to the perceived DP
5: $ExpectedCAToAssign$ = $\text{argmin}_d(SimCAs) = OrderedSimCAs$[0]
6: Send the assignment request to $ExpectedCAToAssign$
7: **if** len($SimCAs$) ≥ 2 **then**
8: $SecondClosestCA \leftarrow OrderedSimCAs$[1]
9: Send the $SecondClosestCA$ to $ExpectedCAToAssign$
10: **end if**
11: **end if**

In the case where there is at least one existing CA similar to the DP of DA, DA **requests for the assignment** by following the steps shown in Algorithm 3. Firstly, DA sorts the list $SimCAs$ in ascending order based on the similarity measure to the perceived DP (*Algo.3, line 4*). The resulting ordered list is denoted as $OrderedSimCAs$. Then, DA identifies the most similar cluster as the expected cluster to be assigned to (*Algo.3, line 5*). Additionally, DA checks if the $SimCAs$ contains more than one similar cluster. In that case, DA retrieves the $SecondClosestCA$ (which is the second closest CA) (*Algo.3, line 8*) and sends it to the $ExpectedCAToAssign$ (which is the closest CA) for further processing.

7.3.2.2 CA's behaviors

In the interaction with DAs, an existing CA in the learned database of the processing entity performs the following behaviors:

1. Evaluating the similarity between its centroid and DP of the required DA
2. Merging with another CA
3. Assigning a DA to the learned database

Algorithm 4 ClusterAgent: evaluating the similarity with DA

1: {—*perceive*—}
2: CA receives a request to compute similarity measure from a DA := (DP, Ts)
3: {—*decide and act*—}
4: d = $similarity\_measure$(CA.centroid, DP)
5: **if** d $\leq \alpha$ **then**
6: Send d to DA and CA's identification
7: **else**
8: Ignore the request
9: **end if**

When receiving **the request of DA for the similarity evaluation**, CA performs the algorithm 4. CA first computes the similarity measure between its centroid and DP of the requesting DA (*Algo.4, line 4*). The similarity measure is defined depending on the types of traffic data contained in DA. This measure must evaluate the discrepancy between different traffic conditions. Then, the computed similarity measure d is compared with the **similarity threshold** α that is adapted to the characteristics of CA (*Algo.4, line 5*). The **similarity threshold** α is defined as a value indicating that two traffic data points are considered similar if the distance between them is smaller than it. Therefore, if d is smaller than α, implying that CA is similar to the interacting DA, then CA sends its identification and the similarity measure d to this DA (*Algo.4, line 6*). Otherwise, CA ignores this request (*Algo.4, line 8*).

In case where **CA is the cluster agent that a DA expects to be assigned to**, and it perceives the information of $SecondClosestCA$ from this DA. CA verifies if merging with $SecondClosestCA$ is necessary before assigning the communicating DA. This function is described by the algorithm 5. Firstly, CA computes the similarity measure d between its centroid and the one of $SecondClosestCA$ (*Algo.5, line 4*). If d is smaller than its similarity threshold α, they merge together. The new centroid is computed as the **mean of their centroids** (*Algo.5, line 6*), and the list of ranges of use $listRUs$ is the **aggregation of their list of ranges of use** (*Algo.5, line 7*). This merging step allows CA's behaviors and the learned database to adapt to new arrival data without depending on initial data.

The merging of clusters is a local process that avoids costly calculations. Then, the learned database of the processing entity is updated (*Algo.5, line 9*).

Algorithm 5 ClusterAgent: merging with another CA

1: {—*perceive*—}
2: CA perceive the information of $SecondClosestCA$ of a DA
3: {—*decide and act*—}
4: d $\leftarrow$ $similarity\_measure$(CA.centroid, $SecondClosestCA$.centroid)
5: **if** d $\leq \alpha$ **then**
6: CA.centroid = (CA.centroid + $SecondClosestCA$.centroid)/2
7: CA.listRUs.append($SecondClosestCA$.listRUs)
8: **end if**
9: Update the learned database of processing entity

Algorithm 6 ClusterAgent: assigning a DA to the learned database

1: {—*perceive*—}
2: CA receives an assignment request from a DA := (DP, Ts)
3: {—*decide and act*—}
4: CA.centroid = adjustCentroid(DA.DP,γ)
5: CA.listRUs = updateRUs(DA.Ts)

If CA is the cluster agent that DA expects to be assigned to, it proceeds to integrate the communicating DA in the learned database (cf. Algorithm 6) using data point DP and communicating timestamp Ts contained in this DA. CA integrates the perceived DP (*Algo.6, line 4*) by adjusting its centroid using γ as **an adjustment coefficient** ($cf.$ Equation 7.1). In this function, γ plays a similar role as **learning rate** in the optimization algorithms as in the gradient descent that makes the chosen cluster gradually move towards the new DP. The communicating timestamp of DA is added to the list of ranges of use of CA (*Algo.6, line 5*).

$$centroid_{adjusted} = centroid + \gamma * \text{sign}(DP - centroid) \tag{7.1}$$

7.3.3 Prediction Process

The prediction process P-ADRIP is performed by AAs, aiming to compute the chain of the next changes of traffic dynamics ($i.e.$ next changes of CAs) for a required prediction horizon. Thus, the predicted data is denoted as: $\mathcal{P} = \{(predCA_1, Ts_1) \longrightarrow (predCA_2, Ts_2) \longrightarrow \cdots \longrightarrow (predCA_H, Ts_H)\}$. Ts_i is the predicted timestamp where the traffic on a given road network entity changes to $predCA_i$. $predCA_1, Ts_1$ is predicted using current traffic state. A prediction

$(predCA_i, Ts_i)$ is computed using the previous prediction $(predCA_{i-1}, Ts_{i-1})$. H is the required prediction horizon.

To complete the functioning of an adequate traffic prediction system, P-ADRIP addresses two points: **spatial dependency and real-time update**.

Spatial dependency in the context of traffic dynamics refers to the **influence of the traffic conditions of neighboring road segments** on the behavior of the considered segment. Considering this dependency is crucial for enhancing the accuracy of traffic prediction methods. It enables the learning of historical traffic patterns not only for the specific segment but also for upstream and downstream road segments. Indeed, the incorporation of spatial dependency allows the model to understand how **traffic dynamics propagate through different road segments over time**. By examining historical traffic patterns in neighboring segments, the model can discover the evolution of traffic dynamics between these road segments. To illustrate the spatial dependency, we consider a simple road network with traffic flows in a specific direction, shown in Figure 7.3. In this scenario, the traffic dynamics on a second road segment are significantly influenced by the number of vehicles entering from the first and on-ramp road segments. Analyzing historical inflow traffic patterns provides valuable insights, leading to an improved estimation of future traffic dynamics on the second road segment.

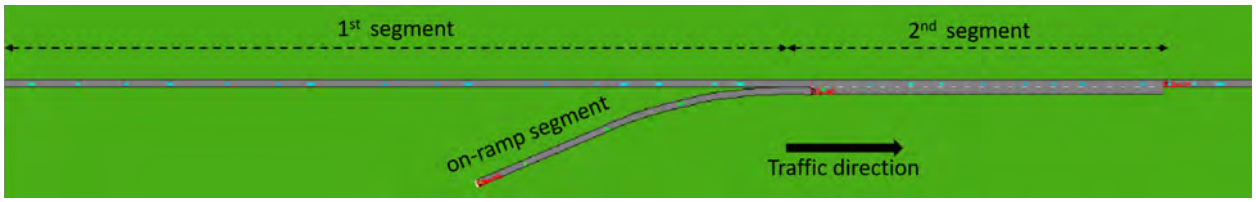


Figure 7.3: Example of a road network

To include **spatial dependency**, AAs are based on the fact that the traffic dynamics on a road network entity are impacted by the traffic of its neighborhood (traffic propagation). Each AA cooperates with the neighboring AAs to collect the historical information required for the estimation of predictions ($cf.$ Algorithm 7). Through these interactions, each AA computes the predictions based on how its traffic dynamics in the past have been influenced by its neighboring AAs. The information of considered AA and its neighbors constitute what we defined as a **local traffic configuration**.

Definition 1. ***Local traffic configuration***

The configuration at a time T under the point of view of an AA is the set of observed traffic dynamic clusters (CAs) with their corresponding ranges of use at T on itself and on its neighboring AAs.

For example, using the road network in Figure 7.3, the local traffic configuration under the point of view of the second road segment is the set of observed traffic dynamic clusters (CAs) with their corresponding ranges of use on itself, the first and on-ramp road segments.

The second characteristic that we aim to integrate into P-ADRIP is **real-time extension**. This

characteristic refers to the **capability of updating traffic predictions dynamically**, ensuring that predictions are always available up to the required time horizon. When the current prediction horizon is shorter than the required one, AAs use the farthest prediction to compute the next one. This extension mechanism is continuous through time. However, as the following prediction is computed based on the previous one, the bad performance of a previous step can propagate and make the prediction accuracy degrade quickly. Thus, the real-time update also includes a self-correction mechanism (presented in Section 7.3.4) using real-time observed data. This ability allows for **the enhancement of the availability and reliability** of the prediction process.

7.3.3.1 AA's behaviors

To proceed the computation of prediction, AA performs two main behaviors:

1. Prediction estimation
2. Solving the non-cooperation situations

Algorithm 7 AnalyzerAgent: Prediction estimation

1: $currentTs \longleftarrow$ Current timestamp of prediction process
2: Ts $\longleftarrow currentTs$
3: **while** Ts $< currentTs$ + H **do**
4: $Config_{Ts} \longleftarrow$ **buildConfig**(AA, neighAAs,Ts)
5: listRUs $\longleftarrow$ **getRUs**(LearnedDatabase.CA_{Ts})
6: histConfigs = []
7: **for** RU $\in$ listRUs **do**
8: histConfigs.add(**buildConfig**(AA, neighAAs, RU.start))
9: **end for**
10: mostSimConfig $\longleftarrow$ **evalutateConfigSim**($Config_{Ts}$, histConfigs)
11: $predCA$, $RU\_predCA \longleftarrow$ **getFollowingCA**(mostSimConfig)
12: Ts $\longleftarrow$ Ts + RU_predCA
13: **end while**

Algorithm 7 details the behaviors of an AA to predict traffic dynamics. It starts from the current timestamp $currentTs$ (the timestamp when the prediction is launched) with the observed configuration until the desired prediction horizon H. The main principle is, given a local traffic configuration constituted by the traffic dynamics on road network entity associated with the considered AA and its neighbors at a given timestamp, to find **the most similar traffic configuration** in the past denoted **historical configuration**. The prediction of AA is then defined as **the succeeding traffic dynamics** that were observed after this historical configuration. This prediction lasts as long as the selected CA lasted in the past (RU_{predCA}).

At each timestamp Ts of P-ADRIP (*Algo.7, line 3*), AA builds (*buildConfig()*) the configuration $Config_{Ts}$ from its CA and by requesting the observations from its neighbors (*Algo.7, line 4*). Then, AA extracts the ranges of use ($listRUs$) of the current CA at Ts (*getRUs()*) (*Algo.7, line 5*) and constructs, for each range of use (*Algo.7, line 7*), the historical configuration containing the CAs of its neighbors at the beginning of this range of use (*Algo.7, line 8*). AA then compares each historical configuration with $Config_{Ts}$ (*evalutateConfigSim()*) based on two ordered criteria:

- the number of neighboring entities with different CAs in both configurations to address the difference in traffic dynamics;
- the time gap between the beginnings of ranges of use in both configurations to address **the difference in dynamic propagation time**. This gap is computed as the difference between the beginning of the CA of neighboring processing entities and the beginning of the CA of the considered processing entity. Indeed, since the change of traffic dynamics on a road segment or an area can result from the changes of traffic dynamics in its neighbors, this criterion aims to evaluate the difference between the two configurations regarding the time of dynamics propagation. Figure 7.4 illustrates the calculation of this criterion.

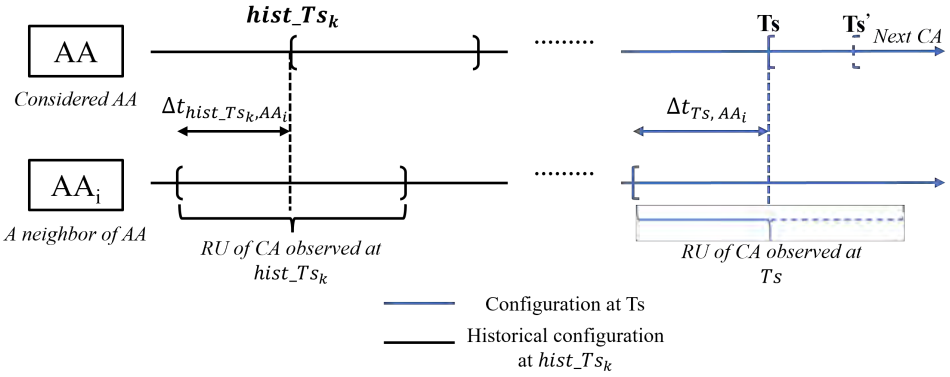


Figure 7.4: Illustration of the calculation of the time gap criterion

For each AA_i belonging to the neighborhood of a given AA, $\Delta t_{Ts,AA_i}$ is the time distance between the beginning of CA observed at Ts on road segment entity associated with AA, and the beginning of the CA observed at Ts on road segment entity associated with AA_i. Similarly, $\Delta t_{hist_{Ts_k},AA_i}$ is computed in the same way with the CAs observed at a historical range of use ($hist\_Ts_k$ that is the beginning of a historical range of use). Thus, the second criterion is expressed as follows:

$$c_2(Config_{Ts}, Config_{hist_{Ts_k}}) = \sum_{i=1}^{M} |\Delta t_{Ts,AA_i} - \Delta t_{hist_{Ts_k},AA_i}| \tag{7.2}$$

where M is the number of neighboring AAs.

We consider a concrete example: an AA_1 who has 2 neighbors AA_2 and AA_3. At Ts, the observed CAs on AA_2 and AA_3 are respectively CA_2^{Ts} and CA_3^{Ts}. The time distance between the beginning of those two CAs with Ts is respectively $\Delta t_{Ts,AA_2}$ = 2 minutes and $\Delta t_{Ts,AA_3}$ = 3 minutes. Now, we consider a historical configuration at $hist_{Ts_k}$ where the observed CAs on AA_2 and AA_3 are respectively $CA_2^{hist_{Ts_k}}$ and $CA_3^{hist_{Ts_k}}$. The time distance between the beginning of those two CAs with $hist_{Ts_k}$ are respectively $\Delta t_{hist_{Ts_k},AA_2}$ = 5 minutes and $\Delta t_{hist_{Ts_k},AA_3}$ = 7 minutes. Consequently, the time gap criterion c_2 is equal to:

$$\begin{aligned} c_2(Config_{Ts}, Config_{hist_{Ts_k}}) &= |\Delta t_{Ts,AA_2} - \Delta t_{hist_{Ts_k},AA_2}| + |\Delta t_{Ts,AA_3} - \Delta t_{hist_{Ts_k},AA_3}| \\ &= |2-5| + |3-7| = 7\ minutes \end{aligned}$$

The most similar historical configuration ($mostSimConfig$) is then defined as the historical configuration minimizing both criteria. Then, AA gets the next CA and its ranges of use of this configuration as the prediction for the next minutes from Ts. AA computes the achieved horizon of this prediction step by forwardly shifting Ts for a time interval equal to the ranges of use of predicted CA. If the desired horizon H is not reached yet, all described steps are repeated.

During interactions between AAs, several **non-cooperation situations** can happen and disturb the functioning of AAs. That requires AAs to **self-adapt**. Thus, a set of **resolutions** allowing AAs to overcome those situations and maintain the adequate functioning of the system has been defined. They are presented as follows: the description of the cooperation failure situation and the cooperative behaviors for the resolution.

- **Incompetence in Responding to Requests**

  Failure description: During the construction of traffic configuration, AA faces an issue where it does not receive responses from its neighbors for their demands of observed CAs. As a result, AA is unable to establish the complete configuration. This situation happens since some neighbors have not yet estimated their prediction at the requested timestamp, preventing them from sending this information to demanding AA.

  Resolution: AAs with shorter prediction horizons launch their calculation first. The remaining AAs wait until the required predictions are computed. Through this mechanism, AAs self-organize their execution based on different situations.

- **Unproductivity in Predictive Calculation with New Information**

  Failure description: In cases where the communicated DA contains a DP that has not been observed yet, AA does not have the historical information necessary for the prediction process.

  Resolution: To address this lack of information, AA uses the newly created CA for this DP as the prediction until the required prediction horizon. When AA detects a significant difference between the centroid of predicted CA and the observed DP, it activates the *self-correction*

mechanism to correct the prediction. Noting that, in parallel, the learning process will cluster this new DP into the learned database, allowing the prediction process to perform better in the future.

- **Ambiguity in Multiple Prediction Propositions**

  Failure description: When several historical configurations are similar to the configuration at time Ts, AA must select the most accurate one for its prediction.

  Resolution: If the configurations are considered equivalent, AA opts for the most recent one. This choice is coherent with the objective of continuous learning, emphasizing the significance of recent changes in the driving environment.

7.3.4 Self-Correction Mechanism

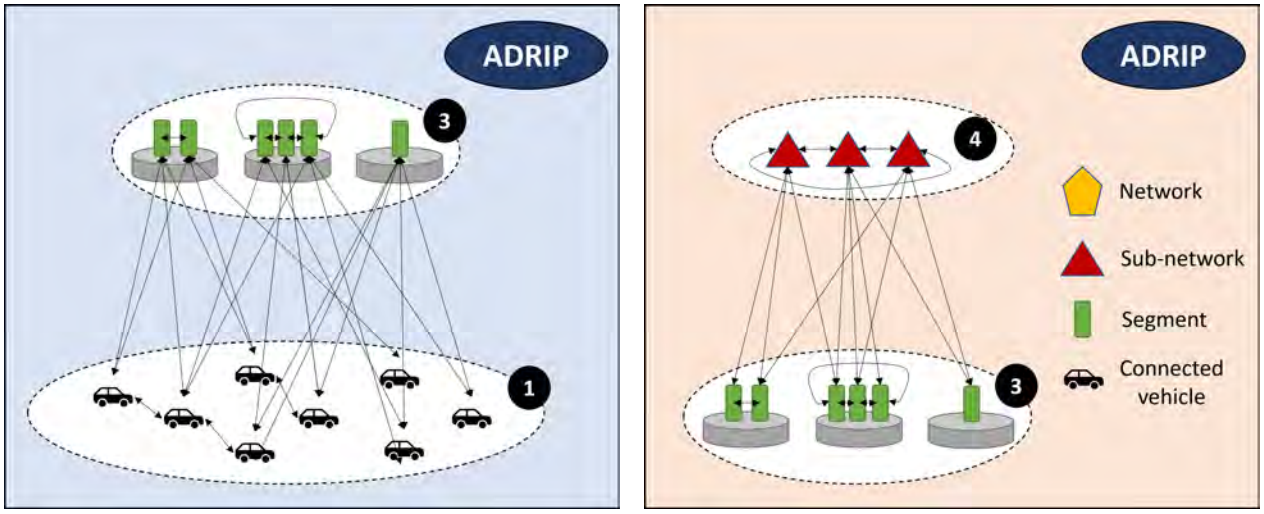
The goal of the self-correction mechanism is to detect when the predicted CA is different from the observed CA and correct it. To do that, when a change in traffic dynamics is detected, AA compares the perceived DP with the centroid of the predicted CA. If the difference between them is larger than the similarity threshold (α), AA relaunches the prediction process using its current configuration. The local self-correction mechanism at an AA level can lead to changes in traffic prediction on its neighboring AAs as they build their traffic configuration using incorrect information. Thus, once the self-correction mechanism of AA detects an error, AA informs its neighboring AA to verify its predictions. This process can propagate to their neighboring AAs for the same reason. This mechanism allows to reduce the degradation of prediction performance due to the dependence of following predicted steps on previous ones.

7.4 Genericity of ADRIP Functioning

ADRIP is presented in the previous section as **a generic solution** capable of addressing traffic prediction problems at different levels of the studied road network architecture, described by the exchange of traffic data from data provider entities to processing entities and the prediction estimation at processing entities.

The application of ADRIP for traffic prediction problems in the multi-level road network architecture is illustrated by the following scenarios. In the first scenario shown in 7.5(a), connected vehicles at level 1 are data provider entities, and road segments at level 3 are processing entities. In the second scenario shown in 7.5(b), road segments at level 3 are data provider entities and sub-networks at level 4 are processing entities. In both scenarios, ADRIP is able to manage traffic data streams sent from data provider entities and perform the prediction computation at processing entities.

In the experiments, the applications of ADRIP need to specify some elements that adapt to the considered scenarios. The first element is **the characteristics of traffic data streams** used as inputs



(a) Scenario 1: ADRIP applied at level 1 and 3

(b) Scenario 2: ADRIP applied at level 1 and 2

Figure 7.5: Two examples of ADRIP functioning

and outputs. The representation of traffic dynamics at different road network levels can be various. For example, traffic data collected from vehicles offer microscopic information such as individual driving profiles. Meanwhile, aggregated data collected from sensors can provide macroscopic traffic information such as traffic flow or density. The diversity of traffic data necessitates considering distinct **similarity measures and thresholds**. The choice of these elements must ensure their representation for different traffic states.

7.5 Conclusions

This chapter presented ADRIP, a MAS-based system for **continuous learning and cooperative prediction**, addressing the multi-level traffic prediction problem. By applying ADRIP, the road network entity is able to study traffic data streams, continuously extract historical patterns, and compute the estimates of future traffic states. This enables the prediction of traffic at different traffic levels in real-time and with continuous updates. Leveraging this prediction, many intelligent services can be developed to improve transportation quality regarding time-saving, efficiency, and safety, leading to a better driving experience for drivers.

ADRIP is highlighted by the following characteristics: **decentralization, distribution, and cooperation**. The **control and decision are decentralized** at three types of agents: DA, CA, and AA. Therefore, it allows the system to perform in an open and dynamic environment, such as a road network, where the entities can be added or removed without re-initialization. The **learning process is distributed**, enabling the local training process on each agent with partial data sets that are interesting for this agent. That avoids the centralization of data collection, leading to

improve data privacy, fasten calculation time and increase the efficiency of data management. Finally, the **cooperation behaviors are embedded in the functions of agents**, allowing to complete the functioning of ADRIP by leveraging their knowledge for the joint decision-making strategy and adaptation.

Chapter 8

ADRIP Instantiations and Evaluations

Objectives of this chapter:

- Underlying ADRIP generality by its instantiation on two study cases:
  1. Traffic prediction for microscopic information level
  2. Traffic prediction for macroscopic information level
- Testing and validating ADRIP instantiation on simulated and real-world data scenarios
- Comparing the results obtained by ADRIP and the state-of-the-art methods

---

The previous chapter presented a generic description of ADRIP for the multi-level traffic prediction problem. This description contains the main processes, agents' behaviors, the generic function of ADRIP, and how ADRIP addresses the multi-level traffic prediction problem.

In this chapter, the experiment presents two instantiations of ADRIP: **traffic prediction for microscopic information level** and **traffic prediction for macroscopic information level**. For each instantiation, we detail the instantiation of ADRIP, including road network entities that play the role of the data provider and processing entities, the definition of traffic dynamics, similarity definition, and adapted similarity threshold. The obtained results of each instantiation for dynamic clustering and cooperative prediction are analyzed and compared to the well-known baselines of related domains. The experiments in this evaluation are conducted in a single iteration since the decisions of the agents are deterministic that leads the same results obtained from the ADRIP when using the same database.

8.1 Traffic Prediction for Microscopic Information Level

This instantiation applies ADRIP to compute the traffic prediction at road segment entities from the stream of data points communicated by connected vehicles, as shown in Figure 8.1.

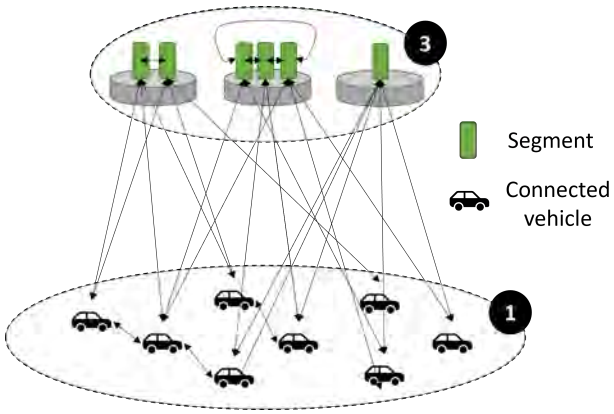


Figure 8.1: Scenario of traffic prediction for microscopic information level instantiation

Therefore, data provider and processing entities are defined as:

- **Data provider entities:** a set of vehicles $V = v_1; v_2; ...; v_n$. Each vehicle follows an itinerary I segmented into a sequence of road segments noted $I = \{rds_1, \ldots, rds_d\}$.
- **Processing entities:** a set of road segments determined according to the road network in Open Street Map (OSM), their starting and ending points located by Global Positioning System (GPS) devices.

Connected vehicles divide their itinerary according to the road segments of OSM. Thanks to GPS equipment, each connected vehicle can locate the entry and exit positions of road segments appearing in its itinerary. At the entry of each road segment (T_{entry}), the connected vehicle starts collecting traffic data. At the exit (T_{exit}), it send these data to the road segment entity it crossed. The entry-exit time interval $[T_{entry}; T_{exit}]$ is also exchanged.

Each road segment entity receives the traffic data stream from the crossing connected vehicle entities that is denoted as:

$DS = \{DP_{Ts_1}, \ldots, DP_{Ts_t}, \ldots, DP_{Ts_N}\}$

consisting of a sequence of N data points arriving during timestamps $Ts_1, \ldots, Ts_t, \ldots, Ts_N$ sent from crossing connected vehicle entities. It is important to note that the set of communicating timestamps of DPs from DS on each road segment is irregular depending on the crossing time of vehicles.

8.1.1 ADRIP Instantiation

From the generic description of ADRIP, we need to instantiate the following elements:

1. **For Data Agent:** the representation of traffic dynamics

2. **For Cluster Agent:** the components of a cluster, the similarity measure of studied traffic dynamics, the definition of similar traffic dynamics and the similarity threshold of associated road segment entity
3. **For Analyzer Agent:** the definition of AA's neighborhood
4. **For Noise detection method:** the method using to detect and eliminate the noise. This element is optional, depending on the expert knowledge about the application scenario

8.1.1.1 Instantiation of Data Agent

Traffic dynamics contain at Data Agent level is defined by:

- An entry-exit time interval $[T_{entry}; T_{exit}]$
- A DP represented by the **Mobility Profile** of a vehicle crossing a road segment entity

  Definition 2. ***Mobility Profile (MP)***.

  The MP of a vehicle on a road segment is the travel time distribution on different speed ranges.

  When a connected vehicle enters a road segment at T_{entry}, it starts collecting its speed. At the exit of road segment at T_{exit}, it computes the ***Mobility Profile (MP)*** which is defined as a representation for traffic dynamic. Previous studies define traffic dynamics as the mean speed, volume, or density of traffic. However, these macroscopic traffic parameters cannot express the variation of vehicle information on a road segment over time. For example, the predicted mean speed at 30km/h can refer to both cases: a constant speed of vehicles on a considered road segment or a homogeneous change of vehicle speeds from 0km/h to 60km/h. Therefore, we aims to introduce MP to provide drivers with the more detailed information.

  From the graphical point of view (Fig.8.2), MP is an histogram whose element is the travel time during which vehicle speeds are within the associated speed range. The speed ranges are sorted in increasing order. Figure 8.2 illustrates the MP of a vehicle crossing a segment of 72m length with the maximum speed of 30km/h using 7 speed ranges.

  Given a MP, perspicuous information such as total travel time, mean speed, or speed variation, can be communicated to drivers. Additionally, compared to the time series of speeds, a MP is more succinct to adapt to memory and calculation time restrictions for continuous learning and real-time prediction.

8.1.1.2 Instantiation of Cluster Agent

Each CA contains a **known MP** ($i.e.$, the centroid of the cluster) that is the combination of all historically assigned MP to this cluster, and a list of **Range of Use** associated to this known MP defined as follows.

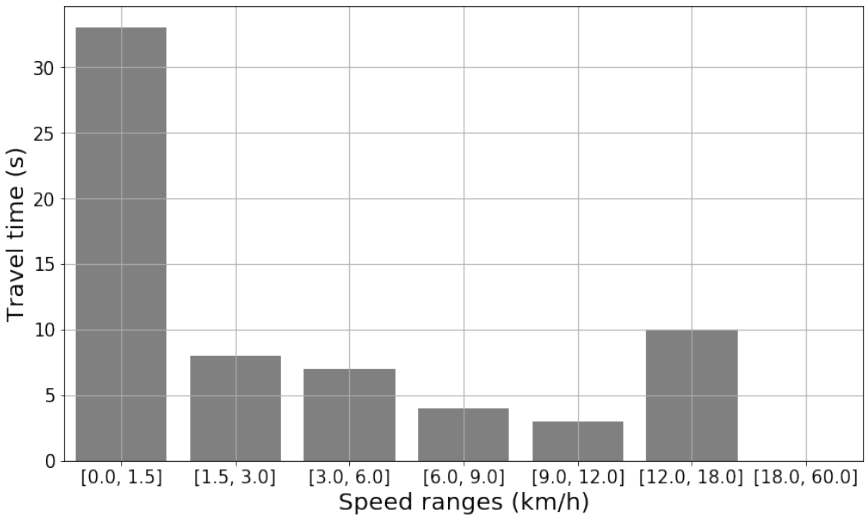


Figure 8.2: Illustration of MP

Definition 3. ***Range of Use (RU)*** *In a given time interval and a traffic dynamic, vehicles follow their previous one, leading to drive with similar MPs. Thus, the range of use of an MP is a time interval* $[T_{start}; T_{end}]$ *during which consecutive vehicles compute similar MPs.* T_{start} *refer respectively to the first vehicle's entry time of that fleet, and* T_{end} *is the entry time of a vehicle that starts communicating a different MP. Each known MP can have many ranges of use, and each range of use indicates the time intervals where consecutive vehicles compute similar MPs to this known MP.*

As an example, consider a connected vehicle entering a road segment at $T_{entry} = 10am$ *and communicating its MP. This MP is then assigned to a CA of this road segment entity. Following connected vehicles crossing this road segment communicate their MPs, all of which are assigned to the same CA as the first connected vehicle, continuing until a vehicle enters at* $T_{entry} = 10:30am$ *and communicates an MP belonging to another CA. Consequently, the range of use of this CA is determined as* $[T_{start}; T_{end}] = [10am; 10:30am]$.

Each CA possesses a **similarity threshold** α to evaluate the similarity between its centroid and an arriving MP. The value of α is pre-defined or computed based on the characteristics of the associated road network entity where it is located such as maximum speed or length. Therefore, it is locally adapted for each road segment entity, according to its length or its maximum speed.

To enable the calculation of MP difference and the evaluation of MP similarity, we introduce the definition of MP similarity ($Def.4$).

Definition 4. *The* ***difference between two MPs (*** MP^k, MP^l***)*** *is an array whose elements are the absolute differences in time travel of respective speed ranges of two MPs. We formulate the expression of MP difference as follows:*

$$MPDiff(MP^k, MP^l) = (|MP_i^k - MP_i^l|)_{i=1,...N} \tag{8.1}$$

where N is the number of speed ranges, MP_i^k *and* MP_i^l *are the values of time travel corresponding to the* i^{th} *speed range.*

Two MPs are similar if all elements of $MPDiff$ ($Eq.8.1$) are smaller than the similarity threshold α. Otherwise they are different.

$$MP^k, MP^l \text{ are similar} \iff \forall i \in \{1, \ldots, N\} : \quad MPDiff(MP^k, MP^l)_i \leq \alpha \tag{8.2}$$

8.1.1.3 Instantiaion of Analyzer Agent

In ADRIP, AA operates the prediction process. To enable the cooperation in the prediction process, each AA must determine its **neighborhood** comprising the AAs with whom it interacts to exchange information necessary for prediction calculation.

In this instantiation, the neighborhood of an AA is defined as the AAs of the direct upstream and downstream road segments. To identify these road segments, we observe two intersections at the entry and exit points of road segment associated with a given AA. Then, all road segments connecting with those intersections are gathered and define the neighborhood of a given AA.

8.1.1.4 Instantiation of Noise Detection

Using trajectory data of vehicles as representative of traffic dynamics requires consistency. Indeed, we can only deduce the dynamics of traffic from the time series of vehicle speeds when vehicle's behaviors **correctly reflect what happens on road segments (no-outlier existence assumption)**. This required consistency is not always guaranteed, for example, emergency vehicles moving with **particular priorities** or vehicles moving with **individual behaviors in free-flow traffic**. In such cases, the learning process is disturbed since it may consider such behaviors as new clusters of traffic dynamics, whereas these discrepancies are due to the diversity of individual vehicle behaviors.

In this instantiation, we explore detecting singular behaviors in free-flow traffic. For that, we compare the distance between the position of the vehicle communicating MP and its leading vehicle to the **Stopping Sight Distance (SSD)**.

Definition 5. ***Stopping Sight Distance (SSD) [56]***

SSD is the minimum distance required on a roadway to enable a vehicle traveling at or near the design speed to stop before reaching a stationary object in its path.

SSD differs from different road properties. Report in [56] analyzes the values of SSD according to the design speed of road segment and braking coefficient. [108] resumes the obtained results reported in Table 8.3.

If the distance between the position of the vehicle communicating MP and its leading vehicle is greater than the SSD of the road segment, that vehicle may have **singular behavior** since it is not constrained by the security restrictions, and the new perceived MP is thus considered as noise. At the system initialization, this noise detection method is not applied on the first vehicle to avoid considering all vehicles as noise on a road segment with few crossing vehicles.

Speed	Stopping Sight Distance, (m)		Typical Emergency Stopping Distance, (m)	
Design Speed (km/h)	Calculated (2.5^s, a=3.4 m/sec^2)	Design (2.5^s, a)	Wet Pave. (1^s, f_{wet})	Dry Pave. (1^s, f_{dry})
30	31.2	35	17.1	14.2
40	46.2	50	27.7	21.6
50	63.5	65	42.0	30.3
60	83.0	85	59.6	40.3
70	104.9	105	81.7	51.6
80	129.0	130	106.1	64.2
90	155.5	160	131.2	78.1
100	184.2	185	163.4	93.4
110	215.3	220	200.6	110.0
120	248.6	250	235.7	127.9

Figure 8.3: Design Stopping Sight Distances and Typical Emergency Stopping Distances [108]

The effectiveness of ADRIP relies on the quality of its learning and prediction processes. Therefore, to evaluate ADRIP's performance, the following evaluation plan consists of two parts: a **Learning Process Evaluation** to assess the clustering quality for traffic dynamics detection and a **Prediction Process Evaluation** to analyze prediction accuracy.

8.1.2 Learning Process Evaluation

8.1.2.1 Data Description

To assess the effectiveness of the proposed dynamic clustering method, we aim to perform it on a data set containing a variety of traffic dynamics. Therefore, the selected scenario consists of two main road segments, with lengths of 1000m and 500m and the maximum speed at 90km/h, respectively. The traffic flow is influenced by a straightforward bottleneck created by an on-ramp segment, as illustrated in Figure 8.4.

Floating Car Data (FCD) are generated using microscopic traffic simulation in MovSim (Multi-model open-source vehicular-traffic Simulator) [176] in which vehicles follow Intelligent Driver Model (IDM) for moving behaviors and Adaptive Cruise Control (ACC) models for lane change behaviors. This simulation generates traffic for about five hours. The entering traffic flow for the main road increases from 1000 vehicles/hour to 1500 vehicles/hour from t_0 to $t_0 + 300s$ and maintains a flow of 1500 vehicles/hour until the end of the simulation. The entering on-ramp flow increases from 0 to 500 vehicles/hour between $[t_0; t_0 + 300s]$, decreases to 300 vehicles/hour between $[t_0 + 300s; t_0 + 600s]$, and maintains this flow for the rest of the simulation. This simulation is run twice. Trajectory data are collected from 10000 vehicles that cross both road segments with

the frequency of speed measurement at 1Hz.

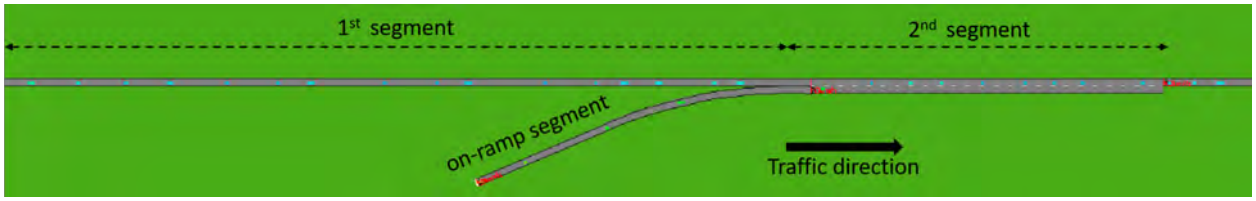


Figure 8.4: Road scenario of data generation using MovSim

8.1.2.2 Experimental Settings

To conduct the experiment, we introduce methods that are compared with ADRIP, detail their parameters and select the evaluation metric.

1. ***Compared Methods and Parameters***

   The performance of dynamic clustering in ADRIP are compared with CluStream using the dataset generated by MovSim. The parameters in ADRIP and CluStream are set as follows:

   - For ADRIP, we base the value's choice of the similarity threshold α on the length of the road segment. In this case, we set $\alpha = 10$ for the 1^{st} road segment and $\alpha = 5$ for the 2^{nd} road segment since the 1^{st} road segment is twice longer than the 2^{nd} road segment. The values of α are selected by experimentation to obtain a reasonable number of clusters on each road segment. The choice of different values of similarity threshold shows the benefit of distributed learning, which is the adaptation of model parameters for each agent.
   - For CluStream, we use its implementation in Python [3]. The maximum numbers of micro-clusters and macro-clusters are respectively set at 50 and 10 as defaults. The time horizon is equal to 10000 (corresponding to the size of used data), and other parameters are set as defaults. The time horizon is divided into 20 frames. Each frame contains 500 MPs.

2. ***Evaluation Metric***

   To measure the effectiveness of clustering algorithms, several measures can be used [139], we focus now on the silhouette coefficient [154] as it indicates if each point is associated with the adequate cluster. The expression of silhouette for one data point i is given as:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \tag{8.3}$$

   where $a(i)$ presents the average distance between i and the other points of its cluster, $b(i)$ presents the average distance between i and the points of the closest cluster to $i's$ cluster. The

silhouette score of a clustering method is obtained by considering the average silhouettes of all data points. The silhouette's value ranges from -1 to 1. This coefficient is interpreted as follows: the higher the silhouette value is, the better the classification is; *i.e.*, the data points are assigned to the best cluster and poorly match other clusters.

To evaluate clustering quality for data stream, the silhouette coefficient of clusters is computed over different frames. The obtained clusters over a frame show the complete performance of an algorithm.

8.1.2.3 Results and Analysis

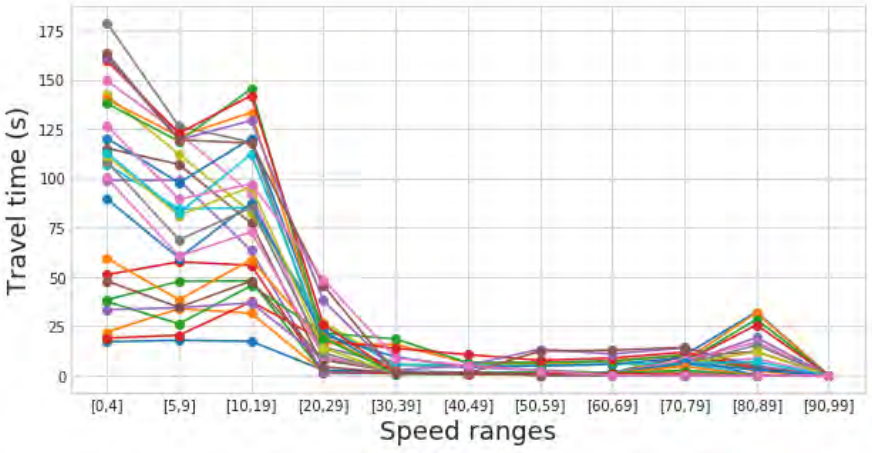
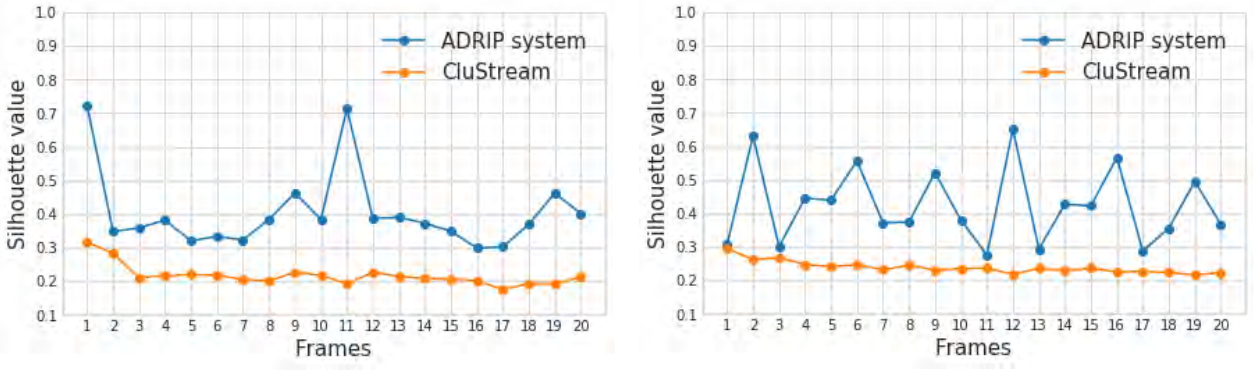


Figure 8.5: The set of known MPs representing the CAs in the 1^{st} road segment obtained by ADRIP after perceiving the 1000 first MPs

Figure 8.5 shows 27 known MPs representing the CAs in the 1^{st} road segment after clustering the 1000 first perceived MPs. Thanks to the clustering mechanism, ADRIP detects a reasonable number of representative traffic dynamics from large amount of communicated data, enabling an efficient data storage.

Figure 8.6 shows the silhouette values over 20 frames on the two main road segments of the selected scenario. On the 1^{st} road segment ($Fig.8.6(a)$), the means of silhouette scores of CluStream and ADRIP over 20 frames are respectively 0.22 and 0.40 (improvement of 82% compared to the silhouette value of CluStream). On the 2^{nd} road segment ($Fig8.6(b)$), the mean of silhouette scores over 20 frames of ADRIP is 0.42 against 0.24 of CluStream (improvement of 75% compared to the silhouette value of CluStream). For both studied road segments, we can observe that the silhouette scores of CluStream are low due to the non-adaptive property (which fixes the number of clusters). Meanwhile, our algorithm's silhouette scores are high but sometimes drop to values worse than the silhouette score of CluStream. This phenomenon is due to the existence of some close clusters that have not been detected to merge yet or the arriving data contain new behaviors. Once ADRIP detects this issue, it modifies the clustering structure (by merging close clusters or creating a new

cluster), the silhouette scores are thus recovered in the following frames.



(a) 1^{st} road segment

(b) 2^{nd} road segment

Figure 8.6: Silhouette values over 20 frames of ADRIP and CluStream

The obtained results from the comparison between ADRIP and CluStream show the efficiency of the dynamic clustering of ADRIP based on the silhouette coefficient using generated data from MovSim within a simple traffic scenario. ADRIP consistently demonstrates higher silhouette values than CluStream across all frames and shows the capacity of dynamic adaptation to new behaviors introduced by incoming data. This observation highlights the efficient performance of ADRIP's dynamic clustering method in handling evolving datasets.

8.1.3 Prediction Process Evaluation

8.1.3.1 Data description

To evaluate the cooperative prediction method, we must extend our analysis to a larger and real road network to account for spatial dependencies between adjacent road segments. However, MovSim does not support the import of real road networks for the simulation. Therefore, for this evaluation, traffic data are generated by the GAMA platform (GIS Agent-based Modeling Architecture) [72].

GAMA is a simulation platform widely used for building explicit agent-based simulations, including traffic simulation with the illustration of many interactions between agents (vehicle, road, infrastructure, people, *etc.*). GAMA allows simulations using real road networks by importing external road networks from shape files or OSM files.

In this simulation, we chose a more extensive road network than the previous one. The selected road network scenario is located on the campus of the University of Toulouse III - Paul Sabatier. Indeed, the campus of the University of Toulouse III - Paul Sabatier can be considered as a *"small city"* as it includes:

- Diverse road types: roundabouts, intersections, pedestrian crossings, car parks, barriers.

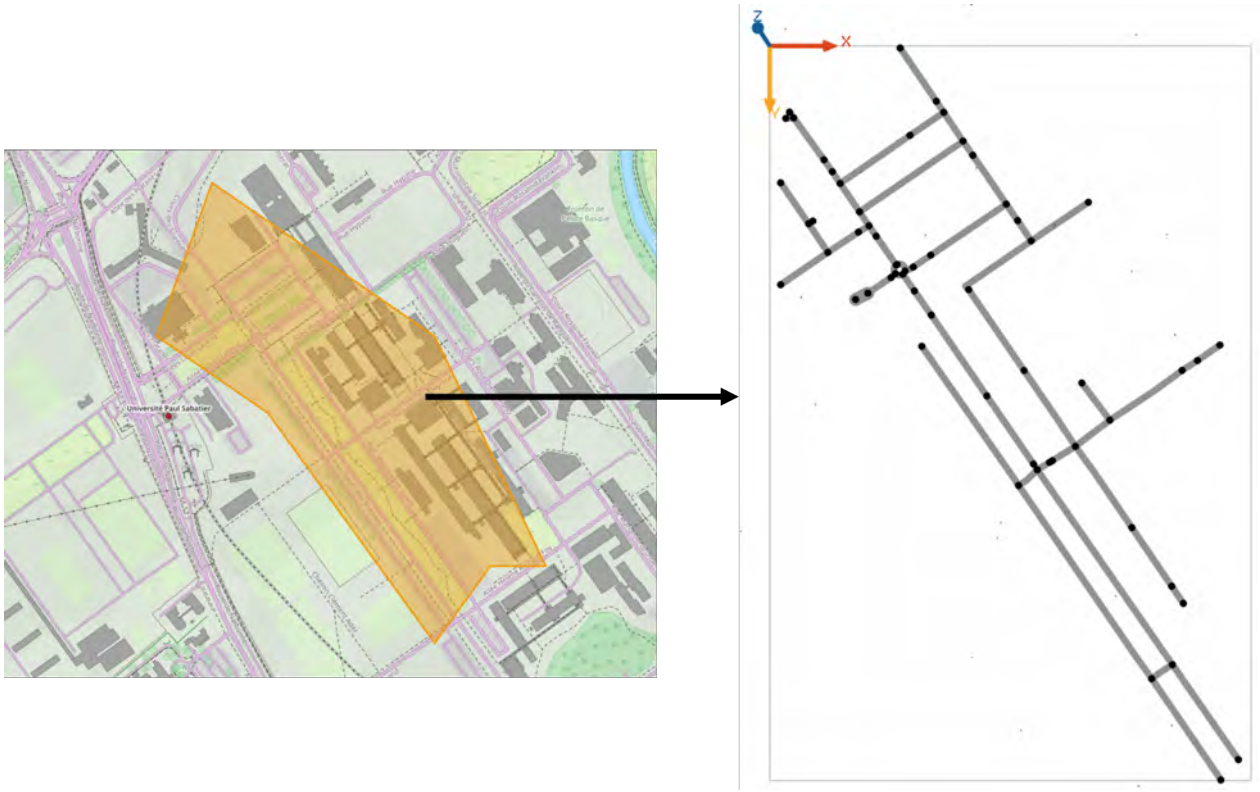


Figure 8.7: The scenario from OSM (left) and the projection of chosen zone in GAMA

- Multi-modal mobility: buses, cars, motorbikes, bicycles, gyro vehicles, pedestrians, metro, cable car.
- Huge number of users: students, staff, visitors, *etc.*
- Infrastructures: building, services, road infrastructures;
- Private territory with two zones: free access, controlled access, authorization, security

In this campus, the platform autoCampus is deployed as a field of experimentation and innovation that will help define the campus of the future with new modes of autonomous, intelligent, and sustainable mobility.

We selected 63 road segments (*Fig* 8.7) with various lengths, including roundabouts and different intersection types. However, many road segments among them are divided by the crosswalks, making them very short and expressing the low variation of traffic dynamics (*Fig* 8.8). As the low variation is also studied on longer road segments, we decided to eliminate those short road segments. Thus, 30 remaining segments were used for this comparison divided into two sets: (1) 9 road segments with high diversity of traffic dynamics mainly located at the main entries and exits of the considered area crossed by many vehicles, (2) 21 road segments with low variation of

traffic dynamics. Figure 8.9 shows the boxplots and standard deviations of travel time on the 30 studied road segments. Roads framed by a dotted line express high variation of traffic dynamics.

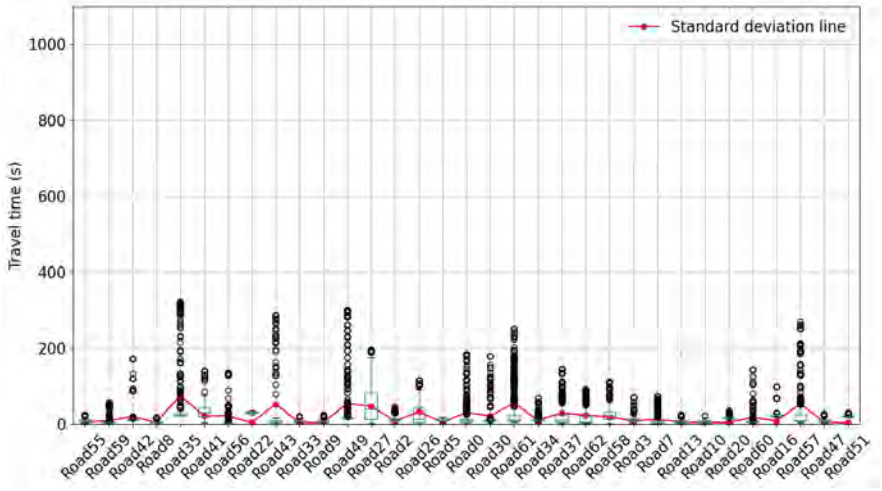


Figure 8.8: Boxplots and standard deviation of travel time on the short road segments

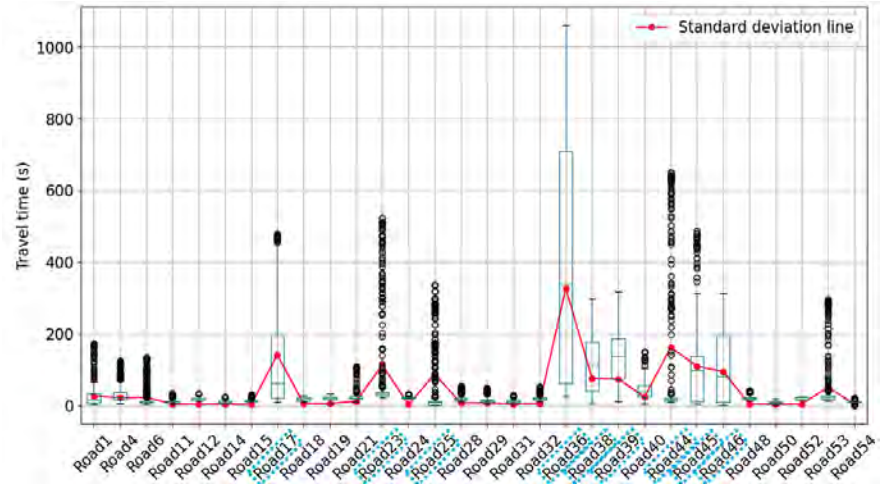


Figure 8.9: Boxplots and standard deviation of travel time on the 30 studied road segments. Roads framed by a dotted line express high variation of data.

The behaviors of vehicles in this simulation follow Advanced Driving Skill [172], which is inspired by the Intelligent Driver Models and the MOBIL model for lane changes. The number of vehicles at every instant of the simulation is inhomogeneously varied between 50 and 200 to get enough diversity in traffic dynamics. The starting position of vehicles and the destination are randomly chosen. Then, GAMA computes all possible paths between these positions and picks the optimal one considered as the trajectory of the vehicle. We simulated the traffic and registered the obtained data for 3 hours, totaling approximately 9300 vehicle trajectories. The generated data include the GPS position, speed, and distance to the closest leading vehicle at every second for

every vehicle.

The complexity of the road network studied in this simulation allows us to observe the traffic propagation along neighboring road segments, enabling the evaluation the cooperative prediction process of ADRIP.

8.1.3.2 Settings

To conduct the experiment, we introduce methods that are compared with ADRIP, detail their parameters, select the evaluation metric and the preprocessing step of data.

1. ***Compared Methods and Parameters***

   ADRIP has two parameters to be defined *a priori* which are the adjustment coefficient γ of the adjustment process of CA's centroid chosen by experimentation at 0.05 and the similarity threshold vector α defined as 20% of the time required to cross the segment with the average speed of each speed range. We compare ADRIP with five models studied in the state-of-the-art. Their implementations are found in Python modules with the parameters set as follows:

   - ARIMA ($statmodel$): Order = (30,1,1), numbers of lags = 30.
   - KNN ($sklearn$): k = 18
   - FFNN ($keras.layers$): Hidden layers = 2, units = 256, learning rate = $1e^{-3}$, dropout rate = 0.1, decay rate = $1e^{-2}$, batch size = 256, optimizer: stochastic gradient descent algorithm.
   - RNN (LSTM, GRU) ($keras.layers.recurrent$): Hidden cells with 64 units, dropout layer = 0.2.

2. ***Evaluation Metrics***

   To evaluate the prediction performance, we adopt 2 metrics: MAE (Mean Absolute Errors) (equation 8.4) and RMSE (Root Mean Squared Error) (equation 8.5) defined as follows.

   $$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{8.4}$$

   $$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{8.5}$$

   where y_i is the i^{th} true value, $\hat{y}_i$ is the i^{th} predicted value and N is the number of data points.

3. ***Data Preprocessing***

   We train ARIMA, KNN, FFNN, RNN models, and ADRIP with the data set corresponding to the first 2 hours of simulation. Learned models are used to estimate future data for the

next hour. All predicted travel time for the next hour will be compared with data from the simulation. Note that the state-of-the-art models do not update themselves during the testing phase, while the learning process of ADRIP continuously self-adapts over time.

The inputs of the compared models are the time series of travel times of all crossing vehicles on each road segment registered every 10 seconds. Those models predict the next value by analyzing its dependence on the previous 5-minute observations. During the testing phase, the vectors of 5-minute previous observations are used to estimate the predictions for the next 5 minutes. Data rescaling transformations such as normalization, standardization, $etc.$ are not applied since it is impractical for continuous learning to set up the rescaling parameters due to the dynamic arrival and generation of data [153].

Remind that ADRIP provides the real-time predictive MP changes denoted as:

$\mathcal{P} = \{(predMP_1, Ts_1) \longrightarrow (predMP_2, Ts_2) \longrightarrow \cdots \longrightarrow (predMP_H, Ts_H)\}$. Ts_i is the predicted timestamp where the traffic on a given road network entity changes to $predMP_i$. $predMP_1, Ts_1$ is predicted using current traffic state. A prediction $(predMP_i, Ts_i)$ is computed using the previous prediction $(predMP_{i-1}, Ts_{i-1})$. In this evaluation, the prediction horizon H is selected at 5 minutes. The predicted MPs are evaluated every 10 seconds, thus the prediction horizon is equivalent to 30 following points, allowing the comparison with other models.

In the ADRIP's version without the self-correction mechanism, the prediction calculation of following step uses the predicted information of previous ones. However, with the self-correction mechanism the prediction can be recomputed using the current observations if the incorrect prediction is detected.

From the predicted MPs, the travel time is computed as the sum of all elements of MP as $T = \sum_{j=1}^{N} MP_j$.

8.1.3.3 Results and Analysis

The prediction performance of ADRIP is studied by three evaluations: the comparison of travel time prediction with state-of-the-art methods, prediction accuracy, performance of self-correction mechanism and data storage

1. ***Travel time prediction***

   Table 8.1 shows the comparison results on road segments with low variation of traffic data and highlights four key points. Firstly, with **the limitation of linear modeling and strict data assumptions**, ARIMA obtains significant prediction errors. Secondly, KNN, FFNN, LSTM, and GRU, thanks to **their capacity to capture long-term temporal dependencies and non-linear modeling**, perform well in travel time prediction for low variation data. High accuracy achieved in KNN is due to the **benefits of static clustering**, which analyzes the

data structure from a complete database, thus having a better understanding for cluster detection. In contrast, the dynamic clustering attempts to detect and gradually update data structure from small samples. That leads to potentially inadequate clustering structures initially, requiring more data and time to improve. Thirdly, **results obtained by both versions of ADRIP are comparable with other models**. In addition, ADRIP **gains in explicability** since we can determine which historical configuration brings this prediction and the impacts of each neighboring road segment on the predictions. That helps to analyze better and understand the traffic evolution. Fourthly, we note that **the self-correction mechanism does not enhance prediction accuracy in traffic with low variations**. Indeed, ADRIP does not frequently launch the self-correction mechanism when dealing with low variation data since no MP change is detected. ADRIP can misunderstand that its predictions are good. Further tests and analysis are required to identify the situations where AAs must launch this mechanism.

Table 8.1: Prediction errors in second of travel time prediction on road segments with low variations of traffic

Method	MAE	RMSE
ARIMA	21.52	26.75
KNN	8.75	14.80
FFNN	10.76	16.71
LSTM	9.05	14.90
GRU	9.11	15.18
ADRIP without self-correction	**10.64**	**16.87**
ADRIP with self-correction	**7.05**	**11.42**

In Table 8.2, the comparative results between ADRIP and other models on road segments with high variation of traffic data are presented (road segments framed by a dotted line in Figure 8.9). Firstly, ARIMA remains obtaining the worst performance. Secondly, we remark that ADRIP without self-correction obtains worse accuracy than KNN, FFNN, LSTM, and GRU. That is resulted from **the emergence of new behavioral patterns of the high dynamics of traffic**. The function of the prediction process in AAs is often disturbed since many new MPs are created, and AAs do not have historical information about them. Thus, the prediction is quickly degraded since the following prediction step depends on the previous one. In such cases, ADRIP proposes the current MP as the prediction and relies on **the self-correction mechanism** to correct them if errors are detected. Therefore, ADRIP with the self-correction mechanism outperforms all compared models in both criteria. This comparison results have proven the advantages of the self-correction mechanism and its importance when dealing with highly dynamic traffic.

2. ***Prediction Accuracy***

Table 8.2: Prediction errors in second of travel time prediction on road segments with high variations of traffic

Method	MAE	RMSE
ARIMA	149.29	190.68
KNN	91.49	122.02
FFNN	90.14	126.23
LSTM	110.24	145.07
GRU	108.06	141.71
ADRIP without self-correction	**134.53**	**181.08**
ADRIP with self-correction	**52.90**	**78.09**

In this section, we evaluate the prediction accuracy of MPs by ADRIP. At every 10 seconds, we verify if the observed MP from the learning and prediction processes are similar and compute the good performance percentage. With the version of ADRIP with the self-correction mechanism, we evaluate the predicted MP computed by the last correction using this mechanism.

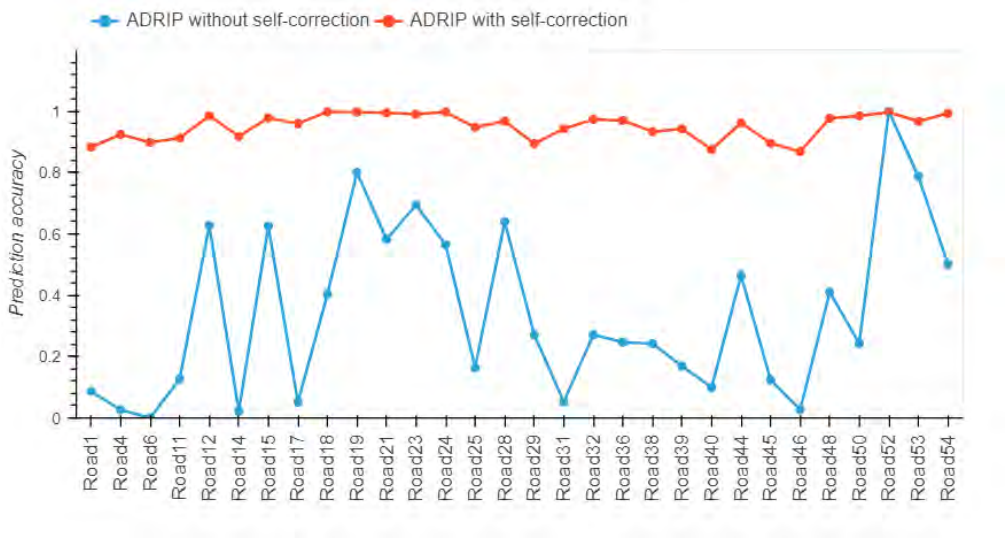


Figure 8.10: Accuracy of MP prediction by ADRIP without and with the self-correction mechanism

Figure 8.10 shows the percentages of good predictions at all road segments obtained by both ADRIP's version. The accuracy of ADRIP without the self-correction mechanism is relatively low with a wide accuracy range between 0.1 and 1, and with an average accuracy of 0.4.

This figure also highlights the variation of ADRIP's performance across different road segments, influenced by several factors:

(a) Number of learned clusters: the higher number of clusters may result in poorer predic-

tions due to diverse transitions between the clusters

(b) Traffic dynamics in the evaluation set: more stable traffic on a segment in the evaluation data set can bring the better prediction accuracy

(c) New traffic dynamics in the evaluation set: new MPs can lead to incorrect predictions as ADRIP requires time to learn and update its predictions

(d) Number of neighboring road segments: ADRIP's prediction process relies on the cooperation with neighboring AAs, thus AAs with few or no neighbor may perform poorly as they lack sufficient data for the prediction process.

Furthermore, the obtained results demonstrate a significant improvement of prediction accuracy when using the self-correction mechanism compared with the non-self-correction version. With ADRIP incorporating the self-correction mechanism, prediction accuracy surpasses 0.9 for all road segments, indicating its effectiveness in overcoming the degradation of prediction performance. Indeed, in ADRIP, the calculation of subsequent prediction steps depend on previous ones. Thus, if the incorrect estimations are computed at the first steps and are not detected, these wrong calculation can spread to the next steps. However, it's noteworthy that employing this mechanism frequently could optimize the prediction decision-making algorithm but might also lead to increased calculation time. The assessment of the self-correction mechanism regarding relaunching frequency becomes important.

3. ***Self-Correction Mechanism Evaluation***

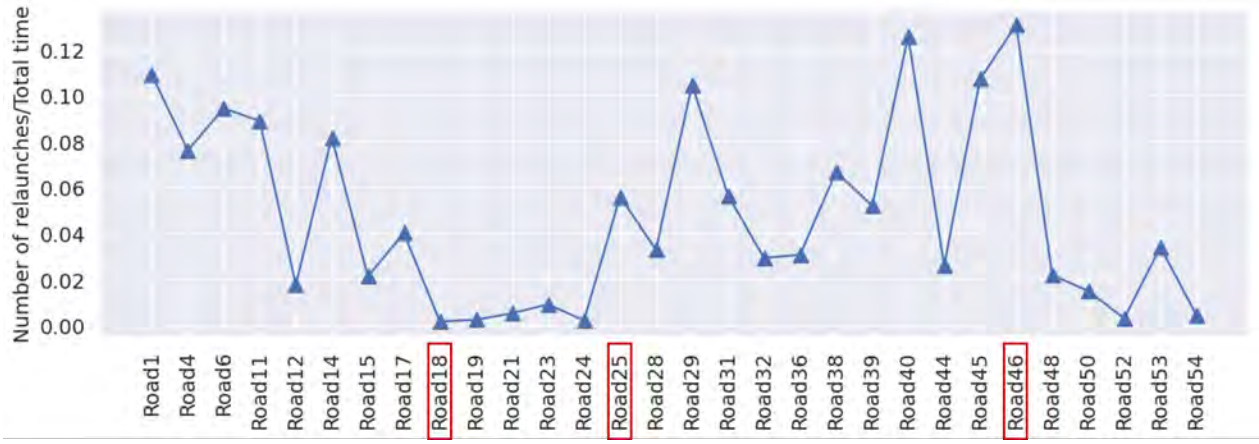
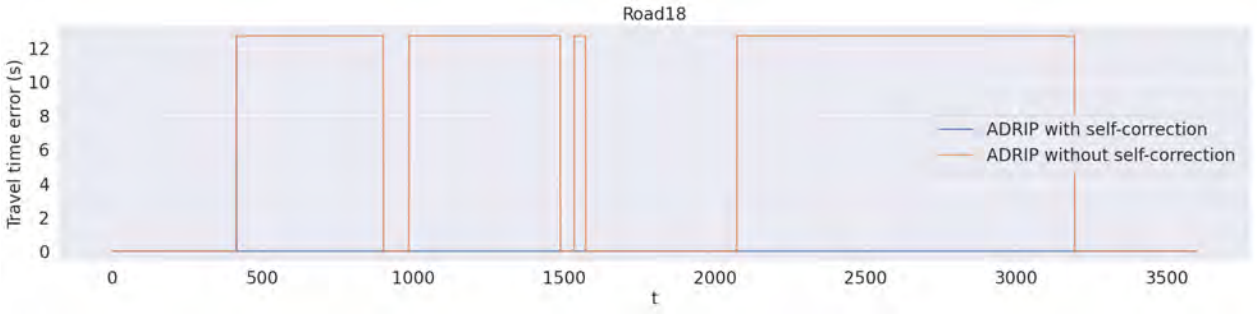


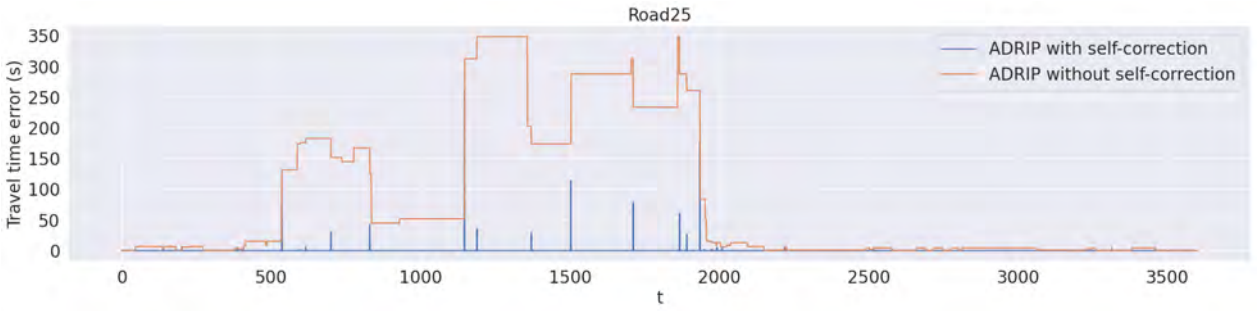
Figure 8.11: Ratio of activation instances of the self-correction mechanism versus total predictions

In this section, we study **the rate between the number of launches of self-correction mechanism versus the total prediction time**, and illustrate **the effectiveness of this mechanism** by plotting, at every second, the prediction errors. To conduct this evaluation, we memorize all predictions of every launch.

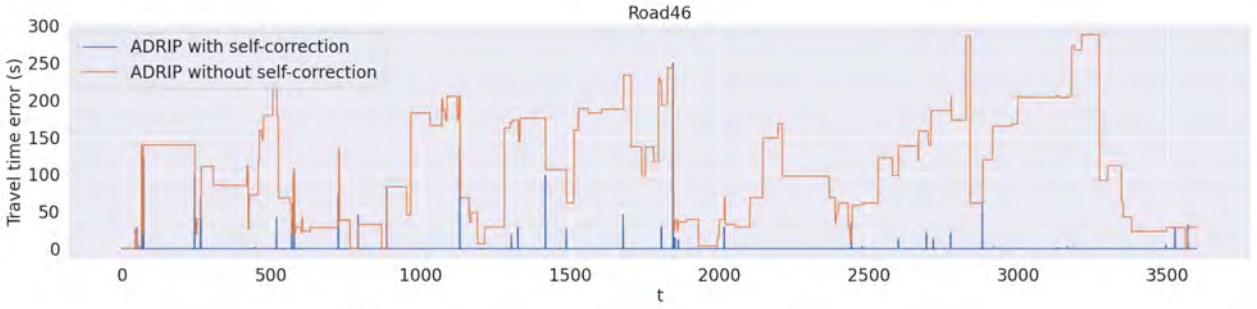
Figure 8.11 shows **the ratios between the instances of activating the self-correction mechanism and the total predictions for each road segment**. In this evaluation, all road segments need to provide traffic predictions at every second during one hour, resulting in a total of 3600 predictions. The self-correction mechanism is used at a rate of 0.17% of the total prediction time for the least-utilized road segment, and at a rate of 13.1% for the most-utilized road segment. These relatively low rates underscore the tolerate usage of the self-correction mechanism, reflecting the reliability of the prediction quality.



(a) Road 18 (road segment with the minimum activation instances of the self-correction mechanism)



(b) Road 25 (road segment with the average activation instances of the self-correction mechanism)



(c) Road 46 (road segment with the maximum activation instances of the self-correction mechanism)

Figure 8.12: Prediction errors over time on 3 representative road segments

To clarify the efficiency of the self-correction mechanism, we selected three representative road segments: **Road 18 (Figure 8.12a), Road 25 (Figure 8.12b) and Road 46 (Figure 8.12c)** respectively with the **minimum, average and maximum activation instances of the self-correction mechanism** and plotted at every second, the differences of travel time between

the observed MP and the predicted MP. In the case Road 18 and Road 25, we observed that prediction errors are significantly smaller and less than those in Road 46 since their traffic data contain less variations than the case of Road 46. Furthermore, the self-correction mechanism consistently identifies prediction errors in these three cases, triggering a recalculation of the predictions. This recalculation effectively corrects prediction errors since the error peaks of blue lines (representing ADRIP with self-correction) have short durations. Most of the time, incorrect predictions are immediately corrected, and errors promptly return to 0.

4. ***Data Storage Evaluation***

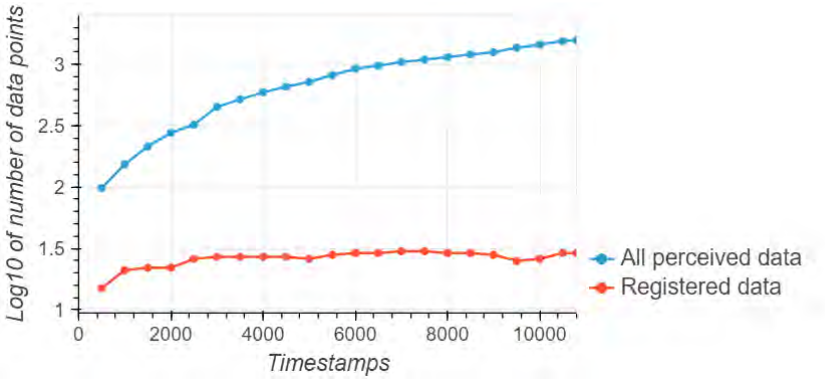


Figure 8.13: Registered data versus perceived data in Segment 46

Regarding data storage, ADRIP can perform the prediction with a small amount of memorized data. Figure 8.13 shows the total perceived MPs on the most crowded road segment versus the registered MPs in its learned database through time. Even though the amount of perceived MPs increases over time, the number of clusters (registered MPs) remains stable when ADRIP detects all the relevant traffic dynamics on the road segment. Therefore, ADRIP proves to be **efficient in reducing the amount of data that needs to be memorized** while achieving an adequate prediction performance. This efficiency is due to its dynamic clustering approach, which allows the system to detect traffic dynamics and adapt effectively to new data patterns.

8.2 Traffic Prediction for Macroscopic Information Level

The deployment of on-board sensors of connected vehicles enables the application of ADRIP to address traffic prediction challenges at a microscopic information level, as detailed in the previous section. Floating car data collected from these sensors are used to predict traffic dynamics to provide drivers with the anticipated mobility profiles on each road segment. However, real-world applications currently face limitations, as the number of connected vehicles and the percentage of them willing to share data remain insufficient for a comprehensive evaluation of ADRIP. Therefore, fixed sensors contribute as stable and primary data sources for real-world applications in ITS due to their wide deployment. Fixed sensors can be installed on roadsides and stably collect traffic

data aggregated from crossing vehicles such as traffic speed, flow, density, etc. These data enable the instantiation of ADRIP for macroscopic traffic information prediction and its application with real-world data.

In this instantiation, ADRIP is applied to compute traffic predictions at road segment entities using traffic data streams communicated by fixed sensors.

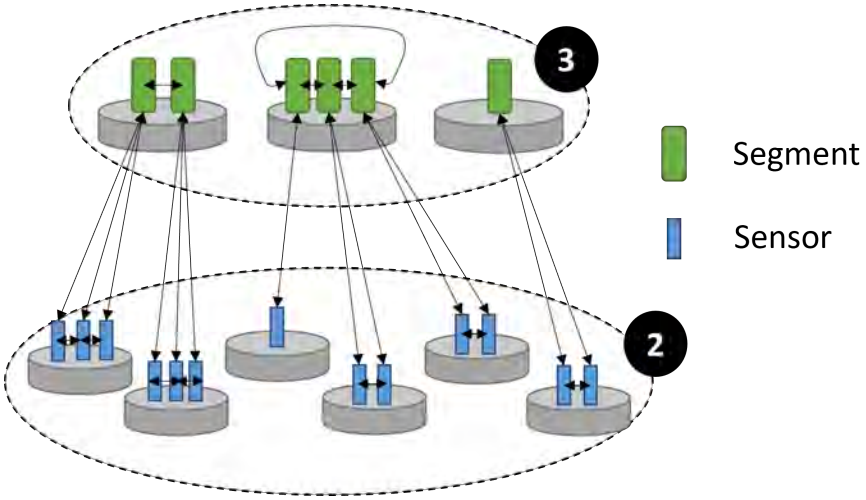


Figure 8.14: Scenario of traffic prediction for macroscopic information level instantiation

Therefore, the data provider and processing entities are determined as follows:

- **Data provider entity:** a set of fixed sensors installed along road segments in the considered network. Fixed sensors can collect the aggregated traffic data from crossing vehicles during a time window, such as: traffic speed, traffic flow, etc. Several sensors can be deployed on the same road segment, allowing to observe different traffic states at different positions on the road.

- **Processing entity:** a set of road segments determined according to the road network in Open Street Map (OSM), its starting and ending points located by GPS devices.

Sensor entities can determine the associated road entity with their location and transmit their traffic measures to it every time interval. Unlike the instantaneous floating car data collected by connected vehicles that are typically recorded and sent immediately, fixed sensors capture traffic information over specific time intervals. For example, to measure traffic flows, sensors count the number of crossing vehicles over a period ($e.g.$ 5 minutes) and record this as an observation. In this instantiation, the communicating timestamp of each DP is the beginning of collecting time interval of sensor.

Each road segment entity receives the traffic data stream from sensors located on them. The data stream perceived at each road segment entity is denoted as:

$DS = \{DP_{Ts_1}, \ldots, DP_{Ts_t}, \ldots, DP_{Ts_N}\}$

consisting of a sequence of N data points arriving during timestamps $Ts_1, \ldots, Ts_t, \ldots, Ts_N$ sent from fixed sensor entities. As the communication between fixed sensor and road segment entities is static, we consider the same set of communicating time intervals for all road segment entities.

Road segment entities apply ADRIP to learn traffic dynamics from their associated DS to build the learned database and compute their predictions. The learned database contains a set of cluster agents CAs representing different traffic dynamics.

8.2.1 ADRIP instantiation

Similar to previous scenario, we need to define following elements to enable the instantiation of ADRIP:

1. **For Data Agent:** the representation of traffic dynamics
2. **For Cluster Agent:** the components of a cluster, the similarity measure of studied traffic dynamics, the definition of similar traffic dynamics and the similarity threshold of associated road segment entity
3. **For Analyzer Agent:** the definition of AA's neighborhood
4. **For Noise detection method:** the method using to detect and eliminate the noise. This element is optional, depending on the expert knowledge about the application scenario

8.2.1.1 Instantiation of Data Agent

Each DA contains

- A communicating timestamp
- A DP: an observation vector O_{Ts}

  Traffic data collected from all sensors located on the same road segment form an observation vector for that road segment entity during the timestamp Ts_t. The observation vector is denoted as $O_{Ts_t} = \{o^1_{Ts_t}, \ldots, o^i_{Ts_t}, \ldots, o^I_{Ts_t}\}$, where $o^i_{Ts_t}$ is traffic data from the sensor i and I is the number of sensors on a given road segment entity. Unlike the previous instantiation, observation vectors on different road segment entities do not have the same size, depending on the number of sensors installed on the considered road segment. For example, road segments with long lengths or multiple lanes can be equipped with multiple sensors to capture traffic data at different locations. To deal with the diversity of observation's size in this instantiation, the distributed and local learning process of ADRIP is an appropriate solution.

8.2.1.2 Instantiation of Cluster Agent

Each CA contains **a centroid** that is the combination of all historically assigned observations to this cluster, and a list of **Ranges of Use (RU)** associated to the cluster. In this instantiation, the range of use is defined as a set of consecutive timestamps when the communicated observation vectors are assigned to a same cluster.

Each CA possesses **a set of similarity thresholds** to evaluate the similarity between its centroid and an arriving observation vector. Their pre-defined or computed based on the characteristics of the associated road network entity where it is located such as maximum speed, length or the number of installed sensors.

To evaluate the similarity between two observation vectors, let O_{Ts_1} and O_{Ts_2} be the observations arriving at time intervals Ts_1 and Ts_2.

$O_{Ts_1} = (o^1_{Ts_1}, \ldots, o^I_{Ts_1})$

$O_{Ts_2} = (o^1_{Ts_2}, \ldots, o^I_{Ts_2})$

We define a threshold α_i for the i communicating sensor in the observation vector O_{Ts}. α_i can be the same if the sensors provide the same traffic information or can be different if the sensors measure different types of traffic data. To evaluate whether two elements in two observations are similar, the difference between them is compared to the corresponding α_i. If the difference between two values is smaller than α_i, they are considered similar and the Comparison Vector (CV) of O_{Ts_1} and O_{Ts_2} denoted $CV(O_{Ts_1}, O_{Ts_2})$ takes the value 1. Otherwise, it takes the value 0.

$$CV_i(O_{Ts_1}, O_{Ts_2}) = \begin{cases} 1 & |o^i_{Ts_1} - o^i_{Ts_2}| \leq \alpha_i, \\ 0 & |o^i_{Ts_1} - o^i_{Ts_2}| > \alpha_i \end{cases}, i = 1, \ldots, I \tag{8.6}$$

If the number of communicating sensors that give different values is smaller than **a similarity threshold** β, then, two observations O_{Ts_1} and O_{Ts_2} are considered **similar**.

$$O_{Ts_1}, O_{Ts_2} \text{ are similar} \Leftrightarrow Card(CV(O_{Ts_1}, O_{Ts_2}) = 0) \leq \beta \tag{8.7}$$

For example, considering a road segment entity with a maximum speed at 90 km/h having four fixed sensor entities located on it, the value of similarity threshold can be defined as $\alpha_i = 30km/h, i = 1, \ldots, 4$, meaning that if the difference of observed speeds at the sensor i is greater than 30 km/h, the value of comparison vector at this sensor is equal to 1 ,$i.e\ CV_i = 1$. The value of β is equal to 2, meaning that if there are more than two sensors giving different speeds ($i.e. CV_i = 1$), two observation vectors are considered different.

8.2.1.3 Instantiation of Analyzer Agent

The definition of AA's neighborhood in this instantiation are the same as the description provided in Section 8.1.1.3. However, it's important to note that only highways within the road network are considered. Therefore, certain road segments may not have neighbors if they are not connected to other highways.

8.2.1.4 Instantiation of Noise Detection

We do not consider noise detection in this instantiation since the data collected from fixed sensors are well-pre-selected.

8.2.2 Learning Process Evaluation

8.2.2.1 Data description

The real-world traffic data set used in this experiment is provided by the Swedish Transport Administration through MMTL (Multimodal traffic management, grant number TRV 2020/118663) project within the collaboration between KTH Royal Institute of Technology, Sweden and IRIT Computer Science Research Institute of Toulouse, France. This collaboration is part of a three-month PhD exchange program associated with this thesis.

This data set is collected from microwave sensors located on highways around Stockholm, Sweden ($Fig.8.15$). There are about 800 sensors collecting **traffic speed** data during 2017 and 2018. The same preprocessing methods are applied as in [37] involving:

- Focusing on the daytime period between 05:00 and 21:30 and aggregating speed observations to 64 time intervals of size 15 minutes.
- Choosing only days and sensors that are active during all time intervals, including about 500 sensors and 349 days in 2017 and 292 days in 2018.

The locations and standard deviation of measured traffic speeds of studied sensors are shown in Figure 8.15(a). Traffic speed variation is very different depending on the sensor's location. For example, three sensors A, B and C located near the on-ramps and off-ramps of Stockholm and Södertälje often experience congestion during morning and afternoon peaks ($Fig.8.15(b), (c), and(d)$). The high variation of traffic speeds of their neighbors also implies a strong dynamic of traffic in these areas. In addition, traffic shows the seasonal property but with different periods and patterns on different sensors. Thus, our clustering approach with flexible time windows is well suited to this case study.

This dataset, obtained from a large-scale road network, enables the validation of ADRIP at multiple traffic levels. Indeed, ADRIP can be applied to the sensor level by incorporating data from all sensors or at high levels, such as road segments or networks, by aggregating data from sensors located at the same considered level.

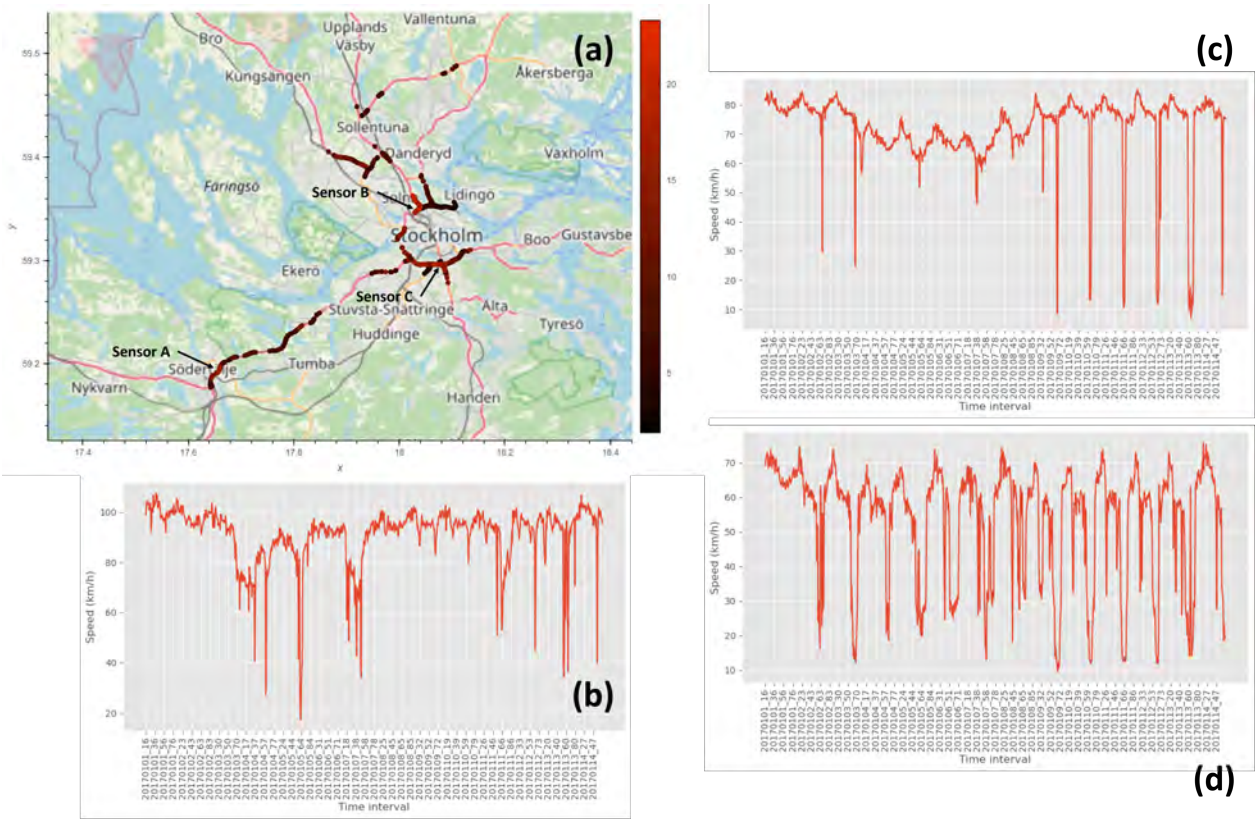


Figure 8.15: Case study in Stockholm, Sweden.
(a) Location of all sensors with their standard deviation of speed measurement;
(b), (c), (d) Speed time series of three example sensors having high speed variations.

8.2.2.2 Settings

To conduct the evaluation, we introduce methods that are compared with ADRIP, detail their parameters, select the evaluation metric, and the preprocessing step of data.

1. ***Compared Methods and Parameters***

   We conduct tests to compare the prediction performance of ADRIP with three well-known distance-based clustering algorithms: **Agglomerative clustering [50], K-Means [125] and Spectral clustering [179].**

   ADRIP requires defining two parameters in advance: similarity thresholds α_i for each communicating sensor and β, defined as the threshold for the number of sensors giving different values. Since our studied scenario includes only the motorways, we define the same value for all $\alpha_i, i = 1, \ldots, I$. In the following experiments, we vary the similarity thresholds: $\alpha_i = [10, 20, 30] km/h$ and $\beta = [5\%, 10\%, 15\%] \times I$ (size of observation vector).

   One of the most important parameters of the compared algorithms is the **number of clusters**. To make the comparison fair, we vary this parameter equal to the number of clusters obtained

Table 8.3: Number of clusters according to similarity thresholds

Similarity thresholds / Data scales	$\alpha_i = 10$ $\beta = 5\%$	$\alpha_i = 10$ $\beta = 10\%$	$\alpha_i = 10$ $\beta = 15\%$	$\alpha_i = 20$ $\beta = 5\%$	$\alpha_i = 20$ $\beta = 10\%$	$\alpha_i = 20$ $\beta = 15\%$	$\alpha_i = 30$ $\beta = 5\%$	$\alpha_i = 30$ $\beta = 10\%$	$\alpha_i = 30$ $\beta = 15\%$
All sensors	9903	3097	868	2678	420	70	667	61	7
Data reduction using correlation	9222	3221	1001	2640	455	88	637	71	9
Data aggregation by GPS clustering	1572	668	309	67	14	7	7	2	1

from ADRIP as it creates clusters dynamically. Table 8.3 shows the number of clusters according to tested thresholds of ADRIP. Other parameters are kept the same as default settings presented in [1]. However, spectral clustering encounters difficulties in reaching convergence due to the high dimensionality of the data. Thus, we modify some parameters from the default setting to accelerate the convergence such as:

- $eigen\_solver = amg$: the eigenvalue decomposition strategy to use.
- $eigen\_tol = 0.01$: stopping criterion for eigen decomposition of the Laplacian matrix.
- $assign\_labels = cluster\_qr$: the strategy for assigning labels in the embedding space

2. ***Evaluation Metrics***

The Mean Absolute Percentage Error (MAPE) is used to evaluate the prediction accuracy in this experiment. The MAPE can be more beneficial than other metrics as MAE or RMSE because it expresses the percentage difference between prediction errors and true values. However, a significant disadvantage of MAPE is that it produces undefined values when the values of data are 0. Therefore, MAPE is not used in the previous case since the simulation contains a lot of traffic jam leading to vehicle speeds can be decreased at 0. In this experiment, real-world data are collected on highways, thus MAPE can be a more appropriate evaluation metric.

The expression of MAPE between the centroid of observed cluster Y_{obs} and Y_{pred} is given as follow:

$$MAPE(Y_{obs}, Y_{pred}) = \frac{1}{I}\sum_{i=1}^{I}\left|\frac{Y_{obs}^{i} - Y_{pred}^{i}}{Y_{obs}^{i}}\right| \tag{8.8}$$

3. ***Data Preprocessing***

We evaluate the dynamic clustering of ADRIP over three data scales using the real-world data set described in the previous section.

- **Data from all sensors:** observations from all sensors will be taken into account. Thus, data in this case express the high variation. An observation vector at Ts_t is summarized

as: $O_{Ts_t} = \{o^1_{Ts_t}, \ldots, o^i_{Ts_t}, \ldots, o^I_{Ts_t}\}$, where $o^i_{Ts_t}$ is traffic data from the sensor i and I is the number of communicating sensors.

- **Data reduction using correlation:** an observation includes only data from representative sensors of groups consisting of correlated sensors. Sensors located on the same road or close to each other can express high correlation ($Fig.8.16$). Road segments observe the measures on their sensors and locally decide or cooperate with their neighbors to identify representative sensors. At this scale, road segment entities show their cooperation ability to select adequate data.

  The sensors in the used dataset are provided with group identification. Sensors within the same group are located in close proximity to each other. For each group, we consider a representative sensor that has the highest mean correlation with other sensors belonging to the same group. By applying this method, the dimension of the observation vector decreases from 500 to 164. An observation vector at Ts_t is summarized as: $O_{Ts_t} = \{o^1_{Ts_t}, \ldots, o^k_{Ts_t}, \ldots, o^K_{Ts_t}\}$, where $o^k_{Ts_t}$ is traffic data from the representative sensor k and K is the number of sensor groups.

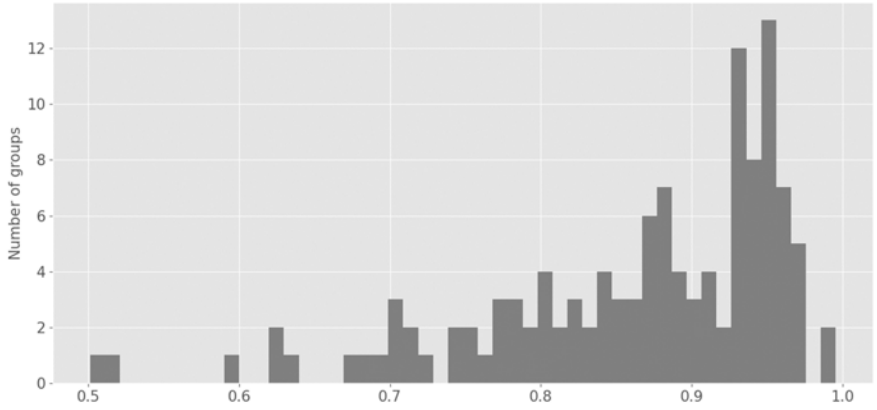


Figure 8.16: Histogram of mean correlation values of observed speeds on sensors belonging to the same group.

Analyzing this data scale is important because learning with high dimensional data (high number of variables or features in observation vector, 500-dimension vector in our case study) can risk computational and statistical issues such as over-fitting due to redundant features.

- **Data aggregation by GPS position clustering:** at this scale, road segments are clustered using K-Means on their GPS locations. The number of clusters depends on the required data granularity. Data from sensors belonging to the same group will be gathered and averaged. An observation vector at Ts_t is summarized as: $O_{Ts_t} = \{o^1_{Ts_t}, \ldots, o^l_{Ts_t}, \ldots, o^L_{Ts_t}\}$, where $o^l_{Ts_t}$ is mean values of observations from sensors within the cluster l and L is the number of clusters.

We first launched the learning process using data of 2017 and started the prediction process simultaneously using data from the beginning of 2018.

8.2.2.3 Results and Analysis

Fig.8.17(a) shows the speed prediction of studied models using data from all sensors. In general, all clustering models tend to decrease MAPE with more clusters. For all tested numbers of clusters, spectral clustering has the worst prediction performance (MAPE equal to 11.3% at 9903 clusters). On the other hand, ADRIP achieves the best performance (MAPE equal to 7.4%). When the number of clusters increases, agglomerative clustering and K-means initially decrease the MAPE to a minimum of 11% before increasing it with higher numbers of clusters. ADRIP shows a continuous drop in MAPE and obtains the best accuracy for all tested numbers of clusters. However, using a large number of clusters is time-consuming, while MAPE improvement is moderate. Therefore, depending on applications, we can choose the adaptive similarity thresholds so that dynamic clustering creates a reasonable number of clusters and achieves the balance between performance and calculation time (using Table 8.3 as a reference). Additionally, when conducting tests involving various road types, the selection of the similarity threshold can be tailored to the specific characteristics of each road segment, enabling the optimized number of cluster and calculation time.

Table 8.4: Calculation times of compared methods for without and with data reduction

Method	ADRIP		Agglomerative		K-Means		Spectral	
Test cases	Case 1	Case 2	Case 1	Case 2	Case 1	Case 2	Case 1	Case 2
Calculation time using data from all sensors	240s	6022s	124s	5790s	30s	1776s	84s	24060s
Calculation time with data reduction using correlation	331s	4737s	42s	1098s	10s	224s	71s	19508s

The prediction results from data reduced by correlation have the same interpretation as the previous case (Fig.8.17(b)). However, we aim to **underline two behaviors in this case**. Firstly, we are interested in **the decrease in calculation time compared to the application using data from all sensors**. The calculation time for the entire data set (containing 46793 observations) is compared through 2 extreme cases that give the smallest (Case 1: $\alpha_i = 30km/h$, $\beta = 15\%$) and biggest (Case 2: $\alpha_i = 10km/h$, $\beta = 5\%$) numbers of clusters (Table.8.4). Secondly, **other methods improve their performance with reduced data, while ADRIP does not improve its performance**. This is due to the used similarity distance. Indeed, other methods use the Euclidean measure that calculates the similarity by averaging the distances at all features. Thus, when having many correlated features

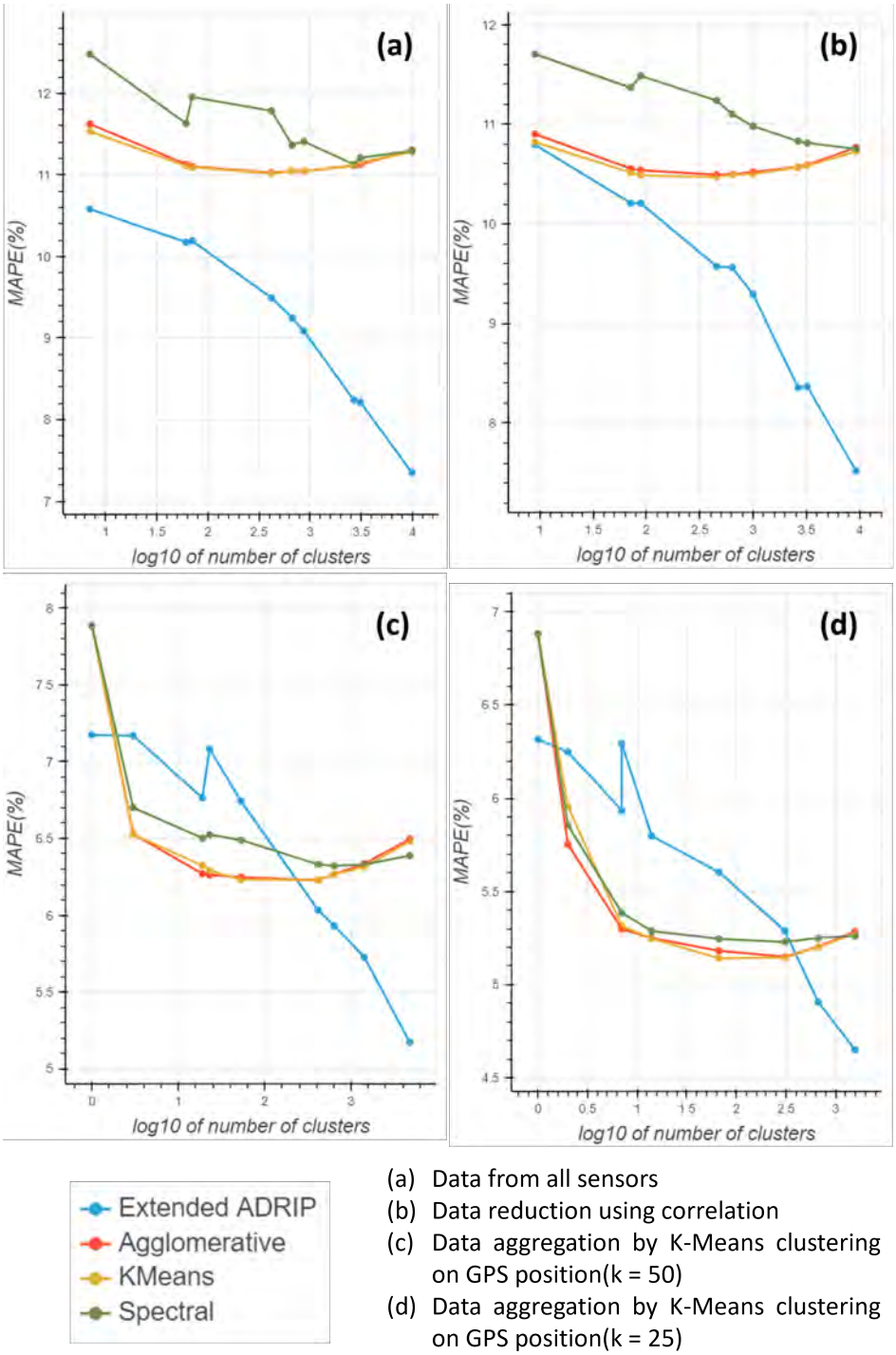


Figure 8.17: MAPE of ADRIP and compared models

with similar data evolution, they dominate in the average distance and ignore the effect of other features that leads to inadequate similarity evaluation. Meanwhile, ADRIP evaluates the similarity at each order in the observation vector, thus, **the redundant features impact on ADRIP less than**

the others.

Fig.8.17(c),(d) show speed prediction using data aggregation by K-Means clustering with $k = 50$ and $k = 25$. Firstly, we notice that with the same number of clusters, **the ranges of errors of aggregated data are smaller than in previous cases** (MAPE varies between 5% and 8% in case $k = 50$ and between 4.5% and 7% in case $k = 25$ while the error range of the two above cases is between 7% and 12%). That is due to the lower variations caused by the data aggregation. Secondly, **when k is smaller**, meaning aggregated data contain lower variation, **other methods outperform ADRIP**. Indeed, the classical clustering methods are well-known to perform well on data with low variations. In these cases, they show the **benefits of static clustering** compared to dynamic clustering since they can understand better data structure from the complete database. Meanwhile, dynamic clustering attempts to detect and gradually update data structure from small samples. However, when clustering involves more clusters, ADRIP obtains prediction with higher accuracy than others.

8.2.3 Prediction Process Evaluation

8.2.3.1 Data Description

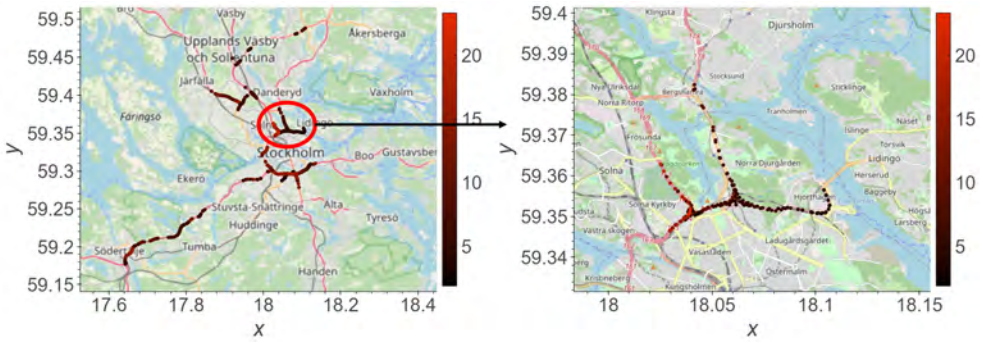


Figure 8.18: Case study in Stockholm, Sweden (left) and the testing area for the traffic prediction at segment level

This evaluation uses the same data set described in Session 8.2.2.1. However, due to the extensive number of road segments (267 road segments) in the studied area, deploying the prediction process becomes prohibitively expensive for laboratory-scale experiments as all agents in ADRIP require an independent and parallel processor to operate. From this reason, we focus our test for the prediction of traffic speeds on the central area of Stockholm (shown in Figure 8.18), containing 75 road segments.

8.2.3.2 Settings

1. ***Compared Methods, Parameters and Evaluation Criteria***

ADRIP has two parameters to be defined *a priori* which are the adjustment coefficient γ fixed at 0.2 and the similarity threshold vector α defined as 20km/h. The value of β is set at 0, signifying that two observation vectors are similar only if all sensors need to provide similar speeds. The parameters of baselines and evaluation criteria are chosen as presented in Section 8.1.3.2.

2. ***Data Preprocessing***

   We train ARIMA, KNN, FFNN, RNN models, and ADRIP with the data of 2017. Learned models are used to estimate future traffic with the horizon at one next hour. The prediction is evaluated using data from 01/01/2018 to 31/05/2018.

   The inputs of compared models are the time series of observation vectors. Those models predict the next value by analyzing its dependence on the 4 previous time intervals (equivalent to one hour). During the testing phase, the vectors of 4 previous observations are used to estimate the predictions for 4 next time intervals (one hour). Likely to the previous application, data rescaling transformations such as normalization, standardization, etc are not applied.

   For ADRIP, the prediction process continuously computes traffic prediction as described in the data preprocessing of Section 8.1.3.2.

   For all evaluated methods, the mean speed is computed as the mean value of observation vectors.

8.2.3.3 Results and Analysis

We categorize the analyzed road segments into two distinct groups: **those with low and high traffic variation**, determined by assessing the standard deviation of traffic speeds. As illustrated in Figure 8.19, road segments with a standard deviation of traffic speeds less than 10km/h are classified as having low traffic variations, while those with a standard deviation exceeding 10km/h are considered as experiencing high traffic variations. This classification results in 60 and 14 road segments with respectively low and high traffic variations.

1. ***Mean Speed Prediction***

   Table 8.5 shows the comparative results for road segments characterized by low traffic data variation between ADRIP with the self-correction mechanism and the baseline models. Note that, as the efficiency of the self-correction is clearly demonstrated in the first instantiation, we have excluded the version of ADRIP without it in this instance. Several key points are highlighted from the obtained results:

   - The table does not include the results of the ARIMA model because it **failed to converge**. The large dataset used for model training poses a significant challenge for ARIMA, causing difficulty in parameter calibration and requiring a long time to achieve the convergence condition.

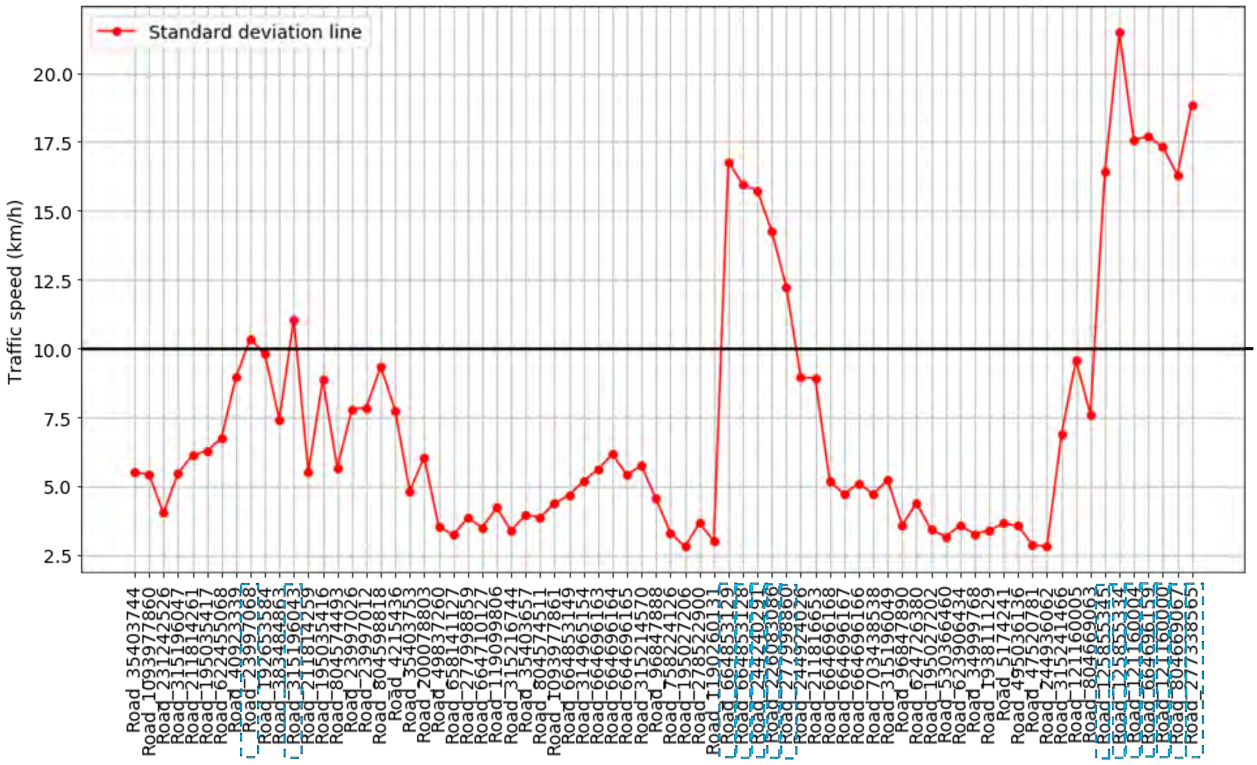


Figure 8.19: Standard deviation of traffic speed data

Table 8.5: Errors of mean speed prediction on roads segments with low traffic variations ($i.e.$ std $<$ 10)

Method	MAE	RMSE
KNN	**2.31**	4.14
FFNN	2.97	4.37
LSTM	2.92	4.90
GRU	2.51	**4.13**
ADRIP with self-correction	**3.64**	**4.87**

- **KNN and GRU models outperform the others** by delivering the best performance with the lowest prediction errors, with KNN achieving the smallest MAE and GRU achieving the smallest RMSE.
- **ADRIP, on the other hand, exhibits the highest prediction errors in both selected metrics**. Similar to the previous evaluation, when using data sets with low standard deviations, well-known baselines achieve a good performance, benefiting from their static modeling capabilities. In contrast, ADRIP's prediction mechanism relies on dynamic clustering, primarily focusing on exploring and updating model knowledge. This property allows ADRIP to maintain its performance in a dynamic environment. However,

in static conditions, the dynamic and exploratory aspects can lead ADRIP to incorrect decisions or insufficient capacity for prediction estimation.

Table 8.6: Errors of mean speed prediction on roads segments with high traffic variations ($i.e.$ std $\geq$ 10)

Method	MAE	RMSE
KNN	5.58	10.27
FFNN	5.50	9.33
LSTM	4.96	8.95
GRU	5.07	9.04
ADRIP with self-correction	**5.36**	**7.17**

Table 8.6 presents the comparative results of ADRIP against baseline models for road segments characterized by high traffic data variation. **ADRIP demonstrates a competitive performance** in this application when compared to other methods. Using MAE, ADRIP obtains smaller errors than KNN and FFNN but more significant errors than RNN-based models. LSTM and GRU consistently demonstrate their ability to capture complex temporal dependencies in high-variation data. However, KNN and FFNN, which lack strong temporal and dynamic properties, perform worse than ADRIP. **When evaluating performance using RMSE, ADRIP stands out as the best performer**, with the lowest error among evaluated methods. Since RMSE results in a higher penalty on large errors than MAE, we can infer from this result that ADRIP tends to **make more small-scale errors but fewer large-scale errors than the other methods**. ADRIP proposes the predictions based on not only the centroid of clusters but also their ranges of use. Consequently, if the range of use of an incorrectly predicted cluster is extensive, this may result in multiple errors stemming from a single prediction calculation. However, these errors are generally smaller than those from other methods. This explains why the MAE obtained from ADRIP is larger than other methods, but the RMSE is smaller, reflecting the trade-off between small and large errors. Lastly, it is important to note that **ADRIP achieves a comparative results with FFNN, LSTM and GRU while gaining a higher level of explainability** in the predictive model.

2. ***Prediction Accuracy***

   Figure 8.20 shows the percentages of good predictions at all road segments obtained by two versions of ADRIP with the real-world data set. In ADRIP without the self-correction, we obtain the mean accuracy for all selected road segments at 0.65 while this value is achieved at 0.90 when integrating the self-correction mechanism. This mechanism consistently demonstrates its efficiency in traffic prediction problem by significantly improving the accuracy.

3. ***Self-Correction Mechanism Evaluation***

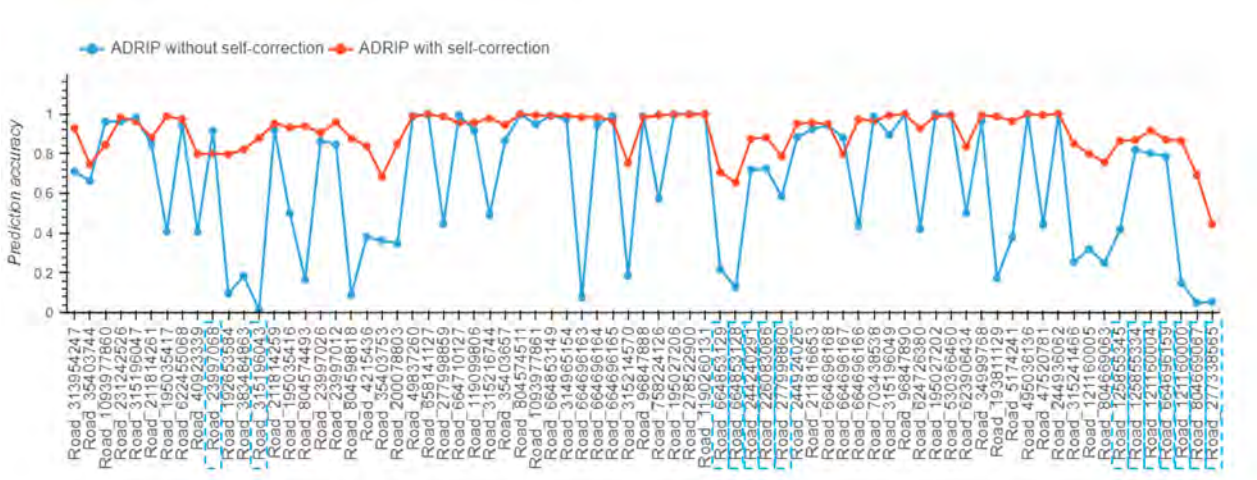


Figure 8.20: Accuracy of cluster prediction of real-world data

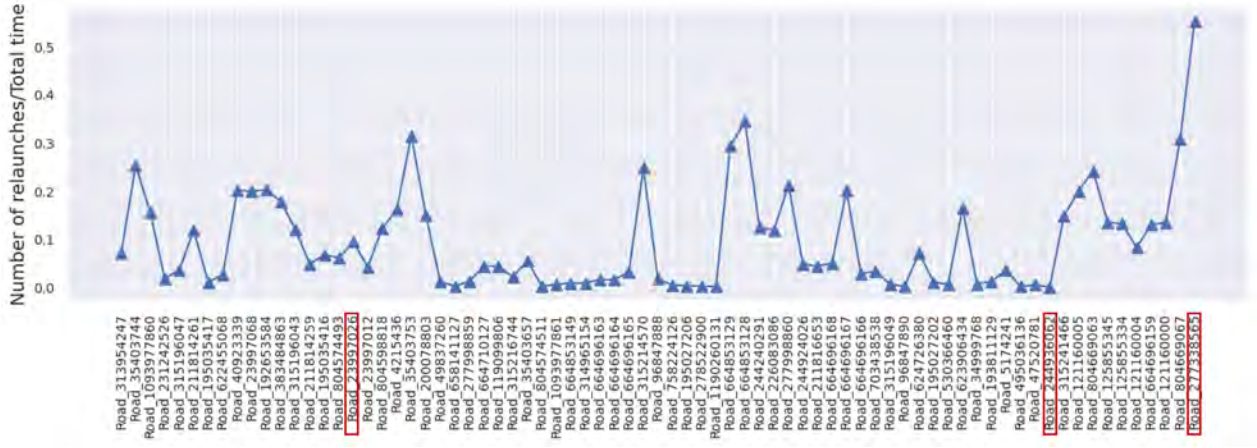
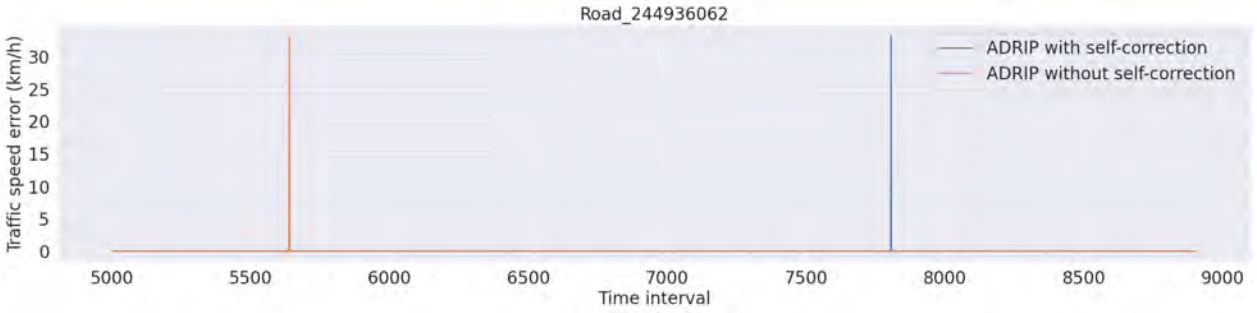


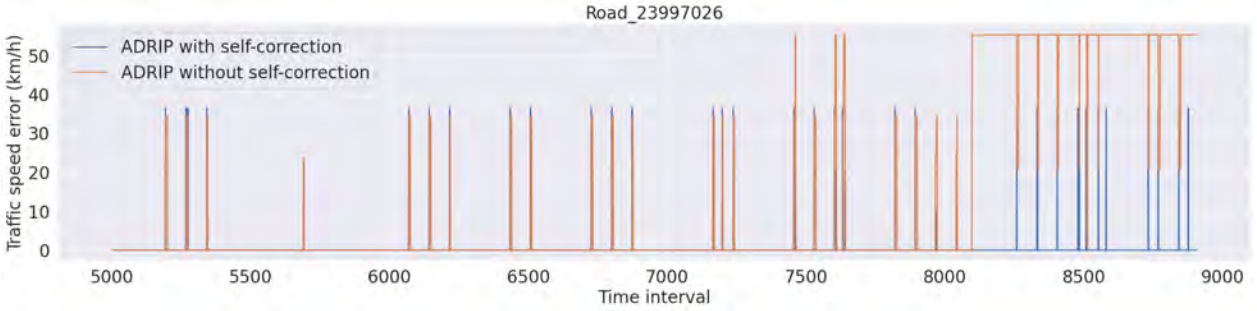
Figure 8.21: Ratio of activation instances of the self-correction mechanism versus total predictions

Figure 8.21 shows the ratios of instances where the self-correction mechanism is activated to the total prediction time for each road segment. In this evaluation, all road segments need to provide traffic predictions at every 15 minutes time interval during the selected period from 01/01/2018 to 31/05/2018, resulting in a total of 8910 predictions. The self-correction mechanism is used at a rate of 0.06% of the total prediction time for the least-utilized road segment, and at a rate of 55.3% for the most-utilized road segment. An exceptional case is observed at *Road 277338565*, where the self-correction mechanism is triggered in over half of the total predictions. This anomaly is attributed by the substantial traffic variations on this road, experiencing the highest number of traffic dynamic transitions (1321 transitions) among the considered roads, and lacking neighboring segments to improve predictions.

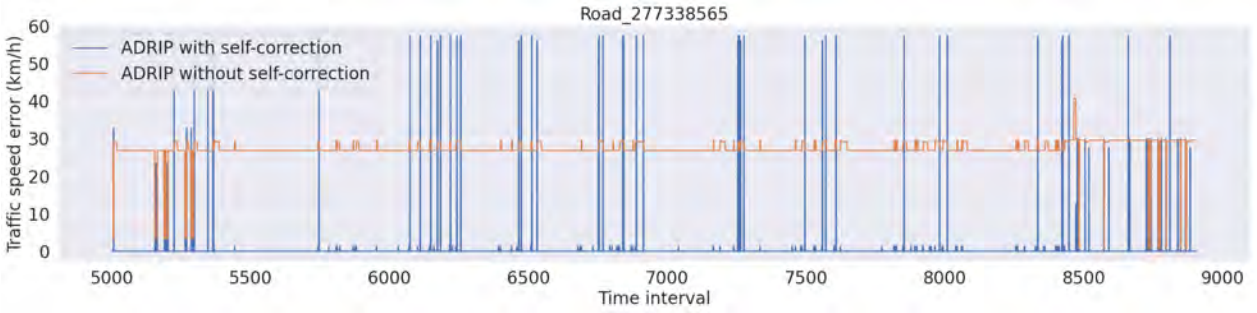
Similar to the first instantiation, we study specific road segments: **Road 244936062 (Figure 8.22a), Road 23997026 (Figure 8.22b) and Road 277338565 (Figure 8.22c)** respectively with the



(a) Prediction errors over time on *Road 244936062* (road segment with the minimum activation instances of the self-correction mechanism)



(b) Prediction errors over time on *Road 23997026* (road segment with the average activation instances of the self-correction mechanism)



(c) Prediction errors over time on *Road 277338565* (road segment with the maximum activation instances of the self-correction mechanism)

Figure 8.22: Prediction errors over time on 3 representative road segments

minimum, average and maximum activation instances of the self-correction mechanism. The differences between observed traffic speed and predicted traffic speed are plotted from the 5000^{th} communication time intervals until the end of the evaluation data set. These time intervals are selected to be able to observe the activation of the self-correction mechanism on these three road segments. For the road segment with the minimum activation instances, no noticeable difference in prediction errors is observed between the two versions of ADRIP. However, in the two other cases, the self-correction mechanism markedly detects bad predictions, prompting a recalculation of predictions. This recalculation consistently demonstrates effective correction for prediction errors, since the short durations of error peaks

in the blue lines is short (representing ADRIP with self-correction). The incorrect predictions are immediately rectified, and errors promptly return to 0 for the majority of the time.

4. ***Data Storage Evaluation***

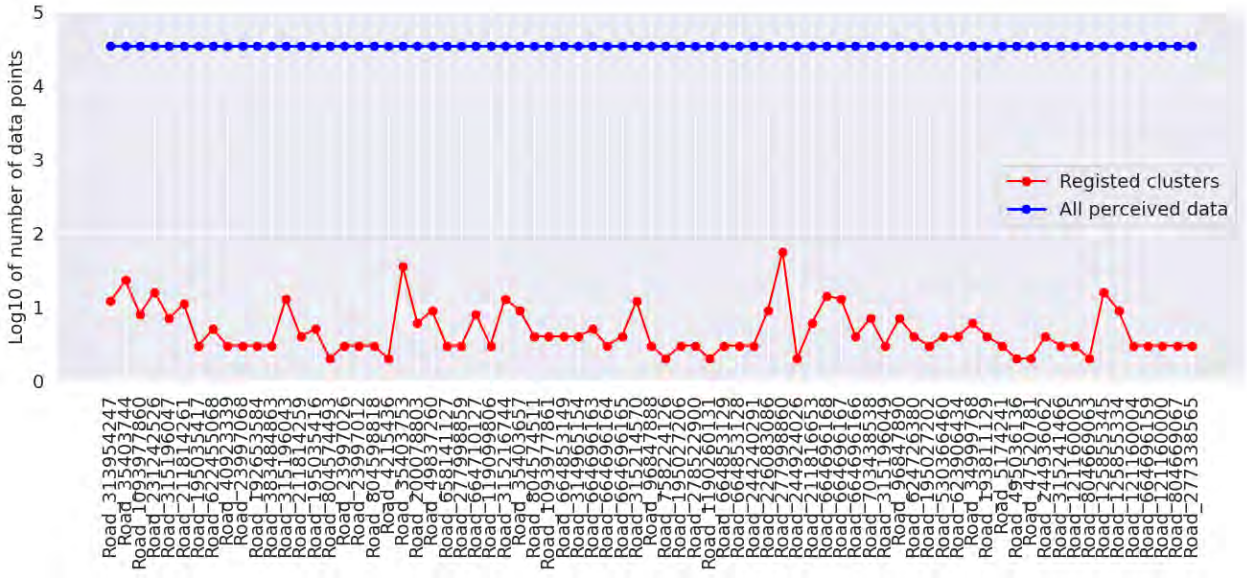


Figure 8.23: Registered clusters versus total perceived data points

Figure 8.23 shows the total perceived DPs versus the number of registered clusters at each road segment entity. All road segments perceive 34383 DPs during this evaluation, however, the number of detected and registered clusters are relatively small, ranging from 2 to 56 clusters on the considered road segment entities.

8.3 Discussion

This chapter presented the instantiation of ADRIP on two cases: **traffic prediction for microscopic and macroscopic information level**. The used data sets are diverse including both simulated and real-world data on different selected road networks. The representation of traffic dynamics is different, adapting to collected traffic data and the predictive information that the applications aim to provide. The instantiation detailed the required components of the generic description of ADRIP, adapting to the considered scenario. Then, ADRIP is evaluated on both learning and prediction processes and compared with the well-known methods of the state of the art.

Each instantiation details **the application context** and **the adaptation of ADRIP** for this instantiation. In the application context, we describe the level of road network architecture where ADRIP is applied with the introduction of data provider and processing entities.

Then, the used traffic data stream is defined, corresponding to the objectives of each scenario. Based on this introduction, the agents of ADRIP are adapted. Then, the similarity measure and

threshold are defined to enable the evaluation of traffic data similarity.

For the experiments, we use three data sets with different scenarios. Data sets are selected to suit with the purposes on each evaluation and the development of the thesis. Performing ADRIP on different testing scenario allows to demonstrate its versatility.

The evaluation of ADRIP is conducted in two steps: **the assessment of the continuous learning process and cooperative prediction process**. The learning process is assessed using both simulated and real-world datasets. In the first dataset, ADRIP obtained a better performance compared to CluStream under silhouette metrics. For the second dataset, a predictive evaluation based on clustering was performed. ADRIP demonstrated its efficiency, particularly in scenarios with traffic data exhibiting high variation, outperforming agglomerative clustering, K-Means, and spectral clustering using the MAPE metric. In both cases, **the continuous learning method of ADRIP showed its advantage when dealing with high-variation data compared to static methods.**

The evaluation of the cooperative prediction process consists of **the comparison of predictive values with actual traffic observations, prediction accuracy assessment, evaluation of the self-correction mechanism, and data storage**. Three key highlights emerge from this evaluation. Firstly, the self-correction mechanism proves efficient in correcting inaccurate predictions and limiting the degradation of prediction quality. Secondly, the number of activation instances of the self-correction mechanism remains reasonable, ensuring the low requirement on time and computational costs of ADRIP. Thirdly, ADRIP demonstrates efficient data storage while maintaining adequate prediction performance.

In summary, ADRIP significantly enhances traffic prediction, particularly in scenarios characterized by high traffic variation, outperforming current state-of-the-art methods. It also performs the timely update of predicted information to ensure accurate predictions. Furthermore, ADRIP demonstrates efficiency in data storage, emphasizing its ability to manage and store information effectively while guaranteeing robust and accurate predictions.

Chapter 9

Conclusion and Perspectives

9.1 General Conclusion

Aiming at mitigating traffic collisions and accidents at the End Of Queue of traffic congestion, the aim of this thesis is to answer the question: **How can we anticipate the existence of traffic congestion and provide this information for drivers/autonomous vehicles?** Nowadays, ITS offers Queue Warning Systems, leveraging V2X connectivity, to inform upcoming vehicles about jam queue status from leading vehicles. However, the performance of this technology is limited by the communication ranges. Therefore, the state-of-the-art has led us to a promising answer by estimating future traffic dynamics.

Nowadays, Big Data Analytics plays an essential role in many services of modern ITS since its development gets along with the advances in data collection and processing technologies. Their benefits have been demonstrated across numerous applications in improving the quality of services compared to traditional modeling approaches. Therefore, our proposal is based on the Big Data Analytics approach.

However, the state-of-the-art also shows the limitations of Big Data Analytics regarding the **data privacy, efficiency of the processing method, data storage, and openness**. To answer these challenges, our proposal ADRIP is designed based on dynamic clustering and adaptive MAS. Their ability for solving big data streams and complex problems across various domains motivates us to build ADRIP with the highlighted characteristics: **dynamic and openness**, answering the data storage and openness challenges by the adaptation of system for changes in the environment with low-rate data storage; **strong interpretability and ability for multi-traffic level application**, showing the efficiency of the processing method and **privacy friendly**, aiming limiting data exchanges and decentralization.

The learning process of ADRIP performs a dynamic clustering algorithm at the road network entity level, enabling distributed and decentralized learning. Our dynamic clustering shows the interactions between **Data Agent (DA)** and **Cluster Agent (CA)** to detect the set of CAs representing

different traffic dynamics at every road network entity. Through this algorithm, ADRIP learns and integrates each DA, representing data arriving in the communicating traffic streams, into the learned database of the associated road entity to adapt to the rapid and continuous changes of the driving environment. Therefore, the learning process in ADRIP addresses the **dynamic** characteristic. Additionally, the distribution of the learning process at the local agent level limits the centralized collection of data, enabling the **data privacy prediction**. Moreover, this local learning also allows ADRIP to function without disruption if agents enter or leave the system, thus enhancing its **openness**.

In parallel with the learning process, ADRIP performs **a cooperative prediction** based on AMAS theory. This process highlights the interactions between the Analyzer Agents (AAs) to compute future traffic dynamics jointly. Each AA uses the local learned database of its associated road entity and cooperates with the AAs of neighboring entities to determine the helpful information for the prediction estimation. The local learning and cooperative prediction processes complete and enhance the **quality of the calculation of prediction** by taking into account the spatio-temporal dependencies while maintaining **a reasonable model complexity and strong interpretability**.

In the conducted experiments, ADRIP is instantiated on two cases: **traffic predictions for microscopic and macroscopic data**. In both cases, ADRIP shows a better performance for traffic prediction when traffic data express high variations. The robust performance of ADRIP for different testing scenarios demonstrates **its ability for multi-data scales towards multi-level traffic predictions**.

9.2 Contributions

9.2.1 Scientific Contributions

The main contribution of this thesis is ADRIP which proposes an effective solution for prediction problems, addressing important components regarding the theoretical aspects: **temporal and spatial analysis, continuous learning for stream analysis, model interpretability**.

The local learning algorithm of each entity in ADRIP is based on the standard principle of the clustering approach. This approach relies on analyzing the historical data, identifying the existing patterns, and grouping the similar ones into clusters. The information within each cluster, representing similar data characteristics, can be used to predict future states by finding the closest cluster with current states and observing the next cluster's evolution after the historical data assigned to that cluster. Therefore, **the temporal dependence is shown across the transitions between clusters**. Besides, **the spatial dependence is studied through the interaction between neighboring entities**. This cooperation is essential to reinforce the necessary information used for computing predictions, especially when applying for traffic predictions to consider traffic propagation.

We tackle common issues encountered in predictive models within dynamic environments: **How can we update models and adapt to incoming data?** To overcome the under-fitting and over-

fitting of static predictive methods, we propose a continuous learning process based on a dynamic clustering approach. The clustering structure and the decision-making self-evolve to accommodate the incoming stream of data. The dynamic property of the dynamic clustering approach, on the one hand, allows the **flexibility of learning model** for the new data behaviors, on the other hand, ensures the fitted representation for all historical data without storing all of them, allowing **the efficient data storage**.

Additionally, ADRIP offers a predictive method characterized by a **high level of model interpretability**, facilitating the explanation of input-output causalities and avoid the black-box nature existing in NN-based models. Indeed, this explanation is clearly obtained by understanding the assignment decision in the dynamic clustering in the local learning process and the exchange protocol between agents in the cooperative prediction process. This characteristic is crucial for managing the quality of predictive information and providing a deeper understanding of the impacts of historical data on future estimates.

Lastly, ADRIP allows to **reduce the processing costs in terms of time, computational costs**. Indeed, the dynamic clustering and prediction algorithms in ADRIP do not contain the hyper-parameter needed to be tuned. Thus, time and computational costs are significantly reduced, enabling real-time prediction.

9.2.2 Industrial Contributions

In this work, ADRIP is applied to deal with traffic prediction problems that play an indispensable role in many ITS services. Overcoming the existing limitations of other solutions, ADRIP makes a significant contribution to enhancing ITS services by offering a robust solution for traffic prediction problems.

In the conducted experiments, ADRIP is tested under two cases using both simulated and real-world data sets under different scenarios. In the first case, ADRIP aims to provide traffic predictions for individual and connected vehicles. Therefore, the processed data is at the microscopic scale to detail the anticipated driving behaviors of vehicles on each road segment. In this scenario, traffic dynamics are defined as the Mobility Profiles. The second case employs traffic speeds as input to derive traffic predictions at the macroscopic scale. Results from comparisons with state-of-the-art methods in both cases consistently demonstrate the outperformance of ADRIP in predicting traffic dynamics within dynamic environments. This robust performance of ADRIP for two study cases with different data scales has highlighted its relevant properties regarding the applicative aspects, allowing it to deal with multi-level traffic prediction problems:

- **Openness:** ADRIP can operate without disruption, although the road network change or traffic entities are added or removed from the transport architecture.
- **Versatility:** The design of ADRIP architecture is generic for different application scenarios.
- **Dynamic:** ADRIP self-evolves to adapt to the changes of the environment.

In the applications for ITS, ADRIP aims to provide a long-term traffic prediction to overcome the limitations of the existing Queue Warning System (QWS) using V2X/V2E connectivity. Indeed, the anticipated traffic information allows vehicles to get informed about the potential traffic condition on road segments ahead in the *dead zone* where no connection is available for V2X/V2E communication. Additionally, the predicted information is computed and updated from historical data, adapting to the changes of the dynamic environment. Lastly, the warning horizon of QWS using V2X/V2E connectivity is limited by the communication range. Therefore, predictions offered by ADRIP with a long horizon can provide to drivers more reaction time to deal with dangerous situations.

9.3 Perspectives

The development of the initial version of ADRIP in this thesis has paved for many potential investigations and further researches.

9.3.1 Scientific Perspectives

The promising results of ADRIP open up many possibilities to investigate in future works to go further in the **learning, prediction processes and optimization of the self-correction mechanism**.

For the learning process, firstly, **a dynamic clustering method for road segment grouping** will be interesting to study to dynamically cluster the roads with similar traffic adapting to current traffic. The aim is to obtain a better traffic network segmentation, leading to a more comprehensive and more interpretative analysis of traffic dependence between urban areas. Furthermore, the definition of neighborhood in the prediction algorithm is only based on the road network, resulting in some road segments having few neighbors. The results obtained from conducted experiments revealed that road segments with fewer neighbors exhibit poorer performance compared to those with a greater number of neighbors. This grouping strategy will enable the identification of road segments with similar traffic evolution, thereby enhancing the prediction accuracy, particularly in the case of isolated road segments. Secondly, **an experiment using real-world data on a scenario with an important diversity of road types** will be interesting to investigate. In this case, the local similarity threshold α will be chosen differently, adapting to the limited speed of each road type that allows to show the adaptability and the dynamic of proposed models. Lastly, **the performance of ADRIP must be evaluated with data distribution shift problem** where the dynamic clustering method is tested on the unseen areas or traffic data arriving with new behaviors.

For the prediction process, we first aim to **extend the evaluation of ADRIP to other levels of road network architecture** to enhance its ability for diverse applications with different data scales. Secondly, the **detection of singular behaviors of vehicle data** will be investigated with different cases. For example, an online anomaly detection mechanism for traffic network data presented in [51] is interesting to explore. Thirdly, **the comparison criteria between configurations can also consider the seasonality of traffic, the fading of outdated data,** $etc.$ to filter the redundant

configurations and improve cooperation. Then, **an experiment of traffic prediction with varying percentages of connected cars sharing their data** will allow ADRIP to approach real-world applications and evaluate its performance when data is missing.

For the self-correction mechanism, optimizing the spread of re-computation in predictions involves investigating two interesting strategies. The first strategy locally occurs at AA that detects prediction errors. In fact, ADRIP predicts a chain of traffic dynamic transitions through different steps until reaching the required prediction horizon. Therefore, AA can verify if relaunching the prediction computation provides the new predicted traffic dynamics at each step. If the AA detects that the re-computation does not change the predictions at a step, it stops the re-computation process. The second strategy aims to restrict the spread of re-computation of predictions to neighboring AAs. As AAs construct their local traffic configurations, they may need to recompute predictions if errors occur at a neighboring AA. In this case, AA can also verify whether the re-computation modifies their predictions. If there is no impact, the AA stops the re-computation, thus limiting unnecessary computational processes.

Finally, conducting a study on the computational cost of ADRIP is essential to complete the evaluation. This cost includes the execution time of functions, the number of exchanged messages, and the response time of agents during their cooperation.

9.3.2 Application Perspectives

ADRIP opens a wide ranges of potential applications. The End Of Queue problems still remain challenging in ITS. The application of a traffic prediction system holds a promising solution for EoQ identification, detection, and anticipation, enabling the enhancement of traffic efficiency and safety. By leveraging historical traffic data and current observations, the system can estimate the future traffic conditions of road networks, including traffic congestion. The anticipated information on traffic congestion allows for predicting the appearance, formation, propagation, and location of the end of jam queues. This predictive insight enables the estimation of time to arrival between vehicles and the approaching queue. Additionally, the change in arrival time can be updated by observing the propagation of traffic jams. Moreover, this system, from the predictive information, can identify the parameters influencing the propagation of a traffic jam's wave and how these properties evolve from a free-flow regime to a congested state.

Integrating a traffic prediction system into collision avoidance systems represents a significant advancement in enhancing road safety and driving experience comfort. The predictive traffic conditions enable the identification of high-risk areas or situations where collisions are more likely to happen. Linking with the context of EoQ, where the jam is not totally formed, and the traffic speed quickly decreases, the predictive traffic conditions empower collision avoidance systems to activate some adaptive functions such as warning drivers, adjusting vehicle speed, or triggering emergency braking systems. Therefore, the application of a traffic prediction system in collision avoidance contributes to the improvement of traffic efficiency by reducing accidents as well as

traffic disruptions.

The two above applications directly address the issues in ITS using traffic prediction information. However, ADRIP can be applied to solve diverse problems that are modeled as a dynamic, distributed learning, and cooperative prediction. For instance, the dynamic clustering in the learning process of ADRIP can be deployed to detect regularities in users' itineraries. Nowadays, the movements of a user during the day can be followed using GPS-equipped mobile devices. Data are retrieved at a given rate and can be combined to identify users' itineraries. Finding itinerary regularities provides information on a user's regular movement behavior and enables adaptive recommendations or improving public transportation management. In this context, the dynamic clustering can be investigated to detect the regularity in user's movement behaviors. A generalized version of the dynamic clustering in ADRIP has been developed and tested in this application [142]. The obtained results show a promising opportunity to proceed to the next step of predictions.

Glossary

- **ITS** – Intelligent Transportation System
- **EoQ** – End Of Queue
- **QWS** – Queue Warning System
- **CV** – Connected Vehicle
- **ADAS** – Advanced Driver Assistance System
- **ATMS** – Advanced Traffic Management System
- **ATIS** – Advanced Traveler Information System
- **OD** – Origin-Destination
- **V2V** – Vehicle-to-Vehicle
- **V2I** – Vehicle-to-Infrastructure
- **V2X** – Vehicle-to-Everything
- **NCS** – Non Cooperative Situation
- **MAS** – Multi-Agent System
- **AMAS** – Adaptive Multi-Agent System
- **DA** – Data Agent

- **CA** – Cluster Agent
- **AA** – Analyzer Agent
- **MP** – Mobility Profile

Own Bibliography

National Conference Paper

- Ngo, Ha-Nhi, Elsy Kaddoum, Marie-Pierre Gleizes, Jonathan Bonnet, and Goursolle Anaïs. *P-ADRIP: un système multi-agent auto-organisateur pour la prévision du trafic routier*. In Journées Francophones sur les Systèmes Multi-Agents: SMA et Smart Cities (JFSMA 2022). 2022.

International Conference Papers

- Ngo, Ha-Nhi, Elsy Kaddoum, Marie-Pierre Gleizes, Jonathan Bonnet, and Goursolle Anaïs. *Life-long learning system of driving behaviors from vehicle data streams*. In 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), pp. 1132-1139. IEEE, 2021.

- Ngo, Ha-Nhi, Elsy Kaddoum, Marie-Pierre Gleizes, Jonathan Bonnet, and Goursolle Anaïs. *P-ADRIP: A multi-agent-based system for traffic forecasting*. In 14th ITS European Congress Toulouse: Smart and Sustainable Mobility for all (2022). 2022.

- Ha Nhi Ngo, Elsy Kaddoum, Matej Cebecauer, Erik Jenelius and Anaïs Goursolle. *Considering Multi-Scale Data for Continuous Traffic Prediction using Adaptive Multi-Agent System*. In 2023 IEEE International Intelligent Transportation Systems Conference (ITSC), IEEE, 2023.

- Ngo, Ha Nhi, Elsy Kaddoum, Marie-Pierre Gleizes, Jonathan Bonnet, and Anais Goursolle. *Continuous learning and cooperative prediction for traffic dynamics by Adaptive Multi-Agent Systems*. In 2023 IEEE International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS), pp. 17-26. IEEE Computer Society, 2023.

- Perles, Alexandre, Ha Nhi Ngo, Elsy Kaddoum, and Valérie Camps. *Adaptive Multi-agent System for Dynamic Clustering Applied to Itineraries Regularities and Traffic Prediction*. In International Conference on Cooperative Information Systems, pp. 79-96. Cham: Springer Nature Switzerland, 2023.

Presentations an Posters

- Session talk and Poster: *End-Of-Queue Prediction System*, Innovation Week at Continental Digital Services France.
- Session talk: *Traffic Prediction System*, Learn&Share day at Continental Digital Services France.
- Presentation: *Model and Predict End of Queue in Traffic Jam using Adaptive Multi Agent Systems*, Transport Planning Division, KTH Royal Institute of Technology, Stockholm, Sweden.
- Poster: *Generalized Traffic Prediction System using Adaptive Multi-Agent Approach for Different Traffic Levels*, Summer School BMW 2023.
- Poster: *Generalized Traffic Prediction System using Adaptive Multi-Agent Approach for Different Traffic Levels*, NeoCampus 2023.

Bibliography

[1] Module clustering in scikit-learn - python package. https://scikit-learn.org/stable/modules/clustering.html. Accessed: 2023-11-23.

[2] National its architecture. https://www.its.dot.gov/arch/index.htm. Accessed: 2023-09-24.

[3] Package clusopt core. http://https://pypi.org/project/clusopt-core/. Accessed: 2021-04-16.

[4] Ali R Abdellah and Andrey Koucheryavy. Vanet traffic prediction using lstm with deep neural network learning. In *Internet of Things, Smart Spaces, and Next Generation Networks and Systems: 20th International Conference, NEW2AN 2020, and 13th Conference, ruSMART 2020, St. Petersburg, Russia, August 26–28, 2020, Proceedings, Part I 20*, pages 281–294. Springer, 2020.

[5] Marcel R Ackermann, Marcus Märtens, Christoph Raupach, Kamil Swierkot, Christiane Lammersen, and Christian Sohler. Streamkm++ a clustering algorithm for data streams. *Journal of Experimental Algorithmics (JEA)*, 17:2–1, 2012.

[6] Charu C Aggarwal. *Data streams: models and algorithms*, volume 31. Springer, 2007.

[7] Charu C Aggarwal, S Yu Philip, Jiawei Han, and Jianyong Wang. A framework for clustering evolving data streams. In *Proceedings 2003 VLDB conference*, pages 81–92. Elsevier, 2003.

[8] Hirotugu Akaike. A new look at the statistical model identification. *IEEE transactions on automatic control*, 19(6):716–723, 1974.

[9] Ahmed Salih Al-Obaidi, Mohammed Ahmed Jubair, Izzatdin Abdul Aziz, Mohd Riduan Ahmad, Salama A Mostafa, Hairulnizam Mahdin, Abdullah Talaat Al-Tickriti, and Mustafa Hamid Hassan. Cauchy density-based algorithm for vanets clustering in 3d road environments. *IEEE Access*, 10:76376–76385, 2022.

[10] Francesco Alesiani, Konstantinos Gkiotsalitis, and Roberto Baldessari. A probabilistic activity model for predicting the mobility patterns of homogeneous social groups based on social network data. In *Proc. 93rd Annu. Meeting Transp. Res. Board*, 2014.

[11] Erik Almlöf, Isak Rubensson, Matej Cebecauer, and Erik Jenelius. Who is still travelling by public transport during covid-19? socioeconomic factors explaining travel behaviour in stockholm based on smart card data. *Preprint, available at SSRN. https://doi. org/10.2139/ssrn*, 3689091, 2020.

[12] Gülfem Işıklar Alptekin and Gülçin Büyüközkan. An integrated case-based reasoning and mcdm system for web based tourism destination planning. *Expert Systems with Applications*, 38(3):2125–2132, 2011.

[13] Flora Amato, Francesco Moscato, Vincenzo Moscato, Francesco Pascale, and Antonio Picariello. An agent-based approach for recommending cultural tours. *Pattern Recognition Letters*, 131:341–347, 2020.

[14] Larisa Angstenberger. *Dynamic fuzzy pattern recognition with applications to finance and engineering*. Springer Science & Business Media, 2001.

[15] Constantinos Antoniou, Haris N Koutsopoulos, and George Yannis. Dynamic data-driven local traffic state estimation and prediction. *Transportation Research Part C: Emerging Technologies*, 34:89–107, 2013.

[16] Afian Anwar, Amedeo Odoni, and Nelson Toh. Busviz: Big data for bus fleets. *Transportation Research Record*, 2544(1):102–109, 2016.

[17] Fabio Arena and Giovanni Pau. An overview of vehicular communications. *Future Internet*, 11(2):27, 2019.

[18] Riyaz Ahamed Ariyaluran Habeeb, Fariza Nasaruddin, Abdullah Gani, Mohamed Ahzam Amanullah, Ibrahim Abaker Targio Hashem, Ejaz Ahmed, and Muhammad Imran. Clustering-based real-time anomaly detection—a breakthrough in big data technologies. *Transactions on Emerging Telecommunications Technologies*, 33(8):e3647, 2022.

[19] David Arthur and Sergei Vassilvitskii. K-means++ the advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035, 2007.

[20] Nii Attoh-Okine. Big data challenges in railway engineering. In *2014 IEEE International conference on big data (Big Data)*, pages 7–9. IEEE, 2014.

[21] Davide Azzalini, Fabio Azzalini, Mirjana Mazuran, and Letizia Tanca. Tracking the evolution of financial time series clusters. In *Proceedings of the 5th Workshop on Data Science for Macro-modeling with Financial and Economic Datasets*, pages 1–5, 2019.

[22] M. Bagchi and P.R. White. The potential of public transport smart card data. *Transport Policy*, 12(5):464–474, 2005. Road User Charging: Theory and Practices.

[23] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[24] Michele Berlingerio, Francesco Calabrese, Giusy Di Lorenzo, Rahul Nair, Fabio Pinelli, and Marco Luca Sbodio. Allaboard: a system for exploring urban mobility and optimizing public transport using cellphone data. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part III 13*, pages 663–666. Springer, 2013.

[25] Amirreza Bigdeli, Abbas Maghsoudi, and Reza Ghezelbash. Application of self-organizing map (som) and k-means clustering algorithms for portraying geochemical anomaly patterns in moalleman district, ne iran. *Journal of Geochemical Exploration*, 233:106923, 2022.

[26] Daniel Billings and Jiann-Shiou Yang. Application of the arima models to urban roadway travel time prediction-a case study. In *2006 IEEE International Conference on Systems, Man and Cybernetics*, volume 3, pages 2529–2534. IEEE, 2006.

[27] Azzedine Boukerche and Jiahao Wang. Machine learning-based traffic prediction models for intelligent transportation systems. *Computer Networks*, 181:107530, 2020.

[28] George EP Box and Gwilym M Jenkins. Time series analysis. forecasting and control. *Holden-Day Series in Time Series Analysis*, 1976.

[29] Karel A Brookhuis, Dick De Waard, and Wiel H Janssen. Behavioural impacts of advanced driver assistance systems–an overview. *European Journal of Transport and Infrastructure Research*, 1(3), 2001.

[30] Pinlong Cai, Yunpeng Wang, Guangquan Lu, Peng Chen, Chuan Ding, and Jianping Sun. A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting. *Transportation Research Part C: Emerging Technologies*, 62:21–34, 2016.

[31] Feng Cao, Martin Estert, Weining Qian, and Aoying Zhou. Density-based clustering over an evolving data stream with noise. In *Proceedings of the 2006 SIAM international conference on data mining*, pages 328–339. SIAM, 2006.

[32] Davy Capera. *Systèmes multi-agents adaptatifs pour la résolution de problèmes: Application à la conception de mécanismes*. PhD thesis, Toulouse 3, 2005.

[33] Davy Capera, J-P Georgé, M-P Gleizes, and Pierre Glize. The amas theory for complex problem solving based on self-organizing cooperative agents. In *WET ICE 2003. Proceedings.*

Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003., pages 383–388. IEEE, 2003.

[34] Matthias Carnein and Heike Trautmann. Customer segmentation based on transactional data using stream clustering. In *Advances in Knowledge Discovery and Data Mining: 23rd Pacific-Asia Conference, PAKDD 2019, Macau, China, April 14-17, 2019, Proceedings, Part I 23*, pages 280–292. Springer, 2019.

[35] Matej Cebecauer, David Gundlegård, Erik Jenelius, and Wilco Burghout. 3d speed maps and mean observations vectors for short-term urban traffic prediction. In *Transportation research board annual meeting (TRB)*, pages 1–20, 2019.

[36] Matej Cebecauer, Erik Jenelius, and Wilco Burghout. Spatio-temporal partitioning of large urban networks for travel time prediction. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 1390–1395. IEEE, 2018.

[37] Matej Cebecauer, Erik Jenelius, David Gundlegård, and Wilco Burghout. Revealing representative day-types in transport networks using traffic data clustering. *Journal of Intelligent Transportation Systems*, 0(0):1–24, 2023.

[38] Santhana Chaimontree, Katie Atkinson, and Frans Coenen. A multi-agent based approach to clustering: Harnessing the power of agents. In *Agents and Data Mining Interaction, May 2-6, 2011, Revised Selected Papers 7*. Springer, 2012.

[39] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

[40] Suresh Chavhan and Pallapa Venkataram. Prediction based traffic management in a metropolitan area. *Journal of traffic and transportation engineering (English edition)*, 7(4):447–466, 2020.

[41] Cen Chen, Kenli Li, Sin G Teo, Xiaofeng Zou, Keqin Li, and Zeng Zeng. Citywide traffic flow prediction based on multiple gated spatio-temporal convolutional neural networks. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 14(4):1–23, 2020.

[42] Dewang Chen, Rong Chen, Yidong Li, and Tao Tang. Online learning algorithms for train automatic stop control using precise location data of balises. *IEEE Transactions on Intelligent Transportation Systems*, 14(3):1526–1535, 2013.

[43] Xiangyang Chen and Ruqing Chen. A review on traffic prediction methods for intelligent transportation system in smart cities. In *2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pages 1–5. IEEE, 2019.

[44] Zhiyi Chen, Yihui Xiong, Bojun Yin, Siquan Sun, and Renguang Zuo. Recognizing geochemical patterns related to mineralization using a self-organizing map. *Applied Geochemistry*, 151:105621, 2023.

[45] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[46] Malik Coleman, Lauren Tarte, Steve Chau, Brian Levine, and Alla Reddy. A data-driven approach to prioritizing bus schedule revisions at new york city transit. *Transportation Research Record*, 2672(8):86–95, 2018.

[47] Zhiyong Cui, Ruimin Ke, Ziyuan Pu, and Yinhai Wang. Deep bidirectional and unidirectional lstm recurrent neural network for network-wide traffic speed prediction. *arXiv preprint arXiv:1801.02143*, 2018.

[48] Guowen Dai, Changxi Ma, and Xuecai Xu. Short-term traffic flow prediction method for urban road sections based on space–time analysis and gru. *IEEE Access*, 7:143025–143035, 2019.

[49] Alaa Daoud, Flavien Balbo, Paolo Gianessi, and Gauthier Picard. A generic multi-agent model for resource allocation strategies in online on-demand transport with autonomous vehicles. In *Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021)*, page 3, 2021.

[50] William HE Day and Herbert Edelsbrunner. Efficient algorithms for agglomerative hierarchical clustering methods. *Journal of classification*, 1:7–24, 1984.

[51] Juliette Dromard and Philippe Owezarski. Integrating short history for improving clustering based network traffic anomaly detection. In *2017 IEEE 2nd International Workshops on Foundations and Applications of Self* Systems (FAS*W)*, pages 227–234, 2017.

[52] Oscar Egu and Patrick Bonnel. Investigating day-to-day variability of transit usage on a multimonth scale with smart card data. a case study in lyon. *Travel Behaviour and Society*, 19:112–123, 2020.

[53] Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

[54] 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.

[55] John C Falcocchio, Herbert S Levinson, John C Falcocchio, and Herbert S Levinson. The costs and other consequences of traffic congestion. *Road Traffic Congestion: A Concise Guide*, pages 159–182, 2015.

[56] Daniel B Fambro, Kay Fitzpatrick, and Rodger J Koppa. *Determination of stopping sight distances*, volume 400. Transportation Research Board, 1997.

[57] Hamed Faroqi, Mahmoud Mesbah, Jiwon Kim, and Ahmad Tavassoli. A model for measuring activity similarity between public transit passengers using smart card data. *Travel Behaviour and Society*, 13:11–25, 2018.

[58] Jacques Ferber, Fabien Michel, and José Báez. Agre: Integrating environments with organizations. In *International Workshop on Environments for Multi-Agent Systems*, pages 48–56. Springer, 2004.

[59] Evelyn Fix and Joseph Lawson Hodges. Discriminatory analysis. nonparametric discrimination: Consistency properties. *International Statistical Review/Revue Internationale de Statistique*, 57(3):238–247, 1989.

[60] Bernd Fritzke. A growing neural gas network learns topologies. *Advances in neural information processing systems*, 7, 1994.

[61] Xiao Fu and Yu Gu. Impact of a new metro line: analysis of metro passenger flow and travel time based on smart card data. *Journal of Advanced Transportation*, 2018, 2018.

[62] Mohamed Medhat Gaber, Arkady Zaslavsky, and Shonali Krishnaswamy. Mining data streams: a review. *ACM Sigmod Record*, 34(2):18–26, 2005.

[63] Angel García-Crespo, Javier Chamizo, Ismael Rivera, Myriam Mencke, Ricardo Colomo-Palacios, and Juan Miguel Gómez-Berbís. Speta: Social pervasive e-tourism advisor. *Telematics and informatics*, 26(3):306–315, 2009.

[64] José García-Rodríguez, Anastassia Angelopoulou, Juan Manuel García-Chamizo, Alexandra Psarrou, Sergio Orts Escolano, and Vicente Morell Giménez. Autonomous growing neural gas for applications with time constraint: optimal parameter estimation. *Neural Networks*, 32:196–208, 2012.

[65] Taher M Ghazal, Nidal A Al-Dmour, Raed A Said, Alireza Omidvar, Urooj Yousuf Khan, Tariq Rahim Soomro, Haitham M Alzoubi, Muhammad Alshurideh, Tamer Mohamed Abdellatif, Abdullah Moubayed, et al. Ddos intrusion detection with ensemble stream mining for iot smart sensing devices. In *The Effect of Information Technology on Business and Marketing Intelligence Systems*, pages 1987–2012. Springer, 2023.

[66] Mohammed Ghesmoune, Mustapha Lebbah, and Hanene Azzag. Clustering over data streams based on growing neural gas. In *Advances in Knowledge Discovery and Data Mining: 19th Pacific-Asia Conference, PAKDD 2015, Ho Chi Minh City, Vietnam, May 19-22, 2015, Proceedings, Part II 19*, pages 134–145. Springer, 2015.

[67] Mohammed Ghesmoune, Mustapha Lebbah, and Hanene Azzag. State-of-the-art on clustering data streams. *Big Data Analytics*, 1:13, 12 2016.

[68] Marie-Pierre Gleizes. Self-adaptive complex systems. In *European Workshop on Multi-Agent Systems*, pages 114–128. Springer, 2011.

[69] Pierre Glize. L'adaptation des systèmes à fonctionnalité émergente par auto-organisation coopérative. *Hdr, Université Paul Sabatier, Toulouse III*, 2001.

[70] Alexey Golubev, Ilya Chechetkin, Konstantin S Solnushkin, Natalia Sadovnikova, Danila Parygin, and Maxim Shcherbakov. Strategway: web solutions for building public transportation routes using big geodata analysis. In *Proceedings of the 17th international conference on information integration and web-based applications & services*, pages 1–4, 2015.

[71] Sergey Grachev, Petr Skobelev, Igor Mayorov, and Elena Simonova. Adaptive clustering through multi-agent technology: Development and perspectives. *Mathematics*, 8(10), 2020.

[72] Arnaud Grignard, Patrick Taillandier, Benoit Gaudou, Duc An Vo, Nghi Quang Huynh, and Alexis Drogoul. Gama 1.6: Advancing the art of complex agent-based modeling and simulation. In Guido Boella, Edith Elkind, Bastin Tony Roy Savarimuthu, Frank Dignum, and Martin K. Purvis, editors, *PRIMA 2013: Principles and Practice of Multi-Agent Systems*. Springer Berlin Heidelberg, 2013.

[73] Matt Grote, Ian Williams, John Preston, and Simon Kemp. Including congestion effects in urban road traffic co2 emissions modelling: Do local government authorities have the right options? *Transportation Research Part D: Transport and Environment*, 43:95–106, 2016.

[74] Maxime Guériau, Frédéric Armetta, Salima Hassas, Romain Billot, and Nour-Eddin El Faouzi. A constructivist approach for a self-adaptive decision-making system: application to road traffic control. In *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 670–677. IEEE, 2016.

[75] Shengnan Guo, Youfang Lin, Shijie Li, Zhaoming Chen, and Huaiyu Wan. Deep spatial–temporal 3d convolutional neural networks for traffic data forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 20(10):3913–3926, 2019.

[76] Jeffrey Hanft, Shrisan Iyer, Brian Levine, and Alla Reddy. Transforming bus service planning using integrated electronic data sources at nyc transit. *Journal of Public Transportation*, 19(2):89–108, 2016.

[77] Hamssa Hasrouny, Abed Ellatif Samhat, Carole Bassil, and Anis Laouiti. Vanet security challenges and solutions: A survey. *Vehicular Communications*, 7:7–20, 2017.

[78] Raheleh Hassannia, Ali Vatankhah Barenji, Zhi Li, and Habib Alipour. Web-based recommendation system for smart tourism: Multiagent technology. *Sustainability*, 11(2):323, 2019.

[79] Dieter Hendricks. Using real-time cluster configurations of streaming asynchronous features as online state descriptors in financial markets. *Pattern Recognition Letters*, 97:21–28, 2017.

[80] Dwight A. Hennessy and David L. Wiesenthal. Traffic congestion, driver stress, and driver aggression. *Aggressive Behavior*, 25(6):409–423, 1999.

[81] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[82] Zhongsheng Hou, Yi Wang, Chenkun Yin, and Tao Tang. Terminal iterative learning control based station stop control of a train. *International Journal of Control*, 84(7):1263–1274, 2011.

[83] Fang-Ming Hsu, Yu-Tzeng Lin, and Tu-Kuang Ho. Design and implementation of an intelligent recommendation system for tourist attractions: The integration of ebm model, bayesian network and google maps. *Expert Systems with Applications*, 39(3):3257–3264, 2012.

[84] Jesper Bláfoss Ingvardson, Otto Anker Nielsen, Sebastián Raveau, and Bo Friis Nielsen. Passenger arrival and waiting time distributions dependent on train service frequency and station characteristics: A smart card data analysis. *Transportation Research Part C: Emerging Technologies*, 90:292–306, 2018.

[85] Charlie Isaksson, Margaret H Dunham, and Michael Hahsler. Sostream: Self organizing density-based clustering over data stream. In *International workshop on machine learning and data mining in pattern recognition*, pages 264–278. Springer, 2012.

[86] Masahiko Itoh, Daisaku Yokoyama, Masashi Toyoda, Yoshimitsu Tomita, Satoshi Kawamura, and Masaru Kitsuregawa. Visual fusion of mega-city big data: an application to traffic and tweets data analysis of metro passengers. In *2014 IEEE International Conference on Big Data (Big Data)*, pages 431–440. IEEE, 2014.

[87] Anil K Jain and Richard C Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[88] Meenal Jain, Gagandeep Kaur, and Vikas Saxena. A k-means clustering and svm based hybrid concept drift detection technique for network anomaly detection. *Expert Systems with Applications*, 193:116510, 2022.

[89] JQ James, Christos Markos, and Shiyao Zhang. Long-term urban traffic speed prediction with deep learning on graphs. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):7359–7370, 2021.

[90] Cynthia Jayapal and S Sujith Roy. Road traffic congestion management using vanet. In *2016 International conference on advances in human machine interaction (HMI)*, pages 1–7. IEEE, 2016.

[91] Yuxuan Ji and Nikolas Geroliminis. On the spatial partitioning of urban transportation networks. *Transportation Research Part B: Methodological*, 46(10):1639–1656, 2012.

[92] Zhibin Jiang, Ching-Hsien Hsu, Daqiang Zhang, and Xiaolei Zou. Evaluating rail transit timetable using big passengers' data. *Journal of Computer and System Sciences*, 82(1):144–155, 2016.

[93] Junchen Jin, Dingding Rong, Yuqi Pang, Peijun Ye, Qingyuan Ji, Xiao Wang, Ge Wang, and Fei-Yue Wang. An agent-based traffic recommendation system: Revisiting and revising urban traffic management strategies. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52(11):7289–7301, 2022.

[94] Sepideh Kaffash, An Truong Nguyen, and Joe Zhu. Big data algorithms and applications in intelligent transportation system: A review and bibliometric analysis. *International Journal of Production Economics*, 231:107868, 2021.

[95] Yiannis Kamarianakis and Poulicos Prastacos. Forecasting traffic flow conditions in an urban network: Comparison of multivariate and univariate approaches. *Transportation Research Record*, 1857(1):74–84, 2003.

[96] Habib M Kammoun, Ilhem Kallel, Jorge Casillas, Ajith Abraham, and Adel M Alimi. Adapt-traf: An adaptive multiagent road traffic management system based on hybrid ant-hierarchical fuzzy model. *Transportation Research Part C: Emerging Technologies*, 42:147–167, 2014.

[97] Christos Karras, Aristeidis Karras, and Spyros Sioutas. Pattern recognition and event detection on iot data-streams. *arXiv preprint arXiv:2203.01114*, 2022.

[98] Christos N. Karras, Aristeidis Karras, Georgios Drakopoulos, Konstantinos Tsakalidis, Phivos Mylonas, and Spyros Sioutas. Weighted reservoir sampling on evolving streams: A sampling algorithmic framework for stream event identification. *Proceedings of the 12th Hellenic Conference on Artificial Intelligence*, 2022.

[99] Jiwon Kim, Jonathan Corcoran, and Marty Papamanolis. Route choice stickiness of public transport passengers: Measuring habitual bus ridership behaviour using smart card data. *Transportation Research Part C: Emerging Technologies*, 83:146–164, 2017.

[100] Akhila Kishore, Anhad Bhasin, Arun Balaji, Chandrasekar Vuppalapati, Divyesh Jadav, Preethi Anantharaman, and Shrutee Gangras. Cense: A cognitive navigation system for people with special needs. In *2017 IEEE third international conference on big data computing service and applications (BigDataService)*, pages 198–203. IEEE, 2017.

[101] Lawrence A Klein, Milton K Mills, David Gibson, and Lawrence A Klein. Traffic detector handbook: Volume ii. Technical report, United States. Federal Highway Administration, 2006.

[102] Teuvo Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.

[103] Taiwo Kolajo, Olawande Daramola, and Ayodele A Adebiyi. Real-time event detection in social media streams through semantic analysis of noisy terms. *Journal of Big Data*, 9(1):1–36, 2022.

[104] Deyu Kong, Xike Xie, and Zhuoxu Zhang. Clustering-based partitioning for large web graphs. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 593–606. IEEE, 2022.

[105] Philipp Kranen, Ira Assent, Corinna Baldauf, and Thomas Seidl. The clustree: indexing micro-clusters for anytime stream mining. *Knowledge and information systems*, 29:249–272, 2011.

[106] Kranti Kumar, Manoranjan Parida, and Vinod Kumar Katiyar. Short term traffic flow prediction in heterogeneous condition using artificial neural network. *Transport*, 30(4):397–405, 2015.

[107] Ibai Lana, Javier Del Ser, Manuel Velez, and Eleni I Vlahogianni. Road traffic forecasting: Recent advances and new challenges. *IEEE Intelligent Transportation Systems Magazine*, 10(2):93–109, 2018.

[108] Robert Layton and Karen Dixon. Stopping sight distance. *Kiewit Center for Infrastructure and Transportation, Oregon Department of Transportation*, 2012.

[109] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[110] Sangsoo Lee and Daniel B Fambro. Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting. *Transportation Research Record*, 1678(1):179–188, 1999.

[111] Guofa Li, Weijian Lai, Xiaoxuan Sui, Xiaohang Li, Xingda Qu, Tingru Zhang, and Yuezhi Li. Influence of traffic congestion on driver behavior in post-congestion driving. *Accident Analysis & Prevention*, 141:105508, 2020.

[112] Hongfei Li, Dhaivat Parikh, Qing He, Buyue Qian, Zhiguo Li, Dongping Fang, and Arun Hampapur. Improving rail network velocity: A machine learning approach to predictive maintenance. *Transportation Research Part C: Emerging Technologies*, 45:17–26, 2014.

[113] Yang Li, Xudong Wang, Shuo Sun, Xiaolei Ma, and Guangquan Lu. Forecasting short-term subway passenger flow under special events scenarios using multiscale radial basis function networks. *Transportation Research Part C: Emerging Technologies*, 77:306–328, 2017.

[114] Zhibin Li, Seongchae Ahn, Koohong Chung, David R Ragland, Wei Wang, and Jeong Whon Yu. Surrogate safety measure for evaluating rear-end collision risk related to kinematic waves near freeway recurrent bottlenecks. *Accident Analysis & Prevention*, 64:52–61, 2014.

[115] Zhibin Li, Koohong Chung, and Michael J Cassidy. Collisions in freeway traffic: influence of downstream queues and interim means to address them. *Transportation research record*, 2396(1):1–9, 2013.

[116] Lijuan Liu and Rung-Ching Chen. A novel passenger flow prediction model using deep learning methods. *Transportation Research Part C: Emerging Technologies*, 84:74–91, 2017.

[117] Long Liu, Jin Xu, Stephen Shaoyi Liao, and Huaping Chen. A real-time personalized route recommendation system for self-drive tourists based on vehicle to vehicle communication. *Expert Systems with Applications*, 41(7):3409–3417, 2014.

[118] Yongxin Liu, Xiaoxiong Weng, Jiafu Wan, Xuejun Yue, Houbing Song, and Athanasios V Vasilakos. Exploring data validity in transportation systems for smart cities. *IEEE Communications Magazine*, 55(5):26–33, 2017.

[119] Clélia Lopez, Ludovic Leclercq, Panchamy Krishnakumari, Nicolas Chiabaut, and Hans Van Lint. Revealing the day-to-day regularity of urban congestion patterns with 3d speed maps. *Scientific Reports*, 7(1):14029, 2017.

[120] Saiqun Lu, Qiyan Zhang, Guangsen Chen, and Dewen Seng. A combined method for short-term traffic flow prediction based on recurrent neural network. *Alexandria Engineering Journal*, 60(1):87–94, 2021.

[121] Joel P Lucas, Nuno Luz, María N Moreno, Ricardo Anacleto, Ana Almeida Figueiredo, and Constantino Martins. A hybrid recommendation approach for a tourism system. *Expert systems with applications*, 40(9):3532–3550, 2013.

[122] Ling Luo, Bin Li, Xuhui Fan, Yang Wang, Irena Koprinska, and Fang Chen. Dynamic customer segmentation via hierarchical fragmentation-coagulation processes. *Machine Learning*, 112(1):281–310, 2023.

[123] Wenchi Ma and Ruijie Wang. Traffic flow forecasting research based on bayesian normalized elman neural network. In *2015 IEEE Signal Processing and Signal Processing Education Workshop (SP/SPE)*, pages 426–430. IEEE, 2015.

[124] Xiaolei Ma, Zhuang Dai, Zhengbing He, Jihui Ma, Yong Wang, and Yunpeng Wang. Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction. *Sensors*, 17(4):818, 2017.

[125] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[126] Guilhem Marcillaud. *Implementation of a cooperative communication within a fleet of connected and autonomous vehicles*. PhD thesis, Université Paul Sabatier-Toulouse III, 2022.

[127] Thomas Martinetz, Klaus Schulten, et al. A" neural-gas" network learns topologies. 1991.

[128] Viktor Mayer-Schönberger and Kenneth Cukier. *Big data: A revolution that will transform how we live, work, and think*. Houghton Mifflin Harcourt, 2013.

[129] Ryszard Stanislaw Michalski, Jaime Guillermo Carbonell, and Tom M Mitchell. *Machine learning: An artificial intelligence approach*. Springer Science & Business Media, 2013.

[130] Wanli Min and Laura Wynter. Real-time road traffic prediction with spatio-temporal correlations. *Transportation Research Part C: Emerging Technologies*, 19(4):606–616, 2011.

[131] Luntian Mou, Pengfei Zhao, Haitao Xie, and Yanyan Chen. T-lstm: A long short-term memory neural network enhanced by temporal information for traffic flow prediction. *Ieee Access*, 7:98053–98060, 2019.

[132] Abdallah Moujahid, Mounir ElAraki Tantaoui, Manolo Dulva Hina, Assia Soukane, Andrea Ortalda, Ahmed ElKhadimi, and Amar Ramdane-Cherif. Machine learning techniques in adas: A review. In *2018 International Conference on Advances in Computing and Communication Engineering (ICACCE)*, pages 235–242. IEEE, 2018.

[133] Shirin Najafabadi, Ali Hamidi, Mahdieh Allahviranloo, and Naresh Devineni. Does demand for subway ridership in manhattan depend on the rainfall events? *Transport Policy*, 74:201–213, 2019.

[134] Nancy L Nihan and Kjell O Holmesland. Use of the box and jenkins time series technique in traffic forecasting. *Transportation*, 9(2):125–143, 1980.

[135] Ann Nowé, Peter Vrancx, and Yann-Michaël De Hauwere. Game theory and multi-agent reinforcement learning. *Reinforcement Learning: State-of-the-Art*, pages 441–470, 2012.

[136] Cheol Oh, Seri Park, and Stephen G Ritchie. A method for identifying rear-end collision risks using inductive loop detectors. *Accident Analysis & Prevention*, 38(2):295–301, 2006.

[137] World Health Organization. *Global status report on road safety 2015*. World Health Organization, 2015.

[138] Qi Ouyang, Yongbo Lv, Yuan Ren, Jihui Ma, and Jing Li. Passenger travel regularity analysis based on a large scale smart card data. *Journal of Advanced Transportation*, 2018, 2018.

[139] Julio-Omar Palacio-Niño and Fernando Berzal. Evaluation metrics for unsupervised learning algorithms. *arXiv preprint arXiv:1905.05667*, 2019.

[140] Brendan Pender, Graham Currie, Alexa Delbosc, and Nirajan Shiwakoti. Social media use during unplanned transit network disruptions: A review of literature. *Transport reviews*, 34(4):501–521, 2014.

[141] Hao Peng, Bowen Du, Mingsheng Liu, Mingzhe Liu, Shumei Ji, Senzhang Wang, Xu Zhang, and Lifang He. Dynamic graph convolutional network for long-term traffic flow prediction with reinforcement learning. *Information Sciences*, 578:401–416, 2021.

[142] Alexandre Perles, Ha Nhi Ngo, Elsy Kaddoum, and Valérie Camps. Adaptive multi-agent system for dynamic clustering applied to itineraries regularities and traffic prediction. In *International Conference on Cooperative Information Systems*, pages 79–96. Springer, 2023.

[143] Phillip E Pfeifer and Stuart Jay Deutrch. A three-stage iterative procedure for space-time modeling phillip. *Technometrics*, 22(1):35–47, 1980.

[144] Giovanny Pillajo-Quijia, Blanca Arenas-Ramírez, Camino González-Fernández, and Francisco Aparicio-Izquierdo. Influential factors on injury severity for drivers of light trucks and vans with machine learning methods. *Sustainability*, 12(4):1324, 2020.

[145] Nicholas G Polson and Vadim O Sokolov. Deep learning for short-term traffic flow prediction. *Transportation Research Part C: Emerging Technologies*, 79:1–17, 2017.

[146] Luo Qi. Research on intelligent transportation system technologies and applications. In *2008 Workshop on Power Electronics and Intelligent Transportation System*, pages 529–531. IEEE, 2008.

[147] Yingchun Qu, Hang Gong, and Pu Wang. Transportation mode split with mobile phone data. In *2015 IEEE 18th international conference on intelligent transportation systems*, pages 285–289. IEEE, 2015.

[148] Jithin Raj, Hareesh Bahuleyan, and Lelitha Devi Vanajakshi. Application of data mining techniques for traffic density estimation and prediction. *Transportation Research Procedia*, 17:321–330, 2016.

[149] Carl Rasmussen. The infinite gaussian mixture model. *Advances in neural information processing systems*, 12, 1999.

[150] D Richards and Richard Cuerden. The relationship between speed and car driver injury severity. *Road Safety Web Publication*, 2009.

[151] Beatriz Rodríguez, Julián Molina, Fátima Pérez, and Rafael Caballero. Interactive design of personalised tourism routes. *Tourism Management*, 33(4):926–940, 2012.

[152] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[153] Stephane Ross, Paul Mineiro, and John Langford. Normalized online learning. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, pages 537–545, 2013.

[154] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[155] Stuart J Russell. *Artificial intelligence a modern approach*. Pearson Education, Inc., 2010.

[156] Mohammadreza Saeedmanesh and Nikolas Geroliminis. Clustering of heterogeneous networks with directional flows based on "snake" similarities. *Transportation Research Part B: Methodological*, 91:250–269, 2016.

[157] Athanasios Salamanis, Giorgos Margaritis, Dionysios D Kehagias, Georgios Matzoulas, and Dimitrios Tzovaras. Identifying patterns under both normal and abnormal traffic conditions for short-term traffic prediction. *Transportation research procedia*, 22:665–674, 2017.

[158] Fernando Martínez Santiago, Francisco Ariza López, Arturo Montejo-Ráez, and Alfonso Ureña López. Geoasis: A knowledge-based geo-referenced tourist assistant. *Expert Systems with Applications*, 39(14):11737–11745, 2012.

[159] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

[160] Sanaz Shaker Sepasgozar and Samuel Pierre. Network traffic prediction model considering road traffic parameters using artificial intelligence methods in vanet. *IEEE Access*, 10:8227–8242, 2022.

[161] Huang Shenghua, Niu Zhihua, and Huang Jiaxin. Road traffic congestion prediction based on random forest and dbscan combined model. In *2020 5th International Conference on Smart Grid and Electrical Automation (ICSGEA)*, pages 323–326. IEEE, 2020.

[162] Jonathan A Silva, Elaine R Faria, Rodrigo C Barros, Eduardo R Hruschka, André CPLF de Carvalho, and João Gama. Data stream clustering: A survey. *ACM Computing Surveys (CSUR)*, 46(1):1–31, 2013.

[163] Isaac J Sledge and James M Keller. Growing neural gas for temporal clustering. In *2008 19th International Conference on Pattern Recognition*, pages 1–4. IEEE, 2008.

[164] Amir Sobhani, William Young, David Logan, and Sareh Bahrololoom. A kinetic energy model of two-vehicle crash injury severity. *Accident Analysis & Prevention*, 43(3):741–754, 2011.

[165] Jérémy Sobieraj. *Methodes et outils pour la conception de Systemes de Transport Intelligents Cooperatifs*. PhD thesis, Université Paris-Saclay; Université d'Evry-Val-d'Essonne, 2018.

[166] Yanchao Song, Siyuan Kou, and Chen Wang. Modeling crash severity by considering risk indicators of driver and roadway: A bayesian network approach. *Journal of safety research*, 76:64–72, 2021.

[167] Veronika Stephanie, MAP Chamikara, Ibrahim Khalil, and Mohammed Atiquzzaman. Privacy-preserving location data stream clustering on mobile edge computing and cloud. *Information Systems*, 107:101728, 2022.

[168] Jau-Ming Su, Nomungerel Erdenebat, Liang-Hua Ho, and Yu-Ting Zhan. Integration of transit demand and big data for bus route design in taiwan. In *Bridging the East and West*, pages 19–26. 2016.

[169] Peng Sun, Azzedine Boukerche, and Yanjie Tao. Ssgru: A novel hybrid stacked gru-based traffic volume prediction approach in a road network. *Computer Communications*, 160:502–511, 2020.

[170] Shiliang Sun, Rongqing Huang, and Ya Gao. Network-scale traffic modeling and forecasting with graphical lasso and neural networks. *Journal of Transportation Engineering*, 138(11):1358–1367, 2012.

[171] Madhar Taamneh, Sharaf Alkheder, and Salah Taamneh. Data-mining techniques for traffic accident modeling and prediction in the united arab emirates. *Journal of Transportation Safety & Security*, 9(2):146–166, 2017.

[172] Patrick Taillandier. Traffic simulation with the gama platform. In *Eighth International Workshop on Agents in Traffic and Transportation*, pages 8–p, 2014.

[173] Adithya Thaduri, Diego Galar, and Uday Kumar. Railway assets: A potential domain for big data analytics. *Procedia Computer Science*, 53:457–467, 2015.

[174] Yongxue Tian and Li Pan. Predicting short-term traffic flow by long short-term memory recurrent neural network. In *2015 IEEE international conference on smart city/SocialCom/SustainCom (SmartCity)*, pages 153–158. IEEE, 2015.

[175] Durga Toshniwal, Narayan Chaturvedi, Manoranjan Parida, Archit Garg, Chirag Choudhary, and Yashpal Choudhary. Application of clustering algorithms for spatio-temporal analysis of urban traffic data. *Transportation Research Procedia*, 48:1046–1059, 2020.

[176] Martin Treiber and Arne Kesting. Traffic flow dynamics. *Traffic Flow Dynamics: Data, Models and Simulation, Springer-Verlag Berlin Heidelberg*, 2013.

[177] Jonathan Tutcher. Ontology-driven data integration for railway asset monitoring applications. In *2014 IEEE International Conference on Big Data (Big Data)*, pages 85–95. IEEE, 2014.

[178] Komkrit Udommanetanakit, Thanawin Rakthanmanon, and Kitsana Waiyamai. E-stream: Evolution-based technique for stream clustering. In *Advanced Data Mining and Applications: Third International Conference, ADMA 2007 Harbin, China, August 6-8, 2007. Proceedings 3*, pages 605–615. Springer, 2007.

[179] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17:395–416, 2007.

[180] Chang-Dong Wang, Jian-Huang Lai, Dong Huang, and Wei-Shi Zheng. Svstream: A support vector-based algorithm for clustering data streams. *IEEE Transactions on Knowledge and Data Engineering*, 25(6):1410–1424, 2011.

[181] Senzhang Wang, Meiyue Zhang, Hao Miao, Zhaohui Peng, and Philip S Yu. Multivariate correlation-aware spatio-temporal graph convolutional networks for multi-scale traffic prediction. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13(3):1–22, 2022.

[182] Xiaoyang Wang, Yao Ma, Yiqi Wang, Wei Jin, Xin Wang, Jiliang Tang, Caiyan Jia, and Jian Yu. Traffic flow prediction via spatial temporal graph neural network. In *Proceedings of the web conference 2020*, pages 1082–1092, 2020.

[183] Yizhe Wang, Xiaoguang Yang, Hailun Liang, Yangdong Liu, et al. A review of the self-adaptive traffic signal control system based on future traffic environment. *Journal of Advanced Transportation*, 2018, 2018.

[184] Yunhao Wang, Yiming Bie, and Qinhe An. Impacts of winter weather on bus travel time in cold regions: case study of harbin, china. *Journal of Transportation Engineering, Part A: Systems*, 144(11):05018001, 2018.

[185] Ming Wei, Jonathan Corcoran, Thomas Sigler, and Yan Liu. Modeling the influence of weather on transit ridership: A case study from brisbane, australia. *Transportation Research Record*, 2672(8):505–510, 2018.

[186] Gerhard Weiss. *Multiagent systems: a modern approach to distributed artificial intelligence*. MIT press, 1999.

[187] Billy M Williams, Priya K Durvasula, and Donald E Brown. Urban freeway traffic flow prediction: application of seasonal autoregressive integrated moving average and exponential smoothing models. *Transportation Research Record*, 1644(1):132–141, 1998.

[188] Billy M Williams and Lester A Hoel. Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results. *Journal of transportation engineering*, 129(6):664–672, 2003.

[189] Chengcheng Xu, Pan Liu, Wei Wang, and Zhibin Li. Evaluation of the impacts of traffic states on crash risks on freeways. *Accident Analysis & Prevention*, 47:162–171, 2012.

[190] Dongkuan Xu and Yingjie Tian. A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2:165–193, 2015.

[191] Miaomiao Yan and Yindong Shen. Traffic accident severity prediction based on random forest. *Sustainability*, 14(3):1729, 2022.

[192] Xin Yang, Qiuchi Xue, Meiling Ding, Jianjun Wu, and Ziyou Gao. Short-term prediction of passenger volume for urban rail systems: A deep learning approach based on smart-card data. *International Journal of Production Economics*, 231:107920, 2021.

[193] Yun Yang, ChongJun Fan, Liang Chen, and HongLin Xiong. Ipmod: An efficient outlier detection model for high-dimensional medical data streams. *Expert Systems with Applications*, 191:116212, 2022.

[194] Hwasoo Yeo, Kitae Jang, and Alexander Skabardonis. Impact of traffic states on freeway collision frequency. 2010.

[195] Jiateng Yin, Dewang Chen, and Yidong Li. Smart train operation algorithms based on expert knowledge and ensemble cart for the electric locomotive. *Knowledge-based systems*, 92:78–91, 2016.

[196] Haiyang Yu, Zhihai Wu, Shuqin Wang, Yunpeng Wang, and Xiaolei Ma. Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks. *Sensors*, 17(7):1501, 2017.

[197] Rose Yu, Yaguang Li, Cyrus Shahabi, Ugur Demiryurek, and Yan Liu. Deep learning: A generic approach for extreme condition traffic forecasting. In *Proceedings of the 2017 SIAM international Conference on Data Mining*, pages 777–785. SIAM, 2017.

[198] Tingting Yuan, Wilson Da Rocha Neto, Christian Esteve Rothenberg, Katia Obraczka, Chadi Barakat, and Thierry Turletti. Machine learning for next-generation intelligent transportation systems: A survey. *Transactions on emerging telecommunications technologies*, 33(4):e4427, 2022.

[199] Noureen Zafar, Irfan Ul Haq, Jawad-ur-Rehman Chughtai, and Omair Shafiq. Applying hybrid lstm-gru model based on heterogeneous data sources for traffic speed prediction in urban areas. *Sensors*, 22(9):3348, 2022.

[200] Allan M Zarembski. Some examples of big data in railroad engineering. In *2014 IEEE International Conference on Big Data (Big Data)*, pages 96–102. IEEE, 2014.

[201] Da Zhang and Mansur R Kabuka. Combining weather condition data to predict traffic flow: a gru-based deep learning approach. *IET Intelligent Transport Systems*, 12(7):578–585, 2018.

[202] Jian Zhang, Zhibin Li, Ziyuan Pu, and Chengcheng Xu. Comparing prediction performance for crash injury severity among various machine learning and statistical methods. *IEEE Access*, 6:60079–60087, 2018.

[203] Lun Zhang, Qiuchen Liu, Wenchen Yang, Nai Wei, and Decun Dong. An improved k-nearest neighbor model for short-term traffic flow prediction. *Procedia-Social and Behavioral Sciences*, 96:653–662, 2013.

[204] Shanqi Zhang, Yu Yang, Feng Zhen, Tashi Lobsang, and Zhixuan Li. Understanding the travel behaviors and activity patterns of the vulnerable population using smart card data: An activity space-based approach. *Journal of Transport Geography*, 90:102938, 2021.

[205] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Birch: an efficient data clustering method for very large databases. *ACM sigmod record*, 25(2):103–114, 1996.

[206] Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE transactions on intelligent transportation systems*, 21(9):3848–3858, 2019.

[207] Yuexu Zhao and Wei Deng. Prediction in traffic accident duration based on heterogeneous ensemble learning. *Applied Artificial Intelligence*, 36(1):2018643, 2022.

[208] Zhili Zhou, Xiaohua Dong, Zhetao Li, Keping Yu, Chun Ding, and Yimin Yang. Spatio-temporal feature encoding for traffic accident detection in vanet environment. *IEEE Transactions on Intelligent Transportation Systems*, 23(10):19772–19781, 2022.

[209] Hao Zhu, Ka-Veng Yuen, Lyudmila Mihaylova, and Henry Leung. Overview of environment perception for intelligent vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 18(10):2584–2601, 2017.

[210] Li Zhu, Fei Richard Yu, Yige Wang, Bin Ning, and Tao Tang. Big data analytics in intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 20(1):383–398, 2018.

[211] Zheng Zhu, Meng Xu, Jintao Ke, Hai Yang, and Xiqun Michael Chen. A bayesian clustering ensemble gaussian process model for network-wide traffic flow clustering and prediction. *Transportation Research Part C: Emerging Technologies*, 148:104032, 2023.