



**HAL**  
open science

# Prédiction des intentions des véhicules pour la conduite autonome en utilisant l'intelligence artificielle

Abdelmoudjib Benterki

► **To cite this version:**

Abdelmoudjib Benterki. Prédiction des intentions des véhicules pour la conduite autonome en utilisant l'intelligence artificielle. Automatique / Robotique. Université Paris-Saclay, 2021. Français. NNT : 2021UPAST077 . tel-03917117

**HAL Id: tel-03917117**

**<https://theses.hal.science/tel-03917117>**

Submitted on 1 Jan 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Prédiction des intentions des véhicules  
pour la conduite autonome en utilisant  
l'intelligence artificielle

*Predicting vehicle intentions for autonomous  
driving using artificial intelligence*

**Thèse de doctorat de l'université Paris-Saclay**

École doctorale n° 575 Electrical, Optical, Bio: PHYSICS-AND-ENGINEERING (EOBE)

Spécialité de doctorat : Génie Electrique

Unité de recherche : ESTACA'Lab, Pôle Systèmes et Energie Embarqués pour le Transport (S2ET), Avenue Paul Delouvrier 78180 Montigny-le-Bretonneux

Réfèrent : Faculté des sciences d'Orsay

**Thèse présentée et soutenue à Montigny-le-Bretonneux,  
le 15 juillet 2021, par**

**Abdelmoudjib BENTERKI**

**Composition du Jury**

**Fethi BEN OUEZDOU**

Professeur, Université de Paris Saclay

Président

**Sébastien GLASER**

Professeur, Université de Queensland, Australie

Rapporteur & Examineur

**Christophe Sabourin**

Maître de conférences HDR, Université Paris Est Créteil

Rapporteur & Examineur

**Michel BASSET**

Professeur, Université de Haute-Alsace

Examineur

**Tarek AHMED-ALI**

Professeur, Université de Caen Normandie

Examineur

**Direction de la thèse**

**Moussa Boukhni**

Maître de conférences HDR, Université de Lorraine

Directeur de thèse

**Choubeila MAAOUI**

Professeure, Université de Lorraine

Co-Directrice de thèse

**Vincent JUDALET**

Enseignant chercheur, ESTACA, Campus Paris-Saclay

Co-Encadrant

**Matthieu DABEK**

Expert ADAS, Recherche et Innovation, groupe Stellantis

Invité



# Remerciements

Ces travaux de recherche ont été pour moi, une occasion d'acquérir un grand nombre de savoir et savoir-faire. Je ne saurais réaliser une telle œuvre sans le concours de certaines personnes auxquelles je tiens à exprimer ma profonde reconnaissance.

Je tiens à remercier très sincèrement mes encadrants pour leur confiance et leur soutien qui m'ont permis de mener à bien ces travaux de thèse. Je commencerai ainsi par remercier M. Moussa Boukhnifer, directeur de cette thèse, pour la confiance qu'il m'a donnée tout du long et même avant de commencer, pour m'avoir transmis sa passion pour la recherche et pour m'avoir apporté de nombreuses connaissances et ses bons conseils. Un immense merci à Mme Choubeila Maaoui, co-directrice, qui, avec à sa grande expertise, m'a guidée, conseillée et encouragée pendant ces années, et dont l'optimisme à toute épreuve font du bien au moral. Merci aussi à M. Vincent Judalet, co-encadrant, qui a parfaitement su m'accompagner et m'encourager, et dont les remarques et suggestions m'ont beaucoup aidée à avancer.

En plus de mes encadrants, je tiens à adresser mes sincères remerciements à l'ensemble des membres du jury. Ainsi, j'exprime ma reconnaissance et mes remerciements à M. Sébastien Glaser et M. Christophe Sabourin pour avoir accepté la charge de rapporter ces travaux de thèse, et pour toutes les remarques et questions pertinentes qu'ils ont su m'adresser à travers leurs rapports. De même, je témoigne toute ma reconnaissance à M. Fethi Ben Ouezdou pour avoir accepté le rôle de président du jury. Je remercie également M. Michel Basset, M. Tarek Ahmed-Ali et M. Mathieu Dabek d'avoir accepté le rôle d'examineur lors de la soutenance.

Je poursuivrai ces remerciements avec mes collègues de travail, pour leur aide, leurs conseils, mais aussi leur convivialité et les bons moments passés en leur compagnie au quotidien. Du côté de ESTACA, je remercie M. Cherif Larouci de m'avoir accueilli dans son équipe et pour toute la confiance qu'il m'a accordée. Merci aussi à tous les permanents, les doctorants et les stagiaires. Pour l'institut VEDECOM, merci à tous les membres de l'équipe VEH08 (actuellement l'équipe VAC) pour leur soutien, conseils, bonne humeur et convivialité.

Il me sera impossible de ne pas remercier tous les membres de ma famille, pour leur soutien moral et tous leurs efforts qu'ils ont entrepris durant ces longues années où j'étais loin d'eux. Qu'ils trouvent ici l'expression de toute mon affection et mes sincères excuses de mon absence et surtout pour ne pas pouvoir partager avec eux toutes leurs joies et leurs peines. En témoignage de l'attachement et de l'amour que je porte pour vous, je vous dédie ce travail avec tous mes vœux de bonheur, santé et de réussite.



# Table des matières

<b>Remerciements</b>	<b>iii</b>
<b>Table des matières</b>	<b>v</b>
<b>Liste des figures</b>	<b>vii</b>
<b>Liste des tableaux</b>	<b>xi</b>
<b>Introduction générale</b>	<b>1</b>
<b>1 État de l'art sur la prédiction des intentions</b>	<b>5</b>
1.1 Introduction à la conduite autonome . . . . .	6
1.2 Description de la problématique . . . . .	11
1.3 État de l'art sur les méthodes de prédiction des intentions des usagers de la route . . . . .	13
1.4 Taxonomie des modèles de prédiction des intentions . . . . .	19
1.5 Travaux de prédiction des intentions réalisés à VEDECOM . . . . .	26
1.6 Jeux de donnée véhiculaire . . . . .	34
1.7 Conclusion . . . . .	40
<b>2 Éléments sur la théorie de l'apprentissage</b>	<b>41</b>
2.1 Introduction . . . . .	42
2.2 Généralité sur l'apprentissage . . . . .	42
2.3 Apprentissage supervisé . . . . .	42
2.4 Exemples d'approches d'apprentissage supervisé . . . . .	44
2.5 Apprentissage non supervisé . . . . .	58
2.6 Approches d'apprentissage non supervisé . . . . .	59
2.7 Apprentissage par renforcement . . . . .	62
2.8 Conclusion . . . . .	64
<b>3 Reconnaissance de situation de conduite</b>	<b>65</b>
3.1 Introduction . . . . .	67
3.2 Principe de la reconnaissance de la situation de conduite . . . . .	67
3.3 Préparation des données . . . . .	71
3.4 Représentation de la manœuvre de changement de voie . . . . .	75

3.5	Détection de la manœuvre . . . . .	78
3.6	Reconnaissance de la phase de la manœuvre . . . . .	85
3.7	Reconnaissance du style de conduite . . . . .	95
3.8	Conclusion . . . . .	105
<b>4</b>	<b>Prédiction de trajectoire et prise de décision</b>	<b>107</b>
4.1	Introduction . . . . .	108
4.2	Approches de prédiction de trajectoire . . . . .	111
4.3	Approche hybride pour la prédiction de trajectoire . . . . .	121
4.4	Prise de décision de changement de voie . . . . .	129
4.5	Conclusion . . . . .	140
	<b>Conclusion générale et perspectives</b>	<b>141</b>
	<b>Résumé/Abstract</b>	<b>157</b>

# Liste des figures

1.1	Les niveaux d'autonomies de conduite (SAE International (Society of Automotive Engineers) ) . . . . .	7
1.2	Concept général des véhicules autonomes dans la ville . . . . .	8
1.3	Architecture d'un système de conduite autonome [Pendleton et al., 2017] . . . . .	9
1.4	Le système de perception embarqué dans un véhicule autonome [Schoettle, Sivak] . . . . .	10
1.5	Aperçu de la structure hiérarchique pour la prise de décision dans la conduite autonome . . . . .	11
1.6	Un exemple d'un scénario possible sur une autoroute. . . . .	12
1.7	Une prédiction de trajectoire basée sur le modèle de la vitesse constante. . . . .	21
1.8	Trajectoire probabiliste prédite pour le véhicule cible. . . . .	21
1.9	Trajectoires observées pour les véhicules venant de la gauche. . . . .	23
1.10	Manœuvres probables pour les véhicules de la scène. . . . .	24
1.11	Résultats de l'observation de la dynamique de la cible. . . . .	29
1.12	Estimation de l'écart latéral à 1 seconde . . . . .	30
1.13	Représentation des trajectoires modèles . . . . .	33
1.14	Comparaison entre la trajectoire réelle et les trajectoires modèles . . . . .	33
1.15	Vraisemblance des trajectoires modèles pendant des changements de voie. . . . .	34
1.16	Classification de la manœuvre en cours. . . . .	34
1.17	Une caméra montée sur le toit d'un bâtiment qui couvre une section de l'autoroute I-80 enregistre les données des trajectoires des véhicules. [FHWA, 2008] . . . . .	37
1.18	Le dispositif d'enregistrement comprend un drone qui vole à côté des autoroutes allemandes et capture le trafic à vue d'oiseau sur un tronçon de route d'environ 420 m de long. . . . .	38
1.19	Le véhicule Agro équipé par deux LiDAR, sept caméras en anneau et deux caméras stéréo frontales [Agroverse, 2021]. . . . .	38
1.20	La nouvelle voiture autonome de Waymo, équipée par un système de perception à trois LiDAR, des Radars et des caméras de conduite autonome [Waymo, 2021] . . . . .	39
1.21	La flotte des véhicules (Citroën Picasso C4) utilisé pour la collection des données MOOVE (VEDECOM). . . . .	40



2.1	Illustration de l'impact de la maximisation de la marge sur la capacité de généralisation. Le point gris est un élément de l'ensemble de test. . . . .	45
2.2	Un perceptron simple avec trois sorties . . . . .	47
2.3	Un perceptron de base multicouche. Nombreux neurones sont combinés pour former des couches, et de nombreuses couches sont combinées pour former le réseau final. . . . .	48
2.4	Un réseau de neurones récurrent simple. . . . .	49
2.5	Les différents types de réseaux de neurones récurrent . . . . .	50
2.6	Une cellule de mémoire à long/court terme LSTM . . . . .	50
2.7	Une cellule récurrente à porte (GRU) . . . . .	53
2.8	Principe de la validation croisée des séries de temps . . . . .	55
2.9	Exemples de modèle de courbe ROC . . . . .	57
2.10	Représentation du principe d'apprentissage par renforcement [Sutton, Barto, 2018] . . . . .	63
3.1	Les méthodes de reconnaissance de la situation de conduite . . . . .	68
3.2	Synoptique de la méthodologie proposée . . . . .	70
3.3	La zone d'étude de NGSIM [FHWA, 2008] . . . . .	71
3.4	Le système local des coordonnées et l'ordre des voies de NGSIM . . . . .	72
3.5	Filtrage de la position latérale et de la vitesse . . . . .	74
3.6	Illustration des processus de maintien de voie (AB et DE) et de changement de voie (BD) d'un véhicule . . . . .	75
3.7	Représentation en boîte à moustaches de l'angle lacet (a), de la vitesse latérale (b), de la vitesse angulaire (c) et de l'accélération latérale (d) pour les trois manœuvres . . . . .	76
3.8	Matrice de confusion des approches utilisées : (a) SVM (b) ANN . . . . .	82
3.9	Trajectoire latérale du véhicule (a) lors d'un changement de voie à droite (b) lors d'un changement de voie à gauche en indiquant l'instant de détection de chaque modèle. . . . .	84
3.10	Représentation du scénario de conduite. Cinq véhicules sont considérés autour du véhicule cible. . . . .	85
3.11	Illustration des processus de changement de voie d'un véhicule : étapes de changement de voie et chronologie. . . . .	86
3.12	Photographie aérienne de la section de route pour l'enregistrement n° 1 dans le jeu de données HighD [Krajewski et al., 2018]. . . . .	86
3.13	Attributs sélectionnés des véhicules environnants . . . . .	88
3.14	Les matrices de confusion de classification . . . . .	91
3.15	Évaluation de la classification des manœuvres à l'aide de la courbe ROC . . . . .	93
3.16	Graphique de synthèse SHAP (analyse de l'importance des attributs) . . . . .	93
3.17	Test du modèle en utilisant deux trajectoires ( $w_{size} = 50$ seconds) . . . . .	94
3.18	Les positions latérales des changements de voie (jeu de données NGSIM) . . . . .	99
3.19	Les profils de vitesse latérale pendant un changement de voie (jeu de données NGSIM) . . . . .	99
3.20	Les profils d'angle de lacet pendant un changement de voie (jeu de données NGSIM) . . . . .	100
3.21	Les résultats de clustering de la vitesse latérale en utilisant K-means . . . . .	101
3.22	Les résultats de clustering de la vitesse latérale en utilisant le clustering spectral . . . . .	102
3.23	Les résultats de clustering de l'angle de lacet en utilisant K-means . . . . .	103

3.24 Les résultats de clustering de l'angle de lacet en utilisant le clustering spectral . . . . .	104
4.1 Principe de prédiction de trajectoire d'un véhicule . . . . .	111
4.2 Une manœuvre de changement de voie : véhicule cible et les véhicules environnants . . . . .	112
4.3 Représentation de la scène de conduite avec six véhicules autour du véhicule cible. . . . .	114
4.4 Structure et architecture de réseau des modèles proposés. . . . .	115
4.5 Courbes d'entraînement du jeux de donnée d'apprentissage des différents modèles. . . . .	119
4.6 La prédiction de la trajectoire de trois véhicules de test : (a), (b) et (c) représentent les positions latérales et (d), (e) et (f) représentent respectivement les positons longitudinales (noir : position réelle, rouge : prédiction LSTM, bleu : prédiction GRU). . . . .	120
4.7 Présentation du système : Intégration de la classification des manœuvres et de l'approche basée sur la prédiction de la trajectoire . . . . .	121
4.8 Architecture du réseau LSTM de prédiction de trajectoire. . . . .	122
4.9 Prédiction de la trajectoire d'un véhicule avec deux changements de voie à droite : (a) et (b) représentent respectivement la trajectoire longitudinale et latérale prévue et réelle. (c) représente les résultats de la classification. . . .	125
4.10 Prédiction de la trajectoire d'un véhicule avec trois changements de voie à gauche : (a) et (b) représentent respectivement la trajectoire longitudinale et latérale prévue et réelle. (c) représente les résultats de la classification. . .	126
4.11 Démonstrateur VEDECOM : capteurs installés et plateforme hardware interne. . . . .	126
4.12 Pistes d'essai Satory (a) et trajectoire du véhicule d'essai(b). . . . .	127
4.13 Prédiction de la trajectoire du démonstrateur VEDECOM :(a) et (b) représentent respectivement la trajectoire longitudinale et latérale prédite et réelle. (c) représente les résultats de la classification. . . . .	128
4.14 Architecture du système de prise de décision proposé. . . . .	130
4.15 Interface de la plateforme de simulation SUMO . . . . .	131
4.16 Exemple d'une scène de conduite du jeux de donnée HighD . . . . .	132
4.17 Six véhicules sont considérés autour de l'ego véhicule . . . . .	132
4.18 Moyennes des récompenses de l'agent pendant l'entraînement) . . . . .	136
4.19 Résultats d'entraînement - La fonction de coût(écart quadratique moyen) .	137
4.20 Nombre de changements de voie effectués par l'agent pendant l'entraînement	138
4.21 Moyenne des non collisions de l'agent pendant l'entraînement . . . . .	139



# Liste des tableaux

2.1	Série temporelle univariante . . . . .	54
2.2	Série temporelle multivariante . . . . .	55
3.1	Description des données contenues dans le jeux de données NGSIM . . . . .	72
3.2	Ensembles des attributs pour la détection des manœuvres . . . . .	79
3.3	Résultats d'évaluation des SVM avec différents noyaux . . . . .	81
3.4	Précision de l'apprentissage pour les ensembles de données E.1 à E.4 . . . . .	81
3.5	Résultats de l'évaluation des SVM et ANN en utilisant les données sélectionnés . . . . .	82
3.6	Temps de prédiction des approches SVM et ANN . . . . .	83
3.7	Les attributs dans les données du jeux de données HighD [Krajewski et al., 2018] . . . . .	87
3.8	Comparaison de HighD et de NGSIM [Krajewski et al., 2018] . . . . .	88
3.9	Attributs sélectionnés pour l'apprentissage . . . . .	89
3.10	Le paramétrage du réseau de neurone proposé . . . . .	90
3.11	Scores moyens de la silhouette pour 3 clusters . . . . .	97
4.1	L'erreur de la position latérale (m) . . . . .	116
4.2	L'erreur de la position longitudinale (m) . . . . .	116
4.3	Évaluation des modèles : Le temps de prédiction et l'erreur de la position latérale et longitudinale de trois véhicules de test . . . . .	116
4.4	Évaluation des modèles : précision de la classification et RMSE de la position latérale et longitudinale. . . . .	124
4.5	Paramètres de l'environnement de simulation . . . . .	132
4.6	Le nombre de véhicule pour chaque type de trafic . . . . .	134



# Introduction générale

## Contexte général

Les travaux de cette thèse ont été réalisés dans le cadre d'un partenariat entre l'institut VEDECOM, l'ESTACA et le laboratoire LCOMS de l'université de Lorraine. La prédiction des intentions, dans le contexte de conduite autonome, est à la fois un sujet de recherche important et une source d'applications possibles à haute valeur ajoutée pour l'industrie. Dans cette première section, nous décrirons les implications scientifiques et les motivations de cette thèse.

Les systèmes de transport ont apporté à l'humanité des avantages sociaux et financiers inestimables, mais ils sont également liés à des effets négatifs tels que les accidents de la route, les émissions de gaz à effet de serre et les embouteillages. Aujourd'hui, comme jamais auparavant, les entreprises technologiques et les laboratoires de recherche du monde entier déploient d'énormes efforts pour réduire ces effets en automatisant les systèmes de transport. Cette technologie a la capacité de transformer radicalement le secteur des transports et de rendre les routes beaucoup plus sûres, en éliminant l'erreur humaine et en transcendant ses limites physiques (temps de réaction, mauvaise visibilité dans des situations météorologiques défavorables, etc.).

La communauté scientifique et l'industrie automobile travaillent sur la prochaine génération de véhicules intégrant des fonctionnalités intelligentes. Avant de présenter le domaine de recherche, nous commençons par la définition du système intelligent de véhicule. Nous le définissons comme un système qui observe l'environnement de conduite, détecte les participants et les informations pertinentes de la scène de conduite, interprète ce qui se passe (ce qui nécessite un certain degré de compréhension) et éventuellement prédit ce qui va se passer, et enfin prend des décisions ou donne un retour au conducteur. Les applications de ces systèmes intelligents sont les suivantes : l'amélioration de la sécurité routière [Maurer et al., 2016; Ren et al., 2019], la réduction de la consommation d'énergie [Liu et al., 2020; Ross, Guhathakurta, 2017] ou l'amélioration du confort du conducteur [Elbanhawi et al., 2015].

## Contributions et objectifs de la thèse

La thématique générale de cette thèse est la prédiction des intentions des véhicules à partir d'un véhicule autonome dans des scénarios d'autoroute avec des voies à chaussées

séparées. Dans le cadre de notre travail, on s'est limité à la prédiction des intentions de changement de voie.

La prédiction des intentions des différents usagers de la route est une étape primordiale vers les véhicules intelligents. C'est également un élément clé pour améliorer la perception et l'interprétation de l'environnement. En prédisant le comportement futur d'un participant au trafic, nous pouvons anticiper sa position dans la future scène de conduite.

Une autre motivation pour prédire le comportement futur des autres véhicules est la réduction de la puissance de calcul nécessaire aux systèmes de détection. La puissance de calcul est une ressource précieuse dans les systèmes embarqués. La connaissance du comportement futur d'un participant au trafic permet au système de concentrer le traitement sur les parties pertinentes de la scène.

Un véhicule intelligent ne pourra pas compter uniquement sur les données de communication pour collecter des informations, car tous les véhicules ne seront pas équipés de telles fonctionnalités. Le système doit donc être en mesure de déduire l'intention des véhicules environnants à l'aide de capteurs embarqués. Dans ce contexte, nous ne prenons pas en compte les données de communication entre les véhicules (V2V/V2X) et nous supposons que les capteurs extéroceptifs (radar, lidar et caméra) sont la seule source d'informations.

Le changement de voie est une succession d'actions critiques qui nécessite une attention constante et une évaluation correcte, ce qui en fait l'une des causes les plus fréquentes d'accidents sur l'autoroute [Fitch et al., 2009]. Les feux clignotants ne peuvent pas être utilisés comme indicateurs de changement de voie car 48,35% des conducteurs ne respectent pas l'utilisation appropriée des clignotants lors de l'exécution d'une manœuvre de changement de voie [Ponziani, 2012]. Si le système d'assistance à la sécurité peut anticiper le changement de voie avant que d'autres véhicules ne franchissent le marquage, le véhicule autonome peut assurer un très haut niveau de sécurité et les taux d'accidents peuvent être considérablement réduits.

De nombreux travaux et projets de recherche sont menés pour une meilleure compréhension des comportements de conduite des véhicules à partir d'un véhicule autonome. Ces projets sont basés sur l'étude des comportements de conduite en situations réelles. L'objectif de nos travaux de thèse est de mettre au point des techniques basées sur l'intelligence artificielle pour prédire les intentions des véhicules en assurant la meilleure précision de prédiction de trajectoire et en anticipant d'avance la réalisation de la manœuvre.

Après cette introduction, le chapitre I donne un aperçu des véhicules autonomes. En étudiant de nombreux aspects de cette nouvelle technologie, il fournit la problématique de la thèse. Une partie importante du chapitre I est dédiée à l'état de l'art sur la prédiction des intentions dans le contexte de la conduite autonome. Cette partie sera consacrée à la prospection des techniques de classification des manoeuvres et de prédiction de trajectoire. Le chapitre se poursuit par un aperçu des différents jeux de données véhiculaires.

Le chapitre II est consacré à la description des notions de base de l'apprentissage automatique. Dans un premier temps, nous décrivons l'apprentissage supervisé, tout en détaillant les deux approches discriminative et générative, ainsi que les deux types de problèmes traités par ce type d'apprentissage à savoir : la classification et la régression. Par la suite, nous allons présenter le principe et les algorithmes d'apprentissage non supervisé. Enfin, un autre type d'apprentissage sera décrit : l'apprentissage par renforcement.

Le chapitre III concerne la problématique de la reconnaissance automatique de situations de conduite à l'aide d'approches de classification et de clustering. La première partie du chapitre est consacrée à la description du principe de la reconnaissance de la

situation de conduite. Après cela, nous allons introduire les étapes de préparation de données et la compréhension de la manoeuvre de changement de voie. La suite du chapitre s'attache à la mise en œuvre des différentes approches pour la reconnaissance de la situation de conduite à savoir la détection de la manoeuvre, la reconnaissance de la phase de conduite et la reconnaissance du style de conduite.

La prédiction de trajectoire d'un véhicule cible et la prise de décision de l'ego véhicule font l'objet du chapitre IV. Le premier volet est consacré au problème de prédiction de trajectoire. Dans cette partie, nous détaillons nos approches basées sur les réseaux de neurones récurrents pour prédire la trajectoire d'un véhicule à partir de son historique de données. Ensuite, nous proposons une approche hybride qui combine la classification des manoeuvres et la prédiction de trajectoire pour concevoir un modèle complet de prédiction des intentions. Le deuxième volet du chapitre est consacré à la prise de décision de changement de voie dans le cadre de la conduite autonome. Dans cette partie, nous présentons notre approche de prise de décision basée sur l'apprentissage par renforcement.

Une conclusion de ce mémoire rappellera les différentes étapes du travail effectué ainsi que les résultats obtenus et les perspectives de recherche découlant de ces travaux de thèse.

Les travaux de cette thèse ont donné lieu aux publications suivantes :

### — Journaux Internationaux

1. A. Benterki, M. Boukhnifer, V. Judalet, C. Maaoui, "Artificial Intelligence for Vehicle Behavior Anticipation : Hybrid Approach based on Maneuver Classification and Trajectory Prediction Hybrid Model". IEEE Access Journal, volume 8, pages 56992-57002, 2020 (IF=4.098).

### — Conférences Internationales avec comité de lecture

1. A. Benterki, V. Judalet, M. Boukhnifer, C. Maaoui, "Driving Intention Prediction and State Recognition on Highway". The 29th Mediterranean Conference on Control and Automation (MED'2021), June 22-25, Bari, Puglia, Italy.
2. A. Benterki, V. Judalet, M. Boukhnifer, C. Maaoui, "Multi-Model and Learning-Based Framework for Real-Time Trajectory Prediction". The 28th Mediterranean Conference on Control and Automation (MED'2020), June 16-19, Saint Raphael, France.
3. A. Benterki, V. Judalet, C. Maaoui, M. Boukhnifer, "Long-Term Prediction Of Vehicle Trajectory Using Recurrent Neural Networks". The 45th Annual Conference of the IEEE Industrial Electronics Society (IES) (IECON'2019), October 14-17, Lisbon, Portugal.
4. A. Benterki, M. Boukhnifer, V. Judalet, C. Maaoui, "Prediction of Surrounding Vehicles Lane Change Intention Using Machine Learning". The 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems : Technology and Applications (IDAACS'2019), September 18-21, Metz, France.





# 1

## État de l'art sur la prédiction des intentions

*« Les voitures autonomes sont le prolongement naturel de la sécurité active et, évidemment, quelque chose que nous devrions faire. »*

---

Elon Musk  
*Président-directeur général de SpaceX et Directeur général de Tesla*

## 1.1 Introduction à la conduite autonome

Au cours du développement technologique actuel, les constructeurs automobiles sont confrontés au défi de maximiser le confort et de minimiser les risques afin que les innovations puissent apporter une réelle contribution à l'amélioration de la sécurité routière. Le développement des systèmes d'aide à la conduite et des fonctions autonomes a augmenté rapidement ces dernières années. Les systèmes et les fonctions actuels visent à assister le conducteur dans certaines situations de conduite et à réduire ainsi le risque d'accidentologie. Des exemples de tels systèmes sont l'assistance au maintien de la voie, l'alerte de collision, le freinage d'urgence et le régulateur de vitesse adaptatif [Elzen et al., 2015], [Kala, 2016], [Panaturak et al., 2009] .

Plusieurs projets de recherche ont été lancés afin de poursuivre le développement de ces technologies et de se rapprocher de l'automatisation totale des véhicules. De nouveaux types de véhicules équipés de technologies innovantes sont en cours de développement et pourraient révolutionner nos déplacements. En offrant une assistance à la conduite voire en déléguant la conduite à la machine, ces véhicules permettraient, à terme, lorsqu'ils circuleront en autonomie, d'améliorer la sécurité routière, mais la coexistence de l'homme et de la machine pose de nombreuses questions, technologiques, juridiques et de sécurité .

Les technologies utilisées dans les systèmes de transport intelligents, notamment dans les véhicules autonomes (VA), sont aujourd'hui au cœur des activités de recherche et d'innovation de nombreuses équipes de recherche au niveau nationale et internationale. Les véhicules autonomes sont largement attendus pour réduire les encombrements routiers, améliorer la sécurité routière en éliminant l'erreur humaine et libérer les conducteurs de la charge de la conduite, en leur permettant d'être plus productifs et/ou de se reposer.

D'après les statistiques [Singh, 2018], plus de 90% des accidents sont provoqués par des erreurs humaines alors que seulement 6% sont causés soit par des défaillances mécaniques des véhicules, des défauts au niveau de l'infrastructure ou par des causes inconnues. Cela a donné l'idée de concevoir des véhicules autonomes afin d'éliminer les erreurs humaines. Les véhicules autonomes devraient jouer un rôle clé dans l'avenir des systèmes de transport, car ils offrent un potentiel de sécurité, une meilleure accessibilité, une meilleure efficacité routière et un impact positif sur l'environnement. La recherche sur les systèmes autonomes a connu des progrès spectaculaires ces dernières années, en raison de l'augmentation de la puissance de calcul et de la réduction des coûts des technologies des capteurs et des ordinateurs de bord. L'objectif de cette première partie est de fournir un aperçu général des systèmes de conduite autonomes.

Afin de rendre la conduite plus sûre et plus pratique, des chercheurs de différents domaines, de la robotique, de l'ingénierie automobile et de domaines connexes, essayent depuis des décennies, de développer des systèmes intelligents pour les véhicules modernes. Ces systèmes sont conçus pour aider à la conduite automatique ou pour surveiller un conducteur humain. Ils peuvent avertir le conducteur en cas de situation dangereuse et lui offrir la possibilité d'éviter les collisions ou d'en atténuer les conséquences si une collision est inévitable. En outre, ces systèmes de véhicules intelligents devraient pouvoir fonctionner dans toutes les situations de circulation, quel que soit l'endroit où ils se trouvent.

Les systèmes d'aide à la conduite (Advanced Driving Assistance Systems (ADAS) [Kala, 2016]) sont des systèmes de sécurité active destinés à informer ou à assister le conducteur afin de réduire les accidents et de faciliter les tâches de conduite. Nous faisons la distinction entre le véhicule automatisé, qui présente des fonctions automatisées avec peu ou

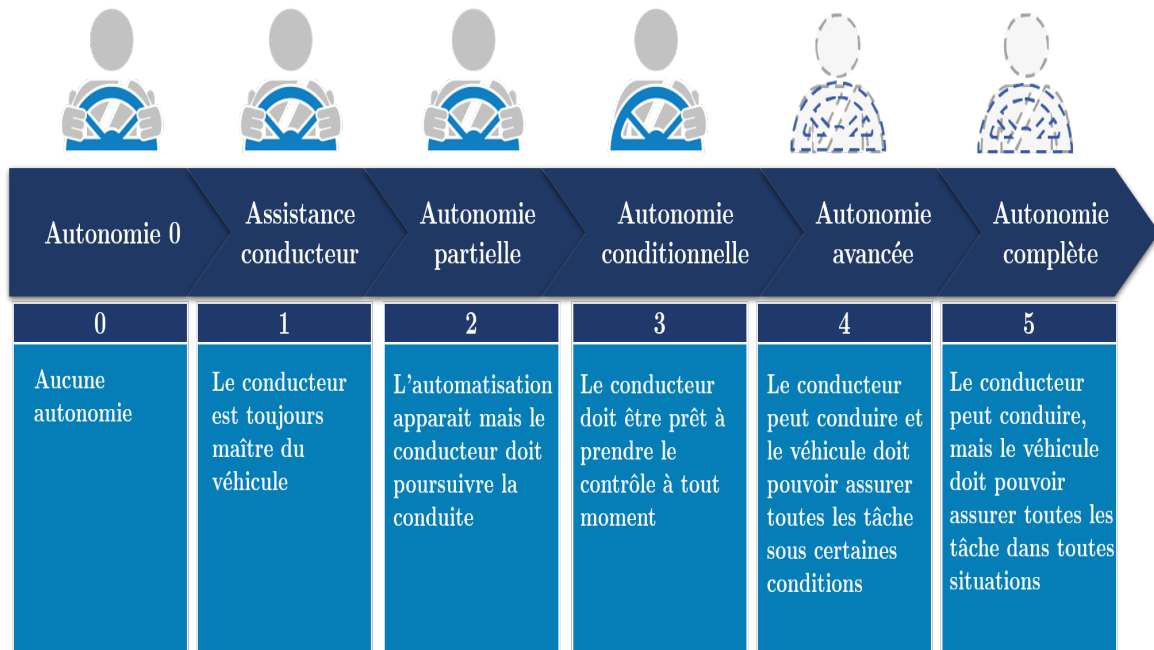


FIGURE 1.1 – Les niveaux d'autonomies de conduite (SAE International (Society of Automotive Engineers) )

pas de contrôle humain direct, et les véhicules autonomes, qui proposent une conduite entièrement automatisée. Si le concept de conduite autonome peut sembler vague, il existe pourtant une nomenclature officielle qui classe les véhicules selon leur capacité à assister les conducteurs. La SAE International (Society of Automotive Engineers) a ainsi identifié 6 niveaux de conduite autonome. La figure 1.1 présente une petite infographie qui résume ces niveaux.

Le rapport du véhicule avec ce qui l'entoure et la manière avec laquelle il peut avoir à réagir aux événements qui se déroulent autour de lui sont des enjeux primordiaux pour l'instauration d'une confiance de la société envers cette innovation (figure 1.2). Le fonctionnement d'une voiture autonome peut être décrit comme suit : le véhicule perçoit son environnement, planifie son trajet puis infère le trajet en fonction des informations des capteurs. Il y a donc quatre sous-composantes : la perception, la planification, le contrôle et la communication (figure 1.3). Un système de conduite doit exécuter ces quatre fonctions afin d'assurer son autonomie totale [Herrmann et al., 2018] :

- **Percevoir son environnement :**

La perception de l'environnement est une fonction fondamentale pour permettre aux véhicules autonomes de détecter et d'identifier tous les objets fixes ou mobiles de son environnement, à savoir l'infrastructure de la route, les positions des obstacles environnants, les vitesses et même des prédictions sur l'évolution de leur état dans le futur [Van Brummelen et al., 2018]. La tâche de perception de l'environnement est abordée en utilisant des capteurs à savoir des Lidars, des Radars, des caméras (figure 1.4). La fusion combine les données fournies par les différents capteurs pour obtenir une information globale pertinente. Cette technique est nécessaire pour exploiter pleinement les avantages de chaque capteur. La fusion de données reste aujourd'hui un grand défi, tant scientifique que technique.



FIGURE 1.2 – Concept général des véhicules autonomes dans la ville

L'analyse et l'interprétation des données est une étape très importante pour que le véhicule autonome puisse comprendre son environnement et identifier sa localisation. La détection et l'évitement d'obstacles est l'un des problèmes les plus difficiles pendant la conduite d'un véhicule autonome. Plusieurs techniques sont utilisées pour analyser l'environnement autour du véhicule pour la segmentation des données fusionnées et la détection d'obstacles [Vu et al., 2014; Aeberhard, Kaempchen, 2011]. Aujourd'hui, les techniques d'apprentissage supervisé permettent la reconnaissance des objets et des obstacles sur la route (véhicules, piétons, infrastructure, etc.). En se basant sur les techniques de localisation, le véhicule doit établir une carte de son environnement, et localiser sa propre position sur une carte locale et globale.

- **Planifier sa trajectoire :**

Dans le contexte du véhicule autonome, la planification est un processus de décision sur une séquence d'actions pour atteindre un objectif spécifié, tout en évitant les collisions avec des obstacles. L'autonomie décisionnelle pose un grand défi technologique et scientifique. Le véhicule autonome doit prendre la meilleure décision dans un temps très réduit. Le planificateur est responsable de la prise de décision pour s'assurer que le véhicule respecte les règles routières stipulées et interagisse avec les autres véhicules de manière conventionnelle et sûre [Frazzoli et al., 2002]. Cette décision comporte la trajectoire à suivre et les manœuvres à réaliser afin d'atteindre les objectifs d'ordre supérieur, pour amener le véhicule d'un emplacement de départ à un emplacement d'arrivée tout en évitant les obstacles et en optimisant le confort et la sécurité d'utilisateur. Le véhicule autonome doit décider des ma-

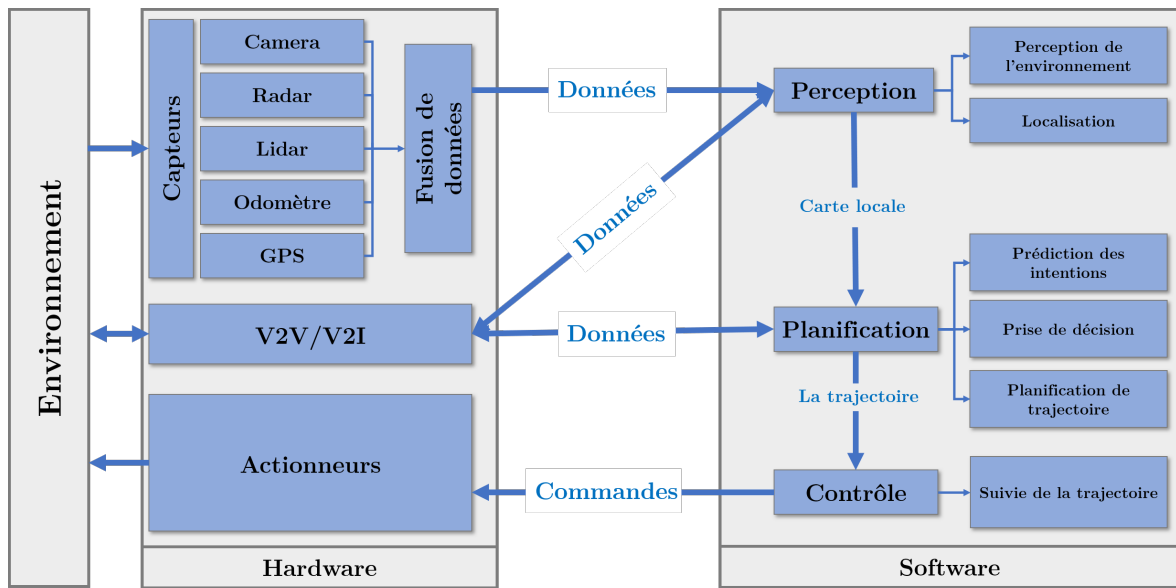


FIGURE 1.3 – Architecture d'un système de conduite autonome [Pendleton et al., 2017]

noeuvres à entreprendre en fonction de l'itinéraire planifié.

La génération de la trajectoire consiste à calculer une trajectoire de référence, qui évite les obstacles et respecte plusieurs autres critères ; comme le confort, la sécurité des passagers et le code de la route [Schwartz et al., 2018]. La planification de trajectoire pour le véhicule autonome nécessite de connaître à tout moment la position des obstacles dans l'environnement proche. Quand les obstacles sont mobiles (autres véhicules), il ne suffit pas de connaître leur position actuelle, mais il est nécessaire de prévoir leurs trajectoires futures, afin de prévenir tout risque de collision avec la trajectoire planifiée de l'égo-véhicule. L'intention d'un usager de la route peut être considérée comme la séquence d'actions qu'il entreprendra pour atteindre un objectif [Lefèvre et al., 2014]. Dans le cas d'un véhicule, l'intention est définie par l'ensemble des manoeuvres et des trajectoires réalisées par un conducteur pour atteindre un point d'arrivée dans un intervalle du temps défini.

La prise de décision se fait généralement en trois niveaux (figure 1.5) : niveau stratégique, niveau tactique et niveau opérationnel. Au niveau stratégique, l'itinéraire global est déterminé à partir des informations d'une carte numérique et du système de localisation. Au niveau tactique, la planification locale est faite à partir de l'itinéraire global et des informations de l'environnement proche du véhicule. En effet, à partir des données de la perception de l'environnement proche et de la localisation du véhicule, une grille d'occupation dynamique (ou une carte locale de la route) est construite avec les informations relatives aux obstacles fixes et mobiles. L'objectif de ce niveau est de définir la manoeuvre qui correspond aux données de la carte locale. Ensuite, la décision finale est faite au niveau opérationnel. Dans ce niveau, plusieurs trajectoires locales possibles sont calculées dans l'espace navigable à l'aide du modèle dynamique du véhicule. La trajectoire de référence est choisie en se basant sur plusieurs critères, de manière à assurer une navigation tout en évitant les obstacles fixes et mobiles.

- **Contrôler ses actionneurs :**

Afin de permettre aux véhicules de se déplacer sur la route, il est nécessaire d'avoir

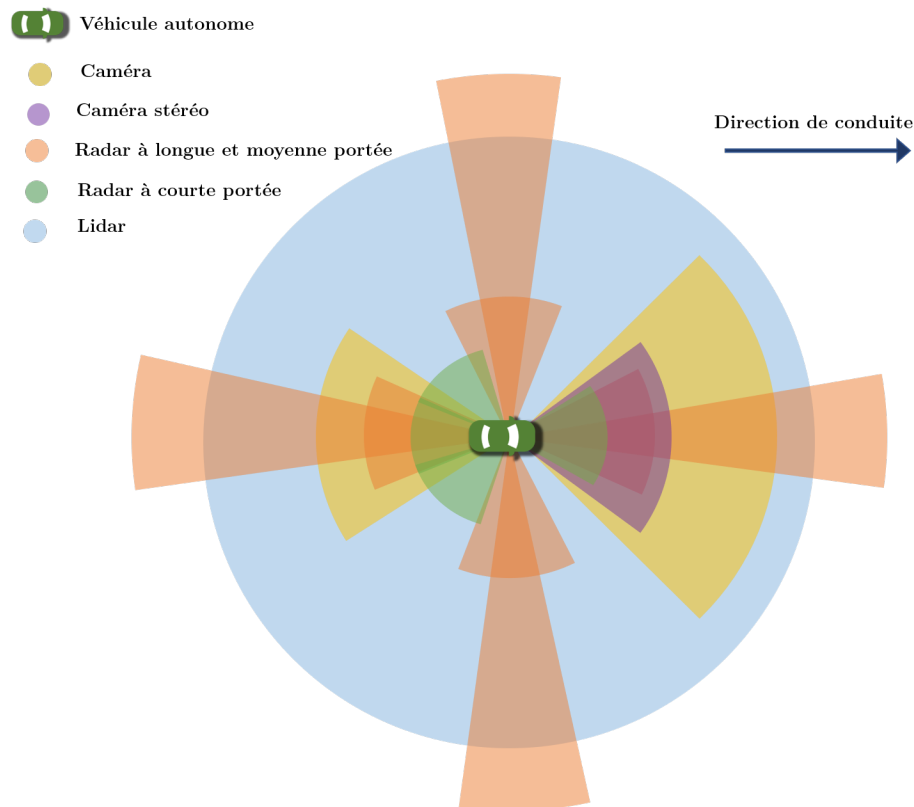


FIGURE 1.4 – Le système de perception embarqué dans un véhicule autonome [Schoettle, Sivak]

une boucle de contrôle pour s'assurer qu'il suivent la trajectoire physique définie par le planificateur. Le module de contrôle est étroitement lié à la perception et au planificateur de trajectoire. De mauvaises trajectoires seront difficiles à exécuter, une mauvaise observation de l'environnement conduira à une divergence entre l'état estimé et la réalité et peut également forcer le planificateur à réaliser des changements brusques dans les trajectoires. Le contrôle du véhicule consiste à manœuvrer le véhicule en utilisant les actionneurs tels que le volant, le frein et l'accélérateur pour suivre la trajectoire de référence. Cette étape peut être subdivisée en deux tâches : le contrôle longitudinal et le contrôle latéral [Khodayari et al., 2010].

Le module de contrôle est responsable du processus de conversion des intentions en actions [Veres et al., 2011]; son objectif principal est d'exécuter les intentions planifiées en fournissant les entrées nécessaires au niveau matériel qui générera les mouvements souhaités. Les contrôleurs représentent l'interaction dans le monde réel en termes de forces et d'énergie. Les mesures à l'intérieur du système de contrôle peuvent être utilisées pour déterminer le bon comportement du système, et par conséquent le contrôleur peut réagir pour rejeter les perturbations et modifier la dynamique du système à l'état souhaité. Des modèles de système peuvent être utilisés pour décrire plus en détail le mouvement souhaité, ce qui est essentiel pour une exécution satisfaisante du mouvement.

- **Communiquer avec son entourage :**

La communications véhicule à véhicule (V2V) peut être exploitée pour réaliser de nouvelles améliorations dans les tâches de perception et / ou de planification grâce à la coopération avec les autres véhicules [Wei et al., 2013]. Dans le cas particulier où l'obstacle est un véhicule autonome équipé d'un système de communication

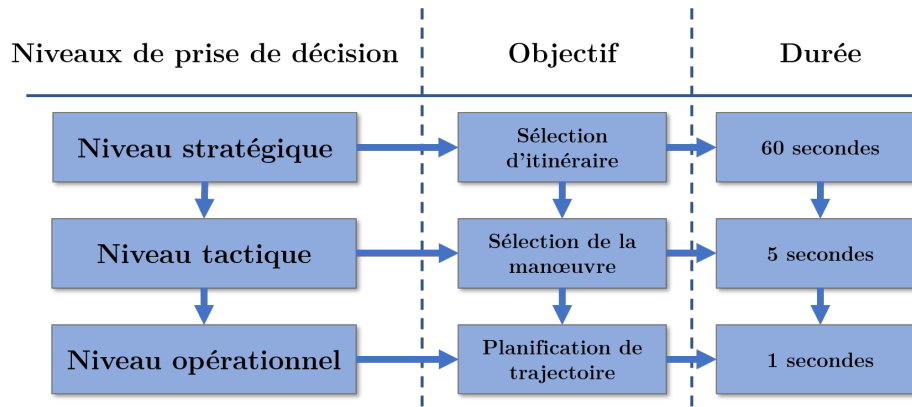


FIGURE 1.5 – Aperçu de la structure hiérarchique pour la prise de décision dans la conduite autonome

véhicule/véhicule (V2V), les véhicules peuvent s'échanger mutuellement leur trajectoire planifiée. Dans le cas d'un véhicule conduit par l'humain, et en prenant en compte les latences de communication, il est impossible de prévoir parfaitement les manœuvres en cours. Il est donc nécessaire de prédire les actions probables des obstacles. Pour cela, il faut mettre en place des algorithmes de prédictions des intentions des autres usagers.

## 1.2 Description de la problématique

Malgré que les humains soient difficiles à modéliser, une meilleure analyse du comportement humain devra permettre aux scientifiques de concevoir des machines capables de prendre des décisions plus «humaines». En outre, aucun humain n'est semblable à un autre, ce qui signifie que les actions possibles sont presque infinies dans tous les scénarios. Au volant, il est presque toujours nécessaire de savoir ce que les voitures environnantes ont l'intention de faire afin d'assurer un contrôle sûr et optimisé du véhicule. Prenons l'exemple illustré à la figure 1.6 : si la voiture arrière sur la voie de gauche (Ego) a l'intention de prendre la sortie sur sa droite, elle doit prédire comment les autres voitures se comporteront afin de se diriger en toute sécurité vers la sortie.

En raison de l'irrationalité de l'être humain et des récents succès remportés dans le domaine de l'intelligence artificielle, cette thèse adoptera une approche de prédiction basée sur les données. L'objectif de la thèse est de voir dans quelle mesure l'intention d'un conducteur humain et le mouvement du véhicule en général peuvent être prédits grâce à des méthodes basées sur l'apprentissage.

En tant qu'être humains, nous possédons une capacité intuitive de « voir » un peu plus loin dans le futur et de prédire le mouvement des objets environnants dans des scénarios complexes du monde réel à travers des années de pratique. Dans le cas de la conduite de voiture, le conducteur humain est capable d'anticiper la manœuvre et la trajectoire des autres véhicules en utilisant des observations de la dynamique comme la vitesse et l'orientation. En utilisant la même approche, une voiture autonome doit être capable de comprendre et de prédire le comportement des autres véhicules. En s'inspirant du fonctionnement du cerveau humain, une nouvelle famille d'approches basées sur l'idée que, dans un environnement donné, les objets ont tendance à suivre des mouvements «typiques» qui dépendent de la nature de l'environnement, ainsi que de la nature des objets



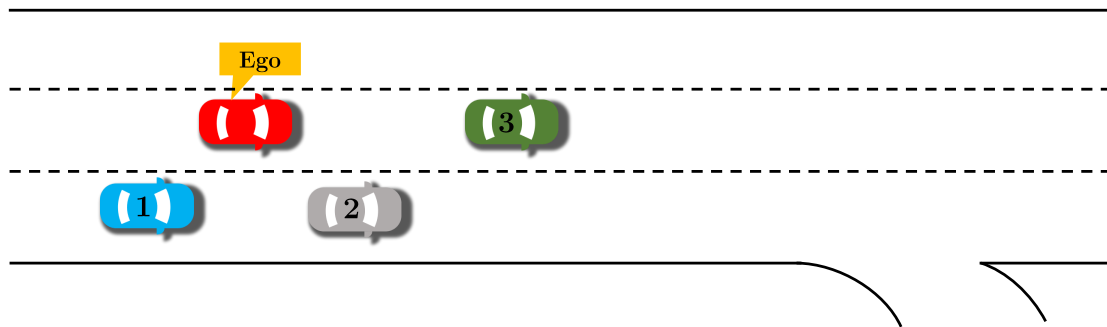


FIGURE 1.6 – Un exemple d'un scénario possible sur une autoroute.

eux-mêmes. Ces approches opèrent en deux étapes distinctes :

- **Apprentissage** : observer les objets dans l'environnement pour identifier et construire des modèles des mouvements typiques.
- **Prédiction** : utiliser les modèles appris pour prédire l'état futur d'un objet donné. La prédiction se devise en trois classes selon le but d'apprentissage : classification, régression et clustering.

Un véhicule autonome déployé dans un trafic complexe doit être capable de prédire le mouvement futur des véhicules environnants. Le temps de réaction des capteurs et le temps de fusion des données constituent le délai de traitement dans le système autonome. Dans les cas critiques, le retard de traitement peut ne pas laisser suffisamment de temps pour éviter la collision. Cependant, la planification des mouvements doit savoir à l'avance comment les participants au trafic, dans le même environnement, se déplaceront.

VEDECOM est un Institut pour la Transition Énergétique (ITE) qui a pour but de développer des technologies de mobilité écologique, autonome et partagée. VEDECOM ambitionne de devenir un leader européen en matière d'innovation dans les domaines des véhicules électrifiés, autonomes et connectés grâce à des infrastructures et services de mobilité et d'énergie partagée.

Les « véhicules autonomes et connectés » sont le champ de recherche consacré du domaine 2 de VEDECOM. Les objectifs principaux de ce domaine se centrent autour du développement de fonctionnalités avancées nécessaires au bon fonctionnement d'un véhicule autonome. Néanmoins, cette thématique est actuellement largement abordée par les différents acteurs académiques et industriels du milieu automobile [Aeberhard et al., 2015; Vlacic et al., 2001; Van Brummelen et al., 2018]. De fait, le positionnement de ce domaine est donc orienté de manière à répondre aux limitations actuelles des véhicules autonomes et se concentrent sur quelques thèmes majeurs de l'automatisation de la conduite, à savoir :

- La possibilité pour un véhicule de se localiser dans des environnements différents et des conditions variées sans discontinuité.
- La détection des obstacles allant d'approches fusionnant des informations hétérogènes bas niveau jusqu'à la prédiction des intentions des usagers de la route et vulnérables.
- La planification et la prise de décision dans des environnements complexes (franchissement d'intersections en zones urbaines, par exemple).

La détection d'obstacles et la prédiction des intentions des usagers de la route sont des objectifs stratégiques du domaine « Véhicules autonomes et connectés ». Ces problématiques sont abordées différemment suivant les cibles : piétons ou usagers de la route. Cette thèse traite la partie « usagers de la route », le sujet vise principalement le cas des voies à chaussées séparées via la classification des manœuvres et la prédiction de trajectoire des autres véhicules. L'objectif est de pouvoir anticiper au maximum les changements de voies et améliorer les prédictions obtenues par les méthodes traditionnelles de suivi utilisées dans ce contexte. Afin d'étendre cet horizon d'anticipation, les pistes suivies se basent sur la mise en relation du suivi des obstacles de la scène avec l'environnement (nombre de voies, etc.) et les interactions entre les obstacles.

### 1.3 État de l'art sur les méthodes de prédiction des intentions des usagers de la route

L'Organisation Mondiale de la Santé (OMS) a signalé 1,35 million de décès sur les routes en 2016 dans le monde [[Organisation mondiale de la santé](#)]. Il est estimé que près de 3 493 personnes seraient décédées en 2019 sur les routes de France, en métropole et en outre-mer [[ONISR](#)]. À ce jour, les accidents de la route sont la huitième cause de décès dans le monde. Le facteur véhicule est fondamental pour la sécurité routière. En effet, les occupants de véhicules représentent 45% des décès, alors qu'environ 90% de tous les accidents sont causés par des erreurs humaines telles que les fausses manœuvres, la surcharge, ou la mauvaise estimation de la situation [[Commission européenne](#)].

Les accidents impliquant des manœuvres de changements de voie et de dépassements sont d'importants problèmes sur les autoroutes et les routes rurales où un véhicule veut dépasser un autre véhicule plus lent dans la même direction. Le danger survient lorsque le véhicule qui se déplace dans la voie opposée n'a pas assez de distance de visibilité dans une situation transitoire. Le changement de voie est un phénomène nécessaire mais un changement de voie inapproprié est une source d'accidents et d'instabilité de la circulation en raison d'un manque d'avertissement et de problèmes de gestion du trafic.

Concernant la compréhension et la prédiction des intentions des autres véhicules dans des scénarios d'autoroute, plusieurs aspects ont été abordés dans la littérature [[Lefèvre et al., 2014](#)]. En conséquence, cette section est subdivisée en quatre parties correspondantes : la section 1.3.1 introduit l'importance de la prédiction des intentions dans les applications de conduite autonome. La section 1.3.1.c présente une vue d'ensemble des approches permettant d'identifier quel type de manœuvre sera exécuté par un véhicule. Cependant, il faut bien noter que des applications telles que les algorithmes d'estimation de risque de collision ou de planification de trajectoire ne peuvent pas traiter directement ce type d'information. Il faut plutôt prévoir des prédictions de positions ou de trajectoires futures des véhicules. Les travaux de recherche sur cette thématique sont présentés dans la section 1.3.2. La section 1.3.3 donne un aperçu des approches hybrides de prédiction qui rassemblent des aspects de classification des manœuvres et de prédiction des trajectoires. La section 1.6 résume les jeux de données utilisés pour la prédiction des intentions

### 1.3.1 Prédiction du comportement pour les applications de conduite autonome

Comme nous le montrerons dans cette section, la prédiction des comportements des différents usagers de la route est une étape primordiale pour les véhicules autonomes. C'est également un élément clé pour améliorer la perception et l'interprétation de l'environnement.

#### 1.3.1.a Prédiction des comportements actuels des usagers de la route

Savoir pourquoi le conducteur ou un autre usager de la route fait ce qu'il est en train de faire est une information précieuse. De même, la différence entre la manière dont le système attend d'un véhicule comment il se comporte et la manière dont il agit réellement contient beaucoup d'informations (comparaison de la prédiction et la situation réelle).

Le système de vision humaine est impressionnant, car il nous permet de détecter facilement les usagers de la route, dans un environnement dynamique et en constante évolution. Malgré des décennies de recherche dans le domaine du traitement d'image de la vision intelligente et l'utilisation de capteurs de plus en plus précis, aucun système de vision artificielle ne peut atteindre des performances comparables. Tout les conducteurs ont des capacités similaires en matière de reconnaissance d'objets, et ils ne commettent généralement pas d'erreur en reconnaissant une voiture ou un piéton dans une scène de conduite. L'attention joue un rôle majeur lors de la conduite d'un véhicule. En tant que conducteurs, nous ne pouvons pas regarder partout en même temps, et parfois nous manquons des informations précieuses qui auraient pu éviter un accident, parce que l'attention est limitée [Tran et al., 2012; Lee et al., 2007]. L'énorme difficulté consiste à évaluer si un conducteur comprend la situation et s'il réagit correctement compte tenu de la scène de conduite.

En sachant comment et quand le conducteur réagit habituellement à certains éléments de la scène, nous pouvons identifier quand il ne réagit pas correctement, ou quand il pourrait être en retard pour réagir. Par exemple, si l'on peut apprendre comment un conducteur s'arrête à un feu rouge, par exemple lorsqu'il commence à décélérer, ou quel est son profil de sa décélération, le système peut alors détecter quand il ne décélère pas. En mesurant la différence entre ce que le système prédit (c'est-à-dire ce que le conducteur fait habituellement) et ce qu'il observe (ce que le conducteur fait réellement), nous pouvons détecter des réactions inhabituelles à une certaine scène. Si le conducteur s'approche d'un passage pour piétons, le système prédit qu'il doit commencer à freiner ou à décélérer à une certaine distance du passage, en fonction de la présence et de la position d'un piéton. S'il ne le fait pas, la différence entre la prédiction et le comportement observé peut déclencher un avertissement un signal ou une décélération planifiée.

#### 1.3.1.b Prédiction des intentions des usagers de la route

Les conducteurs inexpérimentés ne sont pas aussi bons que les conducteurs expérimentés pour anticiper ce qui se passera ensuite dans une scène de conduite. Afin de s'améliorer, le cerveau humain développe des modèles de prédiction basés sur l'expérience [Flanagan et al., 2003]. Une meilleure anticipation de ce que les autres participants de la route vont faire dans un avenir proche donne plus de temps pour planifier notre propre comportement. La même règle s'applique aux véhicules intelligents.

Afin d'anticiper l'évolution de la conduite, nous devons prédire le comportement futur des autres usagers de la route. Les systèmes classiques réagissent à la scène actuelle,

mais nous devons aller au-delà et réagir à une scène future probable. Si un système intelligent peut prédire le comportement futur des autres usagers de la route et anticiper ce qu'ils vont faire, il peut avertir l'ego-véhicule si un autre véhicule est susceptible d'influencer son comportement. Ainsi, le conducteur peut réagir avant qu'une situation dangereuse se produise réellement.

### 1.3.1.c La classification des manœuvres

Les méthodes de classification des manœuvres supposent que les actions du conducteur soient limitées à un ensemble de manœuvres. Étant donné qu'il n'y a qu'un ensemble fini de destinations sur une autoroute, ces approches classeront un véhicule par sa destination, et parfois par la paire origine-destination. Une fois la destination prédite, la manœuvre appropriée (tourner à gauche, tourner à droite, maintien de voie) peut être déterminée. Cela permet de prédire le reste du mouvement du véhicule, car il doit correspondre à la manœuvre.

Les manœuvres des véhicules peuvent être estimées grâce à différentes sources d'entrées de capteurs [Brand et al., 1997; Lefèvre et al., 2014]. La première source commune est celle des données des capteurs orientés vers le conducteur à l'intérieur du véhicule, comme les caméras embarquées pointant vers le visage du conducteur. Dans [Doshi, Trivedi, 2009], le regard et la dynamique de la tête du conducteur ont été collectés par une caméra pour prédire l'intention du conducteur de changer la voie. Bien que l'observation de l'état d'un conducteur donne des indices fiables pour estimer les manœuvres, ce type d'information n'est pas disponible pour estimer les manœuvres des autres conducteurs sur la route. Contrairement à ces approches, nous ne supposons pas la disponibilité de capteurs orientés vers le conducteur dans d'autres véhicules. Nous nous concentrons plutôt sur la reconnaissance des intentions des autres véhicules grâce à la perception basée sur un véhicule autonome.

Les mesures de capteurs proprioceptifs installés sur les véhicules donnent des informations sur le véhicule cible, telles que sa position, sa vitesse et son angle de lacet. Parmi les travaux pertinents, citons [Toledo-Moreo, Zamora-Izquierdo, 2010], où les auteurs utilisent des capteurs embarqués peu coûteux comme le GPS, une centrale inertielle (IMU) et l'odomètre pour prédire les manœuvres latérales et longitudinales. Dans [Berndt et al., 2008], les positions des pédales et des véhicules sont utilisées pour prédire les changements de voie en continu. Ces approches supposent l'accès aux capteurs internes du véhicule, ce qui n'est pas réaliste pour estimer les mouvements des autres véhicules. Dans notre travail, nous utilisons les informations sur l'état du véhicule qui peuvent être collectées à l'aide de capteurs extéroceptifs (caméras, radars et lidars).

La classification des manœuvres ou la reconnaissance des situations sont deux termes utilisés pour résoudre le problème de la compréhension du comportement des autres conducteurs. Dans la communauté des véhicules intelligents, de nombreux travaux ont été consacrés à ce sujet ces dernières années. Dans plusieurs travaux, un module de classification des manœuvres est utilisé afin d'avoir une meilleure prédiction. Dans [Houenou et al., 2013], un ensemble de trajectoires correspondant à différentes manœuvres est prédéfini. Ensuite, un modèle de Markov caché est utilisé pour sélectionner la trajectoire la plus probable d'un objet, en fonction de sa séquence de mesures actuelle. Dans [Campbell, 2012], les machines probabilistes à états finis sont utilisées pour modéliser des manœuvres de conduite complexes comme des séquences d'éléments de base qui sont spécifiés par un ensemble de règles dans un système de logique floue. Les règles sont obtenues à partir d'un ensemble de données d'entraînement comprenant des signaux tels que

la vitesse, l'accélération et l'angle de braquage.

Une approche à filtre Bayésien est utilisée pour reconnaître une manœuvre de conduite en calculant la probabilité de chaque élément de base dans le contexte du modèle de manœuvre. Dans [Castignani et al., 2013], la trajectoire de l'égo-véhicule est prédite en définissant un contexte de conduite qui est un vecteur contenant des signaux pertinents provenant de capteurs du banc d'essai tels que l'utilisation des clignotants ou la direction du regard du conducteur. Le contexte de conduite est enregistré en continu sur une période de 2 secondes et un classificateur fait la distinction entre le changement de voie et le maintien de la voie quelques secondes avant le début de la manœuvre.

Le changement de voie (CV) est une manœuvre qui permet aux conducteurs d'entrer dans une voie qui correspond à leurs besoins et à leur confort. Le processus de changement de voie exige au conducteur d'évaluer la situation dans sa voie d'origine et sa voie cible avant d'entreprendre la manœuvre. Les autres véhicules doivent également s'adapter pour changer de voie en toute sécurité. La trajectoire est déterminée par la précision de ces évaluations subjectives ainsi que par l'état du trafic. Une évaluation erronée par l'égo-véhicule ou des véhicules voisins ou une manœuvre incorrecte peut provoquer une collision. La collision peut être évitée si la future manœuvre est prédite.

Les problèmes de prédiction de changement de voie sont principalement traités comme des problèmes de classification [Meyer-Delius et al., 2009; Mandalia, Salvucci, 2005; Kasper et al., 2012; Kumar et al., 2013; Woo et al., 2016; Schlechtriemen et al., 2015]. Cette manœuvre peut être divisée en trois classes principales : changement de voie gauche (CVG), changement de voie droite (CVD) et maintien de voie (MV). Pour la détection du changement de voie effectué par d'autres conducteurs, les données des capteurs de perception doivent être utilisées comme des caractéristiques pour le classifieur. Les caractéristiques correctement sélectionnées déterminent les performances de détection [Woo et al., 2016; Schlechtriemen et al., 2015; Peng et al., 2015; Tomar, Verma, 2013].

Ce problème est abordé dans la littérature sous différents points de vue, dont la plupart sont basés sur des méthodes probabilistes. L'approche des réseaux bayésiens propose de reconnaître les manœuvres de conduite sur les autoroutes et les expériences sont réalisées sur des données réelles [Kasper et al., 2012]. Les auteurs de [Meyer-Delius et al., 2009] ont utilisé des modèles de Markov cachés (HMM) pour la modélisation et la reconnaissance des situations. Une manœuvre est définie comme une distribution sur des séquences d'états, les HMM tentent d'identifier chacun de ces états. L'approche a été évaluée sur des données réelles pour la conduite sur autoroute dans de nombreuses situations, elle a été capable de reconnaître et de suivre de multiples situations. Cependant, nous notons que les travaux les plus récents utilisent des approches basées sur l'apprentissage : Support Vector Machine (SVM) [Mandalia, Salvucci, 2005; Kumar et al., 2013; Woo et al., 2016] et Random Forests [Schlechtriemen et al., 2015]. Ce type de modèles s'appuie sur les données dynamiques du véhicule pour détecter l'intention du conducteur de changer de voie et, finalement, une sortie probabiliste multi-classe est donnée comme résultat.

Les réseaux de neurones artificiels (ANN) ont également été envisagés pour identifier l'intention du conducteur de changer de voie [Peng et al., 2015; Tomar, Verma, 2013; Leonhardt, Wanielik, 2017]. L'ANN apprend le schéma du raisonnement humain, l'apprentissage et les capacités cognitives de la conduite en utilisant les facteurs influençant la voie du conducteur.

Ces dernières années, et sur la base de leur succès dans plusieurs domaines, d'autres types plus avancés des réseaux de neurones sont utilisés, notamment les réseaux de neurones récurrents (RNN) [Dogan et al., 2011], le Deep Learning (apprentissage profond)

[Olabiya et al., 2017; Gurghian et al., 2016; Kim, Lee, 2014] et l'apprentissage par renforcement [Kuefler et al., 2017]. Les réseaux de neurones à convolution (CNN) sont également utilisés pour la détection de changement de voie en utilisant la prédiction d'image [Mänttari et al., 2018; Lee et al., 2017a]. Les approches basées sur l'apprentissage nécessitent des jeux de données de mouvement de haute qualité contenant des scénarios de conduite interactifs dans le monde réel pour l'apprentissage et les tests. Un bref aperçu des jeux de données utilisés sera donné dans les sections suivantes.

### 1.3.2 Prédiction de trajectoire

Un véhicule autonome circulant dans un environnement complexe rencontre de nombreux véhicules circulant à proximité. Pour une interaction sûre avec ces participants de la route, le véhicule autonome devra planifier son trajet afin d'éviter toute collision potentielle. Par conséquent, il est essentiel que le véhicule autonome prédise la trajectoire future des véhicules circulant autour de lui. Les conducteurs humains font cette prédiction de manière intuitive en observant les états des véhicules environnants ainsi que l'environnement. Les véhicules autonomes peuvent prédire la trajectoire future des véhicules environnants en utilisant des données sur leurs états passés pour connaître leur comportement, ainsi qu'en tenant compte des facteurs qui pourraient affecter leur trajectoire future, telle que la structure de la route. Cela nécessiterait un modèle capable d'analyser la trajectoire passée d'un véhicule et de prédire une trajectoire future probable.

Un tel modèle serait chargé d'apprendre le comportement d'un véhicule à partir de sa trajectoire passée. Par exemple, à partir des données d'un véhicule qui se déplace à une vitesse à peu près constante ou accélère, le modèle est capable de prédire les prochaines positions qui correspondent à l'historique de l'état du véhicule. Pour intégrer la structure de la route dans la prédiction de trajectoire, le modèle devrait être capable d'exploiter la description de mouvement dans le système de coordonnées curvilignes ainsi que d'associer les variations de la trajectoire du véhicule à la courbure de la route et de refléter cela dans la prédiction de trajectoire. Par exemple, la trajectoire future d'un véhicule roulant sur une courbe peut être prédite en obtenant sa vitesse et son accélération à partir de la trajectoire passée et en anticipant le ralentissement du véhicule en raison de la courbure de la route lorsqu'elle suit la courbe.

Il est utile de prédire la manœuvre, et donc la destination d'un véhicule, mais il serait beaucoup plus utile de prédire la trajectoire entière du véhicule. Avoir une estimation de la position où vont les véhicules environnants permet au véhicule autonome de planifier sa trajectoire. Les modèles de prédiction de trajectoire tentent de produire une trajectoire future d'un véhicule compte tenu de certaines données sur le passé du véhicule. Souvent, des modèles cinématiques ou dynamiques à base de la physique sont utilisés en conjonction avec des filtres de Kalman [Veeraraghavan et al., 2006], des simulations de Monte Carlo [Althoff, Mergel, 2011] ou des modèles de mélange gaussien variationnel [Wiest et al., 2012]. Ces méthodes produisent une proposition de prédiction unique et donne une mesure de l'incertitude. Un système de reconnaissance de manœuvre est combiné avec un prédicteur de trajectoire basé sur un processus gaussien pour permettre une prédiction multimodale [Tran, Firl, 2014]. Plus récemment, l'accent a été mis sur les approches basées sur les données pour cette tâche, car il y a eu des avancées récentes dans les calculateurs conçu pour l'apprentissage en profondeur.

Un jeu de données couramment utilisé pour ce travail est le jeu de données routières NGSIM (Next Generation Simulation), US-101 et I-80 [FHWA, 2008]. Ce jeu de données a été collecté en 2005 à l'aide de plusieurs caméras aériennes observant des sections de

route. Ces données ont été prises sur trois sections de 15 minutes chacune et contiennent les trajectoires d'environ 5000 véhicules. Des techniques de suivi visuel ont été utilisées pour extraire les trajectoires des véhicules des données d'image à une fréquence de 10 Hz. Étant donné la nature du suivi des véhicules à partir d'une caméra distante, les données souffrent d'un bruit de suivi considérable [Thiemann et al., 2008], et même avec des techniques de filtrage et de correction agressives, des problèmes importants persistent, telles que les collisions multiples [Coifman, Li, 2017].

En utilisant le jeu de données NGSIM, Morton et al. [Morton et al., 2016] ont proposé un modèle basé sur le réseau de neurones récurrent RNN de type LSTM (Long Short Term Memory) pour prédire le profil d'accélération des conducteurs sur une fenêtre de dix secondes. L'utilisation des profils d'accélération pour déterminer la vitesse et donc la position est utile, mais elle n'inclut aucune autre manœuvre, comme les changements de voie. Ils étendent leur modèle en utilisant des réseaux alternatifs génératifs [Kuefler et al., 2017] pour produire des trajectoires complètes. Les trajectoires des véhicules sur les autoroutes peuvent également être prédites à l'aide d'une grille d'occupation. En ajoutant une couche d'embarquement (embedding layer) avant le RNN, Kim et al. [Kim et al., 2017] ont codé les positions des véhicules à l'aide d'une carte unique. Pour prédire les positions du véhicule à différents horizons temporels (0,5 seconde, 1 seconde, 2 secondes), un réseau entièrement nouveau est entraîné pour chaque horizon. Les données employées dans ces expériences sont collectées à l'aide de radars Delphi longue portée et se composent de 1325 véhicules cibles. Le travail est prolongé par Park et al. [Park et al., 2018] en exploitant un codeur-décodeur RNN, permettant à un seul réseau de produire la prédiction d'emplacement du véhicule cible pour tous les horizons temporels. Cette configuration de réseau particulière permet d'utiliser la recherche en faisceau [Neubig, 2017], ce qui augmente l'espace de recherche de sortie du réseau, ce qui contribue à son tour à la précision du modèle. La configuration de codage de la grille d'occupation ne permet pas de représenter correctement la similitude des cellules voisines et cela doit donc être appris par le réseau.

Les travaux de Deo et al. [Deo et al., 2018] se basent sur un modèle de mélange gaussien variationnel combiné à un modèle de Markov caché. Cette méthode génère une prédiction de la manœuvre que le véhicule cible effectuera, puis utilise le modèle de mélange gaussien variationnel pour produire une prédiction de trajectoire que le véhicule prendra, avec incertitude. Cette méthode est validée sur un ensemble de données collectées par les auteurs à l'aide d'un système basé sur la vision. Les séquences d'entraînement sont composées de 48 enregistrements, qui sont ensuite étendus avec des techniques d'augmentation des données. Quatre enregistrements sont utilisés pour l'ensemble de données d'évaluation.

### 1.3.3 La prédiction multi-modèle

La trajectoire future d'un véhicule peut être décrite comme un problème multimodal. C'est-à-dire qu'il existe plusieurs modes (destinations) qu'un véhicule peut emprunter, mais la combinaison de ces modes n'est pas garantie d'être une autre solution valable. En tant que tel, une trajectoire unique avec incertitude est inappropriée, donc plusieurs prédictions possibles doivent être proposées.

Il est beaucoup plus utile non seulement de prédire une seule trajectoire avec incertitude, mais de prédire toutes les manœuvres possibles qu'un véhicule peut effectuer et de lui attribuer une probabilité. Cela permet au planificateur de trajectoire de considérer toutes les trajectoires possibles, pour prendre une décision plus éclairée. Deo et al. [Deo,

Trivedi, 2018] ont amélioré leurs travaux antérieurs en utilisant le schéma Social-LSTM combiné avec un classificateur de manœuvre. Ce classificateur est formé pour prédire les manœuvres dans l'ensemble de [changement de voie à gauche, changement de voie à droite, garder la voie], combiné avec [garder la vitesse, réduire la vitesse]. En entraînant un réseau par classe de manœuvre, ils sont capables de prédire une trajectoire par manœuvre possible, et sont classés par le classificateur.

Les Autoencodeurs Variationnels Conditionnels (AVC) pour la prédiction de trajectoire sont une autre technique qui a été adoptée à partir de l'écriture simulée. Le travail de Google SketchRNN [Ha, Eck, 2017] ajoute la couche Mixture Density Network (MDN) à un autoencodeur variationnel (AV) pour reproduire les dessins. Un AV est un type de réseau capable d'apprendre à reproduire des motifs. L'entrée variationnelle permet d'introduire un niveau de stochasticité, qui à son tour permet de produire différents dessins. Un AVC [Sohn et al., 2015] permet à un autoencodeur variationnel d'être conditionné sur une entrée. En utilisant ce modèle, la trajectoire d'un véhicule peut être prédite en conditionnant le trajet du véhicule. Cette technique a été récemment adoptée pour une utilisation dans la prédiction de trajectoire de véhicule et la modélisation d'interaction de véhicule. Schmerling et al. [Schmerling et al., 2018] utilisent ce modèle pour planifier le comportement d'un véhicule sur la base d'échantillons de données provenant d'un simulateur humain. En échantillonnant plusieurs fois à partir de l'AVC, une collection de trajectoires peut être produite, mais celles-ci ne sont pas classées. La moyenne et la variance de cet ensemble de trajectoires sont utilisées pour calculer l'incertitude. Lee et al. [Lee et al., 2017b] ont produit une modification des AVC pour classer ces sorties. Leur modèle «DESIRE» est testé à la fois sur une partie du jeu de données KITTI et du jeu de données Stanford Drone, démontrant l'adaptation de ces techniques entre la prédiction des véhicules et des piétons, si les données sont suffisantes. Bien que leur sortie de prédiction semble être multimodale, on ne sait pas combien de fois le réseau doit être échantillonné pour exposer tous les modes de la prédiction. Les échantillons peuvent être classés, mais l'incertitude de chaque prédiction n'est pas donnée.

Les réseaux de neurones profonds se sont également révélés utiles dans la planification de la trajectoire des véhicules. Cependant, comme ces méthodes impliquent une interaction avec la scène, la collecte de données et la validation du système sont extrêmement difficiles. Ces travaux utilisent généralement des simulateurs complets [Dosovitskiy et al., 2017; Shalev-Shwartz et al., 2016], des simulateurs humains sont également exploités [Schmerling et al., 2018], un parcours d'essai étroitement contrôlé, ou une combinaison de tous [Galceran et al., 2017]. La validation à l'aide de simulateurs est difficile lors de la modélisation de scénarios réels impliquant des risques car il n'y a pas de risque équivalent présent dans le simulateur [Rouzikhah et al., 2010]. Les simulateurs qui sont des sources fermées sont encore plus difficiles à valider, car souvent les détails sur la manière dont les données synthétiques sont générées sont omis. Sans ces informations, il n'y a aucun moyen de savoir si les scénarios générés par le simulateur sont réalistes, et il ne donne donc aucune indication de performances réelles de l'algorithme testé.

## 1.4 Taxonomie des modèles de prédiction des intentions

Afin d'anticiper le mouvement des véhicules et d'augmenter le niveau de sécurité, de nombreuses solutions sont proposées pour prédire l'intention des véhicules cibles. Pour résoudre ce problème, de nombreuses études ont tenté d'intégrer différents modèles pour identifier le mouvement futur des véhicules. Dans la section précédente, nous avons présenté l'état de l'art des trois aspects qui reformule le problème de la prédiction des inten-



tions (la prédiction de trajectoire, la classification des manoeuvres et la prédiction multimodale). Dans cette section, nous allons aborder les différents modèles proposés pour résoudre ce problème. Ces modèles sont classés en trois grandes classes [Lefèvre et al., 2014] : modèles basés sur la physique, modèles basés sur les manoeuvres et des modèles basés sur les interactions. Nous ajoutons dans cette thèse une quatrième classe : les modèles basés sur les données.

### 1.4.1 Modèles basés sur la physique

Les modèles d'évolution fondés sur la physique représentent les véhicules comme des entités dynamiques soumis aux lois de la physique. Le mouvement futur est prédit à l'aide de modèles dynamiques et cinématiques reliant les entrées de commande (l'angle de braquage, l'accélération), les propriétés de la voiture (le poids) et les conditions extérieures (coefficient de frottement de la surface de route) à l'évolution de l'état du véhicule (position, cap, vitesse). De nombreux travaux ont été effectués sur les modèles d'évolution pour les véhicules légers et ils restent couramment utilisés pour l'estimation de la trajectoire.

#### 1.4.1.a Les modèles cinématiques

Les modèles cinématiques décrivent le mouvement d'un objet en se basant sur les relations entre sa position, sa vitesse et son accélération, sans prendre en compte les forces qui agissent sur cet objet. Différents modèles peuvent être utilisés pour le suivi de véhicule [Schubert et al., 2008]. Les plus simples sont les modèles « Constant Velocity » (CV) et « Constant Acceleration » (CA) qui considèrent le déplacement linéaire. Les modèles « Constant Turn Rate and Velocity » (CTRV) et « Constant Turn Rate and Acceleration » (CTRA) intègrent les variables d'angle de braquage et de vitesse de lacet afin de prendre en compte les variations angulaires des cibles.

#### 1.4.1.b Les modèles dynamiques

Les modèles dynamiques prennent en compte l'effet des actionneurs (accélération, freinage, braquage). Ils se basent souvent sur une représentation dynamique simplifiée du véhicule (modèle bicyclette) [Lin et al., 2000]. Les modèles dynamiques prennent en compte l'effet des actionneurs pour estimer les forces qui agissent sur le déplacement du véhicule, comme les forces de contact pneu/chaussée ou les forces aérodynamiques. Dans le cas des voitures, différents modèles dynamiques sont décrits dans la littérature scientifique. Les modèles complexes prenant en compte individuellement chaque roue de la voiture sont généralement réservés à des applications de contrôle. Pour du suivi de cible, un modèle plus simple dit « modèle bicyclette » est généralement mis en œuvre. Ce modèle linéaire représente la voiture comme un véhicule deux-roues évoluant sur un plan 2D.

#### 1.4.1.c Les méthodes de génération de trajectoires prédites

Ces méthodes consistent à observer la dynamique actuelle du véhicule à l'aide d'un observateur (typiquement un filtre probabiliste), et à extrapoler sa trajectoire future en considérant que le véhicule gardera la même dynamique. En considérant l'état actuel du véhicule (position, vitesse) connu, les modèles d'évolutions cinématiques ou dynamiques

décrits précédemment sont mis en œuvre pour prédire la trajectoire future des véhicules. Les trajectoires prédites peuvent être de différentes formes :

**A) Trajectoire unique**

La méthode la plus simple est d'utiliser le modèle d'évolution décrit précédemment à partir de l'état actuel du véhicule pour générer une trajectoire unique. Cette stratégie peut être mise en œuvre à partir des modèles dynamiques ou cinématiques. Cependant, la prédiction ne prend pas en compte les incertitudes du modèle ni les évolutions possibles de la dynamique du véhicule. La trajectoire prédite n'est donc fiable qu'à court terme (de l'ordre de la seconde).

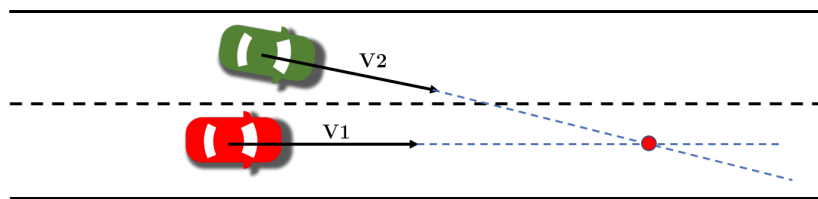


FIGURE 1.7 – Une prédiction de trajectoire basée sur le modèle de la vitesse constante.

**B) Trajectoires probabiliste (gaussiennes)**

Les observateurs probabilistes pour l'estimation de l'état d'un véhicule fournissent une estimation des incertitudes sur l'état actuel du véhicule et sur son évolution. Ces incertitudes sont généralement décrites par une distribution normale (gaussienne), représentée par les ellipses sur la figure ci-après.

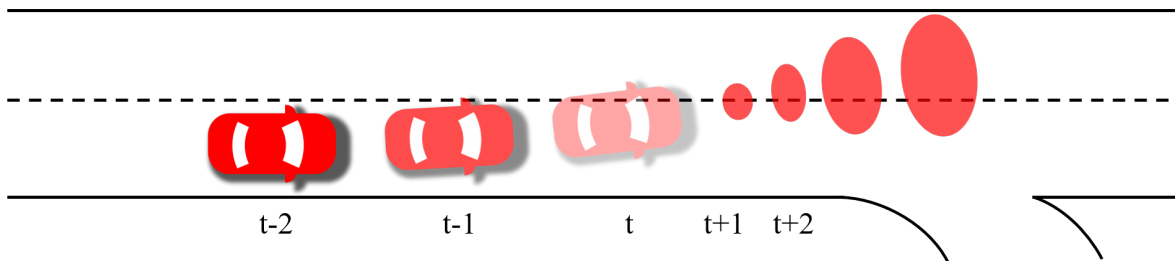


FIGURE 1.8 – Trajectoire probabiliste prédite pour le véhicule cible.

En itérant la phase de prédiction des filtres probabilistes à partir de l'état actuel, on obtient une estimation de l'état et des incertitudes à un instant futur. En comparaison à une trajectoire unique, cette technique présente l'avantage d'estimer l'incertitude de la trajectoire prédite [Ammoun, Nashashibi, 2009]. Cependant, elle conserve l'inconvénient de ne pas prendre en compte les évolutions possibles de la dynamique du véhicule.

**C) Trajectoires multiples type Monte Carlo**

Dans le cas général, c'est-à-dire quand aucune hypothèse n'est faite sur la linéarité des modèles ou sur la distribution des incertitudes, l'expression analytique pour la distribution sur les états prédits n'est habituellement pas connue. Les méthodes de Monte Carlo fournissent des outils pour approximer cette distribution [Althoff, Mergel, 2011]. L'idée est de faire un échantillonnage aléatoire à partir des variables

d'entrée du modèle d'évolution afin de générer des trajectoires potentielles futures. Pour prendre en compte la forme de la route, des pondérations peuvent être appliquées aux trajectoires générées pour pénaliser celles qui ne respectent pas les contraintes de la topologie routière. Les entrées sont généralement l'accélération et l'angle de braquage (quand il est connu) ou l'écart latéral. Pour tenir compte de la faisabilité d'une manœuvre, on peut soit retirer les échantillons de trajectoires générés avec une accélération latérale plus élevée, ce qui est physiquement possible, soit tenir compte des limitations physiques d'un véhicule dans le modèle d'évolution de sorte que les entrées soient réparties de manière plus réaliste, dans quel cas l'étape de post-traitement pour enlever des trajectoires irréalisables n'est pas nécessaire.

Les méthodes de Monte Carlo peuvent être utilisées pour prédire la trajectoire d'un véhicule soit à partir d'un état actuel parfaitement connu, soit à partir d'un état de courant incertain estimé par un filtre de Kalman.

### **Limites des modèles basés sur un modèle physique**

Comme elles ne considèrent que la dynamique actuelle du véhicule cible, les méthodes basées sur les modèles physiques sont limitées à court terme (moins d'une seconde) pour la prédiction de la trajectoire. Généralement, elles sont incapables d'anticiper tout changement dans le mouvement de la voiture causé par l'exécution d'une manœuvre particulière (par exemple freinage avant un virage), ou des changements causés par des facteurs externes (par exemple ralentissement en raison d'un véhicule à l'avant).

## **1.4.2 Modèles basés sur les manœuvres**

Les méthodes basées sur les manœuvres considèrent que le mouvement d'un véhicule sur le réseau routier correspond à une série de manœuvres exécutées indépendamment des autres véhicules. La prédiction de la trajectoire est ainsi basée sur l'identification précoce des manœuvres que les conducteurs ont l'intention d'effectuer. Si on peut identifier l'intention de manœuvre d'un conducteur, on peut prédire le mouvement futur du véhicule correspondra à cette manœuvre. Les trajectoires dérivées de ce schéma sont plus pertinentes et fiables à long terme que celles dérivées des modèles de mouvement basés sur les modèles physiques. Les modèles de mouvement basés sur la manœuvre reposent soit sur des trajectoires « types » prédéfinies (prototype), soit sur la base d'une estimation de l'intention de manœuvre.

### **A) Modèles basés sur des trajectoires prédéfinies**

L'idée est que les trajectoires des véhicules sur le réseau routier peuvent être regroupées dans un ensemble fini de trajectoires typiques. Les trajectoires réalisables sur un élément d'infrastructure sont représentées à l'aide de trajectoires prototypes qui sont généralement apprises à partir de données lors d'une phase d'entraînement. Par la suite, la prédiction peut être effectuée à partir d'une trajectoire partielle en trouvant le ou les trajectoires prototypes les plus probables en utilisant ces trajectoires comme modèle de mouvement futur. Il existe plusieurs possibilités pour représenter un modèle de mouvement basé sur les trajectoires des échantillons. Une solution consiste à calculer une trajectoire prototype unique pour chaque type de manœuvre (par exemple, l'ensemble des manœuvres réalisables sur une intersection, voir figure 1.9), en agglomérant les trajectoires précédemment observées.

Cette trajectoire prototype est représentée sous forme stochastique en calculant la moyenne et l'écart-type des trajectoires observées lors de l'apprentissage.

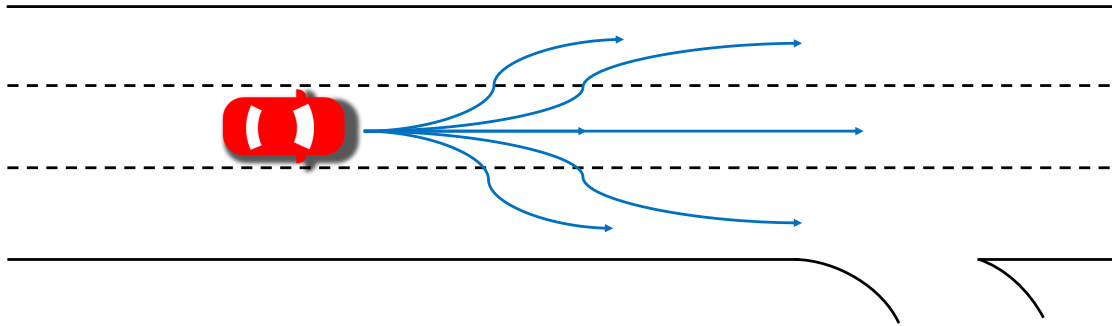


FIGURE 1.9 – Trajectoires observées pour les véhicules venant de la gauche.

Une autre possibilité pour représenter les trajectoires de référence est l'utilisation des Processus Gaussiens (Gaussian Processes, GPS) qui semblent bien adaptés pour représenter les manœuvres dans le contexte de la circulation routière [Tay, 2009; Tran, Firl, 2014]. Les GPS peuvent être considérés comme une généralisation des distributions de probabilité gaussiennes. Ils modélisent un processus comme une distribution gaussienne sur une fonction. Lorsqu'il est appliqué dans le contexte des trajectoires de véhicules, l'hypothèse est que les trajectoires dans l'ensemble de données d'apprentissage sont des fonctions d'échantillonnage d'un processus gaussien. Par conséquent, l'apprentissage consiste à ajuster une distribution gaussienne sur ces fonctions. Les avantages principaux des GPS sont leur robustesse au bruit dans les trajectoires observées (par rapport aux approches présentées précédemment) et leur capacité à représenter les variations dans l'exécution d'un modèle d'évolution d'une manière cohérente et probabiliste.

A partir de la trajectoire partielle exécutée par un véhicule, la prédiction peut être effectuée en comparant cette trajectoire partielle avec les trajectoires de références (prototypes, ou GPS) apprises, en sélectionnant le ou les trajectoires les plus probables. La similarité entre la trajectoire partielle et une trajectoire modèle est estimée en définissant une métrique de distance.

Lorsque les motifs de mouvement sont représentés par un ensemble fini de trajectoires prototypes, la distance d'une trajectoire partielle à un motif de mouvement est mesurée par sa similarité avec les trajectoires de prototype. Un certain nombre de métriques ont été définies pour mesurer la similarité entre deux trajectoires, notamment la distance Euclidienne moyenne entre les points des trajectoires, ou la plus longue sous-séquence commune (LCS) [Buzan et al., 2004]

## B) Modèles de classification des manœuvres en cours

Une alternative à l'utilisation des trajectoires de références (prototypes ou GPS) est d'estimer l'intention de manœuvre du conducteur (s'arrêter à un stop, suivre un autre véhicule, changer de voies, exécuter un virage à gauche, etc. . .) et ensuite prédire les états physiques successifs correspondant à une possible exécution des manœuvres (Figure 1.10). Un avantage majeur par rapport aux trajectoires prototypes est qu'il n'est pas nécessaire de faire correspondre la trajectoire partielle avec une trajectoire observée précédemment.

Différentes variables d'entrée peuvent être utilisées pour estimer l'intention de manœuvre d'un conducteur, par exemple l'état physique du véhicule (position, vitesse, cap, accélération, vitesse de lacet, clignotant, etc.), les informations sur le réseau routier (géométrie et topologie de la route, limite de vitesse, règles de circulation, etc.), et le comportement du conducteur (mouvement de la tête, style de conduite, etc.).

La classification des manœuvres peut être réalisée en se basant sur des algorithmes d'apprentissage discriminatifs (par exemple Multi-Layer Perceptrons (MLP) [Ortiz et al., 2011], ou alors en décomposant chaque manœuvre en une chaîne d'événements consécutifs et de représenter cette séquence d'événements en utilisant un modèle de Markov caché (Hidden Markov Model, HMM) [Berndt et al., 2008].

Une fois la manœuvre identifiée, les trajectoires futures sont prédites de sorte qu'elles correspondent à la manœuvre identifiée. Cela peut être fait de manière déterministe, en dérivant les commandes d'entrée correspondant à la manœuvre reconnue, puis en générant une seule trajectoire à partir d'un modèle de mouvement cinématique [Tamke et al., 2011]. Un problème avec cette approche strictement déterministe est qu'il ne prend pas en compte les incertitudes sur l'état actuel du véhicule, sur la classification de la manœuvre ou sur l'exécution de cette manœuvre. Afin d'explorer l'espace des exécutions potentielles d'une manœuvre de manière probabiliste, on peut notamment utiliser le GPS.

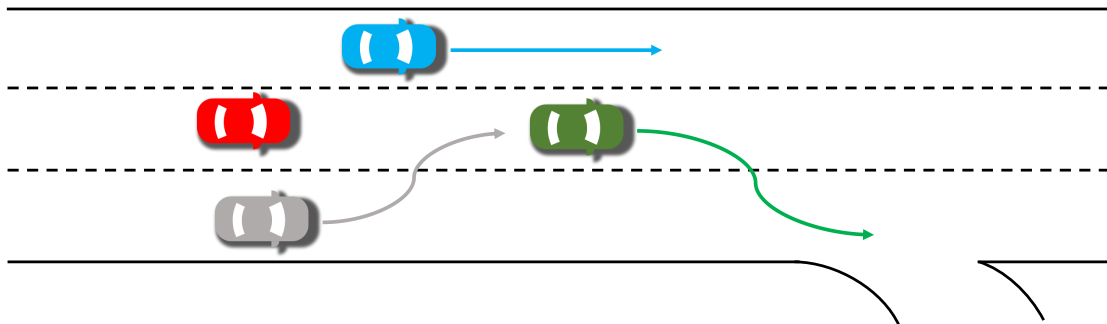


FIGURE 1.10 – Manœuvres probables pour les véhicules de la scène.

### Limites des modèles basées sur les manœuvres

Pendant longtemps, la principale limitation de l'utilisation des trajectoires prototypes était leur représentation strictement déterministe du temps. En effet, lorsque les motifs de mouvement sont représentés à l'aide d'un ensemble fini de trajectoires, il faudrait un très grand nombre de prototypes pour modéliser la grande variation dans l'exécution d'une manœuvre. Manipuler des variations plus subtiles de la vitesse comme celles provoquées par la circulation congestionnée est toujours un problème pour de tels modèles. Dans une certaine mesure, l'introduction des processus gaussiens a résolu ce problème en permettant une représentation indépendante de l'instant de réalisation de la manœuvre.

Cependant les GPS souffrent d'autres limitations. En plus de leur charge de calcul importante, ils n'ont pas la capacité de prendre en compte les limitations physiques d'un véhicule et peuvent donc générer des trajectoires d'échantillons irréalistes. Une solution à ce problème pourrait être l'utilisation d'un algorithme RRT (Rapidly Exploring Random Tree) [Aoude et al., 2011] pour générer de façon aléatoire des points vers des trajectoires réalisables de manière dynamique, en utilisant comme entrées l'état actuel du véhicule et

les trajectoires d'échantillons générées par les GP. Une autre difficulté lors de l'utilisation de trajectoires prototypes est leur adaptation à différentes dispositions routières, en particulier lorsqu'elles sont appliquées à des intersections routières. Parce que chaque modèle de mouvement est formé pour une géométrie et une topologie d'intersection spécifiques, ils ne peuvent être réutilisés qu'à des intersections avec une disposition similaire.

### 1.4.3 Modèles basés sur les interactions entre les véhicules

La prise en compte des interactions entre les différents véhicules de la scène conduit à une meilleure interprétation de leurs trajectoires par rapport aux trajectoires de référence décrites dans la section précédente. Elle contribue ainsi à une meilleure compréhension de la situation et à une estimation plus fiable des trajectoires futures. Malgré cela, il existe peu de méthodes qui permettent de prendre en compte ces interactions dans la littérature. Elles sont basées sur des trajectoires prototypes ou sur des Réseaux Bayésiens Dynamiques (RBD).

#### A) Utilisation de trajectoires prototypes

Pour les méthodes reposant sur des trajectoires prototypes, les influences intervéhiculaires ne peuvent pas être prises en compte au cours de la phase d'apprentissage, car le nombre de trajectoires de référence deviendrait rapidement intraitable. Cependant, il est possible de prendre en compte les influences mutuelles pendant la phase d'appariement en supposant que les conducteurs ont une forte tendance à éviter les collisions quand ils le peuvent [Käfer et al., 2010]. Les paires de trajectoires qui entraînent une collision sont pénalisées dans le processus d'appariement et, par conséquent, les trajectoires sûres sont toujours considérées comme plus probables que les trajectoires dangereuses. Cette approche est une solution pour prendre en compte les interdépendances lors de l'utilisation de trajectoires prototypes.

#### B) Modèles par réseaux bayésiens dynamiques

La plupart des modèles de mouvements interactifs sont basés sur des réseaux bayésiens dynamiques (RBD). Les dépendances (risques de collisions) entre les trajectoires de véhicules différents peuvent être modélisées avec des HMM couplés (CHMM) [Brand et al., 1997]. Cependant, comme le nombre de dépendances possibles croît quadratiquement avec le nombre d'entités, la complexité est difficilement gérable dans le contexte de situations de trafic complexes. Une solution pour simplifier le modèle est de rendre les CHMM asymétriques en supposant que le trafic environnant affecte le véhicule d'intérêt, mais pas l'inverse [Tay, 2009]. L'hypothèse de dépendances asymétriques réduit considérablement la complexité de calcul du problème. Il a été utilisé dans un certain nombre d'ouvrages, notamment en ce qui concerne les changements de pistes et les manœuvres de dépassement. Le fait que les interactions entre véhicules soient régies par les règles de la circulation peut également être exploité.

### Limites des modèles basées sur les interactions

Les méthodes basées sur les interactions permettent des prédictions à plus long terme par rapport aux méthodes basées sur des modèles physiques et sont plus fiables que les méthodes de mouvement à manœuvre car ils tiennent compte des dépendances entre les véhicules. Cependant, cette exhaustivité présente certains inconvénients : le calcul de

toutes les trajectoires potentielles des véhicules avec tous modèles est coûteux en terme de temps de calcul.

#### 1.4.4 Modèles basés sur les méthodes d'apprentissage avancées

Les nouvelles approches de prédiction des intentions consistent à créer des modèles qui recherchent des relations dans les données. Le Machine Learning ou l'apprentissage de la machine représente une famille de méthodes où un modèle paramétré apprend à modéliser un processus ou à trouver des modèles dans les données par un apprentissage supervisé, donc basé sur des exemples de paires entrée-sortie. Les réseaux neuronaux profonds (Deep Neural Network DNN) deviennent de plus en plus dominants dans plusieurs domaines, tels que la vision par ordinateur, la reconnaissance vocale et le traitement du langage naturel. Ces dernières années, ces méthodes ont fait leur entrée dans la prédiction des intentions [Park et al., 2018; Deo et al., 2018; Neubig, 2017; Kim et al., 2017].

Les approches fondées sur l'apprentissage approfondi n'exigent pas la définition des relations entre les caractéristiques et les comportements de conduite. Ces modèles tentent plutôt de trouver les relations dans les données qui peuvent expliquer le comportement de conduite, en examinant des données d'exemple. Il n'est pas nécessaire de créer des modèles physiques ou des modèles de graphes authentiques où les relations entre les états et les futurs comportements de conduite doivent être bien comprises et définies manuellement. Au contraire, si l'on dispose d'une quantité et d'une diversité suffisantes de données d'apprentissage pour la tâche souhaitée, ces relations peuvent être apprises par le modèle. La quantité de données d'apprentissage est déterminée par la complexité du modèle, où les tâches complexes qui nécessitent des modèles plus grands exigent plus de données pour entraîner le modèle.

Le type de données et la complexité de la tâche définit l'architecture du modèle qui sera le plus efficace, plutôt que la signification physique des données. Pour la classification des manœuvres ou pour la prédiction de trajectoire, on utilise principalement des outils qui analysent une trajectoire passée pour prédire le comportement futur. Une trajectoire est alors considérée comme toute autre série temporelle, une séquence d'états dans le temps avec un ensemble de caractéristiques qui décrivent l'état du véhicule à chaque pas de temps. Après l'entraînement avec ces séries temporelles, le modèle devient capable de prédire la trajectoire et d'identifier la future manœuvre.

## 1.5 Travaux de prédiction des intentions réalisés à VEDECOM

Cette thèse est une suite des travaux réalisés au sein de l'institut VEDECOM dans le cadre de la thématique " estimation des intentions des usagers de la route pour le véhicule autonome". Cette section synthétise l'avancée des travaux au sein de l'équipe VEH08 sur la prédiction. Les travaux se concentrent sur des situations de conduite interurbaines avec des véhicules légers.

La première partie présente la reconstruction et la prédiction de la trajectoire du véhicule cible avec compensation du déplacement de l'égo-véhicule via un filtre de Kalman étendu. La deuxième partie détaille le principe de l'identification et de la classification de la manœuvre initiée par la cible, ainsi que les résultats sur des essais de changement de voie réalisés sur piste.

### 1.5.1 Reconstruction de trajectoire d'un véhicule cible

Lorsqu'un véhicule cible est suivi (algorithme de tracking) dans le temps, il est possible de reconstruire sa trajectoire passée en mémorisant ses positions successives. La position des cibles issue des capteurs extéroceptifs est donnée en référence à la position de l'égo véhicule. Lorsque l'égo-véhicule est en mouvement, la référence change. Si on souhaite exprimer la trajectoire de la cible dans un repère fixe, il faut donc compenser le déplacement de l'égo-véhicule ( $dx$ ,  $dy$ ) par rapports aux mesures précédents. Il est pour cela nécessaire d'estimer le déplacement de l'égo véhicule entre deux mesures.

#### 1.5.1.a Compensation du déplacement de l'égo véhicule

La trajectoire d'un véhicule est définie par ses différentes positions au cours du temps, données dans un même repère immobile (lié à la route, par exemple le référentiel terrestre). La position des véhicules cibles sont mesurées par les algorithmes de perception mis en œuvre pour les différents capteurs extéroceptifs de l'égo-véhicule (Caméra, Lidar, Radar). Les sorties de ces algorithmes sont fusionnées afin d'assurer le suivi des cibles dans le temps, et de filtrer les bruits de mesure.

Cependant, les positions des cibles issues des algorithmes de perception sont données relativement à la position de l'égo-véhicule. Si l'égo-véhicule est en mouvement, il est donc nécessaire de projeter ces données dans un repère immobile. Pour cela, il faut connaître précisément le déplacement de l'égo-véhicule dans ce repère. Dans un premier temps, un GPS RTK (précision centimétrique) embarqué sur l'égo véhicule est utilisé pour mesurer directement les déplacements ( $dx$ ,  $dy$ ) de l'égo-véhicule, et ainsi déduire le déplacement de la cible dans un repère fixe.

Dans un second temps, ces déplacements pourront être estimés par un observateur d'état pour l'égo-véhicule. Les observateurs se basent sur des capteurs proprioceptifs standards disponibles sur le bus CAN (vitesse véhicules, accélération, angle de braquage). L'utilisation des infos GPS, lorsqu'elles sont disponibles, permet d'initialiser l'observateur et de limiter la dérive sur les positions de l'égo-véhicule. L'observateur de l'égo-véhicule est en cours de réalisation. Une fois le déplacement de l'égo véhicule estimé, les données capteurs ( $x_{target}$ ,  $y_{target}$ ) sont converties dans le repère fixe. Ces données sont alors utilisées comme données d'entrée pour l'observation de la dynamique de la cible.

En considérant à la fois les incertitudes sur les positions des cibles liés aux bruits de mesure des capteurs de perceptions, et les incertitudes liées à la position de l'égo-véhicule, il est possible d'estimer les incertitudes sur les positions des véhicules cibles dans ce repère fixe.

#### 1.5.1.b Observation de la dynamique de la cible

Pour comprendre les intentions du conducteur, il est intéressant de connaître ses actions actuelles, c'est-à-dire les consignes de conduite qu'il est en train de donner au véhicule. Ces consignes prennent la forme d'une accélération (négative dans le cas d'un freinage et positive dans le cas d'une accélération) et d'un angle de braquage, qui définit la courbure de la trajectoire du véhicule (la courbure est l'inverse du rayon de courbure) elle est nul quand le véhicule va tout droit, et généralement positive en cas de virage à gauche et négative en cas de virage à droite.

En connaissant les positions successives d'une cible dans le repère fixe, la trajectoire passée est construite. Cette trajectoire passée va permettre d'estimer non seulement sa position actuelle, mais également sa vitesse, son cap et son accélération, voire même sa



courbure. Pour cela, le filtre de Kalman étendu (EKF) décrit ci-après est mis en œuvre [Julier, Uhlmann, 1997]. le vecteur d'état considéré suivant :

$$X = [x_{\text{ref}}, y_{\text{ref}}, v_{\text{longi}}, \Psi, c, a_{\text{longi}}]^T \quad (1.1)$$

Avec :

- $(x_{\text{ref}}, y_{\text{ref}})$  : position de la cible dans le repère fixe (en m)
- $v_{\text{longi}}$  : vitesse longitudinale (en  $m/s$ )
- $\Psi$  : angle de cap du véhicule cible (en radian, orienté vers l'Est, sens trigonométrique)
- $c$  : courbure de la trajectoire de la cible (l'inverse du rayon de courbure, en  $m^{-1}$ )
- $a_{\text{longi}}$  : accélération longitudinale de la cible (en  $m/s^2$ ).

Et le modèle d'évolution non linéaire  $X_{k+1} = f(X_k)$  :

$$f : \begin{cases} x_{k+1} = x_k + \cos(\Psi_k) * (v_k \Delta t + \frac{1}{2} a_k \Delta t^2) - \sin(\Psi_k) * \frac{1}{2} v_k^2 c_k \Delta t^2 \\ y_{k+1} = y_k + \sin(\Psi_k) * (v_k \Delta t + \frac{1}{2} a_k \Delta t^2) + \cos(\Psi_k) * \frac{1}{2} v_k^2 c_k \Delta t^2 \\ v_{k+1} = v_k + a_k \Delta t \\ \Psi_{k+1} = \Psi_k + v_k c_k \Delta t \\ c_{k+1} = c_k \\ a_{k+1} = a_k \end{cases} \quad (1.2)$$

Le vecteur de mesure est composé des coordonnées  $(x_{\text{ref}}, y_{\text{ref}})$  de la cible dans le repère fixe, issue du capteur LIDAR. Quand une communication V2V est présente entre l'égo est la cible, les informations échangées (vitesse, accélération, angle de braquage) pourront être intégrées au vecteur de mesure.

### 1.5.1.c Résultats de l'observation d'état de la cible

La figure 1.11 montre les résultats de l'observateur lors d'un roulage sur la piste d'essai de Satory. L'égo-véhicule roule à vitesse constant (environ 35 km/h), la cible réalise une accélération de  $2 m/s^2$  ( $t=65s$ ) puis un changement de voie ( $t=78s$ ).

Les figures montrent les estimées de la vitesse (figure 1.11 (a)), de l'orientation (figure 1.11(b)), de l'accélération (figure 1.11 (c)), et de la courbure (figure 1.11 (d)) du véhicule cible. La cible est détectée par l'égo-véhicule à  $t=50s$  et disparaît à  $t=90s$ . Les signaux de référence (vitesse, orientation et accélération) sont donnés par la centrale inertielle ATLANS du véhicule cible, la courbure de référence est estimée à partir de la mesure de l'angle volant disponible sur le bus CAN et de l'empattement du véhicule. L'observateur met environ 3 secondes à converger. A partir de  $t=53s$ , les estimations sont cohérentes. On note une très bonne estimation de la vitesse et de l'angle de cap du véhicule cible, avec des retards inférieurs à la demi-seconde. En ce qui concerne l'accélération longitudinale, l'estimation est également précise, mais avec un retard de l'ordre de 1 seconde et de léger dépassement. Ce retard dans l'estimation de l'accélération explique le phénomène de dépassement pour l'estimation de la vitesse (par exemple à  $t=66s$ ) en phase de forte accélération ou de freinage ( $> 1 m/s^2$ ). L'estimation de la courbure est un peu moins précise (précision de l'ordre  $5.10^{-3} m^{-1}$ ) avec un retard de quasiment 2 secondes. Ce retard explique la présence d'un dépassement pour l'estimation de l'orientation quand la courbure varie brusquement.

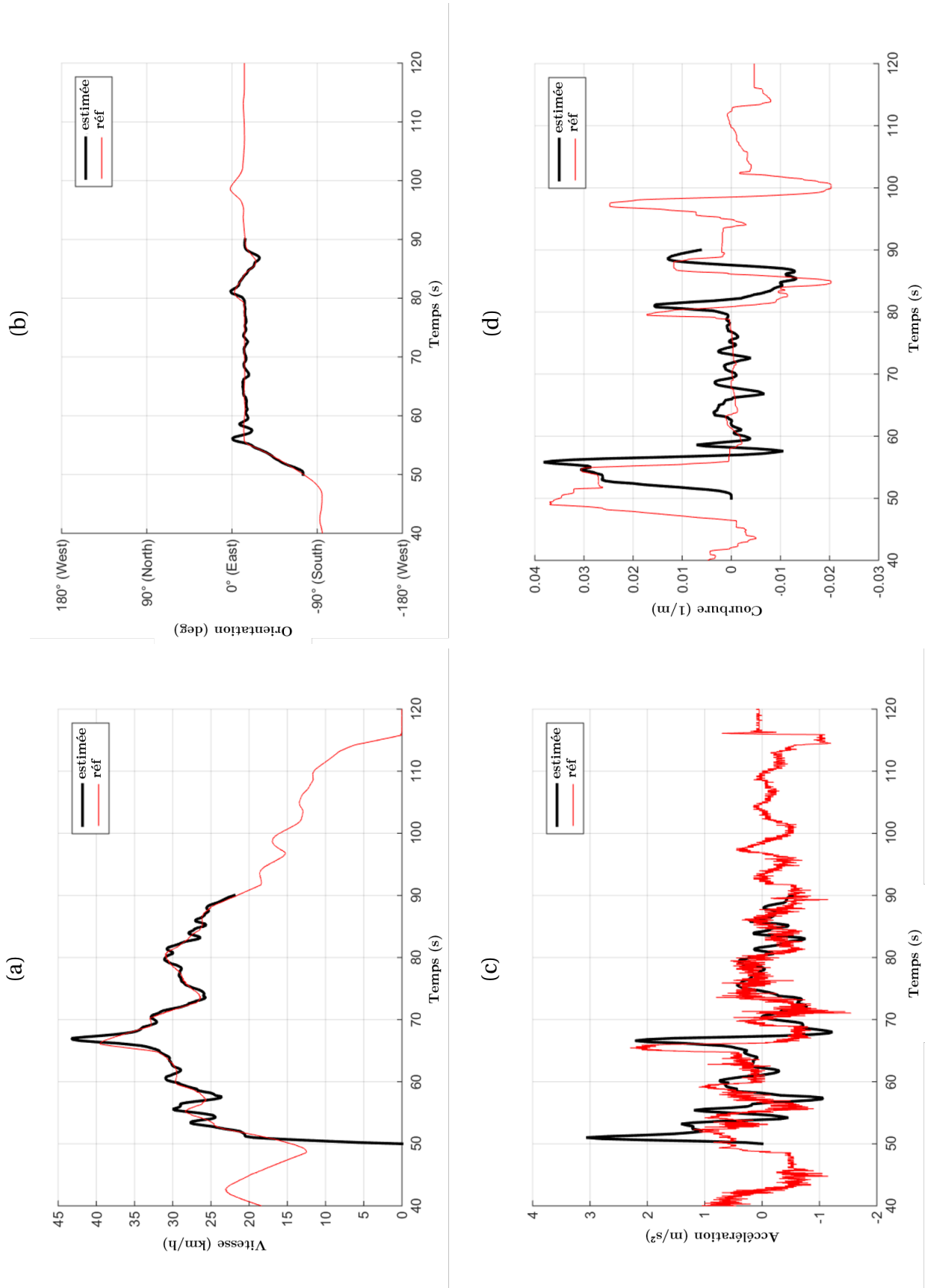


FIGURE 1.11 – Résultats de l'observation de la dynamique de la cible.

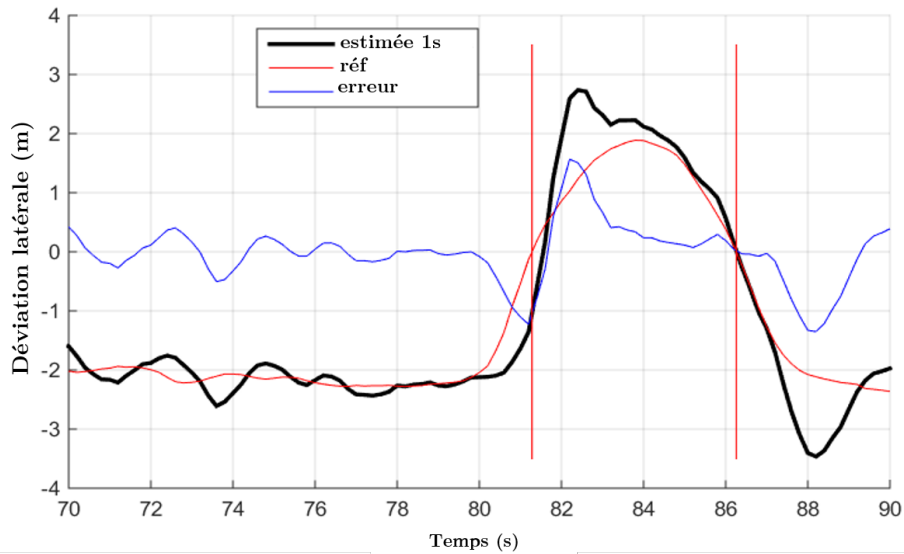


FIGURE 1.12 – Estimation de l'écart latéral à 1 seconde

### 1.5.2 Prédiction de la trajectoire future

L'observateur décrit précédemment a permis d'estimer la dynamique actuelle de la cible (vitesse, cap, accélération, courbure). La trajectoire future peut être estimée en supposant cette dynamique constante sur un horizon temporel suffisamment court. On compare ci-après (figure 1.12) l'estimation à 1 seconde de l'écart latéral de la cible par rapport aux marquages. A l'instant  $t$  (courbe noire). Ensuite l'écart latéral réel est mesuré 1 seconde plus tard (courbe de référence en rouge). L'écart latéral est nul lorsque la cible roule sur le marquage entre les 2 voies, il est positif quand la cible roule sur la voie de gauche et négatif quand elle roule sur la voie de droite. L'erreur d'estimation (différence entre les 2 courbes) est affichée en bleu. Lors de cet enregistrement, la cible effectue d'abord un changement de voie, de la voie de droite vers la voie de gauche, puis elle retourne sur la voie de droite. Les instants de franchissement des lignes sont indiqués par un trait à  $t = 81.2s$  et  $t = 86.2s$ .

L'erreur d'estimation à 1 seconde est de l'ordre de 50cm pendant le suivi de voie. Lors d'un changement de voie, le début de la manœuvre est détecté avec un retard d'environ 1s (le retard de l'observateur). Cependant, l'instant de franchissement de la voie est prédit avec un retard faible ( $< 0.5s$ ). On observe un dépassement sur l'estimation de l'écart latéral en fin de manœuvre de changement de voie ( $t = 82s$  et  $t = 88s$ ).

### 1.5.3 Classification des manœuvres en cours

Si l'estimation de la trajectoire par un estimateur de la dynamique reste cohérente à 1 seconde, elle perd en précision sur un horizon plus important car l'hypothèse que la dynamique actuelle reste constante n'est plus valable. Au-delà de la seconde, il est alors nécessaire d'identifier la manœuvre que la cible est en train de réaliser. Dans cette partie l'état des travaux sur la classification des manœuvres sont présentés.

#### 1.5.3.a Présentation de l'algorithme de classification

Les dispositifs de perception à bord de l'égo-véhicule (caméra, radar, LIDAR) permettent de reconstruire une carte locale autour de l'égo-véhicule, comprenant des in-

formations sur l'infrastructure (nombre de voie, position des marquages, vitesse limites, etc.) et sur les obstacles présents (position et dynamique actuelle). Cette carte locale sert de support aux algorithmes de planification de trajectoire de l'égo véhicule. En positionnant la cible sur cette carte locale, on obtient une carte locale autour de la cible. Il est ainsi possible d'utiliser ces mêmes algorithmes de planification de trajectoire, cette fois appliqués à la cible. Afin d'estimer la manœuvre engagée par une cible, on génère différentes trajectoires possibles correspondant aux manœuvres que la cible est susceptible de réaliser. Pour une voiture circulant sur la voie de droite sur une route interurbaine à voies multiples, les 5 manœuvres envisagées sont :

1. Suivi de la voie de droite à vitesse constante (nommée RCV pour right, constant velocity)
2. Suivi de la voie de droite avec un freinage important ( $5 \text{ m/s}^2$ ) (nommée RCB pour right, constant breaking)
3. Suivi de la voie de droite avec une forte accélération ( $2 \text{ m/s}^2$ ) (nommée RCA pour right, constant acceleration)
4. Changement de voie, de la voie de droite vers la droite (en cas de présence d'une troisième voie) (nommée R2R, from right to right)
5. Changement de voie vers la gauche (nommée R2L from right to left)

Les mêmes manœuvres sont définies pour la voie de gauche :

6. Suivi de la voie de gauche à vitesse constante (LCV)
7. Suivi de la voie de gauche avec un freinage important (LCB),
8. Suivi de la voie de gauche avec une forte accélération (LCA),
9. Changement de voie, de la voie de gauche vers la droite (L2R),
10. Changement de voie, de la voie de gauche vers la gauche (en cas de présence d'une troisième voie) (L2L).

On cherche à mesurer la probabilité  $t_{j,k}$  que la cible exécute actuellement la manœuvre  $j$  à l'instant  $k$  (pour  $j$  allant de 1 à 10). Lorsque la manœuvre est impossible à réaliser, la probabilité associée pourrait être réduite. Par exemple, quand la cible circule déjà sur la voie de gauche, elle ne peut pas réaliser un changement de voie vers la gauche (manœuvre n°5). Dans le cadre de cette étude, une hypothèse est faite que l'information sur le nombre de voie n'est pas disponible. La probabilité des autres manœuvres est initialisée au moment de la première détection de la cible à  $\mu_{j,0} = 1/n$ , avec  $n$  égal au nombre de trajectoires possibles. Ces probabilités sont ensuite mises à jour à chaque nouvelle estimation de la position de la cible selon l'algorithme suivant :

- Pour chacune des autres manœuvres possibles à réaliser, une trajectoire modèle correspondant à la manœuvre est calculée. Ces trajectoires modèles sont alors comparées à la trajectoire actuelle de la cible. Pour cela, on définit une fonction de distance ( $d$ ) entre deux trajectoires comme la somme des distances euclidiennes entre les points de la trajectoire modèle et de la trajectoire de la cible :

$$d_j = \sum_i \sqrt{(x_{i,j} - \hat{x}_i)^2 + (y_{i,j} - \hat{y}_i)^2}$$

Où  $(x_{i,j}, y_{i,j})$  sont les coordonnées du  $i$ ème point de la trajectoire modèle correspondant à la manœuvre  $j$ , et  $(\hat{x}_i, \hat{y}_i)$  sont les coordonnées de la trajectoire actuelle de la cible.

- En fonction des distances entre les trajectoires modèles et la trajectoire actuelle, il est alors possible d'associer une vraisemblance à chacune des manœuvres. On propose la définition de la vraisemblance suivante

$$L_j = 1 - \frac{d_j}{\sum_i d_i}$$

Les vraisemblances des trajectoires sont comprises entre 0 et 1. Lorsque la distance entre la trajectoire modèle et la trajectoire actuelle est faible, la vraisemblance s'approche de 1, et elle tend vers 0 quand la distance avec la trajectoire actuelle augmente.

- Finalement, on se sert de la vraisemblance calculée pour chaque trajectoire modèle afin de mettre à jour la probabilité associée à la manœuvre correspondante :

$$\mu_{j,k+1} = \frac{\mu_{j,k} L_j}{\sum_i \mu_{i,k} L_i}$$

Le résultat de la classification correspond à la manœuvre avec la probabilité la plus élevée.

### 1.5.3.b Génération des trajectoires modèles

Les trajectoires modèles sont générées à chaque itération de l'algorithme afin de les caler sur la vitesse actuelle de la cible. on considère 2 voies, et sur chaque voie, les 5 manœuvres citées précédemment (voir figure 1.13) sont converties en trajectoires modèles. Les trajectoires sont calculées pour une durée de 2.8 secondes, avec un période d'échantillonnage de 0.2s, soit 15 points. Elles débutent au centre de la voie avec une vitesse initiale égale à la vitesse estimée de la cible. Pour les 3 premières trajectoires (suivi de voie), l'écart latérale reste nul et la vitesse évolue en fonction de l'accélération considérée : 0m/s<sup>2</sup> (en bleu), +2 m/s<sup>2</sup> (en noir), et -5m/s<sup>2</sup> (en magenta). Pour les changements de voie (en cyan et en vert), la vitesse est considérée constante et l'écart latéral suit une fonction sinusoïdale. La période du signal sinusoïdal est calculée afin d'obtenir un changement de voie complet en 3 seconde. La distance parcourue pendant le changement de voie est ainsi proportionnelle à la vitesse de la cible.

### 1.5.3.c Résultats de la classification de manœuvres

On compare la trajectoire mesurée et prédite de la cible avec les différentes trajectoires modèles. Par exemple, sur la figure 1.14, on voit que la trajectoire mesurée (en rouge) est très proche de la trajectoire modèle correspondant à un changement de voie droite -> gauche (en cyan). La distance sera donc minimale pour ce modèle, ce qui résulte en une vraisemblance proche de 1.

On calcul ainsi la vraisemblance (figure 1.15) des différents modèles pour la manœuvre de double changement de voie vue précédemment. Les traits rouges correspondent aux franchissements de la ligne médiane. Quand la cible suit la voie de droite (t < 80s), la vraisemblance du modèle correspondant reste maximal avec une valeur supérieure à 0.9. En cas de changement de voie, plus d'une seconde avant le franchissement de la ligne, les vraisemblances des modèles correspondants aux changements de voie dépasse celle des modèles suivis de voie. La figure 1.16 représente les manoeuvres choisies en fonction du maximum vraisemblance.

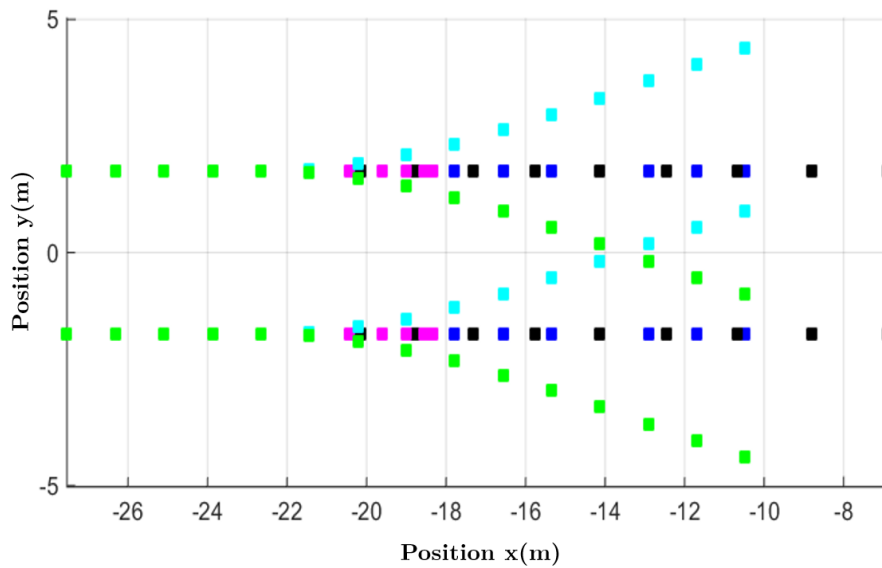


FIGURE 1.13 – Représentation des trajectoires modèles

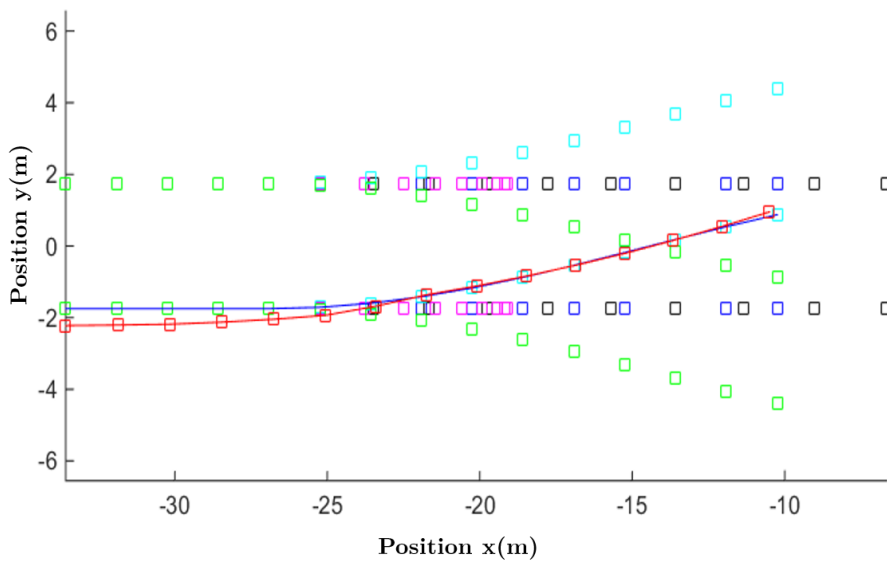


FIGURE 1.14 – Comparaison entre la trajectoire réelle et les trajectoires modèles

Les probabilités associées aux modèles varient selon les vraisemblances. On retrouve bien que le modèle 1 (suivi de voie à droite) est le plus probable jusqu'à  $t = 80s$ . A l'instant  $t = 80.4s$ , le modèle « changement de voie droite  $\rightarrow$  gauche devient le plus probable. La détection du changement de voie survient environ 1 seconde avant le franchissement du marquage par la cible. Le premier coup de volant correspondant à la manœuvre a été enregistré à  $t = 79s$ . La détection a donc lieu environ 1.5 seconde après que la cible ait initié le changement de voie. Les résultats sont semblables lors du changement de voie gauche  $\rightarrow$  droite. On remarque cependant que suite à ce changement de voie ( $t = 87s$ ), le modèle changement de voie droite  $\rightarrow$  droite devient le plus probable pendant 1 seconde, On a donc une fausse détection. Ceci est dû au dépassement observé dans l'estimation de l'écart latéral à 1 seconde.

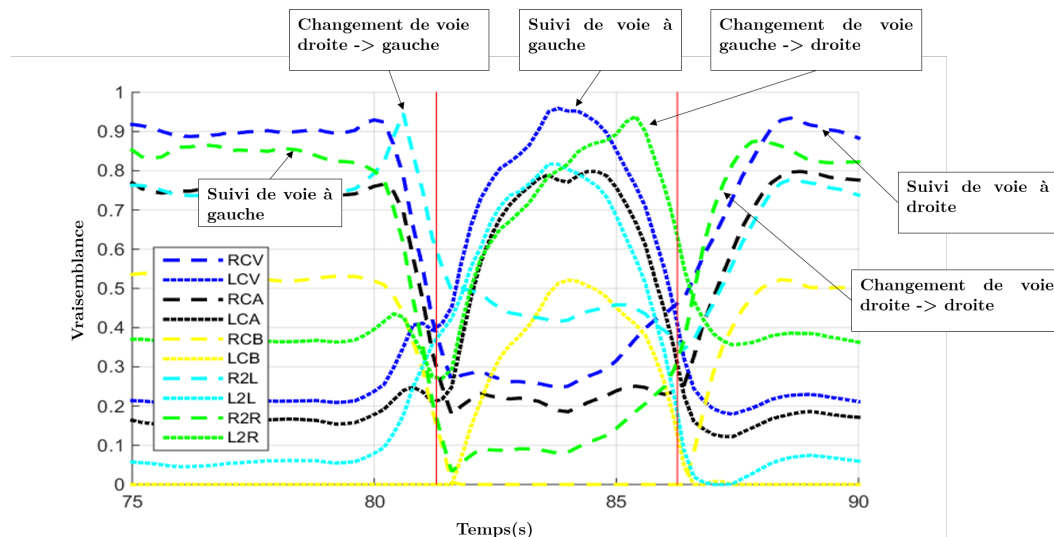


FIGURE 1.15 – Vraisemblance des trajectoires modèles pendant des changements de voie.

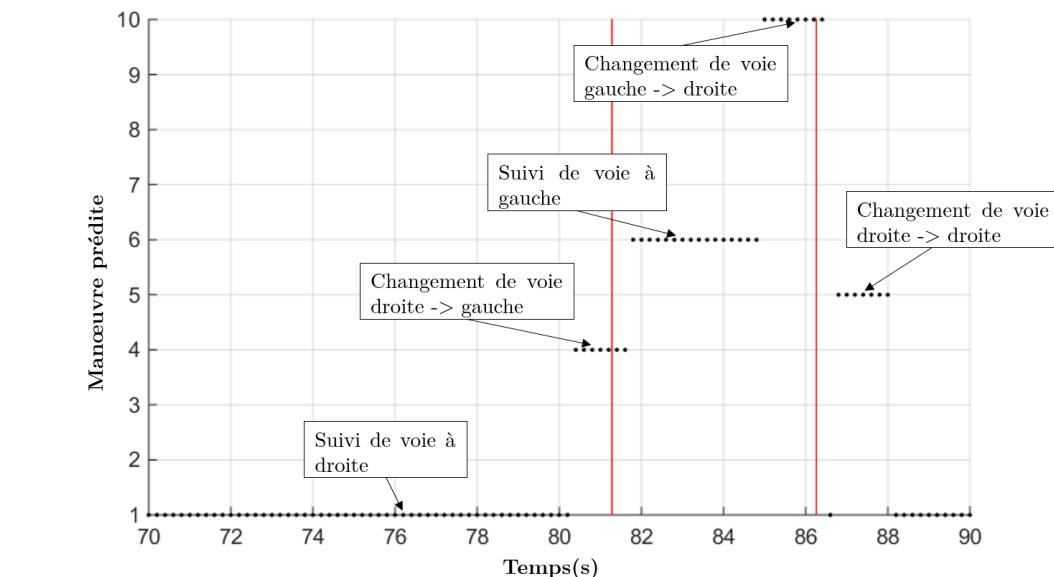


FIGURE 1.16 – Classification de la manœuvre en cours.

## 1.6 Jeux de donnée véhiculaire

Les jeux de données sont essentiels pour le développement et l'évaluation d'une méthode basée sur l'apprentissage. Il faut veiller à ce que les données collectées permettent de représenter correctement le problème à résoudre. Pour la conduite autonome, de plus en plus de jeux de données deviennent disponibles. Ces jeux de données sont souvent des enregistrements de données sensorielles différents en taille et en type de données enregistrées. Pour évaluer clairement l'impact des méthodes proposées sur la prédiction des intentions, un jeu de données approprié doit être choisi. Dans l'idéal, les données collectées sont totalement naturelles, le conducteur doit être à l'aise dans le véhicule de manière à conduire comme il le fait normalement. En outre, les données représentatives du problème devraient être collectées publiquement, lorsque les citoyens conduisent leur propre véhicule et n'ont pas connaissance qu'ils font l'objectif d'un jeu des données collectées. Seulement dans ce cas, le jeu de données peut capturer des nuances dans le com-

portement du conducteur, telles que le dépassements dangereux ou l'excès de vitesse. Toutes sortes de comportements du conducteur humain doivent être pris en compte si l'on veut passer à une conduite autonome. Concrètement, un certain nombre de critères du jeu de données ont été mis en place pour entraîner et tester correctement les modèles.

- **Taille du jeu de données**

L'apprentissage profond est une technique d'apprentissage automatique, où une quantité suffisante de données doit être présente. L'entraînement sur une quantité suffisante de données diverses empêche le sur-apprentissage du modèle. Cela implique que le modèle se généralise mieux, en apprenant les tendances importantes réelles dans les données observées. Pour cette raison, l'ensemble de données doit être de taille importante pour bien former le modèle.

- **Type de données**

De nombreux jeux de données de véhicules autonomes se concentrent sur la perception, fournissant des ensembles complets d'informations visuelles concernant la scène de conduite. Pour la prédiction des intentions, l'ensemble de données doit contenir des tracklets, c'est-à-dire les états suivis des véhicules dans la scène au fil du temps. Ceux-ci peuvent être donnés en coordonnées GPS globales, dans un référentiel local ou par rapport à un l'égo-véhicule. De plus, les véhicules sur les lieux devraient être suivis pendant une durée suffisante. La prédiction de trajectoire est une tâche d'apprentissage semi-supervisée, ce qui signifie que les données n'ont pas besoin d'être étiquetées. Au contraire, les trajectoires enregistrées peuvent être découpées en une partie qui est utilisée comme trajectoire observée et une partie qui est la trajectoire future. Cela nécessite une séquence suffisamment longue pour fournir à la fois l'entrée et la réalité terrain.

- **Informations cartographiques**

Une carte sémantique de la scène doit être construite. Cela signifie que les informations cartographiques de la zone où les données sont enregistrées soient disponibles. En fournissant directement des fonctionnalités de carte, on obtient une carte détaillée qui peut être utilisée pour créer une carte sémantique. Les cartes open source telle que Open Street Maps ne fournissent pas d'informations cartographiques suffisamment précises pour extraire des entités basées sur des géolocalisations.

- **Isolement de l'environnement**

Notre travail se concentre sur un des défis de la prédiction. Le plan vise à améliorer la prédiction de intentions en incluant l'interaction et la géométrie de la route dans les prévisions. Pour évaluer correctement si les prédictions sont réellement améliorées, le jeu de données doit se qualifier sur deux aspects. Premièrement, il devrait être un jeu de données suffisamment difficile en ce qui concerne les aspects géométriques interactifs et routiers. Pour l'interaction, le trafic doit être suffisamment dense pour obtenir un fort niveau d'interaction dans le comportement de conduite. Pour savoir si les modèles peuvent obtenir de bonnes performances de prédiction quelle que soit la scène de la route, plusieurs scénarios avec des environnements différents doivent être disponibles. Deuxièmement, il est souhaitable de limiter les influences de la scène qui sortent du cadre de ce travail. Cela inclut les feux de circulation, les piétons, les obstacles sur la route et plus encore. En isolant un environnement de conduite qui ne contient que les facteurs modélisés, les effets des méthodes proposées peuvent être mieux évalués.



### 1.6.1 Jeux de données de la vue à vol d'oiseau (Datasets from Bird's Eye View (BEV))

Les véhicules autonomes s'appuient généralement sur des cartes à vol d'oiseau très détaillées de leur environnement, qui capturent à la fois des éléments statiques de la scène tels que le marquage de la route ainsi que des éléments dynamiques tels que les voitures et les piétons. Plutôt que de capturer le monde 3D complet dans son intégralité, une solution populaire consiste à représenter le monde sous la forme d'une carte à vol d'oiseau (BEV), qui fournit un moyen compact de capturer la configuration spatiale de la scène. Ces cartes sont pratiques; elles sont simples à visualiser, traiter, et exploiter dans de nombreux scénarios. Les informations essentielles pour la circulation sont largement liées au plan du sol.

La construction de cartes à vol d'oiseau est cependant actuellement un pipeline de traitement complexe à plusieurs étapes, impliquant un processus de multiples tâches de vision par ordinateur : estimation du plan du sol, segmentation de la route, détection de voie, détection d'objet 3D, et bien d'autres. Intuitivement, toutes ces tâches sont liées : connaître le marquage de la route devrait nous informer de l'endroit sur l'image où nous devons chercher les voitures; de même, une voiture sortant de derrière un bâtiment peut indiquer la présence d'une route latérale cachée au-delà. Dans cette partie, on cite quelques jeux de données de la vue à vol d'oiseau utilisés pour la prédiction des intentions.

#### **Le jeu de données « Next Generation Simulation (NGSIM) »**

Le jeu de données NGSIM [FHWA, 2008] est le jeu de données de mouvement de véhicule le plus populaire dans les communautés de recherche liées au comportement des véhicules. NGSIM est un projet financé par la Federal Highway Administration (FHWA) des États-Unis. Les données brutes ont été collectées par 8 caméras numériques synchronisées montées sur le toit de bâtiments adjacents aux autoroutes US-101 et I-80, dont 600 mètres de zone d'étude (Figure 1.17). Des techniques de suivi visuel et traitement d'images ont été utilisées pour extraire les trajectoires des véhicules depuis des données d'image à une fréquence de 10 Hz. Les données sont traitées automatiquement pour extraire des informations détaillées sur les trajectoires des véhicules comme les positions, les vitesses, les accélérations, numéro de voies ... etc. Les données traitées sont disponibles sous forme de représentations vectorielles des scènes suivies.

#### **Le jeu de données « HighD »**

Le jeu de données HighD [Krajewski et al., 2018] est un nouveau jeu de données des trajectoires réelles des véhicules enregistrées sur les autoroutes allemandes. À l'aide d'un drone (figure 1.18), les limites typiques des méthodes de collecte de données de trafic établies telles que les occlusions sont surmontées par la perspective aérienne. Le trafic a été enregistré à six endroits différents et comprend plus de 110 500 véhicules. La trajectoire de chaque véhicule, y compris le type, la taille et les manœuvres du véhicule, sont automatiquement extraites. En utilisant des algorithmes de vision par ordinateur de pointe, l'erreur de positionnement est généralement inférieure à dix centimètres. Bien que l'ensemble de données ait été créé pour la validation de la sécurité des véhicules hautement automatisés, il convient également à de nombreuses autres tâches telles que l'analyse des modèles de trafic ou le paramétrage des modèles de conducteur.



FIGURE 1.17 – Une caméra montée sur le toit d'un bâtiment qui couvre une section de l'autoroute I-80 enregistre les données des trajectoires des véhicules. [FHWA, 2008]

### 1.6.2 Jeux de données des capteurs embarqués

En plus des jeux de données de la vue à vol d'oiseau, deux types de données basées sur des capteurs embarqués sont également accessibles au public. L'un inclut les données de circulation des usagers de la route adjacents à partir d'un véhicule équipé par des Lidars, Radars et Caméras embarqués. L'autre ne contient que les données de circulation de nombreux véhicules de collecte de données provenant du GPS embarqué et des capteurs odométriques. Il existe deux avantages majeurs pour les jeux de données provenant de capteurs embarqués : Une variété de scénarios de conduite avec une durée d'enregistrement des données relativement longue sont généralement inclus dans ces jeux de données, telles que la conduite urbaine aux intersections signalées / non signalées et la conduite sur autoroute avec rampe d'insertion, etc. Les occlusions des Lidars, radars et des caméras sont enregistrées de sorte que les occlusions réelles du point de vue de l'égo-véhicule puissent être partiellement récupérées.

L'interaction entre les usagers de la route est un problème majeur lors de l'utilisation des jeux de données provenant des capteurs embarqués. Pour les jeux de données de circulation avec des flottes basées sur le GPS, il est difficile de déterminer si les véhicules présentent dans un environnement "interactif" interagissaient réellement les uns avec les autres puisqu'il n'y a pas d'enregistrement de mouvement des autres véhicules environnants sans dispositifs GPS installés. Pour les jeux de données de circulation construits à partir de Lidar, Radar et de Caméras embarqués, il est difficile de garantir que tous les objets environnants ayant un impact sur le comportement des autres véhicules sont inclus dans le jeu de données lors de la prédiction des intentions. Par conséquent, les interactions complètes sont relativement rares dans ce type de jeux de données. Si les capteurs ne peuvent pas couvrir tout le champ de vision, il sera même impossible de garantir l'exhaustivité des informations pour les entités environnantes du véhicule de collecte de données. On cite quelques jeux de données collectées à partir des capteurs embarqués :

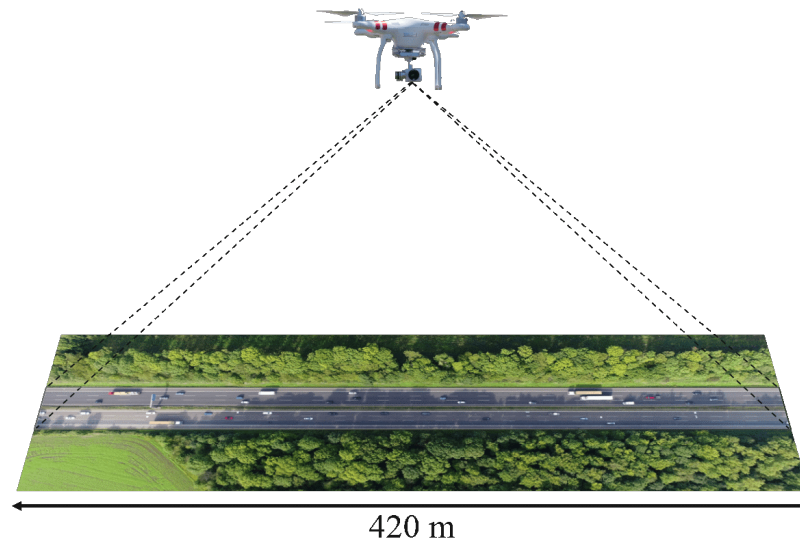


FIGURE 1.18 – Le dispositif d’enregistrement comprend un drone qui vole à côté des autoroutes allemandes et capture le trafic à vue d’oiseau sur un tronçon de route d’environ 420 m de long.

#### Le jeu de données « Argoverse »

Le jeu de données Argoverse [Chang et al., 2019] est conçu pour développer les tâches de perception des véhicules autonomes, y compris le suivi 3D et la prédiction de mouvement. Argoverse comprend des données de capteurs collectées par une flotte de véhicules autonomes à Pittsburgh et à Miami ainsi que des annotations de suivi 3D, 300k de trajectoires de véhicules intéressantes extraites et de riches cartes sémantiques. Les données du capteur se composent d’images à 360° de 7 caméras avec des champs de vision qui se chevauchent, de l’imagerie stéréo orientée vers l’avant, de nuages de points 3D provenant de LiDAR à longue portée et d’une pose à 6 ddl. Les 290 km de voies cartographiées contiennent de riches métadonnées géométriques et sémantiques qui ne sont actuellement disponibles dans aucun jeu de données public. Toutes les données sont publiées sous une licence Creative Commons sur le site officiel de Argoverse.

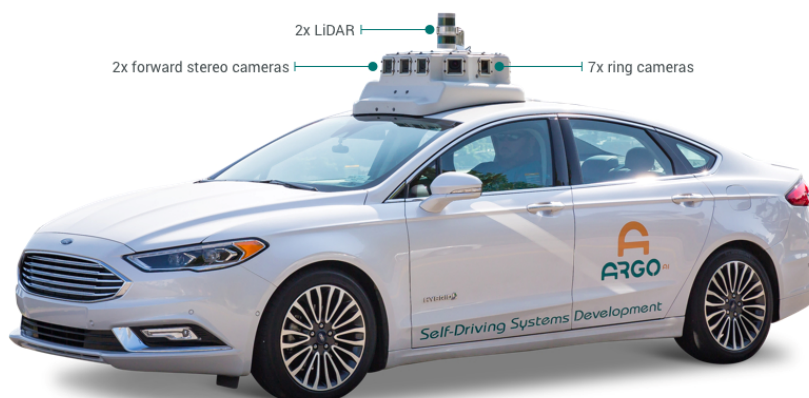


FIGURE 1.19 – Le véhicule Agro équipé par deux LiDAR, sept caméras en anneau et deux caméras stéréo frontales [Argoverse, 2021].

#### Le jeu de données « Waymo »

Le jeu de données Waymo [Sun et al., 2020] est composé de données de capteur haute

résolution collectées par les voitures autonomes Waymo (Figure 1.20) dans une grande variété de conditions. Ce jeu de données est disponible publiquement pour aider la communauté des chercheurs à faire des progrès dans la technologie de conduite autonome. Il se compose de 1150 scènes d'une durée de 20 secondes chacune, constituées de données LiDAR et de caméras de haute qualité bien synchronisées et calibrées capturées dans une gamme de zones géographiques urbaines et suburbaines. Ces données contiennent environ 12 millions d'annotations de boîtes LiDAR et environ 12 millions d'annotations de boîtes de caméras, ce qui donne environ 113 000 d'objets détectés avec le LiDAR et environ 250 000 objets détectés avec des images de caméras. Toutes les annotations ont été créées puis révisées par des étiqueteurs formés à l'aide d'outils d'étiquetage de niveau production.



FIGURE 1.20 – La nouvelle voiture autonome de Waymo, équipée par un système de perception à trois LiDAR, des Radars et des caméras de conduite autonome [Waymo, 2021]

### **Le jeu de donnée « MOOVE» (MONitoring Outillé pour le Véhicule dans son Environnement)**

Le projet MOOVE vise à collecter une grande quantité de données liées à la conduite en utilisant des véhicules non automatisés. Les technologies de détection implémentées sur les véhicules MOOVE ont été spécifiquement choisies pour avoir la perception de l'environnement la plus proche à la réalité. Ces technologies sont très proches de celles utilisées sur les futurs véhicules autonomes. Les véhicules MOOVE (Figure 1.21) sont équipés d'une grande capacité de perception similaires à celles des véhicules autonomes, avec caméras, radars et lidars, mais elles restent conduites par un conducteur humain. Leur vocation : rouler dans tous les temps et en tous lieux pour enregistrer le maximum de données sur les différentes situations routières auxquelles se trouve confronté un véhicule, en particulier pour identifier les situations de risque pour le véhicule autonome. Depuis 2016, elles ont à leur actif un million de kilomètres, parcourus dans 17 pays européens en quelques 15 000 heures de conduite, ayant généré l'enregistrement de 250 téraoctets de données. L'analyse de ces informations doit servir aux constructeurs automobiles, pour spécifier et valider les lois de commande du véhicule autonome.

La méthodologie MOOVE repose sur les étapes suivantes :

**L'acquisition de données :** Les données collectées sur chacun des véhicules MOOVE sont synchronisées, stockées sur un stockage SSD puis importées sur le Datacenter pour effectuer des analyses. Les résultats peuvent être utilisés par tous les partenaires du projet MOOVE. Tous les véhicules MOOVE sont équipés de :

- Radars à courte portée 360 °
- Radar longue portée avant



FIGURE 1.21 – La flotte des véhicules (Citroën Picasso C4) utilisé pour la collection des données MOOVE (VEDECOM).

- LIDAR avant et arrière
- Caméra intelligente avant
- Récepteurs du système de positionnement par satellites (GNSS)
- Unité de mesure inertielle (IMU).

**Le traitement des données :** Les données sont analysées pour extraire les informations suivantes :

- Les objets dynamiques et statiques entourant le véhicule MOOVE
- L'infrastructure routière
- Les conditions environnementales (y compris météorologiques).

## 1.7 Conclusion

La prédiction des intentions des véhicules est un défi technologique complexe, en raison des nombreux facteurs qui influencent le comportement de conduite des usagers de la route. Au cours de ce chapitre, nous avons introduit les véhicules intelligents puis les différents défis de la conduite autonome sont présentés. Ensuite, un aperçu est donné sur l'importance de la prédiction des intentions des véhicules en conduite autonome. Ce premier chapitre consiste tout d'abord en un recensement des travaux réalisés, l'objectif était d'identifier les principaux modèles et méthodes de classification des manœuvres et la prédiction de trajectoire des véhicules. À partir de l'état de l'art réalisé, nous avons pu identifier les différentes pistes d'amélioration pour le positionnement de notre recherche qui sera dédié à la méthode de machine, ces approches feront l'objectif du chapitre suivant.

# Éléments sur la théorie de l'apprentissage

## Sommaire

---

<b>1.1 Introduction à la conduite autonome</b> . . . . .	<b>6</b>
<b>1.2 Description de la problématique</b> . . . . .	<b>11</b>
<b>1.3 État de l'art sur les méthodes de prédiction des intentions des usagers de la route</b> . . . . .	<b>13</b>
1.3.1 Prédiction du comportement pour les applications de conduite autonome . . . . .	14
1.3.2 Prédiction de trajectoire . . . . .	17
1.3.3 La prédiction multi-modèle . . . . .	18
<b>1.4 Taxonomie des modèles de prédiction des intentions</b> . . . . .	<b>19</b>
1.4.1 Modèles basés sur la physique . . . . .	20
1.4.2 Modèles basés sur les manœuvres . . . . .	22
1.4.3 Modèles basés sur les interactions entre les véhicules . . . . .	25
1.4.4 Modèles basés sur les méthodes d'apprentissage avancées . . . . .	26
<b>1.5 Travaux de prédiction des intentions réalisés à VEDECOM</b> . . . . .	<b>26</b>
1.5.1 Reconstruction de trajectoire d'un véhicule cible . . . . .	27
1.5.2 Prédiction de la trajectoire future . . . . .	30
1.5.3 Classification des manœuvres en cours . . . . .	30
<b>1.6 Jeux de donnée véhiculaire</b> . . . . .	<b>34</b>
1.6.1 Jeux de données de la vue à vol d'oiseau (Datasets from Bird's Eye View (BEV)) . . . . .	36
1.6.2 Jeux de données des capteurs embarqués . . . . .	37
<b>1.7 Conclusion</b> . . . . .	<b>40</b>

---

## 2.1 Introduction

Le but de ce chapitre est de fournir au lecteur une vue d'ensemble des notions fondamentales intervenant lors de la mise en œuvre d'un système de prédiction des intentions à base d'apprentissage automatique. Dans un premier temps, nous décrivons l'apprentissage supervisé tout en détaillant quelques méthodes utilisées dans le cadre de cette thèse, ainsi que les deux types de problèmes traités par ce type d'apprentissage à savoir la classification et la régression. Par la suite, deux autres catégories d'apprentissage seront décrites : l'apprentissage non supervisé et l'apprentissage par renforcement.

## 2.2 Généralité sur l'apprentissage

Dans un cadre général, un apprentissage est l'acquisition de nouveaux savoirs (souvent sous forme de connaissances), c'est-à-dire le processus d'acquisition de compétences, par l'expérience ou l'enseignement [Mitchell, 1997]. L'apprentissage est un terme utilisé dans plusieurs domaines (philosophie, linguistique, psychologie, biologie, . . . etc), mais chacun a sa propre définition de cette notion. Comme notre travail s'inscrit dans le domaine des sciences de l'ingénieur (informatique, automatique, robotique) et également dans le domaine de l'Intelligence Artificielle nous parlerons d'apprentissage automatique plutôt que d'apprentissage tout court.

Au cours des deux dernières décennies, l'apprentissage automatique est devenu l'un des piliers de la technologie. Avec la croissance des quantités de données disponibles créées, stockées, manipulées, il y a de bonnes raisons de croire que l'analyse automatique des données deviendra encore omniprésente comme un ingrédient nécessaire pour le progrès technologique. Les domaines d'applications sont divers et multiples. Nous en citerons ici quelques-uns en biologie [Cho, Won, 2003; Lapedes et al., 1988; Bhardwaj et al., 2005], en finance [Atiya, 2001; Zhang et al., 1999] et dans l'industrie automobile [Mitrović, 2004; Torkkola et al., 2004]. Dans ce qui suit, nous allons présenter les deux approches de l'apprentissage statistique, à savoir l'apprentissage supervisé et l'apprentissage non supervisé.

## 2.3 Apprentissage supervisé

Dans la littérature, l'apprentissage supervisé est défini de plusieurs façons. Parmi ces définitions, on trouve celle donnée par [Mitchell, 1997] qui précise que l'apprentissage supervisé peut être caractérisé par trois éléments : i) la base d'apprentissage, ii) la tâche, iii) la mesure de performance.

L'apprentissage supervisé fait intervenir les éléments suivants [Bishop, 2006] :

- Des variables d'entrées ( $\mathbf{X} \in \mathcal{X}$ ) appelées covariables appartenant à l'ensemble  $\mathbb{R}$  des réels de dimension  $p$ , noté  $\mathbb{R}^p$ .
- Des variables de sorties ( $\mathbf{Y} \in \mathcal{Y}$ ) appelées variables d'intérêts appartenant à l'ensemble des réels  $\mathbb{R}$ . Les variables d'intérêts peuvent être continues (problème de régression) ou discrètes, variables catégorielles (problème de classification).
- Une fonction de prédiction notée  $f_{\theta}(x)$  à paramètre  $\theta$  appartenant à l'ensemble des paramètres  $\Theta$ . Cette fonction a pour rôle d'associer des variables de l'espace d'entrée  $\mathbf{X}$  avec des variables de l'espace de sortie  $\mathbf{Y}$ .

- Un ensemble d'apprentissage  $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$  constitué par l'ensemble de réalisations du couple  $(\mathbf{X}, \mathbf{Y}) = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), (\mathbf{x}_3, y_3), \dots, (\mathbf{x}_N, y_N)\}$  supposées indépendantes et identiquement distribuées (i.i.d) issues de la loi de probabilité jointe inconnue  $p(\mathbf{x}; y) = p(\mathbf{x})p(y | \mathbf{x})$ .
- Un ensemble de test constitué par l'ensemble de nouvelles réalisations de  $\mathbf{X}$ , et l'objectif est de prédire la variable  $y = f_\theta(\mathbf{x})$ .
- Une fonction coût de l'erreur notée  $L(f_\theta(\mathbf{x}), y)$  permettant de traduire l'erreur d'attribution de la variable d'entrée à sa variable de sortie. La forme de cette fonction dépend essentiellement du problème étudié.

L'apprentissage supervisé est un cadre de recherche de la relation entre les attributs d'entrées et de sorties à partir d'une base  $\mathcal{S} \subset (\mathcal{X} \times \mathcal{Y})$ ; dite base d'apprentissage définie par :

$$\mathcal{S} \triangleq \{(\mathbf{x}_i, y_i)\}_{i=1}^N \quad (2.1)$$

Dans l'apprentissage supervisé, l'objectif est d'estimer la fonction de prédiction  $f_\theta(\mathbf{x})$  qui minimise le coût de l'erreur (ou fonction de perte<sup>1</sup>) pour les futures réalisations du couple  $(\mathbf{X}, \mathbf{Y})$  ainsi que le risque total moyen défini par :

$$\mathbb{E}[L(f_\theta(\mathbf{x}), y)] = \int_{\mathcal{X}} \int_{\mathcal{Y}} L(f_\theta(\mathbf{x}), y) p(\mathbf{x}, y) d\mathbf{x} dy \quad (2.2)$$

Le calcul du risque moyen donné par l'équation 2.2 suppose que la loi jointe du couple  $(\mathbf{X}, \mathbf{Y})$  est connue. Malheureusement, en pratique cette loi n'est pas connue et la seule connaissance qu'on a sur le problème est relative aux valeurs de ces réalisations. Cependant, le risque empirique est calculé en remplaçant l'espérance par une moyenne empirique. Formellement, le risque empirique est défini comme suit :

$$R_{emp}(f_\theta(\mathbf{x})) = \frac{1}{N} \sum_{i=1}^N [L(f_\theta(\mathbf{x}_i), y_i)] \quad (2.3)$$

Comme indiqué précédemment, l'apprentissage supervisé traite souvent deux types de problèmes liés essentiellement à la nature de la sortie  $\mathbf{Y}$ , à savoir le problème de classification et le problème de régression. Selon cette sortie, on peut également classifier les approches en deux catégories à savoir les approches génératives et les approches discriminatives.

### 2.3.1 Le problème de la classification

On parle de problème de classification quand la variable d'intérêt  $y$  appartient à un ensemble fini  $\mathcal{Y} = \{c_1, c_2, c_3, \dots, c_k\}$ . En d'autres termes le but de la classification est de prédire l'appartenance d'un objet  $x$  décrit par le vecteur d'attributs à une classe donnée  $\{c_i\}_{i=1}^k$ . Dans ce cas de figure, la fonction de coût  $L$  prend souvent la forme suivante :

$$L(f_\theta(\mathbf{x}), y) \triangleq \begin{cases} 0 & \text{si } f_\theta(\mathbf{x}) = y, \\ 1 & \text{sinon.} \end{cases} \quad (2.4)$$

1. En anglais "loss" function



### 2.3.2 Le problème de régression

Dans le problème de régression,  $y$  est une variable continue qui appartient à l'ensemble des réels  $\mathbb{R}$ . Le but de la régression est d'associer une valeur numérique à une réalisation de la variable d'entrée  $x$ . La fonction de coût  $L$ , la plus utilisée dans la littérature est de la forme suivante :

$$L(f_{\theta}(x), y) = (f_{\theta}(x) - y)^2 \quad (2.5)$$

### 2.3.3 Approches génératives/discriminatives

Dans l'apprentissage statistique, on distingue deux groupes d'approches à savoir les approches génératives et les approches discriminatives. Dans l'approche générative, l'algorithme apprend un modèle à partir de la distribution conjointe  $p(x, y)$ . Ce modèle consiste à modéliser pour tout  $Y = \{c_1, c_2, c_3, \dots, c_k\}$ , les densités conditionnelles  $p(x | y)$  ainsi que la probabilité a priori  $p(y)$ . Les probabilités a posteriori de chaque classe sont ensuite calculées en utilisant le théorème de Bayes [Bayes, 1958] :

$$p(x | y) = \frac{p(x)p(y | x)}{\sum_{y'} p(y') p(y | y')} \quad (2.6)$$

L'approche discriminative modélise directement la distribution conditionnelle  $p(y | x)$ . A partir de cette distribution conditionnelle, nous pouvons faire des prédictions de  $y$ , pour toute nouvelle valeur de  $x$ , en utilisant la règle du Maximum A Posteriori (MAP) :

$$\hat{y} = \arg \max_{y \in Y} p(y | x) \quad (2.7)$$

## 2.4 Exemples d'approches d'apprentissage supervisé

Dans cette section, nous présentons quelques approches d'apprentissage supervisé adopté dans le cadre de cette thèse. Dans un premier temps, nous détaillons quelques algorithmes de classification et de régression utilisés pour la prédiction des intentions des véhicules. Ensuite, nous abordons les techniques de validation et d'évaluation métrique.

### 2.4.1 Machine à Vecteurs de Support

Les Machines à Vecteurs de Support ou les SVM<sup>2</sup> ont été introduites par V.Vapnik en 1995 dans son ouvrage "The nature of statistical learning theory" [Vapnik, 2013]. Les SVM représentent depuis une dizaine d'années une approche largement reconnue dans la communauté de la classification supervisée pour ses excellentes performances globales se hissant généralement parmi les meilleurs. Depuis, différentes variantes de l'algorithme SVM ont été proposées par plusieurs chercheurs et appliquées à divers domaines [Maaoui et al., 2016; Mandalia, Salvucci, 2005].

Cette approche est basée sur la théorie de la Minimisation du Risque Empirique (MRE) et celle du Risque Structurel (MRS). Dans le cas où les données sont séparables, l'idée principale est de trouver un hyperplan (séparateur)  $f(x) = w^T x + b$  (où  $w$  est le vecteur des poids) qui maximise la distance entre les exemples positifs  $y_i = +1$  et négatifs  $y_i = -1$

2. En anglais "Support Vectors Machines", certains auteurs utilisent l'expression "Séparateurs à Vaste Marge". Nous préférons "Machines à Vecteurs de Support" qui nous semble plus adéquate.

et dont la séparation (ou marge : distance séparant la frontière du plus proche exemple) est aussi grande que possible (la marge entre les deux classes vaut  $\frac{2}{\|\mathbf{w}\|}$ ). Comme le montre la figure 2.1a, on a un hyperplan optimal avec une grande marge, ce qui nous permet de bien classer un nouvel exemple malgré le fait qu'il soit localisé dans la marge. Cependant, sur la figure 2.1b malgré le fait que l'on ait trouvé un hyperplan optimal, l'exemple se voit mal classé, du fait que la marge est petite. Maximiser la marge revient à minimiser  $\|\mathbf{w}\|$  ce qui équivaut à résoudre un problème d'optimisation quadratique sous contraintes.

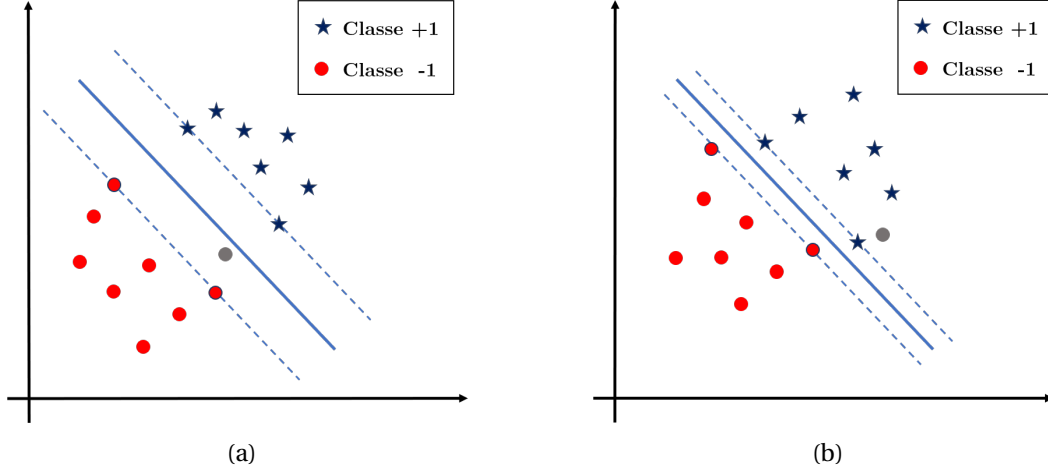


FIGURE 2.1 – Illustration de l'impact de la maximisation de la marge sur la capacité de généralisation. Le point gris est un élément de l'ensemble de test.

Dans le cas des données non linéairement séparables (pas d'hyperplan séparant tous les exemples), on introduit des variables  $\xi_i$  de relaxation "ressort" pour assouplir les contraintes et contrôler à travers de la variable  $C$ . Sous sa forme primaire et en intégrant ces variables de relaxation, la fonctionnelle à optimiser s'écrit :

$$\begin{cases} \min_{\mathbf{w}, b} & \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_i \xi_i \\ & y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i, \forall i \end{cases} \quad (2.8)$$

Ce problème peut se réécrire sous forme duale en introduisant des multiplicateurs de Lagrange pour chaque contrainte.

$$\begin{cases} \max_{\alpha_i} & \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) \\ & 0 \leq \alpha_i \leq C, \forall i \\ & \sum_{i=1}^N \alpha_i y_i = 0 \end{cases} \quad (2.9)$$

La fonction de décision associée s'exprime par :  $f(\mathbf{x}) = \text{sign}(\sum_{i=1}^N \alpha_i^* y_i (\mathbf{x}_i \cdot \mathbf{x}) + b)$  à partir des paramètres  $\alpha^*$  et  $b$  solutions du problème précédent.

Vapnik propose de projeter les points dans un nouvel espace  $\mathcal{H}$  de Hilbert muni d'un produit scalaire  $\langle \cdot, \cdot \rangle$  via une fonction  $\varphi(\cdot)$ . En pratique, cette transformation est implicite à travers le calcul du noyau de Mercer au travers de produits scalaires tels que  $K(\mathbf{x}_1, \mathbf{x}_2) = \langle \varphi(\mathbf{x}_1), \varphi(\mathbf{x}_2) \rangle$  [Vapnik, 2013]. Plusieurs noyaux peuvent être envisagés à savoir : noyau linéaire, noyau polynomial, noyau RBF (Radial Basis Function) . . . .

On en déduit le problème dual qui a la même forme que dans le cas séparable à savoir :

$$\begin{cases} \max_{\alpha_i} & \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \\ & 0 \leq \alpha_i \leq C, \forall i \\ & \sum_{i=1}^N \alpha_i y_i = 0 \end{cases} \quad (2.10)$$

et la frontière de décision s'exprime sous la forme :  $f(\mathbf{x}) = \text{sign}(\sum_{i=1}^N \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b)$ .

Dans notre cas, plusieurs noyaux seront utilisés pour la classification des situations de conduite.

La forme générique de chaque noyau est présenté par [Vapnik, 2013] :

Le noyau linéaire :

$$K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \cdot \mathbf{x}_j \quad (2.11)$$

Le noyau polynomial :

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \cdot \mathbf{x}_j + 1)^d \quad (2.12)$$

Le noyau sigmoïde :

$$K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\mathbf{x}_i^T \cdot \mathbf{x}_j + c) \quad (2.13)$$

Le noyau RBF

$$\begin{aligned} K(\mathbf{x}, \mathbf{z}) &= \exp\left(-\frac{\|\mathbf{x} - \mathbf{z}\|^2}{2\sigma^2}\right) \\ &= \exp(-\gamma\|\mathbf{x} - \mathbf{z}\|^2) \end{aligned} \quad (2.14)$$

avec  $\gamma = \frac{1}{2\sigma^2}$ , où  $\|\cdot\|$  désigne la norme Euclidienne. La fonction de projection induite par ce type de noyau permet de projeter un exemple sur une fonction gaussienne représentant la similarité de l'exemple avec tous les points de  $\mathcal{X}$ .

Le paramètre  $\sigma$  permet de régler la largeur de la gaussienne. En prenant un  $\sigma$  grand, la similarité d'un exemple par rapport à ceux qui l'entourent sera assez élevée, tandis qu'en prenant un  $\sigma$  qui tend vers 0, l'exemple ne sera similaire à aucun autre. En resserrant fortement la gaussienne, le classifieur peut apprendre n'importe quel ensemble d'apprentissage sans commettre d'erreur. En fait,  $\sigma$  peut être considéré comme un autre paramètre permettant de contrôler la capacité du classifieur.

Ainsi, les deux hyper-paramètres à ajuster sont la constante contrôlant le compromis entre une erreur empirique et la marge (C) et le paramètre  $\gamma$ .

## 2.4.2 Contexte des réseaux de neurones

Les travaux de cette thèse font appel aux réseaux de neurones (ANN<sup>3</sup>) et aux réseaux de neurones récurrents (RNN<sup>4</sup>). Pour comprendre les RNN, il faut d'abord décrire les réseaux de neurones standards. Cette section décrira les réseaux neuronaux simples, les réseaux neuronaux profonds, les réseaux neuronaux récurrents, et abordera l'optimisation par lots et la sélection d'hyperparamètres. Les réseaux de neurones sont une technique basée sur les données et sont devenus populaires ces dernières années, en partie en raison de l'augmentation des données disponibles, ainsi que des optimisations du matériel informatique. Ces réseaux ont été utilisés pour résoudre une pléthore de problèmes, tels que la classification d'images, la génération de texte et la détection d'objets. La composante fondamentale d'un réseau profond est un perceptron unique, qui est expliqué dans la section suivante.

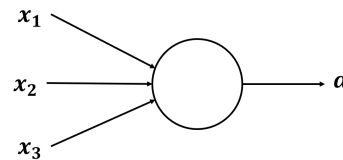


FIGURE 2.2 – Un perceptron simple avec trois sorties

### 2.4.3 réseaux de neurones simple

Un réseau de neurones est constitué de plusieurs neurones artificiels (appelés perceptrons) reliés en tant que nœuds dans un graphe (figure 2.2). Chacun de ces nœuds aura une activation de sortie  $\alpha$  selon la fonction :

$$\alpha = f \left( \sum_{i=1}^N \mathbf{W}_i x_i + \mathbf{b} \right) \quad (2.15)$$

où  $N$  est le nombre d'entrées,  $\mathbf{W}$  est un vecteur de poids, et  $\mathbf{b}$  de biais, qui sont des paramètres appris du modèle. Celui-ci est ensuite enveloppé dans une fonction d'activation  $f$  pour augmenter la complexité du modèle car il ajoute de la non-linéarité. Les exemples courants de la fonction d'activation comprennent le tanh, le sigmoïde ou l'unité linéaire rectifiée.

Un seul neurone n'est pas particulièrement utile car il est trop simpliste d'apprendre une fonctionnalité de niveau supérieur. Cependant, ces neurones sont souvent regroupés en couches, et ces couches sont empilées pour former un perceptron multicouche, comme le montre la figure 2.3. Un perceptron multicouche est une forme de base de réseau neuronal profond (DNN<sup>5</sup>) où chaque couche est entièrement dense, c'est-à-dire que tous les nœuds d'une couche sont connectés à chaque nœud de la couche précédente.

Il existe trois types de couches qui sont généralement adressées, à savoir la couche d'entrée, la couche cachée et la couche de sortie. Les couches d'entrée et de sortie s'expliquent d'elles-mêmes, et les couches cachées sont toutes les couches du milieu qui ne sont pas directement exposées aux données d'entrée ou de sortie. Les DNN sont entraînés pour être utilisés comme de grandes fonctions non linéaires, et sont généralement entraînés pour résoudre des problèmes de régression ou de classification. Les problèmes de régression sont ceux où le modèle tente de produire une seule sortie sous la forme d'un nombre à virgule flottante, comme l'estimation de la position d'un véhicule à partir de données historiques. Les problèmes de classification sont ceux où le modèle attribue une seule étiquette aux données d'entrée d'un ensemble donné d'étiquettes, comme la classification de manœuvres.

Ces réseaux sont entraînés grâce à un processus d'optimisation appelé rétro-propagation. Un DNN est d'abord initialisé avec des poids aléatoires et l'inférence est exécutée sur le réseau en entrant un échantillon d'apprentissage. Cela propage ensuite les données via les nœuds jusqu'à ce qu'une solution finale soit produite. Cette solution sera comparée aux données de vérité terrain, qui est l'étiquette correcte pour ces données d'entrée. Cela produit l'erreur du réseau, qui est générée par une fonction de perte. En différenciant à travers chaque couche du réseau, un gradient peut être trouvé, permettant à l'optimiseur de mettre à jour les poids du réseau pour cet exemple d'apprentissage. Cette opération est

3. En anglais Artificial Neural Network  
 4. En anglais Recurrent Neural Network  
 5. En anglais Deep Neural Network

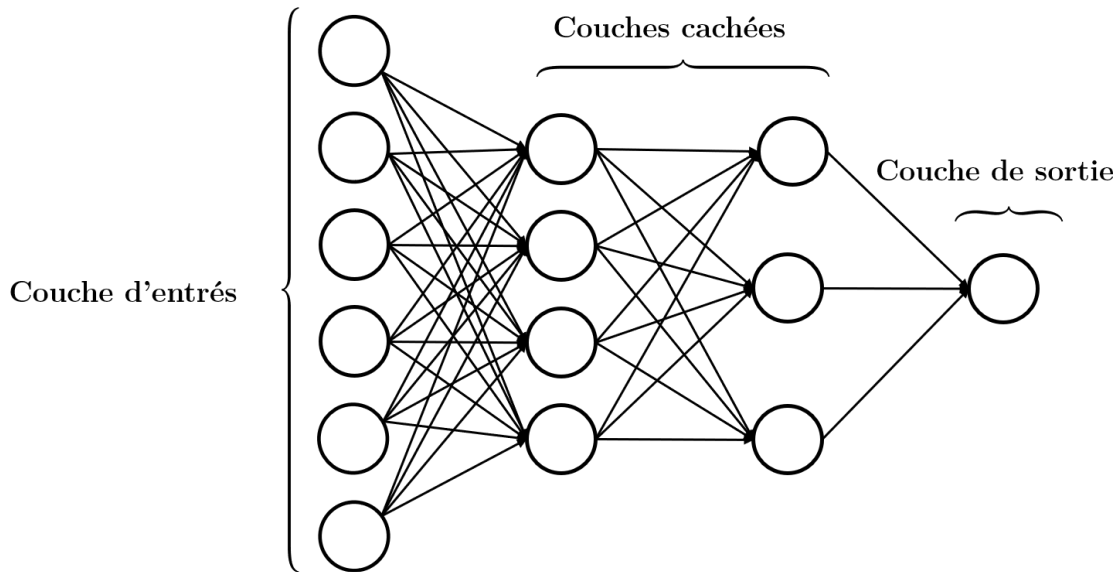


FIGURE 2.3 – Un perceptron de base multicouche. Nombreux neurones sont combinés pour former des couches, et de nombreuses couches sont combinées pour former le réseau final.

répétée pour chaque échantillon de l'ensemble d'apprentissage. Comme avec tous les optimiseurs, cela se fait avec une très petite taille de pas. Si la taille du pas est très grande, les poids peuvent gonfler et le modèle ne convergera pas. Pour accélérer le processus d'apprentissage, plusieurs échantillons peuvent être utilisés comme données de formation à la fois, dans un processus appelé traitement par lots où les poids de mise à jour sont additionnés.

Les hyper-paramètres sont un autre aspect des DNN. De nombreux paramètres sont utilisés à la fois dans la définition du modèle et dans l'apprentissage ; les valeurs de ces paramètres doivent être trouvées. Certains de ces paramètres incluent : le nombre et le type de couches, la largeur de chaque couche, le taux d'apprentissage, l'optimiseur et la taille du lot. Alors que certains paramètres ont peu d'influence sur le résultat final, comme la taille du lot, d'autres doivent être ajustés. Cela se fait principalement par une recherche par grille, où chaque valeur est essayée à son tour, et la performance finale de chaque réseau comparée. Les optimiseurs bayésiens existent pour optimiser les hyper-paramètres, mais dans la pratique, ils sont souvent moins performants par rapport à une recherche par grille [Balaprakash et al., 2018].

#### 2.4.4 réseaux de neurones récurrent

Les réseaux de neurones récurrents (en anglais Recurrent Neural Network (RNN)) sont des modèles "connectistes" [Lipton et al., 2015] conçus pour des tâches d'apprentissage machine séquentielles, c'est-à-dire des problèmes dont les données d'entrée et/ou de sortie sont liées de manière séquentielle. Par exemple dans la classification vidéo [Yue-Hei Ng et al., 2015], la reconnaissance d'action [Du et al., 2015] ou les tâches de traitement du langage naturel comme la traduction automatique [Sutskever et al., 2014]. Les réseaux neuronaux récurrents (RNN) sont capables de transmettre sélectivement des informations sur une fenêtre contextuelle relativement longue, ce qui leur permet d'établir des dépendances temporelles entre les données d'entrée et de sortie. Dans le domaine de la prédiction de trajectoire, les RNN ont été utilisés avec succès pour le suivi d'objets [Milan

et al., 2017] avec des performances comparables aux filtres de Kalman. Cela fait des RNN un choix approprié pour étudier leurs performances pour la prédiction des trajectoires des véhicules.

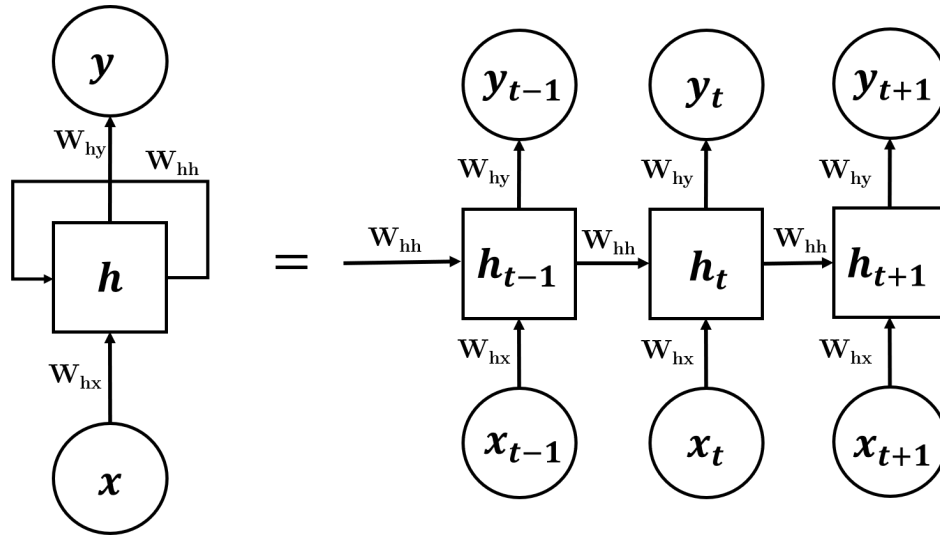


FIGURE 2.4 – Un réseau de neurones récurrent simple.

Le fonctionnement d'un RNN dans le temps peut être vu sur la figure 2.4. La clé du fonctionnement du RNN est l'état caché  $h_t$ , qui est transmis à chaque pas de temps. Cet état caché est utilisé en même temps que l'entrée actuelle pour calculer la sortie. Les équations suivantes régissent le calcul d'un RNN :

$$\begin{aligned} h_t &= f(W_{hx} * x_t + W_{hh} * h_{t-1}) \\ y_t &= W_{yh} * h_t \end{aligned} \quad (2.16)$$

où  $x_t$  est l'entrée,  $h_{t-1}$  est l'état caché du pas de temps précédent,  $h_t$  est l'état caché du pas de temps actuel,  $y_t$  est la sortie,  $f$  est une fonction d'activation,  $(W_{hx}, W_{hh}, W_{yh})$  sont des poids d'apprentissage. Les poids sont entraînés à l'aide d'un algorithme de rétropropagation [Rumelhart et al., 1985]. La rétropropagation fonctionne dans un cadre supervisé pour obtenir la dérivée d'une fonction de perte par rapport à chaque paramètre du réseau, puis ajuste les poids par descente de gradient pour réduire la perte.

Les états cachés d'un réseau peuvent être considérés comme la mémoire du réseau, qui est utilisée pour capturer les informations des étapes temporelles précédentes. À chaque pas de temps, l'état caché est mis à jour pour inclure les nouvelles informations disponibles. Le RNN peut capturer des dépendances temporelles à long terme, théoriquement pour un nombre quelconque de pas de temps [Lipton et al., 2015]. En utilisant cette mémoire, le réseau peut déduire des relations séquentielles entre les entrées sur plusieurs pas de temps.

Un RNN effectue essentiellement la même tâche à chaque étape, c'est-à-dire qu'il produit un résultat souhaité en fonction de l'entrée actuelle et de l'état caché. Cela permet au RNN d'utiliser le même ensemble de paramètres pour tous les pas de temps, comme le montre la figure 2.4. Cela offre de multiples avantages, d'une part, le nombre de paramètres à entraîner est considérablement réduit; d'autre part, un RNN peut fonctionner avec des données de n'importe quelle longueur à partir de n'importe quel pas de temps, car il effectue la même opération à chaque pas de temps.

Il existe de nombreuses dispositions qu'un RNN peut avoir, ce qui peut être vu dans la figure 2.5. Le premier est le plus basique et il représente un RNN typique, avec une entrée

et une sortie. Le type suivant représenté est un réseau un-à-plusieurs, où la sortie est une séquence et l'entrée est un échantillon de données unique. Un exemple de ceci est un générateur de légende d'image où une seule image est servie comme entrée au réseau, et le réseau produit une phrase (une séquence de mots). Cette phrase n'a pas de longueur fixe, le réseau fonctionne jusqu'à ce que le caractère de "fin de ligne" soit produit par le réseau.

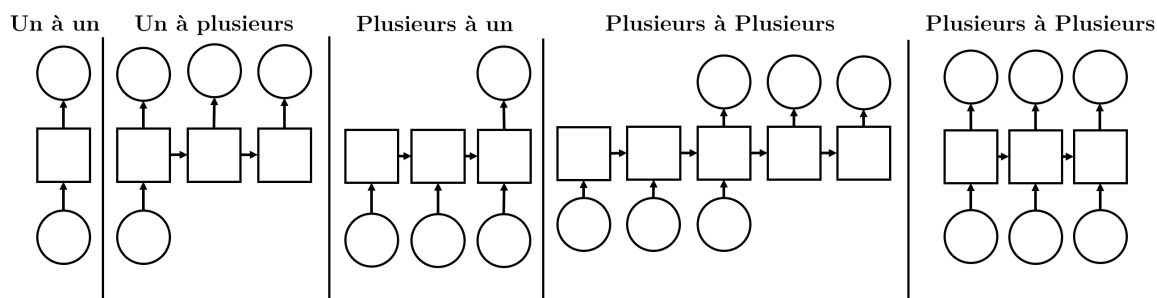


FIGURE 2.5 – Les différents types de réseaux de neurones récurrent

Le troisième réseau illustré est une configuration plusieurs-à-un, ce type est très utilisé pour la classification. Les deux types suivants de RNN sont tous les deux plusieurs à plusieurs. Le premier type, une séquence entière est entrée avant qu'une séquence ne soit sortie, qui est le format du traducteur de langue par exemple. Le dernier exemple est celui où il existe une sortie pour chaque entrée pendant la séquence, comme la classification vidéo qui peut changer à mesure que la vidéo progresse.

### 2.4.5 Cellules de réseaux de neurones récurrents

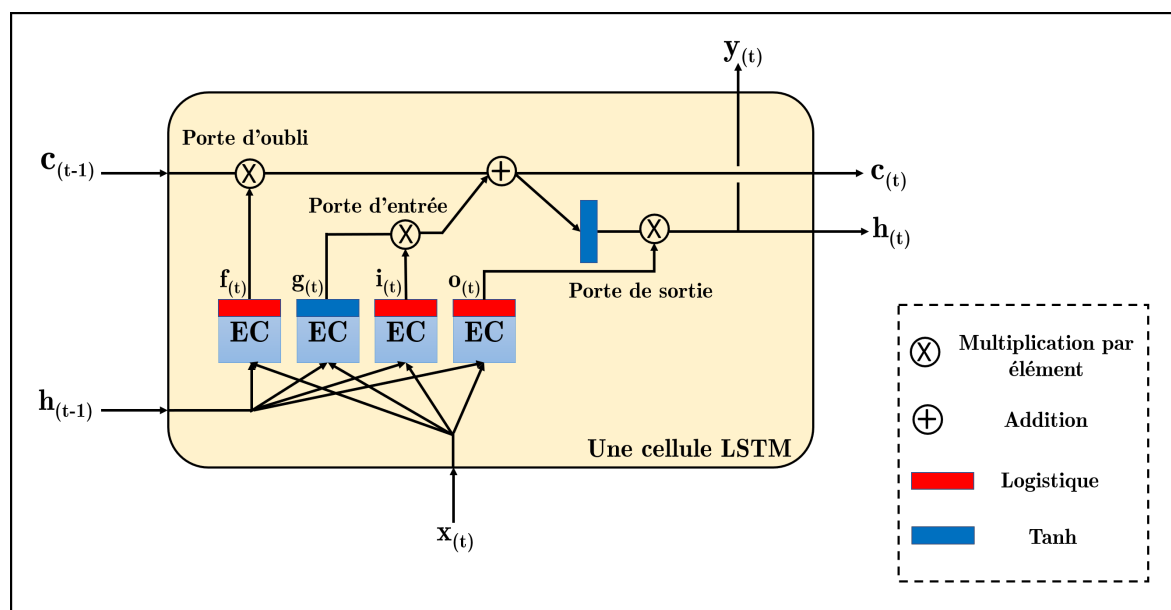


FIGURE 2.6 – Une cellule de mémoire à long/court terme LSTM

L'utilisation de couches denses standard dans une configuration récurrente ne fonctionne pas très bien. Ce problème est similaire à la raison pour laquelle les réseaux très

profonds échouent, ce problème est connu sous le nom de disparition de gradient. Il existe de nombreuses couches que la rétro-propagation doit mettre à jour et ces gradients diminuent pour chaque couche traversée, ce qui fait qu'ils convergent vers zéro assez rapidement. Pour résoudre ce problème, un nouveau type de cellule de réseau doit être utilisé, le plus courant est le module LSTM. Il y a quatre composants dans un LSTM [Hochreiter, Schmidhuber, 1997], la porte d'entrée  $i_t$ , la porte de sortie  $o_t$ , la porte d'oubli  $f_t$  et la cellule  $c_t$ . Celles-ci peuvent être vues dans la figure 2.6. La porte d'entrée contrôle si les données d'entrée sont stockées dans la cellule. La porte de sortie contrôle si les données de la cellule sont transmises à la sortie LSTM  $h_t$ . En conservant les informations dans la cellule, les activations peuvent être maintenues entre les pas de temps et sont influencées par la porte d'entrée et la porte d'oubli. Cela permet au module LSTM de conserver des informations pendant une durée arbitraire, car la dérivation du contenu de la cellule est constante, ce qui permet aux gradients de se propager à travers chaque couche du réseau. Mathématiquement, ils sont également similaires aux connexions sautées utilisées dans les réseaux très profonds. Il existe de nombreux autres types de cellules qui résolvent également le problème du gradient de fuite comme les unités récurrentes à portes [Chung et al., 2014], ou les LSTM modifiés, comme l'hyper-LSTM [Ha et al., 2016], et le LSTM normalisé par lots [Cooijmans et al., 2016].

#### 2.4.5.a Long-Short Term Memory LSTM (Les unités de mémoire à long/court terme)

Les réseaux de mémoire à long/court terme (LSTM) sont une extension des réseaux de neurones récurrents, ce qui facilite la mémorisation des données passées en mémoire (Figure 2.6). Par conséquent, il est bien adapté pour apprendre des expériences importantes qui ont des délais très longs entre les deux séquences. Les unités d'un LSTM sont utilisées comme unités de construction pour les couches d'un RNN, qui est alors souvent appelé réseau LSTM. Les LSTM permettent aux RNN de se souvenir de leurs entrées sur une longue période. Ceci est dû au fait que les LSTM conservent leurs informations dans une mémoire, qui est très similaire à la mémoire d'un ordinateur car le LSTM peut lire, écrire et supprimer des informations de sa mémoire [Chung et al., 2014].

L'idée clé est que le réseau peut apprendre ce qu'il faut stocker à long terme, ce qu'il faut jeter et ce qu'il faut en lire. Lorsque l'état à long terme  $c_{(t-1)}$  parcourt le réseau de gauche à droite, il est possible de voir qu'il passe d'abord par une porte d'oubli, supprimant quelques mémoires, puis il ajoute de nouvelles mémoires par l'opération d'ajout (qui ajoute les mémoires qui ont été sélectionnées par une porte d'entrée). Le résultat  $c_{(t)}$  est envoyé directement, sans aucune autre transformation. Ainsi, à chaque étape, certaines mémoires sont supprimées et d'autres sont ajoutées. De plus, après l'opération d'addition, l'état à long terme est copié et passé par la fonction  $\tanh$ , puis le résultat est filtré par la porte de sortie. Cela produit le résultat à court terme état  $h_{(t)}$  (qui est égal à la sortie de la cellule pour ce pas de temps,  $y_{(t)}$ ). Voyons maintenant d'où viennent les nouveaux mémoires et comment fonctionnent les portes.

Tout d'abord, le vecteur d'entrée actuel  $x_{(t)}$  et l'état à court terme précédent  $h_{(t-1)}$  sont transmis à quatre couches différentes Entièrement Connectées (EC). Elles servent toutes un objectif différent :

- La couche principale est celle qui produit  $g_{(t)}$ . Elle a le rôle habituel d'analyser les entrées actuelles  $x_{(t)}$  et l'état précédent (à court terme)  $h_{(t-1)}$ . Dans une cellule de base, il n'y a rien d'autre que cette couche, et sa sortie va directement vers  $y_{(t)}$  et  $h_{(t)}$ . En revanche, dans une cellule LSTM, la sortie de cette couche ne va pas directement, mais ses parties les plus importantes sont stockées dans l'état à long terme



(et le reste est supprimé).

- Les trois autres couches sont des contrôleurs de porte. Comme ils utilisent la fonction d'activation logistique, leurs sorties vont de 0 à 1. Comme on peut le voir, leurs sorties sont envoyées à des opérations de multiplication par éléments, donc si elles émettent des 0, elles ferment la porte, et si elles émettent des 1, elles l'ouvrent. Plus précisément :
  - La porte d'oubli (contrôlée par  $f(t)$ ) contrôle quelles parties de l'état à long terme doivent être supprimées.
  - La porte d'entrée (contrôlée par  $i(t)$ ) contrôle quelles parties de  $g(t)$  doivent être ajoutées à l'état à long terme.
  - Enfin, la porte de sortie (contrôlée par  $o(t)$ ) contrôle quelles parties de l'état à long terme doivent être lues et sorties à ce stade, à la fois vers  $h(t)$  et vers  $y(t)$ .

En bref, une cellule LSTM peut apprendre à reconnaître un apport important (c'est le rôle de la porte d'entrée), le stocker à long terme, le conserver aussi longtemps que nécessaire (c'est le rôle de la porte de l'oubli), et l'extraire chaque fois que cela est nécessaire. Cela explique pourquoi ces cellules ont étonnamment bien réussi à capturer des modèles à long terme dans les séries temporelles, des trajectoires, des textes longs, des enregistrements audio, et plus encore.

L'équation 2.17 résume comment calculer l'état à long terme de la cellule, son état à court terme et son résultat à chaque pas de temps pour une seule instance (les équations pour un ensemble mini-lots sont très similaires) [Chung et al., 2014].

$$\begin{aligned}
 \mathbf{i}(t) &= \sigma(\mathbf{W}_{xi}^\top \mathbf{x}(t) + \mathbf{W}_{hi}^\top \mathbf{h}(t-1) + \mathbf{b}_i) \\
 \mathbf{f}(t) &= \sigma(\mathbf{W}_{xf}^\top \mathbf{x}(t) + \mathbf{W}_{hf}^\top \mathbf{h}(t-1) + \mathbf{b}_f) \\
 \mathbf{o}(t) &= \sigma(\mathbf{W}_{xo}^\top \mathbf{x}(t) + \mathbf{W}_{ho}^\top \mathbf{h}(t-1) + \mathbf{b}_o) \\
 \mathbf{g}(t) &= \tanh(\mathbf{W}_{xg}^\top \mathbf{x}(t) + \mathbf{W}_{hg}^\top \mathbf{h}(t-1) + \mathbf{b}_g) \\
 \mathbf{c}(t) &= \mathbf{f}(t) \otimes \mathbf{c}(t-1) + \mathbf{i}(t) \otimes \mathbf{g}(t) \\
 \mathbf{y}(t) &= \mathbf{h}(t) = \mathbf{o}(t) \otimes \tanh(\mathbf{c}(t))
 \end{aligned} \tag{2.17}$$

$\mathbf{W}_{xi}, \mathbf{W}_{xf}, \mathbf{W}_{xo}, \mathbf{W}_{xg}$  sont les matrices de poids de chacune des quatre couches pour leur connexion au vecteur d'entrée  $x(t)$ .

$\mathbf{W}_{hi}, \mathbf{W}_{hf}, \mathbf{W}_{ho},$  and  $\mathbf{W}_{hg}$  sont les matrices de poids de chacune des quatre couches pour leur connexion à l'état à court terme précédent  $h(t-1)$ .

$\mathbf{b}_i, \mathbf{b}_f, \mathbf{b}_o,$  et  $\mathbf{b}_g$  sont les termes de biais pour chacune des quatre couches.

#### 2.4.5.b Unité récurrente à porte (Gated Recurrent Unit GRU)

Un autre type de réseau récurrent est le Gated Recurrent Unit (GRU). Il est légèrement différent du LSTM et a un poids moindre. L'architecture LSTM est très efficace, mais aussi assez compliquée. La complexité du système le rend difficile à analyser et coûteux en terme de calcul. Les unités récurrentes à portes sont les unités les plus sophistiquées qui sont mises en œuvre, récemment introduites [Cho et al., 2014] comme alternative au LSTM. Elle se compose de deux portes, une porte de réinitialisation  $r$ , et une porte de mise à jour  $z$ . Intuitivement, la réinitialisation détermine comment combiner la nouvelle entrée avec la mémoire précédente, et la porte de mise à jour fixe le montant de la mémoire antérieure à conserver.

La cellule GRU est une version simplifiée de la cellule LSTM, et elle semble fonctionner tout aussi bien (ce qui explique sa popularité croissante). On résume ci-dessous les principales simplifications [Cho et al., 2014] :

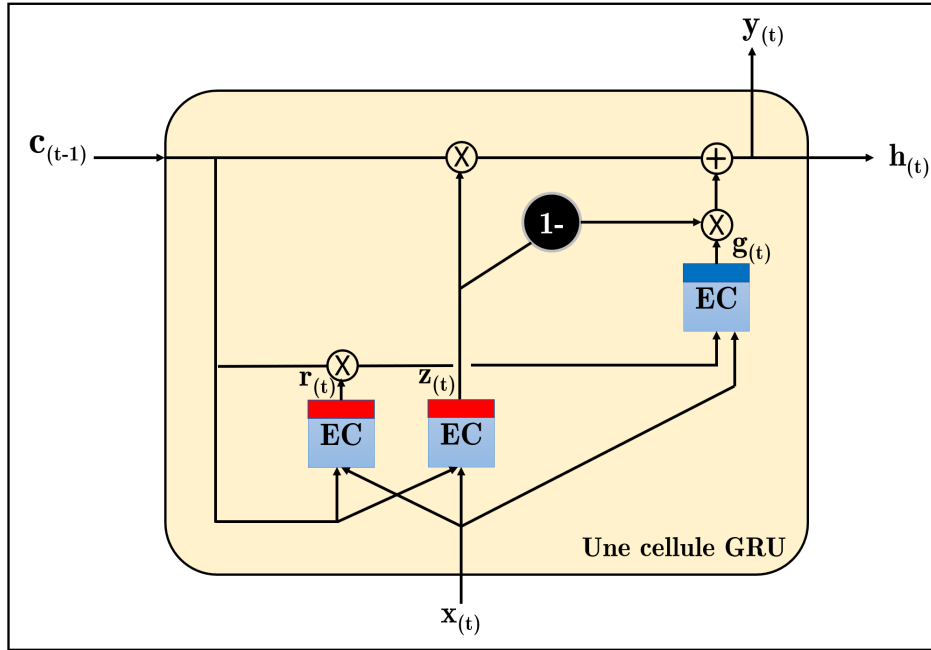


FIGURE 2.7 – Une cellule récurrente à porte (GRU)

- Les deux vecteurs d'état sont fusionnés en un seul vecteur  $h(t)$ .
- Un contrôleur de porte unique  $z(t)$  contrôle à la fois la porte d'oubli et la porte d'entrée. Si le contrôleur de porte émet un 1, la porte d'oubli est ouverte ( $= 1$ ) et la porte d'entrée est fermée ( $1 - 1 = 0$ ). S'il émet un 0, c'est le contraire qui se produit. En d'autres termes, chaque fois qu'une mémoire doit être stockée, l'emplacement où elle sera stockée est d'abord effacé. Il s'agit en fait d'une variante fréquente de la cellule LSTM en soi.
- Il n'y a pas de porte de sortie; le vecteur d'état complet est sorti à chaque pas de temps. Cependant, il existe un nouveau contrôleur de porte  $r(t)$  qui contrôle quelle partie de l'ancien sera affichée dans la couche principale ( $g(t)$ ).

L'équation 2.18 résume la manière de calculer l'état de la cellule à chaque pas de temps pour une seule instance [Cho et al., 2014].

$$\begin{aligned}
 \mathbf{z}(t) &= \sigma(\mathbf{W}_{xz}\mathbf{x}(t) + \mathbf{W}_{hz}^\top \mathbf{h}(t-1) + \mathbf{b}_z) \\
 \mathbf{r}(t) &= \sigma(\mathbf{W}_{xr}^\top \mathbf{x}(t) + \mathbf{W}_{hr}^\top \mathbf{h}(t-1) + \mathbf{b}_r) \\
 \mathbf{g}(t) &= \tanh(\mathbf{W}_{xg}^\top \mathbf{x}(t) + \mathbf{W}_{hg}^\top (\mathbf{r}(t) \otimes \mathbf{h}(t-1)) + \mathbf{b}_g) \\
 \mathbf{h}(t) &= \mathbf{z}(t) \otimes \mathbf{h}(t-1) + (1 - \mathbf{z}(t)) \otimes \mathbf{g}(t)
 \end{aligned} \tag{2.18}$$

#### 2.4.6 Les séries de temps

Les données de séries temporelles se composent de deux composants principales, à savoir les unités de temps et la valeur de consonne attribuée pour l'unité de temps fournie. Les séries temporelles suivent les tendances des points de données dans une période donnée à intervalles réguliers. Dans les séries temporelles, il n'y a aucune spécification obligatoire de la taille des données à inclure, ce qui permet de rassembler les données de manière à fournir les informations que l'analyste ou la personne qui examine l'activité doit fournir [Blanco et al., 2018]. Les séries temporelles sont connues pour être un

problème répandu d'intérêt pratique significatif car elles permettent de prédire les valeurs futures des séries avec une certaine marge d'erreur par rapport à ses valeurs passées [Tealab, 2018].

Les données de séries temporelles sont classées en deux types :

- **Séries temporelles stationnaires :** Les séries temporelles stationnaires sont celles dont les propriétés statistiques telles que la moyenne, la variance et l'autocorrélation sont toutes constantes dans le temps.
- **Séries temporelles non stationnaires :** les séries temporelles non stationnaires sont les séries temporelles dont les propriétés statistiques telles que la moyenne, la variance et l'autocorrélation changent avec le temps.

Les données non stationnaires sont imprévisibles et ne peuvent pas être modélisées ou prévues en raison du changement de la moyenne, de la variance et de la covariance au cours du temps. Afin d'obtenir des résultats cohérents et fiables, il faut d'abord les convertir en données stationnaires avant d'effectuer une analyse statistique plus approfondie. Par exemple, si la série augmente constamment avec le temps, la moyenne et la variance de l'échantillon augmenteront avec la taille de l'échantillon et elles sous-estimeront toujours la moyenne et la variance dans les périodes futures.

#### 2.4.6.a Série temporelle univariée

Une série temporelle univariée est une série avec une seule variable dépendante du temps. Dans ce type, une seule variable variera dans le temps. Par exemple, l'échantillon ci-dessous comprend les valeurs de vitesse d'un véhicule au cours du temps. Ici, la vitesse est la variable dépendante du temps [Brownlee, 2017].

	Temps_Global	Vitesse
0	1163028100	22.14
1	1163028000	22.14
2	1163027900	22.14
3	1163027800	22.14
4	1163027700	22.17
5	1163027600	22.25
6	1163027500	21.84

TABLEAU 2.1 – Série temporelle univariante

#### 2.4.6.b Série temporelle multivariée

Une série temporelle multivariée se compose de plusieurs variables dépendant du temps. Chaque variable dépend non seulement de ses valeurs passées, mais a également une certaine dépendance vis-à-vis d'autres variables. Pour le même exemple dans l'analyse univariée, les données ci-dessous ont été incluses avec plus de facteurs ainsi que la vitesse. Dans ce cas, plusieurs variables doivent être prises en compte pour prédire la position par exemple de manière optimale. Une série comme celle-ci est dans la catégorie des séries temporelles multivariées. Le tableau 2.2 représente ceci :

La prédiction de séries temporelle à variables multiples ne peut être effectuée qu'en fonction de la disponibilité des données historiques des variables d'entité. Il est possible

	Temps_Global	Position_X	Position_Y	Vitesse	Accélération
0	1163028100	-28.209	176.508	22.14	0.00
1	1163028000	-25.792	177.147	22.14	0.00
2	1163027900	-23.316	177.490	22.14	0.00
3	1163027800	-20.947	177.658	22.14	5.75
4	1163027700	-18.637	177.821	22.17	-6.13
5	1163027600	-16.430	178.113	22.25	-10.02
6	1163027500	-14.257	178.540	21.84	-9.13

TABLEAU 2.2 – Série temporelle multivariante

de prédire la variable cible à l'aide de données de séries temporelles univariées, également appelées prédictions de séries temporelles univariées, mais dans l'analyse de séries temporelles à plusieurs variables, il n'est pas possible de prédire la variable cible sans les données des variables de caractéristiques. Dans nos travaux, on s'intéresse à la prédiction de séries temporelles à variables multiples, où les données historiques des véhicules sont prises en compte pour la prédiction des intentions.

### 2.4.7 Validation croisée des séries de temps

La validation croisée des séries temporelles est une méthode d'estimation de fiabilité d'un modèle fondé sur une technique d'échantillonnage. Cette méthode consiste à diviser les données d'apprentissage en un ensemble d'apprentissage plus petit et un ensemble de validation, puis entraîner le modèle avec le petit ensemble d'apprentissage et l'évaluer avec l'ensemble de validation [Hyndman, Athanasopoulos, 2018]. La précision de la prédiction est calculée en faisant la moyenne sur les ensembles des validations et cette procédure est parfois appelée «évaluation sur une origine de prédiction glissante» [Hyndman, Athanasopoulos, 2018].

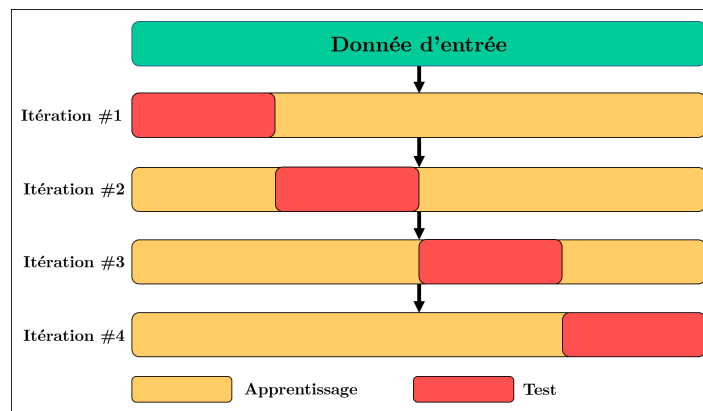


FIGURE 2.8 – Principe de la validation croisée des séries de temps

La validation croisée des séries temporelles est différente de la validation croisée des données indépendantes. Les séries temporelles sont constituées d'une structure temporelle et les valeurs ne peuvent pas être mélangées aléatoirement. Avec la randomisation, les dépendances temporelles entre les observations seront perdues. Pour surmonter ce

problème, il faut utiliser une approche plus fiable en commençant par un sous-ensemble de données d'apprentissage, en faisant des prédictions sur les points de données ultérieurs et en vérifiant ensuite la précision des points de données estimés. Ces points de données sont ensuite inclus dans le prochain ensemble de données d'apprentissage et les points de données ultérieurs sont estimés.

En termes plus simples, le modèle doit être entraîné avec un petit segment de séries temporelles du début jusqu'à un instant  $t$ , les prédictions et l'erreur pour les étapes  $t+n$  suivantes doivent être calculées. Ensuite, l'ensemble d'apprentissage doit être décalé jusqu'à la valeur  $t+n$ , les prédictions de  $t+n$  jusqu'à  $t+2n$  doivent être calculées, et la même procédure de déplacement du segment test de la série temporelle est poursuivie jusqu'à la dernière donnée disponible.

## 2.4.8 Métriques d'évaluation

### 2.4.8.a Erreur Moyenne Absolue

L'Erreur Moyenne Absolue (EMA, en anglais MAE) est une mesure de la différence entre deux variables continues souvent utilisée pour évaluer les prédictions de séries temporelles. Elle est la métrique la plus utilisée pour la mesure de la performance des modèles car elle est simple et son interprétation est claire, contrairement à l'erreur quadratique moyenne. Le EMA est défini mathématiquement comme suit [Willmott, Matsuura, 2005] :

$$\text{EMA} = \frac{\sum_i^n |y_i - \hat{y}_i|}{n} \quad (2.19)$$

où  $y_i$  est la valeur réelle,  $\hat{y}_i$  est la valeur estimée et  $n$  le nombre total de variables.

### 2.4.8.b Erreur Quadratique Moyenne

En statistique, l'Erreur Quadratique Moyenne (EQM, en anglais MSE) d'un modèle mesure la moyenne des carrés des erreurs - c'est-à-dire la différence quadratique moyenne entre les valeurs estimées et les valeurs réelles. MSE est une fonction de risque, correspondant à la valeur attendue de la perte d'erreur au carré [Mood et al., 1963].

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (2.20)$$

où  $y_i$  est la valeur réelle,  $\hat{y}_i$  est la valeur estimée et  $n$  le nombre total de variables.

### 2.4.8.c Précision, rappel et F-mesure

La précision est définie comme le rapport entre les échantillons pertinents ou réellement positifs et le nombre total de résultats positifs prévus, formulé mathématiquement comme suit [Makhoul et al., 1999] :

$$\text{Précision} = \frac{\text{Vrais Positifs}}{\text{Vrais Positifs} + \text{Faux Positifs}} \quad (2.21)$$

Ainsi, une valeur de précision de 1 indique que l'algorithme n'a pas produit de faux positifs, ce qui signifie que les résultats de la prédiction sont très fiables.

Le rappel est une mesure décrivant la fraction des échantillons pertinents qui ont été récupérés, mathématiquement [Makhoul et al., 1999]

$$\text{Rappel} = \frac{\text{Vrais Positifs}}{\text{Vrais Positifs} + \text{Faux Négatifs}} \quad (2.22)$$

Une valeur de rappel de 1 signifie que l'algorithme a pu identifier correctement tous les échantillons pertinents de l'ensemble de données.

Le score F1 d'une classification est la moyenne harmonique de la précision et du rappel, plus précisément il peut être décrit comme suite [Makhoul et al., 1999] :

$$\text{F1-Mesure} = 2 \cdot \frac{\text{précision} \cdot \text{rappel}}{\text{précision} + \text{rappel}} \quad (2.23)$$

Un score F1 de 1 signifie que le classificateur a une précision et un rappel parfaits pour la quantité mesurée.

#### 2.4.8.d La courbe ROC

La courbe ROC (Receiver Operating Characteristics) [Bradley, 1997] est une méthode graphique qui permet de visualiser les performances des algorithmes d'apprentissage statistique. Sur cette courbe est représentée en abscisse la proportion de faux positifs et en ordonnée la proportion de vrais positifs voir figure 2.9.

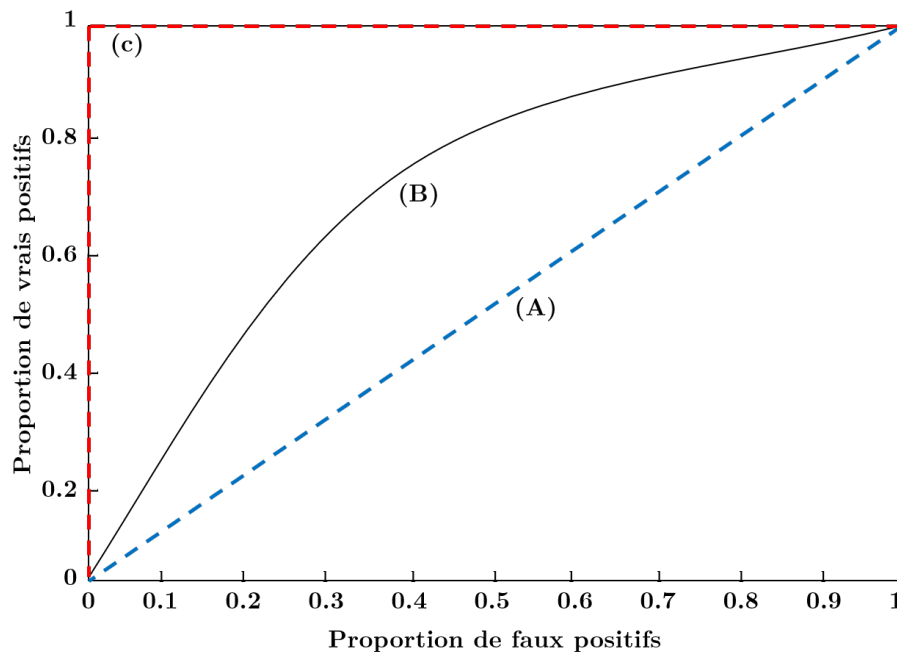


FIGURE 2.9 – Exemples de modèle de courbe ROC

- La courbe ROC idéale est représentée par la courbe (C). Dans ce cas de figure, l'algorithme prédit parfaitement toutes les étiquettes sans jamais se tromper. Un bon système d'aide à la décision essaiera de tendre vers cette limite.
- Une courbe (B) représente un classifieur qui arrive à prédire les bonnes étiquettes en se trompant dans certains cas.
- Une courbe diagonale (la courbe(A)) représente le classifieur aléatoire, c'est un cas non informatif car les chances sont équiprobables pour que le classifieur arrive à prédire les bonnes étiquettes ou à se tromper.

- Si on obtient une courbe sous la diagonale, cela est souvent synonyme d'un problème sur les sorties du classifieur (présence d'un biais).

Un classifieur sera donc d'autant meilleur que sa courbe se situera proche de la courbe (C) et loin de la diagonale. Souvent, à partir de cette courbe, le critère (AUC) (aire sous la courbe ROC) est calculé. Ce critère permet de connaître le lien entre le taux de vrais positifs et le taux de faux positifs. Plus, l'aire est importante, plus le classifieur possède un pouvoir discriminant sur les classes. L'utilisation du critère (AUC) peut se révéler fort intéressante dans le cas où une classe est sous représentée par rapport à une autre.

## 2.5 Apprentissage non supervisé

L'apprentissage non supervisé consiste à chercher un partitionnement homogène d'une manière qui reflète la structure statistique de l'ensemble des motifs d'entrées [Dayan et al., 1999]. Contrairement à l'apprentissage supervisé, dans l'apprentissage non supervisé, il n'y a pas de variables de sortie associée à chaque variable d'entrée  $X$ .

L'algorithme de l'apprentissage non supervisé apprend à partir d'une base d'apprentissage définie de la façon suivante :

$$\mathcal{S} = \{(\mathbf{x}_i)\}_{i=1}^N \quad (2.24)$$

Les problèmes traités par l'apprentissage non supervisé sont : le regroupement automatique et la réduction de la dimension. Les solutions proposées pour résoudre ces deux problèmes s'appuient généralement sur des arguments géométriques ou sur une modélisation probabiliste.

### 2.5.1 Regroupement automatique

Le problème du regroupement automatique<sup>6</sup> consiste à regrouper les données en sous-groupes ayant des caractéristiques statistiques semblables. Autrement dit, le but est de séparer les données d'entrées en groupes d'individus qui présentent des caractéristiques similaires. Le clustering est riche par ces méthodes (les approches hiérarchique [Ward Jr, 1963] (Classification Ascendante Hiérarchique, CAH), l'algorithme des k-means [MacQueen, others, 1967], les modèles de mélange gaussien [McLachlan et al., 2019],...). L'intérêt de cette méthode réside dans le fait qu'elle permet d'avoir les mêmes sorties qu'un apprentissage supervisé en se basant juste sur les variables d'entrées.

### 2.5.2 Réduction de la dimension

Dans les problématiques de reconnaissance de formes, lorsque la dimension de l'espace des variables d'entrées d'un système est élevée, il est important et nécessaire de réduire cette dimension. En effet, les variables décrivant le problème à modéliser ne sont pas toutes de la même importance pour la compréhension du phénomène étudié. Il existe des variables qui sont pertinentes pour l'identification du modèle et d'autres qui ne le sont pas.

Toutefois, la réduction de la dimension de l'espace des variables d'entrées a pour objectifs de faciliter l'analyse des données, d'améliorer les performances des classifieurs,

6. En anglais Clustering. Dans la suite de ce manuscrit, nous préférons utiliser le terme communément employé "clustering" au lieu de "regroupement automatique".

de sélectionner les ensembles de données les plus cohérents à la fois pour plus de précision et pour la réduction du temps de calcul des algorithmes, enlever les informations redondantes ou non pertinentes, etc . . . Les méthodes de la réduction de la dimensionnalité sont généralement regroupées en deux approches : les approches linéaires (ACP Analyse en Composantes Principales, ALD Analyse Linéaire Discriminante, MDS Multi-Dimensional Scaling) et les approches non linéaires (Isomap, LLE (Locally Linear Embedding)), pour plus de détail voir [Lee, Verleysen, 2007].

## 2.6 Approches d'apprentissage non supervisé

Dans ce paragraphe, nous avons cité quelques exemples d'approches d'apprentissage non-supervisé. Les techniques d'apprentissage non supervisé : K-means et le clustering spectrale se sont avérées efficaces pour trouver des modèles dans les données véhiculaires. Dans cette section, ces deux techniques utilisées dans cette thèse seront décrites.

### 2.6.1 La méthode de K-means

K-means<sup>7</sup> [MacQueen, others, 1967] est l'algorithme d'apprentissage automatique non supervisé le plus couramment utilisé pour partitionner un ensemble de données donné en un ensemble de  $k$  groupes (c'est-à-dire  $k$  clusters), où  $k$  représente le nombre de groupes spécifié au préalable par l'analyste. Il classe les objets dans plusieurs groupes (c'est-à-dire des grappes), de sorte que les objets d'une même grappe soient aussi semblables que possible (c'est-à-dire une forte similarité intra-classe), tandis que les objets de grappes différentes sont aussi dissemblables que possible (c'est-à-dire une faible similarité inter-classe). Dans le clustering k-means, chaque cluster est représenté par son centre (centroïde) qui correspond à la moyenne des points assignés au cluster.

L'idée de base du clustering k-means consiste à définir des clusters de manière à minimiser la variation totale intra-cluster (appelée variation totale intra-cluster). Il existe plusieurs algorithmes de k-means. L'algorithme standard est l'algorithme de Hartigan-Wong (1979), qui définit la variation totale à l'intérieur du cluster comme la somme des distances euclidiennes au carré entre les éléments et le centroïde correspondant :

$$W(C_k) = \sum_{x_i \in C_k} (x_i - \mu_k)^2 \quad (2.25)$$

$x_i$  désigne un point de données appartenant au cluster  $C_k$ .

$\mu_k$  est la valeur moyenne des points assignés au cluster  $C_k$ .

Chaque observation  $x_i$  est assignée à un cluster donné de telle sorte que la somme des distances carrées de l'observation par rapport aux centres de grappes qui lui sont assignés  $\mu_k$  soit minimale.

Nous définissons la variation totale au sein du cluster comme suit :

$$\text{Totale} = \sum_{k=1}^k W(C_k) = \sum_{k=1}^k \sum_{x_i \in C_k} (x_i - \mu_k)^2 \quad (2.26)$$

La somme totale des carrés à l'intérieur des clusters mesure la compacité (c'est-à-dire la qualité) du clustering et nous voulons qu'elle soit aussi petite que possible.

7. En français K-moyenne



### 2.6.1.a L'algorithme de K-means

La première étape de l'utilisation du clustering k-means est d'indiquer le nombre de clusters ( $k$ ) qui seront générés dans la solution finale. L'algorithme commence par sélectionner de manière aléatoire  $k$  objets dans l'ensemble des données pour servir de centres initiaux pour les clusters. Les objets sélectionnés sont également appelés moyens ou centroïdes des clusters.

Ensuite, chacun des objets restants est affecté à son centroïde le plus proche, où le plus proche est défini en utilisant la distance euclidienne entre l'objet et la moyenne du cluster. Cette étape est appelée "étape d'affectation des clusters". Notez que, pour utiliser la distance de corrélation les données sont entrées sous forme de z-scores. Après l'étape d'affectation, l'algorithme calcule la nouvelle valeur moyenne de chaque cluster. Le terme cluster "mise à jour des centroïdes" est utilisé pour désigner cette étape. Une fois que les centres ont été recalculés, chaque observation est vérifiée à nouveau pour voir si elle pourrait être plus proche d'un cluster différent. Tous les objets sont réaffectés à nouveau en utilisant les moyens de cluster mis à jour.

Les étapes d'affectation des clusters et de mise à jour des centroïdes sont répétées de manière itérative jusqu'à ce que les affectations des clusters cessent de changer (c'est-à-dire jusqu'à ce que la convergence soit atteinte). C'est-à-dire que les clusters formés dans l'itération actuelle sont les mêmes que ceux obtenus dans l'itération précédente.

L'algorithme K-means peut être résumé comme suit :

---

**Algorithm 1:** L'algorithme de clustering K-means

---

1. Spécifier le nombre de clusters ( $K$ ) à créer.
  2. Sélectionner de façon aléatoire  $k$  objets dans l'ensemble des données comme centres ou moyennes des clusters initiaux.
  3. Affecter chaque observation à son centroïde le plus proche, en fonction de la distance euclidienne entre l'objet et le centroïde.
  4. Pour chacun des  $k$  clusters, mettre à jour le centroïde du cluster en calculant les nouvelles valeurs moyennes de tous les points de données dans le cluster. Le centroïde d'un  $K_i^{\text{ème}}$  cluster est un vecteur de longueur  $p$  contenant les moyennes de toutes les variables pour les observations dans le  $K_i^{\text{ème}}$  cluster;  $p$  est le nombre de variables.
  5. Minimiser de manière itérative la somme totale des carrés internes. C'est-à-dire, itérer les étapes 3 et 4 jusqu'à ce que les affectations de grappes cessent de changer ou que le nombre maximal d'itérations soit atteint.
- 

### 2.6.2 Le clustering spectral

Le clustering spectral est un algorithme de clustering en plein essor qui a donné de meilleurs résultats que de nombreux algorithmes de clustering traditionnels dans de nombreux cas. Il traite chaque point de données comme un nœud de graphe et transforme ainsi le problème du clustering en un problème de partitionnement de graphe. Une mise en œuvre typique se résume en trois étapes fondamentales :

1. Construction du graphique de similarité : Cette étape construit le graphe de similarité sous la forme d'une matrice d'adjacence représentée par  $A$ . La matrice d'adjacence peut être construite de la manière suivante :-

- Graphique de voisinage d'épsilon : Un paramètre epsilon est fixé au préalable. Ensuite, chaque point est connecté à tous les points qui se trouvent dans son rayon d'épsilon. Si toutes les distances entre deux points quelconques sont d'échelle similaire, alors généralement les poids des arêtes c'est-à-dire la distance entre les deux points ne sont pas stockés puisqu'ils n'apportent aucune information supplémentaire. Ainsi, dans ce cas, le graphe construit est un graphe non orienté et non pondéré.
  - K-Proches Voisins : Un paramètre k est fixé au préalable. Alors, pour deux sommets u et v, une arête est dirigée de u vers v seulement si v est parmi les k plus proches voisins de u. Notez que cela conduit à la formation d'un graphe pondéré et dirigé car il n'est pas toujours le cas que pour chaque u ayant v comme un des k plus proches voisins, il en sera de même pour v ayant u parmi ses k plus proches voisins. Pour rendre ce graphe non dirigé, on suit l'une des approches suivantes :
    - (a) Diriger une arête de u vers v et de v vers u si v est parmi les k plus proches voisins de u OU si u est parmi les k plus proches voisins de v.
    - (b) Diriger une arête de u vers v et de v vers u si v est parmi les k plus proches voisins de u ET u est parmi les k plus proches voisins de v.
  - Graphe entièrement connecté : Pour construire ce graphe, chaque point est relié par une arête non dirigée, pondérée par la distance entre les deux points et chaque autre point. Comme cette approche est utilisée pour modéliser les relations de voisinage local, la métrique de similarité gaussienne est généralement utilisée pour calculer la distance.
2. Projection des données sur un espace de dimension inférieure : Cette étape est effectuée pour tenir compte de la possibilité que les membres d'un même cluster soient éloignés dans l'espace dimensionnel donné. Ainsi, l'espace dimensionnel est réduit de manière à ce que ces points soient plus proches dans l'espace dimensionnel réduit et puissent donc être regroupés par un algorithme de clustering traditionnel. Cela se fait en calculant la matrice laplacienne du graphique. Pour calculer cette dernière, il faut d'abord définir le degré d'un nœud. Le degré du ième nœud est donné par :

$$d_i = \sum_{j=1 | (i,j) \in E}^n w_{ij} \quad (2.27)$$

Notez que  $w_{ij}$  est l'arête entre les noeuds i et j comme défini dans la matrice d'adjacence ci-dessus.

La matrice des degrés est définie comme suit :-

$$D_{ij} = \begin{cases} d_i, & i = j \\ 0, & i \neq j \end{cases} \quad (2.28)$$

Ainsi, la matrice laplacienne du graphique est définie comme suit :

$$L = D - A \quad (2.29)$$

Cette matrice est ensuite normalisée pour des raisons d'efficacité mathématique. Pour réduire les dimensions, on calcule d'abord les valeurs propres et les vecteurs

propres respectifs. Si le nombre de clusters est  $k$ , les premières valeurs propres et leurs vecteurs propres sont pris et empilés dans une matrice telle que les vecteurs propres sont les colonnes.

3. Regroupement des données : Ce processus consiste principalement à regrouper les données réduites en utilisant n'importe quelle technique de regroupement traditionnelle - typiquement le regroupement K-Means. Tout d'abord, chaque nœud se voit attribuer une ligne de la matrice normalisée du laplacien du graphique. Ensuite, ces données sont regroupées à l'aide de toute technique traditionnelle. Pour transformer le résultat du clustering, l'identifiant du nœud est conservé.
  - (a) Sans hypothèse : Cette technique de clustering, contrairement aux autres techniques traditionnelles, ne suppose pas que les données suivent une certaine propriété. Ainsi, cela permet à cette technique de répondre à une classe plus générique de problèmes de clustering.
  - (b) Facilité de mise en œuvre et rapidité : Cet algorithme est plus facile à mettre en œuvre que les autres algorithmes de clustering et est également très rapide car il consiste principalement en des calculs mathématiques.
  - (c) Non-sollicitable : Comme il implique la construction de matrices et le calcul de valeurs propres et de vecteurs propres, cet algorithme prend beaucoup de temps pour les ensembles de données denses.

## 2.7 Apprentissage par renforcement

Dans l'apprentissage par renforcement (RL<sup>8</sup>), un agent est piloté par une fonction de récompense  $R$ . La fonction de récompense donne à l'agent un feedback basé sur la qualité d'une action  $a$ , étant donné que l'agent est dans l'état  $s$ . Grâce à ce retour d'information, l'agent apprend quelle est la meilleure action qu'il peut entreprendre à chaque état pour maximiser sa future récompense [Sutton, Barto, 2018]. Ce processus est décrit dans la figure 2.10. RL a été présenté sous de nombreuses formes au fil des ans. Les travaux dans ce domaine comprennent par exemple la programmation dynamique adaptative [Gosavi, 2009], l'apprentissage TD (Temporal Difference) [Gosavi, 2009] et SARSA (State-Action-Reward-State-Action) [Corazza, Sangalli, 2015]. Ces dernières années, l'apprentissage par renforcement profond a connu de grandes avancées. Comme ses versions non "profondes", l'apprentissage par renforcement profond a de nombreuses variantes. Pour ne citer que quelques nouvelles variantes, le réseau Q profond [Mnih et al., 2013], l'optimisation de la politique de la région de confiance [Schulman et al., 2015], l'acteur-critique à avantage asynchrone [Arulkumaran et al., 2017] et les algorithmes d'optimisation de la politique proximale [Schulman et al., 2017]. Toutes ces variantes sont utilisées dans le même objectif, c'est-à-dire développer un agent qui apprend à se comporter de manière optimale dans son environnement.

Une politique  $\pi$  est une règle de décision qui indique à l'agent l'action à entreprendre compte tenu d'un certain état  $s$ . Un déploiement de politique est le processus d'utilisation d'une politique pour évaluer de nombreuses actions possibles pour atteindre la prochaine instance temporelle et pour sélectionner l'action  $a$  qui maximise la récompense attendue à long terme. La politique reçoit ensuite l'instance temporelle suivante après avoir sélectionné l'action  $a$ , ce processus est ensuite répété pour un certain nombre d'étapes.

8. En anglais Reinforcement Learning

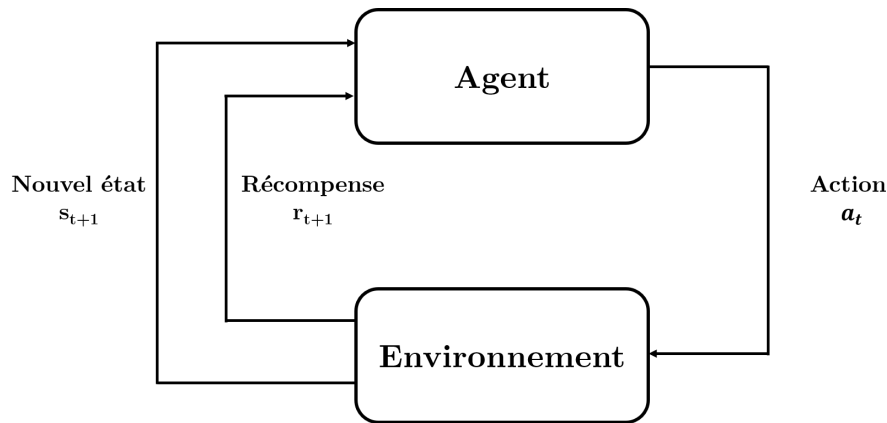


FIGURE 2.10 – Représentation du principe d'apprentissage par renforcement [Sutton, Barto, 2018]

Nous avons mentionné que le RL est le processus qui consiste à apprendre à un agent à prendre la meilleure action  $a$  à un état  $s$  où il dispose d'une fonction de récompense  $R$  qui décrit la qualité d'un  $s$  particulier. Ce processus est un processus de décision de Markov (MDP<sup>9</sup>). Un MDP est une façon de formaliser un problème qui implique l'apprentissage à partir de l'interaction pour atteindre un objectif, où l'environnement dans lequel le problème est posé est entièrement observable (ce qui est le cas dans cette thèse), sinon nous avons un processus de décision de Markov partiellement observable (POMDP). Un MDP est défini par cinq paramètres,  $S, A, P, R$  et  $\gamma$  [Sutton, Barto, 2018].

- $S$  est un ensemble fini d'états.
- $A$  est un ensemble fini d'actions.
- $P = p(s' | s, a)$  est la probabilité de transition vers l'état  $s' \in S$ , à partir de l'état  $s \in S$  en prenant l'action  $a \in A$ .
- $R = r(s, a)$  est une fonction de récompense qui renvoie la récompense immédiate attendue de l'état  $s \in S$  après l'action  $a \in A$ .
- $\gamma$  est un facteur d'actualisation,  $\gamma \in [0, 1]$ . Intuitivement, un facteur de 0 rend un agent "opportuniste" en ne considérant que les récompenses actuelles, tandis qu'un facteur approchant 1 le fera s'efforcer d'obtenir une récompense élevée à long terme.

Il existe différentes approches pour apprendre la politique qui optimise la fonction de récompense ( $R$ ). Ces approches peuvent être divisées en deux catégories, celles basées sur la valeur et celles basées sur la politique.

Les approches basées sur la politique apprennent directement la politique  $\pi$  et les approches basées sur la valeur apprennent la qualité d'un certain état  $s$  dans un environnement par la fonction de récompense fournie, et utilisent cette information pour extraire  $\pi$ .

### 2.7.1 Le Q-learning

Le Q-learning ou l'apprentissage Q est une méthode classique d'apprentissage par renforcement de l'erreur TD [Sutton, Barto, 2018], visant à trouver la fonction de valeur d'action optimale :  $Q_\pi(s, a) = E[R_t | s_t, a_t = a, \pi]$ . On peut facilement trouver la politique optimale dans un environnement. Voici la formule de l'apprentissage Q :

9. Markov Decision process

$$Q(s, a) = Q(s, a) + \alpha \left[ r + \gamma \max_{a'} Q(s, a) - Q(s, a) \right] \quad (2.30)$$

Avec :

$\alpha$  représente les taux d'apprentissage déterminant la quantité d'erreur TD que Q peut apprendre.

### 2.7.2 Deep Q Network

DeepMind a proposé une nouvelle méthode appelée Deep Q Network [Mnih et al., 2015] pour combiner l'apprentissage Q avec l'apprentissage profond. Le réseau exerce  $Q(s, a; \theta)$  pour approximer  $Q_*(s, a; \theta)$ . La combinaison a considérablement amélioré les performances de l'agent d'apprentissage par renforcement, car il peut non seulement approximer efficacement la fonction de valeur d'action, mais aussi libérer la méthode d'apprentissage Q (Q-learning) de la grande table de Q valeur (Q-values). La fonction de perte de chaque itération est montrée comme suit :

$$L_t(\theta_t) = E_{\pi} \left[ \left( r + \gamma \max_{a'} Q^*(s_{t+1}, a_{t+1}; \theta_{t'}) - Q^*(s_t, a_t; \theta_t) \right)^2 \right] \quad (2.31)$$

Avec :

$\theta_t$  représentent deux réseaux paramétrés avec la même structure.

L'algorithme DQN les utilise pour diminuer la corrélation entre la récompense et la politique afin de s'assurer que l'agent peut pleinement chercher et apprendre dans différentes conditions.

## 2.8 Conclusion

Dans ce chapitre, nous avons introduit les notions de base de l'apprentissage statique (apprentissage supervisé et apprentissage non supervisé). Ces différentes approches sont à utiliser dans le cadre de notre thèse pour la prédiction des intentions des véhicules. Nous avons aussi abordé les méthodes d'évaluation des modèles. Pour finir nous avons présenté le principe de l'apprentissage par renforcement.

Le Machine Learning est un outil très puissant qui permet d'effectuer de multiples actions. Néanmoins, il n'a pas que des qualités, il doit être constamment adapté au problème qu'il tente de résoudre. L'apprentissage automatique peut être supervisé ou non supervisé. Si les données sont clairement labellisées, on opte pour l'apprentissage supervisé. L'apprentissage non supervisé donnerait généralement de meilleures performances et de meilleurs résultats pour le regroupement des données non-labellisées. Si la taille du jeu de données est importante, le choix peut se porter sur des techniques d'apprentissage profond.

L'apprentissage par renforcement aborde le problème de l'apprentissage des stratégies de prise de décision pour les agents autonomes avec moins ou sans données. Contrairement à l'apprentissage supervisé, les systèmes d'apprentissage par renforcement ne nécessitent pas de paires d'entrée-sortie explicites pour l'entraînement. Il apprend, en continue, à s'améliorer pour trouver le meilleur modèle d'un agent qui accomplit la tâche et maximise la récompense.

# Reconnaissance de situation de conduite

*« Un seul neurone dans le cerveau est une machine incroyablement complexe que nous ne comprenons pas encore aujourd'hui. Un seul neurone dans un réseau neuronal est une fonction mathématique incroyablement simple qui capture une infime fraction de la complexité d'un neurone biologique. »*

Adrew Ng  
Chercheur en intelligence artificielle

## Sommaire

<b>2.1</b>	<b>Introduction</b>	<b>42</b>
<b>2.2</b>	<b>Généralité sur l'apprentissage</b>	<b>42</b>
<b>2.3</b>	<b>Apprentissage supervisé</b>	<b>42</b>
2.3.1	Le problème de la classification	43
2.3.2	Le problème de régression	44
2.3.3	Approches génératives/discriminatives	44
<b>2.4</b>	<b>Exemples d'approches d'apprentissage supervisé</b>	<b>44</b>
2.4.1	Machine à Vecteurs de Support	44
2.4.2	Contexte des réseaux de neurones	46
2.4.3	réseaux de neurones simple	47
2.4.4	réseaux de neurones récurrent	48
2.4.5	Cellules de réseaux de neurones récurrents	50
2.4.6	Les séries de temps	53
2.4.7	Validation croisée des séries de temps	55
2.4.8	Métriques d'évaluation	56
<b>2.5</b>	<b>Apprentissage non supervisé</b>	<b>58</b>
2.5.1	Regroupement automatique	58
2.5.2	Réduction de la dimension	58
<b>2.6</b>	<b>Approches d'apprentissage non supervisé</b>	<b>59</b>
2.6.1	La méthode de K-means	59
2.6.2	Le clustering spectral	60

<b>2.7 Apprentissage par renforcement</b> . . . . .	<b>62</b>
2.7.1 Le Q-learning . . . . .	63
2.7.2 Deep Q Network . . . . .	64
<b>2.8 Conclusion</b> . . . . .	<b>64</b>

---

## 3.1 Introduction

La conduite est une tâche délicate qui nécessite une bonne compréhension de l'environnement pour prendre les décisions appropriées. Comme un conducteur qui perçoit l'environnement et traite les informations dans son cerveau pour prendre une décision logique [Engström et al., 2005], le véhicule autonome doit être capable de reconstruire l'environnement en utilisant les informations provenant des capteurs embarqués. L'état actuel de l'environnement et des participants autour du véhicule ne suffit pas toujours pour prendre des décisions sûres. Il faut donc anticiper les déplacements des participants et prédire l'état futur des usagers de la route autour du véhicule autonome.

La prédiction de l'intention d'un véhicule peut être formulée comme un problème de régression ou de classification [Lefèvre et al., 2014]. La formulation de la classification est la plus courante dans la prédiction de l'intention. Elle est appliquée en discrétisant les états du véhicule en termes de manœuvres, comme le changement et le maintien de voie, ce qui permet une classification plus rapide et en temps réel entre les intentions de manœuvre. La prédiction de trajectoire utilise également la formulation de régression pour prédire les valeurs de position et de vitesse, de manière à cartographier le mouvement du véhicule sur la route comme dans la prédiction de trajectoire [Deo et al., 2018].

Dans ce chapitre, nous présentons une des contributions de ces travaux de thèse qui concerne la reconnaissance de situation de conduite d'un véhicule à partir d'un véhicule autonome. La première partie du chapitre est consacrée à l'explication de la problématique, abordant la question de détection des manœuvres que nous avons formalisé comme un problème de classification supervisée. Par la suite, nous évoquons les différentes étapes et les algorithmes pouvant être envisagés pour mettre en place cette procédure de classification de situation de conduite d'un véhicule. Nous présentons par la suite notre approche de reconnaissance des phases de changement de voie. Dans cette partie, on considère que chaque manœuvre est un enchaînement de trois phases : préparation, insertion et ajustement. Dans la dernière partie du chapitre, nous évoquons la problématique de la reconnaissance du style de conduite, pour laquelle nous proposons une approche de clustering pour distinguer les conducteurs agressifs et les conducteurs calmes.

## 3.2 Principe de la reconnaissance de la situation de conduite

La modélisation des manœuvres des véhicules fait référence au problème de la génération de modèles de mouvement pour le contrôle haut niveau des véhicules, telles que les changements de voie ou les virages, qui peuvent être exécutées par un conducteur. L'entrée est un ensemble de trajectoires de démonstration enregistrées par l'utilisateur pour chaque manœuvre et style de conduite, où chaque trajectoire est une séquence de positions continues du véhicule. La sortie est une bibliothèque de modèles de mouvement pour chaque manœuvre et style de conduite qui saisit les incertitudes des mouvements du véhicule à différentes étapes. La bibliothèque apprise est stockée pour une utilisation future dans le problème de reconnaissance des intentions et le problème de planification des manœuvres.

La reconnaissance des intentions fait référence au problème de l'estimation et de la prévision des styles de conduite, des manœuvres et des trajectoires des véhicules. L'entrée est une séquence de positions observées d'un véhicule cible et une bibliothèque de modèles de véhicules composée de modèles de manœuvres appris hors ligne, et la sortie est une distribution de probabilité conjointe sur les intentions du véhicule, constituée des



manœuvres et des styles de conduite actuels, ainsi qu'une séquence de manœuvres et de trajectoires futures des véhicules agents.

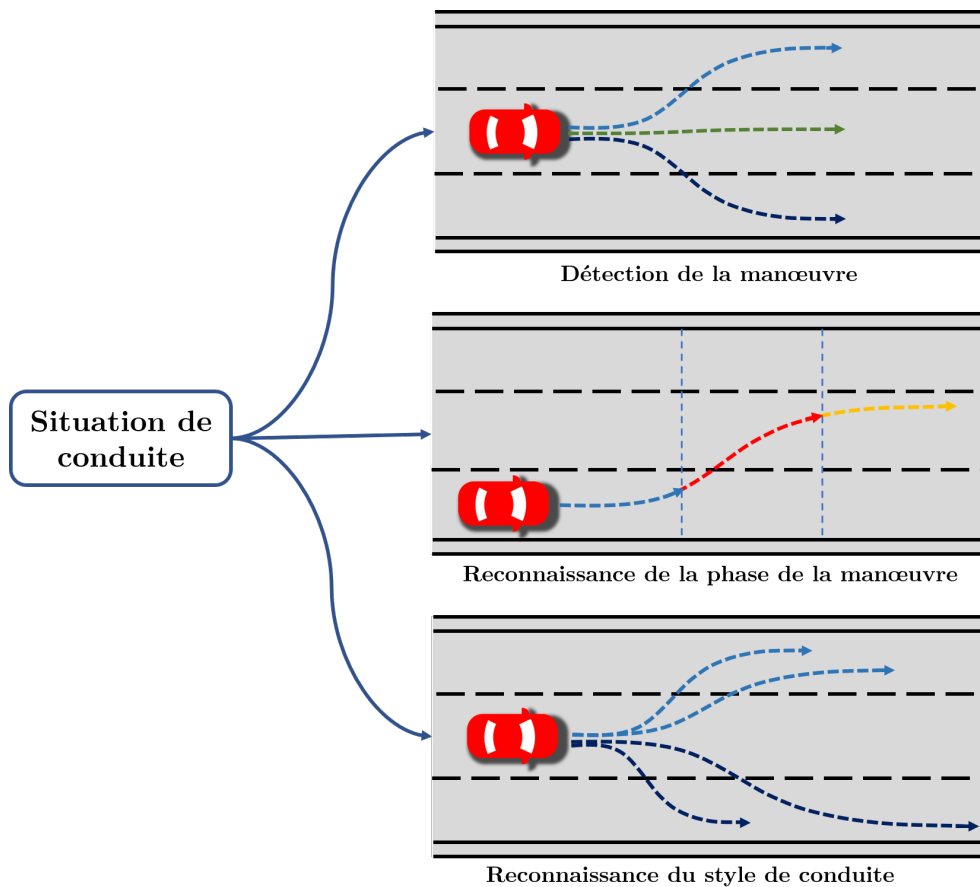


FIGURE 3.1 – Les méthodes de reconnaissance de la situation de conduite

La planification des manœuvres fait référence au problème de la génération d'une séquence de manœuvres qui satisfont l'exigence d'optimalité sous certaines contraintes de risque, où l'optimalité signifie que l'ego véhicule passe le minimum de temps à se déplacer vers le lieu de destination, et le risque est défini comme la probabilité de collisions avec des obstacles tels que les véhicules adjacents. La manœuvre de changement de voie est l'une des actions essentielles qu'un conducteur effectue lorsqu'il circule sur une autoroute. Les conducteurs effectuent certaines actions et réactions pendant la conduite sur les autoroutes en fonction de leur destination ou d'événements liés au temps. Cette réaction est perçue par l'ego véhicule. Le temps de perception-réaction constitue le délai de propagation. Dans les cas critiques, le délai de propagation peut ne pas laisser suffisamment de temps de manœuvre pour éviter les collisions si l'espacement entre les véhicules n'est pas adéquat. Les erreurs dans le processus de décision de changement de voie, généralement causées par l'incapacité du conducteur à interpréter avec précision et en temps utile les informations concernant les autres véhicules à proximité, sont souvent causes d'accidents.

Afin d'éliminer ces erreurs, ou du moins de minimiser leur impact, et d'augmenter le niveau de sécurité, les véhicules du futur devront intégrer des solutions intelligentes qui permettront d'examiner avec précision tous les aspects d'une manœuvre de changement de voie. Pour cela, un certain nombre de paramètres en temps réel doivent être pris en compte et estimés. Premièrement, l'acquisition de données tels que la vitesse, l'accélération, la décélération et la position des véhicules voisins; deuxièmement, déterminer le

début de la manœuvre de changement de voie; et troisièmement, l'estimation des trajectoires optimales et la distance de sécurité minimale entre les véhicules. La solution de l'apprentissage automatique promet de fournir des fonctionnalités telles que la sécurité des véhicules et la stabilité du trafic sur les autoroutes grâce la reconnaissance de la situation de conduite.

Pour la reconnaissance de la situation de conduite, on utilise un processus de prédiction à trois niveaux (figure 3.1) à partir d'observations de données : dans un premier temps, des méthodes d'apprentissage supervisé sont utilisées pour apprendre les intentions de conduite en relation avec le maintien ou le changement de voie. Le premier niveau de notre processus consiste à détecter, identifier et anticiper la manœuvre d'un véhicule en se basant sur son historique. Afin d'optimiser la décision prise par le véhicule, nous avons proposé dans un deuxième stage une méthode de reconnaissance de chaque phase de la manœuvre de conduite à l'aide d'un modèle profond d'apprentissage. Les performances d'un système de conduite autonome dépendent fortement du contexte de conduite actuel. Il est crucial de développer un système de conduite en tenant compte des différentes situations et environnements de trafic ainsi que du comportement des conducteurs. Dans un troisième niveau de notre processus, nous avons proposé des méthodes basées sur l'apprentissage non supervisé afin de reconnaître le style de conduite.

Afin d'atteindre notre objectif, nous avons adopté une méthodologie modulaire, en utilisant une approche à base d'apprentissage automatique. Cette approche permet de proposer des solutions mieux adaptées à ce type de problème grâce à la prise en compte de l'aspect multi-varié des données et la corrélation mécanique qui existe entre les paramètres mesures sur la dynamique d'un véhicule cible. La démarche que nous proposons pour la classification de situations de conduite d'un véhicule est une démarche classique de classification pour la reconnaissance des formes. A partir des données collectées, nous avons mis une stratégie qui se définit en trois niveaux de traitement :

1. Une étape de pré-traitement et de labellisation des données où il s'agit essentiellement de mettre en forme les données;
2. Une étape de mise en œuvre d'un ensemble de classifieurs;
3. Une étape finale de classification et d'évaluation des méthodes utilisées.

La méthodologie proposée est présentée avec ses différentes étapes dans la figure 3.2.

La première partie du chapitre sera consacrée à la description des approches d'apprentissage utilisées à savoir : machine à vecteurs de support, réseaux de neurone de classification et les réseaux de neurones récurrents. La partie suivante sera consacrée à la mise en œuvre de ces différentes approches pour la reconnaissance de situations de conduite.

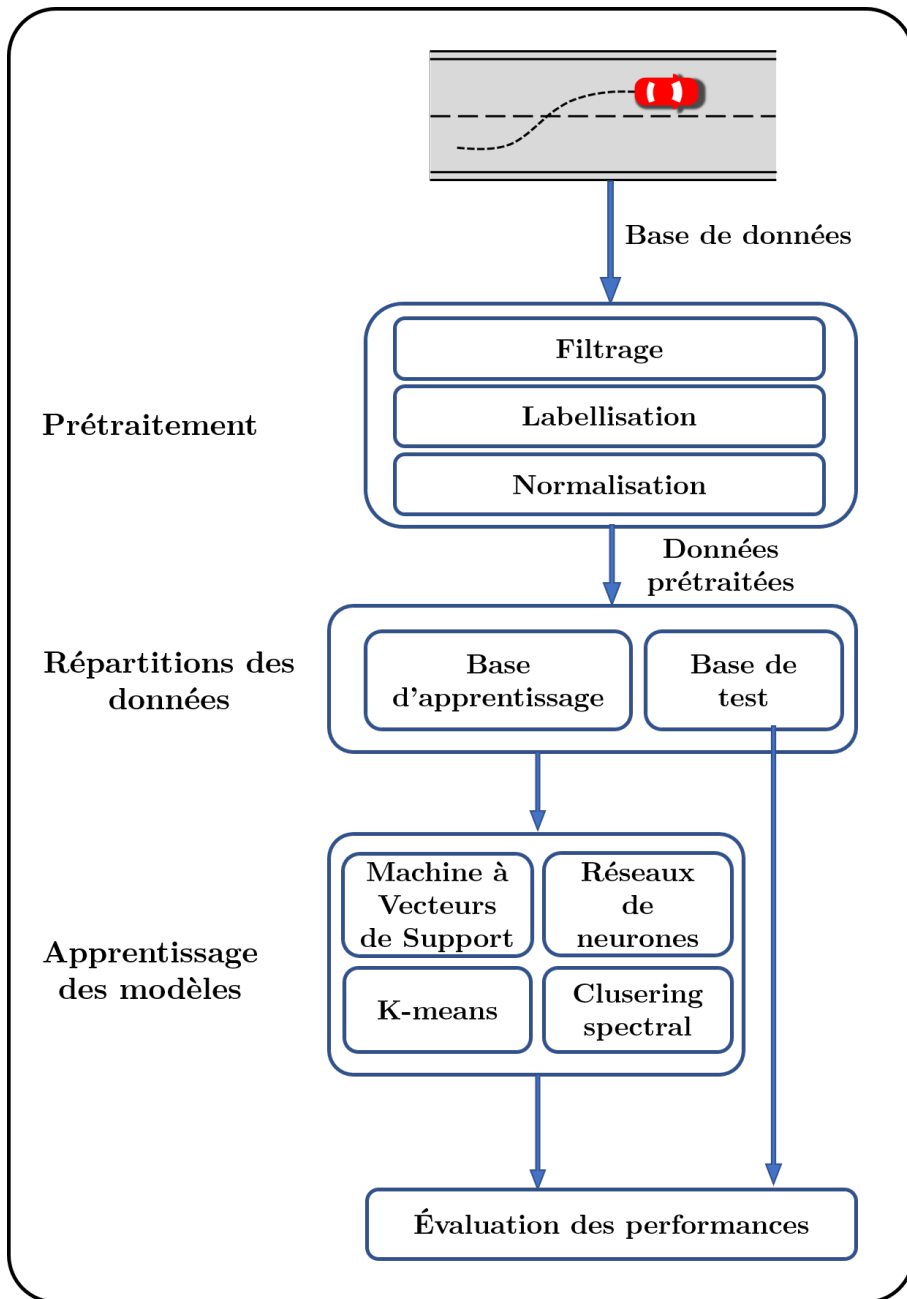


FIGURE 3.2 – Synoptique de la méthodologie proposée

### 3.3 Préparation des données

Dans les applications d'apprentissage automatique, il ne suffit pas d'avoir beaucoup de données. Pour que le modèle soit correctement entraîné, les données utilisées doivent être propres, précises, complètes et bien labellisées. La préparation de ces données est donc une étape cruciale. Les données utilisées dans cette partie proviennent du jeu de données de Next Generation Simulation (NGSIM). NGSIM se compose de données enregistrées sur plusieurs autoroutes aux États-Unis. Plus précisément, l'enregistrement de US-101 en Californie sera utilisé. La zone d'étude mesurait environ 640 mètres de longueur et se composait de cinq voies principales sur toute la section. Les données sont capturées par plusieurs caméras situées au-dessus de l'autoroute enregistrant les informations des véhicules sur l'autoroute. Sept caméras vidéo numériques synchronisées, montées en haut d'un bâtiment de 36 étages adjacent à l'autoroute, ont enregistré des véhicules passant dans la zone d'étude. Les données de trajectoire de chaque véhicule ont ensuite été extraites à partir de la vidéo à l'aide de NG-VIDEO, une application logicielle personnalisée développée pour le programme NGSIM. A chaque échantillon, les coordonnées d'objets sont enregistrées. La fréquence d'échantillonnage des données est de 10 Hz et l'enregistrement dure 45 minutes.

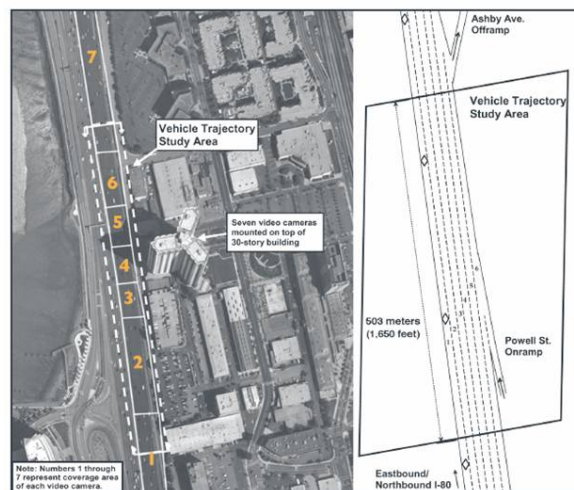


FIGURE 3.3 – La zone d'étude de NGSIM [FHWA, 2008]

Le jeu de données NGSIM fournit les trajectoires de véhicule sous la forme de coordonnées  $(X, Y)$  du centre avant du véhicule dans un repère global et de coordonnées locales  $(x, y)$  du même point sur un repère relative sur la route (figure 3.4). Dans cette thèse, pour des raisons de précision et pour faciliter l'intégration des modèles sur les véhicules, nous utilisons le système de coordonnées locales (colonnes 5 et 6 du jeu de données), où  $x$  est la position latérale du véhicule par rapport au bord à gauche de la route, et  $y$  sa position longitudinale. De plus, l'ensemble de données contient le numéro de voie de chaque véhicule à chaque échantillon, ainsi que des informations sur les dimensions et le type de véhicule (moto, voiture ou camion). Enfin, les données contiennent également d'autres informations sur le véhicule comme l'identifiant du véhicule précédent et suivant.

Chaque ligne du jeu de données NGSIM représente les données temporelles et spatiales pour un seul véhicule. La description des données incluses dans chaque champ est donnée dans le tableau 3.1

Afin de pouvoir utiliser de façon convenable les données issues de NGSIM, il est né-

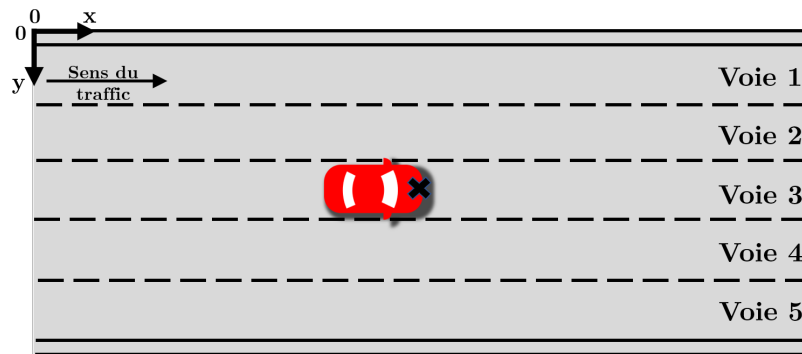


FIGURE 3.4 – Le système local des coordonnées et l'ordre des voies de NGSIM

Colonne	Intitulé	Unité	Description
1	Identifiant véhicule	-	Numéro d'identification du véhicule (croissant en fonction de l'heure d'entrée dans la section)
2	Identifiant d'image	-	Numéro d'identification de l'image temporelle de 0,1 s (ascendant par l'heure de départ)
3	Nombre dimages	-	Nombre total de trames temporelles de 0,1 s dans lesquelles le véhicule apparaît dans l'ensemble de données
4	Temps global	Millisecondes	Temps basée sur le nombre de secondes écoulées depuis le 1 <sup>er</sup> janvier 1970 00 :00 :00 UTC
5	Position latérale locale	Pieds	Coordonnée latérale (X) de l'avant-centre du véhicule par rapport un repère relative à la route
6	Position longitudinale locale	Pieds	Coordonnée longitudinale (Y) de l'avant-centre du véhicule par rapport un repère relative à la route.
7	Position latérale globale	Pieds	Coordonnée latérale (X) du centre avant du véhicule basée sur le système de coordination globale de l'État de Californie
8	Position longitudinale globale	Pieds	Coordonnée longitudinale (Y) du centre avant du véhicule basée sur le système de coordination globale de l'État de Californie
9	Longueur du véhicule	Pieds	Longueur du véhicule
10	Largeur du véhicule	Pieds	Largeur du véhicule
11	Classe de véhicule	-	Type de véhicule : 1 - moto, 2 - voiture, 3 - camion
12	Vitesse	Pieds/seconde	Vitesse instantanée du véhicule
13	Accélération	Pieds/seconde	Accélération instantanée du véhicule
14	Numéro de voie	-	Position actuelle du véhicule dans la voie de circulation
15	Identifiant du véhicule leader	-	Numéro d'identification du véhicule leader sur la même voie. Une valeur de "0" signifie qu'il n'y a pas de véhicule leader"
16	Identifiant du véhicule suiveur	-	Numéro d'identification du véhicule suiveur sur la même voie. Une valeur de "0" signifie qu'il n'y a pas de véhicule suiveur"
17	Interdistance spatiale	Pieds	Distance entre l'avant-centre d'un véhicule et l'avant-centre du véhicule leader
18	Temps inter-véhiculaire	Secondes	Temps nécessaire pour se déplacer du centre avant d'un véhicule (à la vitesse du véhicule) au centre avant du véhicule leader

TABLEAU 3.1 – Description des données contenues dans le jeux de données NGSIM

cessaire de réaliser certains prétraitements de ces données. Comme il est possible de le voir sur le tableau de description des données (Tableau 3.1), plusieurs variables sont exprimées dans des unités américaines et non standard. Il faut donc opérer une conversion des valeurs dans les unités du système international, à savoir passer les longueurs en mètres et non en pieds et également les vitesses et les accélérations en *mètre/seconde* et en *mètre/seconde*<sup>2</sup>. Nous rappelons qu'un pied vaut 30,48 centimètres, soit 0,3048 mètres. Une fois cette étape de changement d'unités réalisée, il est nécessaire de normaliser certaines données comme le temps qui est établi ici en millisecondes et qui prend pour origine le 1<sup>er</sup> janvier 1970. Nous ramènerons donc les données à une origine des temps prise comme le temps d'entrée du premier véhicule dans la section d'étude. Nous traduirons également ce temps en secondes.

### 3.3.1 Filtrage de données

Une limitation connue du jeu de données NGSIM est que les données de la position du véhicule ont été obtenues à partir d'un traitement des flux vidéo et que les trajectoires enregistrées sont bruitées [Montanino, Punzo, 2013] (voir figure 3.5). Les vitesses, obtenues par dérivation numérique, souffrent encore plus de ce bruit. Pour cette raison, nous avons utilisé un filtre Savitzky-Golay [Savitzky, Golay, 1964].

Le filtre Savitzky-Golay (SG) est un filtre numérique à Réponse Impulsionnelle Finie (RIF), dont la fonction de transfert est représentée par des équations polynomiales. Bien que le filtre SG soit plus efficace pour préserver les composantes haute fréquence pertinentes, les filtres SG sont moins efficaces pour éliminer les bruits de haut niveau dans un signal. L'idée principale présentée par Savitzky et Golay [Savitzky, Golay, 1964] est d'éviter les problèmes rencontrés avec les moyennes glissante, tout en maintenant le lissage des données et en préservant les caractéristiques de la distribution telles que les maxima, les minima et la largeur relatifs. Savitzky et Golay ont proposé une méthode de lissage des données basée sur l'approximation polynomiale des moindres carrés locaux. L'avantage du filtre de Savitzky-Golay est qu'il préserve certaines caractéristiques des séries temporelles comme les minima et maxima locaux. L'algorithme calcule une régression polynomiale locale sur les données d'entrée en résolvant le polynôme qui s'exprime sous la forme :

$$Y = a_0 + a_1z + a_2z^2 + \dots + a_kz^k \quad (3.1)$$

Où  $k$  et  $a$  sont respectivement l'ordre et le coefficient de la fonction polynomiale. Pour le filtrage nous avons utilisé un SG de troisième ordre. Ce filtre fonctionne bien pour la différenciation des signaux avec une longueur de fenêtre de 21 échantillons (correspondant à une fenêtre de temps de 2 s) pour lisser les positions longitudinales et latérales et calculer les vitesses correspondantes, comme l'illustre la figure 3.5.

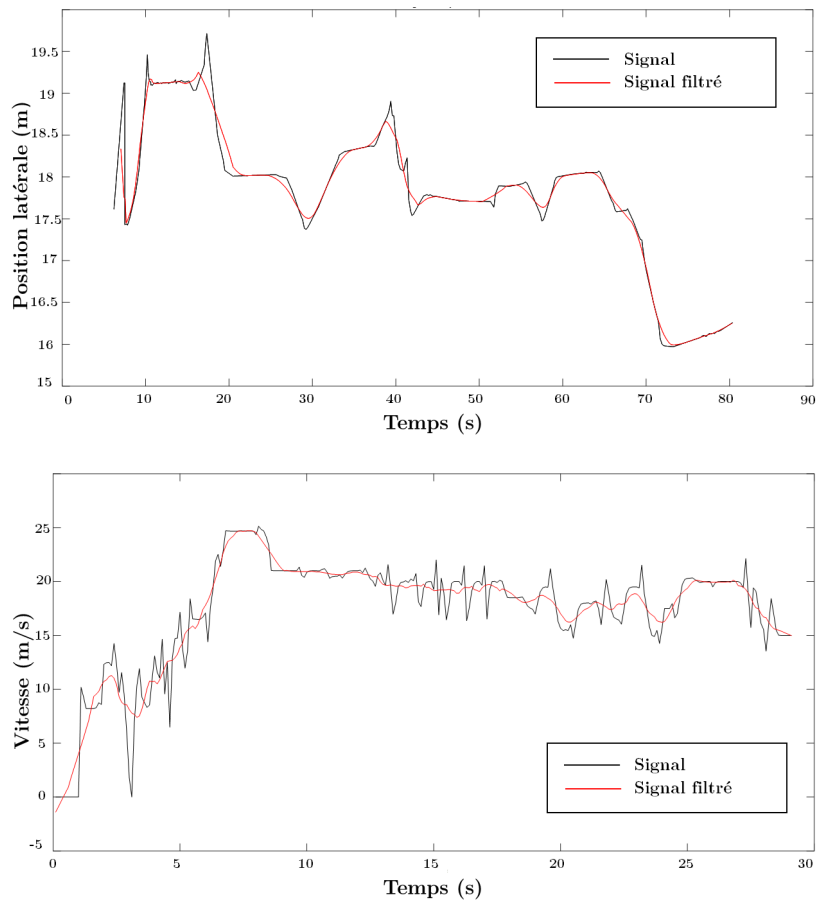


FIGURE 3.5 – Filtrage de la position latérale et de la vitesse

### 3.4 Représentation de la manœuvre de changement de voie

L'intention de conduite peut être définie comme la manœuvre préférée par le conducteur parmi un ensemble de manœuvres disponibles en fonction de l'environnement perçu. Selon [Lefèvre et al., 2014], le raisonnement au niveau de la manœuvre extrait les caractéristiques de haut niveau, ce qui conduit à une estimation fiable du mouvement et du risque à long terme et plus facile à généraliser.

En ce qui concerne le contrôle latéral du véhicule sur les autoroutes, le maintien de la voie et le changement de voie sont des aspects fondamentaux de la conduite sur autoroute [Eskandarian, 2012]. Le maintien de la voie décrit la tâche consistant à conduire dans la voie actuelle sans avoir l'intention de la quitter. En d'autres termes, si le véhicule reste dans la même voie, il appartient à la manœuvre de maintien dans la voie. D'autre part, le changement de voie est la manœuvre associée au fait de quitter la voie actuelle et d'entrer dans une nouvelle voie. La manœuvre d'un véhicule est qualifiée de changement de voie si, au cours de la trajectoire enregistrée, une manœuvre complète de changement de voie a lieu. La figure 3.6. illustre un processus de changement de voie afin de distinguer les deux catégories. Un véhicule se déplace en ligne droite sur l'autoroute dans une voie spécifique, puis décide de changer. Au point B, le véhicule commence à exécuter une manœuvre de changement de voie qui est décrite par son mouvement latéral. Au point C, le changement de voie se produit lorsque le véhicule passe d'une voie à l'autre, tandis qu'au point D, le conducteur termine la manœuvre. Au point D, le conducteur termine le processus de changement de voie et revient au maintien de la voie. Par conséquent, la trajectoire du véhicule se compose de deux étapes de maintien de la voie (AB et DE) et d'une étape de changement de voie (BD). Un changement de voie est entièrement exécuté si tous les points (c'est-à-dire B, C et D) se trouvent dans la section observée de l'autoroute.

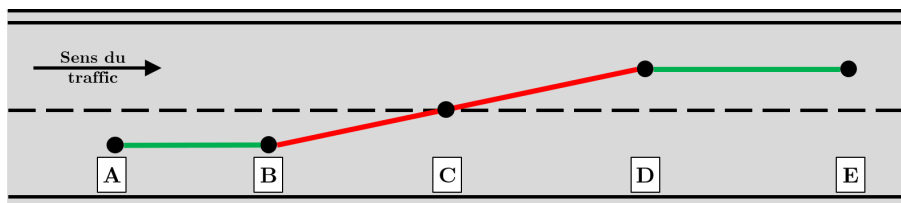


FIGURE 3.6 – Illustration des processus de maintien de voie (AB et DE) et de changement de voie (BD) d'un véhicule

Notre travail consiste à classifier trois manœuvres : changement de voie vers la droite (CVD), changement de voie vers la gauche (CVG) et maintien de voie (MV). Les changements de voie peuvent être extraits et labellisés automatiquement en suivant plusieurs critères. Dans notre travail, le critère de « numéro de voie » est utilisé pour extraire tous les changements de voie disponibles dans le jeu de données NGSIM. L'incrémentations de cette fonction peut faire référence à un changement de voie à gauche. Au contraire, la diminution fait référence au changement de voie à droite. Par conséquent, nous pouvons extraire automatiquement les changements de voie et nous pouvons également étiqueter la direction des changements (droite, gauche).

Afin de concevoir le modèle de prédiction, on doit bien comprendre et présenter chaque variable d'état du véhicule cible et des véhicules adjacents. Les véhicules changent de vitesse pendant le processus de changement de voie. Le changement de vitesse commence par une variation importante pendant la phase de préparation où la vitesse est ajustée en fonction de la vitesse des véhicules dans la voie cible. Elle est suivie d'une accélération/décélération constante dans les directions longitudinale et latérale pendant la phase



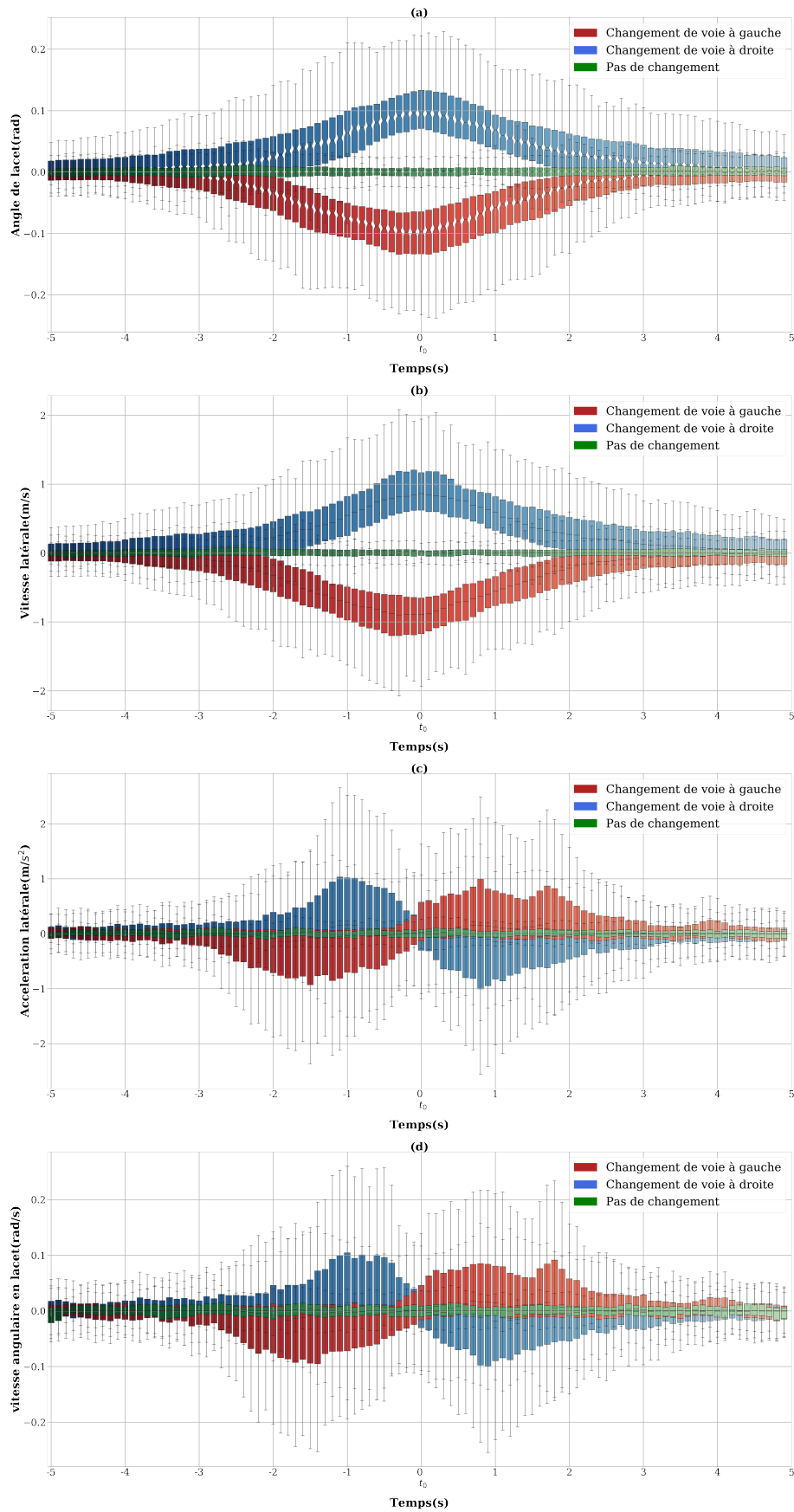


FIGURE 3.7 – Représentation en boîte à moustaches de l'angle lacet (a), de la vitesse latérale (b), de la vitesse angulaire (c) et de l'accélération latérale (d) pour les trois manœuvres

de croisement. Enfin, le véhicule ajuste sa position dans la voie cible par un changement de vitesse approprié. Il a été observé, à partir des données NGSIM sur les changements de voie, que les trajectoires des véhicules peuvent être divisées en sections correspondant à des profils de vitesses longitudinales et latérales distinctes (voir figure 3.7). Ceci a été utilisé pour améliorer la précision des prédictions et choisir la meilleure taille de la série temporelle utilisée pour l'apprentissage.

Pour chaque manœuvre (CVD, CVG, MV), on prend 50 échantillons (5 secondes) avant et 50 échantillons (5 secondes) après le franchissement de voie. Ainsi, l'instant d'intersection entre la trajectoire et le marquage de la nouvelle voie est considéré comme le temps de référence  $t_0$ , et les données pour chaque manœuvre sont extraites sur l'intervalle  $[t_0 - 5 \text{ s}, t_0 + 5 \text{ s}]$ . Sur la base des critères choisis, 380 changements de voie à droite, 420 changements de voie à gauche et plus de 1000 maintien de voie ont pu être extraits de NGSIM. Afin de créer un jeu de données homogène, nous prenons le même nombre de trajectoires pour chaque classe. Nous sélectionnons aléatoirement 300 trajectoires de chaque manœuvre pour l'entraînement et 80 trajectoires pour les tests.

Pour étudier l'influence des différents attributs du véhicule sur le changement de voie dans le temps, nous proposons une représentation graphique en boîte à moustaches de toutes les attributs extraits (figure 3.7). Chaque boîte à moustaches décrit la moyenne et l'écart type des variables calculées sur 180 manœuvres. La figure 3.7 illustre les quatre attributs choisis (la vitesse latérale, l'accélération latérale, l'angle de lacet et vitesse angulaire en lacet). Chaque représentation de boîte à moustaches affiche :

- Trois types de manœuvre (LCL, LCR, LK).
- 300 sujets pour chaque manœuvre.
- 100 pas de temps (10s) à partir du 50ème pas de temps avant le franchissement de voie et se terminent au 50ème pas de temps après le franchissement de voie.

À partir des représentations en boîte à moustaches, nous choisissons les attributs les plus pertinentes qui représentent une différence importante entre les trois classes. Les changements de voie sont détectés à l'aide de l'observation des attributs. Avant le changement de voie (entre 5s et 4s avant le franchissement de voie  $t_0$ ) l'angle de lacet (3.7 (a)), la vitesse angulaire en lacet (3.7 (b)), vitesse latérale (3.7 (c)) et accélération latérale (3.7 (d)) des trois manœuvres ont à peu près la même forme. À partir de l'instant -4s (4s avant le franchissement de voie), la différence entre les trois représentations devient significative, ce qui montre clairement que les changements de voie commencent à partir de 4s avant le franchissement de la voie. Cela réfute l'hypothèse citée dans [Schlechtriemen et al., 2014], que le changement de voie est caractérisé par une période fixe de 2s avant le franchissement du changement de voie.

## 3.5 Détection de la manœuvre

Dans cette partie du chapitre, nous présentons et analysons les résultats de classification des différentes approches supervisées décrites précédemment (machine à vecteurs de support et réseaux de neurones). Rappelons que trois situations de conduites ont été considérées : changement de voie vers la droite (CVD), changement de voie vers la gauche (CVG) et maintien de voie (MV). Le but de la classification des situations de conduite est de détecter et anticiper la future manœuvre. La reconnaissance de ces situations de conduite est réalisée par un apprentissage supervisé. Pour cela, nous avons proposé de construire des modèles pour chaque situation de conduite à reconnaître puis les comparer avec de nouvelles séquences inconnues.

### 3.5.1 Données d'apprentissage

Comme mentionné dans la section précédente, le jeu de données NGSIM offre de nombreuses attributs liés à la trajectoire du véhicule, nous extrayons donc les attributs qui sont les plus pertinentes pour le changement de voie : position locale, vitesse et accélération. Sur la base des attributs extraits et en utilisant la structure de la route, nous déterminons d'autres attributs pour satisfaire les contraintes du modèle de changement de voie :

- **Distance par rapport au marquage de la voie** : La distance au marquage de la voie dans notre cas représente la distance entre le centre avant du véhicule et le marquage de la voie. Nous calculons cette distance en fonction du marquage de droite et de gauche. Cette caractéristique donne une idée des changements de mouvement du véhicule liés à la voie actuelle et future. Nous prenons la voie initiale (avant le changement de voie) comme référence, et nous changeons cette référence pour être la nouvelle voie après la réalisation de la manœuvre.
- **Angle de lacet par rapport à la route** : Le lacet est l'angle de direction de l'orientation du véhicule par rapport à la route. Les changements d'angle de lacet peuvent être un signe très important pour détecter un changement de voie.
- **Vitesse de lacet** : La vitesse de lacet est la dérivée première de la vitesse de lacet, cette caractéristique est utilisée pour calculer la vitesse des changements d'orientation du véhicule par rapport à la route.
- **Vitesse et accélération latérales** : La dynamique latérale est une caractéristique très significative qui donne beaucoup d'informations avant que le changement de voie ne se produise. Par conséquent, nous avons utilisé les coordonnées locales liées à la route pour calculer l'accélération et la vitesse latérales.

#### 3.5.1.a La création d'un vecteur d'attributs

La conception d'un vecteur d'attributs approprié est une étape très importante pour tout classificateur discriminant, car la sélection des attributs est directement liée aux capacités discriminantes du classificateur. Bien qu'il n'y ait pas de théorie fixe derrière la conception ou le choix des vecteurs d'attributs, nous devons choisir les meilleures caractéristiques en fonction des données disponibles et de notre compréhension du problème.

Sur la base des critères choisis pour l'apprentissage, nous prenons le même nombre de trajectoires pour chaque classe, donc nous prenons 180 trajectoires de chaque manœuvre, en utilisant 8 attributs (vitesse longitudinale, vitesse latérale, accélération longi-

tudinale, accélération latérale, distance au marquage gauche, distance au marquage droit, angle de lacet et vitesse de lacet par rapport à la route).

Pour la classification des manœuvres de changement et de maintien de la voie, nous avons testé plusieurs combinaisons d'attributs (Tableau 3.2). Ces quatre ensembles d'attributs ont été sélectionnés pour couvrir toutes les données dans un scénario de conduite. Le choix s'est également porté sur les attributs qui ont montré une différence notable sur la représentation de chaque manœuvre par boîte de moustache (figure 3.7).

Les données d'entrée de notre système sont un vecteur composé d'un ensemble d'attributs sélectionnés 3.2. Où : **LongV** est la vitesse longitudinale, **LatV** est la vitesse latérale, **LongAcc** est l'accélération longitudinale, **LatAcc** est l'accélération latérale, **LeftDist** est la distance au marquage gauche, **RightDist** est la distance au marquage droit, **Yaw** est l'angle de lacet et **YawRate** est la vitesse de lacet liée à la route.

<i>Ensemble</i>	<i>Attributs</i>
E.1	LatV, LatAcc, Yaw
E.2	LatV, LatAcc, Yaw, Yaw Rate
E.3	LatV, LatAcc, Yaw, Yaw Rate, LeftDist, RightDist
E.4	LatV, LongiV, LatAcc, LongiAcc, Yaw, Yaw Rate, LeftDist, RightDist

TABLEAU 3.2 – Ensembles des attributs pour la détection des manœuvres

Après le choix des attributs, un deuxième point important lors de la création d'un vecteur d'attribut pour les données de séries temporelles est la longueur du vecteur d'entrée. Comme la préparation de changement de voie est un processus continu qui peut durer 1 seconde, 3 secondes ou même plus ou moins (figure 3.7) selon la situation ou le conducteur. Un seul point de données à un seul moment ne peut pas représenter précisément le processus complet de changement de voie. Par conséquent, pour générer des données d'entrée, une fenêtre est sélectionnée autour du moment où le véhicule passe d'une situation de maintien de voie à un changement de voie (la phase de préparation) et tous les points de données dans cette fenêtre sont étiquetés comme étant les points de données réels représentant les changements de voie correspondants. La taille de cette fenêtre devrait être suffisante pour capturer complètement un processus de changement de voie. Dans cette étude, nous avons testé plusieurs taille de fenêtres afin de trouver le meilleur compromis entre la précision de la classification et l'anticipation de la manœuvre.

### 3.5.2 La conception du classifieur

#### 3.5.2.a Conception du classifieur SVM (Machine à vecteurs de support)

La conception de l'algorithme SVM est important pour la performance globale. L'objectif est d'obtenir une grande précision lors de la classification de nouvelles données, et pas nécessairement une solution parfaite pour classer les données d'apprentissage. Le réglage de l'algorithme implique plusieurs étape à suivre. Pour implémenter l'algorithme SVM, on utilise la conception du classificateur suggérée par [Hsu et al., 2003] :

La motivation derrière l'utilisation d'un noyau est qu'il fait correspondre les données d'un espace de basse dimension à un espace de haute dimension, convertissant un problème de classification non linéaire à basse dimension à un problème de classification linéaire à haute dimension [Cristianini et al., 2000]. Pour la classification des situations de conduite, nous avons testé les quatre noyaux de base utilisés dans la littérature (linéaire, sigmoïde, RBF et polynomial à plusieurs degrés).

---

**Algorithm 2:** Les étapes d'implémentation d'un SVM

---

1. Transformer les données au format spécifié par l'algorithme SVM
  2. Effectuer la mise à l'échelle des données d'entrée
  3. Choisir la fonction noyau du SVM
  4. Utiliser la validation croisée pour trouver les paramètres optimaux
  5. Utiliser les paramètres optimaux pour entraîner le jeu de données d'apprentissage
  6. Tester
- 

**3.5.2.b Conception du classifieur ANN (Réseau de neurones artificiels)**

La complexité des réseaux de neurones se mesure par le nombre de neurones dans la couche cachée. Plus un réseau de neurones comporte de neurones, plus le système sera flexible mais également complexe. Des réseaux de neurones flexibles permettent d'approcher des fonctions de n'importe quelle complexité entre les variables d'entrée et les classes de sortie. Par conséquent, pour modéliser un ensemble de données, il est important d'utiliser des réseaux de neurones relativement flexibles, contenant suffisamment de neurones dans la couche cachée. Le choix du nombre optimal de neurones dépend en grande partie du domaine de la problématique, mais d'une manière générale, il est lié au nombre d'entrées. La conception de modèles ANN suit un certain nombre de procédures systématiques. En général, il y a sept étapes fondamentales [Gurney, 1997], ces étapes sont résumées ci-dessous :

---

**Algorithm 3:** Les étapes d'implémentation d'un ANN

---

1. Sélectionner un petit nombre de neurones cachés dans la couche.
  2. Entraînement du réseau de neurones et évaluation de ses performances du réseau de neurones.
  3. Vérification de la valeur de momentum acceptable du réseau de neurone.
  4. Si la valeur de momentum acceptable n'est pas atteinte, entraîner à nouveau le réseau de neurone et évaluer la performance du réseau de neurone.
  5. Si la valeur de momentum acceptable est atteinte, vérifiez la valeur d'erreur quadratique moyenne acceptable du réseau de neurone.
  6. Si la valeur de l'erreur quadratique moyenne acceptable n'est pas atteinte alors sélectionnez à nouveau un petit nombre de neurones cachés dans la couche.
  7. Si la valeur d'erreur quadratique moyenne acceptable est atteinte, alors le processus de sélection d'un petit nombre de neurones cachés dans la couche s'arrête. La couche s'arrête.
- 

**3.5.3 Résultats et discussions**

Dans cette partie, on présente les résultats des classifications pour la détection des trois manœuvres, en utilisant l'algorithme SVM et ANN implémentés et les ensembles de données. L'implémentation du SVM a été faite en utilisant libSVM [Chang, Lin, 2011], différentes fonctions de noyau peuvent être choisies pendant la classification, correspon-

dant aux différents espaces de données transformés. Quatre types de noyaux sont testés : Linéaire, polynomial(Poly), fonction de base radiale (RBF) et sigmoïde.

TABLEAU 3.3 – Résultats d'évaluation des SVM avec différents noyaux

Noyau	Linéaire	Sigmoïde	RBF	Polynomial 1	Polynomial 2	Polynomial 3
Précision	0.949	0.873	0.932	0.944	0.953	0.957

Le jeu de données final est constitué de 180 trajectoires pour chacune des trois classes. La recherche en grille utilisant la validation croisée à quatre niveaux pour déterminer le paramètre optimal du noyau  $\sigma$  et la taille de la fenêtre  $\gamma$  a nécessité le calcul de plusieurs combinaisons en modifiant à chaque itération le noyau et la taille de la fenêtre. Initialement, on fixe la taille de la fenêtre à  $\gamma = 2s$  pour choisir le meilleur noyau. En comparant le rappel de chacun (tableau 3.3), nous constatons que la meilleure performance de classification est obtenue en utilisant un noyau polynomial du troisième ordre [Benterki et al., 2019].

Six tailles de fenêtre entre 1 et 3 secondes, ont été sélectionnées pour la recherche de la meilleure combinaison (ensemble d'attributs/taille de fenêtre). Les résultats de l'utilisation de ces tailles de fenêtre, présentés dans le tableau 3.4, ont montré que la plus grande précision a été obtenue pour l'ensemble E.4 avec  $\gamma = 3s$ . Avec  $\gamma = 3s$ ,  $\gamma = 2.5s$  et  $\gamma = 2s$ , les précisions les plus élevées ont été atteintes avec plus de 98,3% pour les ensembles E.4. Ces trois précisions ont été obtenues avec un noyau polynomiale d'ordre 3. Lors de la sélection des tailles de fenêtre, la précision semblait s'améliorer avec l'augmentation des tailles de fenêtre. Cependant, agrandir la taille de la fenêtre peut affecter le temps de prédiction et engendrer un retard de détection de la manœuvre. Par conséquent, le choix optimale qui combine le meilleur compromis entre la précision et le temps de détection de la manœuvre est  $\gamma = 2s$  avec une précision de 98.4%. Comme conclusion, la meilleure combinaison pour notre algorithme SVM est la suivante : un noyau polynomial d'ordre 3, un ensemble d'attributs E.4 (LatV, LongiV, LatAcc, LongiAcc, Yaw, Yaw Rate, LeftDist, RightDist) avec une fenêtre de taille  $\gamma = 2s$  [Benterki et al., 2019].

TABLEAU 3.4 – Précision de l'apprentissage pour les ensembles de données E.1 à E.4

Taille de fenêtre[s]	Précision [%]			
	E.1	E.2	E.3	E.4
1	81.3	83.6	86.4	91.4
1.5	82.5	84.6	89.9	95.6
2	85.0	89.4	93.8	98.3
2.5	86.7	92.7	95.3	98.4
3	87.9	93.6	96.7	98.6

Concernant notre classifieur ANN, après avoir entraîné plusieurs réseaux en variant à chaque fois le nombre de couches cachées et le nombre des nœuds, et pour optimiser le temps de calcul, nous avons retenu un réseau qui comprend deux couches cachées. La fonction d'unité linéaire rectifiée est utilisée comme fonction d'activation, avec un solveur Broyden-Fletcher-Goldfarb-Shanno (BFGS) à mémoire limitée. Pour prendre en compte les relations entre les éléments de traitement, une architecture à connexion complète (Dense) est utilisée. Pour l'entraînement de notre réseau ANN, nous avons utilisé le

meilleur ensemble d'attributs (E.4) et la meilleure taille de fenêtre ( $\gamma = 2s$ ) trouvés avec la classification SVM.

Nous avons évalué les systèmes proposés de prédiction et de classification des changements de voie en utilisant l'ensemble de données présenté précédemment. Des situations de changement de voie et de maintien de la voie dans NGSIM sont utilisés pour l'entraînement et l'évaluation des algorithmes. Afin de montrer la performance de classification de chaque méthode proposée, la matrice de confusion est calculée, ainsi que toutes les mesures d'évaluation. Un exemple illustratif de changement de voie de véhicules sera présenté en utilisant les modèles proposées. Nous présentons les résultats en termes de métriques d'évaluation des classes prédites par rapport aux manœuvres réelles.

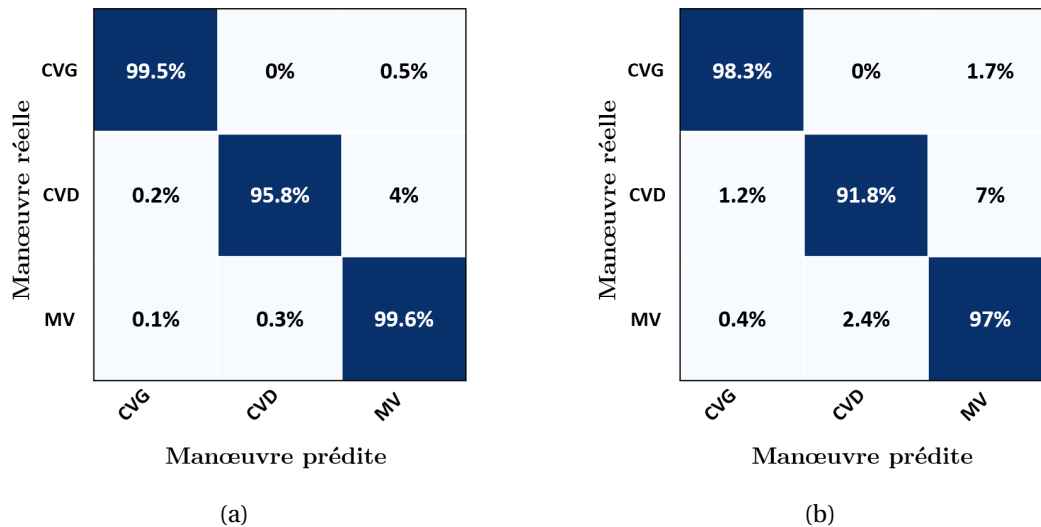


FIGURE 3.8 – Matrice de confusion des approches utilisées : (a) SVM (b) ANN

La représentation de la matrice de confusion est utilisée pour évaluer les résultats obtenus. La matrice de confusion est un outil permettant de mesurer la qualité d'un système de classification. Chaque colonne de la matrice représente une classe réelle (ou de référence), tandis que chaque ligne représente la classe prédite. L'un des intérêts de la matrice de confusion est qu'elle permet de conclure rapidement si un système de classification est capable de classer correctement.

TABLEAU 3.5 – Résultats de l'évaluation des SVM et ANN en utilisant les données sélectionnés

Approches	Rappel	exactitude	Précision	Score F1
SVM	0.957	0.971	0.958	0.957
ANN	0.983	0.988	0.983	0.983

La figure 3.8 représente les matrices de confusion et les tableaux 3.5 et 3.6 montrent la performance du classifieur SVM ainsi que celle du ANN. Concernant le SVM, la précision de prédiction de cette approche est de 97,1 %. La sensibilité, la précision et le score F1 sont presque identiques (95,7 %). Pour l'approche ANN, il est évident que ANN est plus précis avec 98.8%. En ce qui concerne les autres métriques, la classification a une sensibilité, une précision et un score F1 importants (98,3 %).

Le temps de prédiction dépend du mode de conduite, un conducteur calme commence le changement de voie très tôt et prend plus de temps pour traverser le marquage

Approche	Temps de prédiction moyen (secondes)		
	CVG	CVD	Moyenne
<b>SVM</b>	2	1.9	1.95
<b>ANN</b>	2.2	2.4	2.30

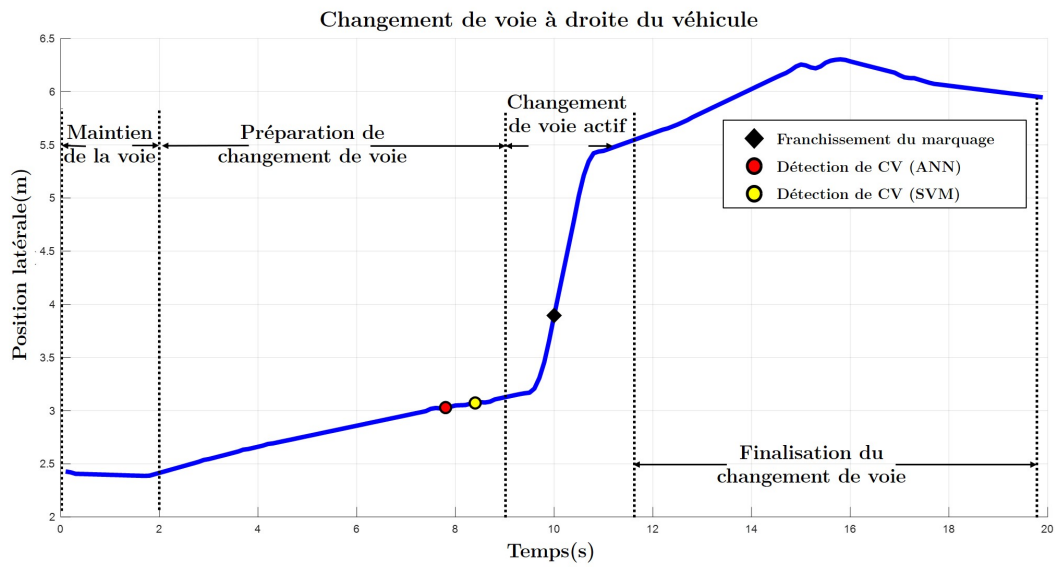
TABLEAU 3.6 – Temps de prédiction des approches SVM et ANN

de la voie, ce qui rend le temps de prédiction plus grand. A l'inverse, un conducteur agressif prend moins de temps pour changer de voie, donc le temps entre la détection du changement de voie et le franchissement de la voie est très court. Dans ce travail, nous remarquons que l'ANN détecte le changement de voie plus tôt que le SVM (tableau 3.6). Le temps de prédiction moyen pour le changement de voie à gauche est de 2.4 secondes avec l'ANN et de 2 secondes avec le SVM, en ce qui concerne le changement de voie à droite, le temps de prédiction moyen est de 2.2 secondes pour l'ANN et de 1.9 secondes avec le SVM.

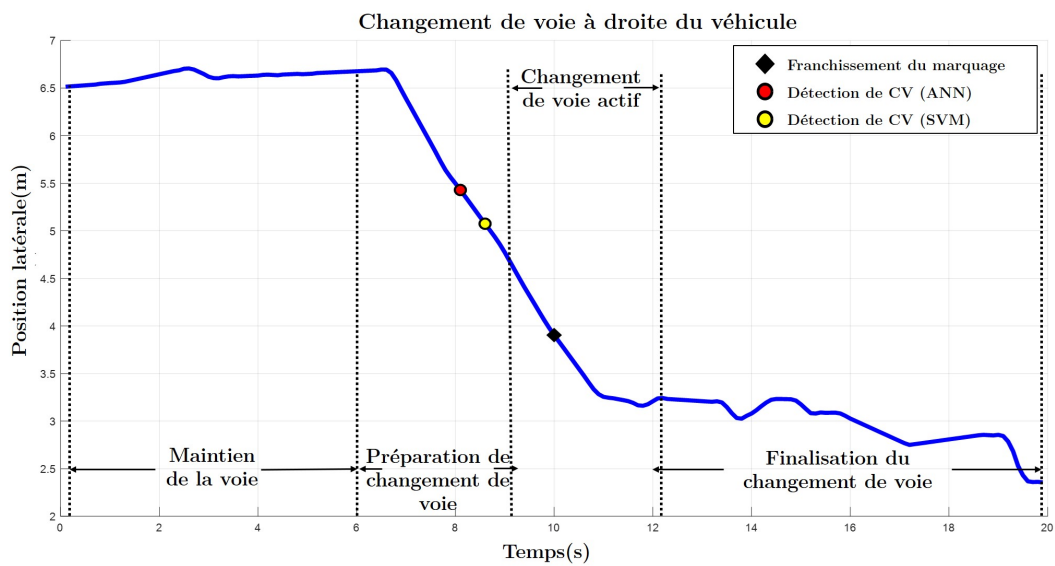
Bien que les résultats de prédiction de ANN soient meilleurs que ceux de SVM, la différence n'est pas très importante en termes de métriques d'évaluation et de temps de prédiction. Les résultats montrent clairement que les attributs sélectionnés et la taille de fenêtre sont très efficaces car elles utilisent des observations de la dynamique du véhicule et de la structure de la route.

Afin d'illustrer davantage les performances des méthodes proposées, nous représentons les positions latérales du centre avant du véhicule par rapport le bord le plus à gauche de la section d'autoroute dans le sens de la circulation pendant l'évolution du changement de voie (figures 3.9). Dans chaque figure, les instants de changement de voie, de détection de changement de voie par SVM et de détection de changement de voie par ANN sont présentés. Chaque figure représente un exemple des événements de changement de voie étudiés (CVG et CVD). En se concentrant sur les instants de détection de changement de voie, on constate que l'ANN est capable de prédire la manœuvre plus tôt que le SVM, mais les deux approches présentent un temps de prédiction significatif (supérieur à 2 secondes).





(a)



(b)

FIGURE 3.9 – Trajectoire latérale du véhicule (a) lors d'un changement de voie à droite (b) lors d'un changement de voie à gauche en indiquant l'instant de détection de chaque modèle.

### 3.6 Reconnaissance de la phase de la manœuvre

Dans cette section, on se concentre sur la reconnaissance des manœuvres de changement de voie des véhicules environnants (cibles) adjacents au véhicule autonome, comme le montre la figure 3.10. Deux hypothèses sont faites : i) l'égo véhicule peut accéder aux états dynamiques et cinématiques de base des véhicules cibles, par exemple la position, l'accélération, etc., par l'intermédiaire de capteurs embarqués ; ii) l'égo véhicule ne peut pas obtenir directement l'état des conducteurs des véhicules cibles. L'égo véhicule reçoit d'abord les mesures de l'état des véhicules cibles, puis prédit et actualise périodiquement leurs manœuvres.

Pour chaque mise à jour des manœuvres, la classification du comportement du conducteur est traitée pour définir sa situation actuelle afin de fournir au système de prise de décision toutes les informations nécessaires pour prendre la meilleure décision. Comme le comportement du véhicule cible est également influencé par les participants au trafic environnant, les états de cinq véhicules environnants (précédent dans la même voie), précédent gauche/droite, et suivant gauche/droite) sont pris en compte dans le système de classification (Fig 3.10).

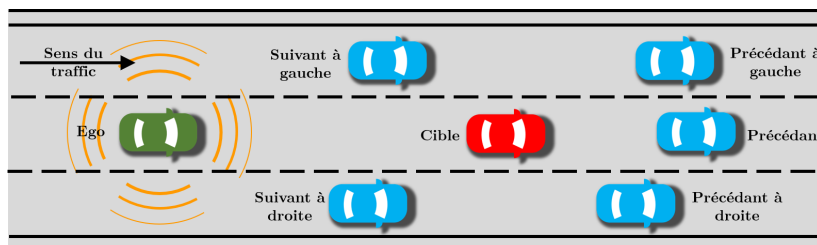


FIGURE 3.10 – Représentation du scénario de conduite. Cinq véhicules sont considérés autour du véhicule cible.

Dans cette partie, nous adoptons un processus de classification multi-classes à partir d'observations de données brutes. Les données de trajectoire contiennent généralement des informations sur la position par rapport à l'ego-véhicule, la vitesse, l'accélération et la position sur la voie d'un véhicule. Un véhicule doit appartenir à la classe "changement de voie" si, au cours de la trajectoire enregistrée, une manœuvre de changement de voie complète a eu lieu. En revanche, si le véhicule reste dans la même voie, il appartiendra à la classe "maintien de la voie". La figure 3.11 illustre un processus de changement de voie, un changement de voie est composé de trois étapes : préparation, insertion et ajustement. Un véhicule circule en ligne droite sur l'autoroute dans une voie spécifique et décide ensuite de changer la voie. Pendant la phase de préparation, le véhicule commence à exécuter une manœuvre de changement de voie, qui est décrite par son mouvement latéral. En arrivant à la phase d'insertion, le changement de voie se produit lorsque le centre du véhicule passe d'une voie à une autre, tandis que pendant la phase de l'ajustement, le conducteur termine le processus de changement de voie et suit sa nouvelle voie.

Un changement de voie est entièrement exécuté si toutes les étapes (préparation, insertion et ajustement) sont réalisées dans la section observée de l'autoroute. Dans cette section, nous développons une approche permettant de reconnaître l'intention du véhicule. Contrairement à l'état de l'art et aux approches de classification des manœuvres précédemment citées, l'avantage de notre travail est la détection de chaque étape lors du changement de voie. Comme les changements de voie à droite et à gauche sont divisés en trois étapes, la prédiction des manœuvres est reformulée, passant d'un problème de classification à trois classes à un problème de classification à sept classes. Par consé-

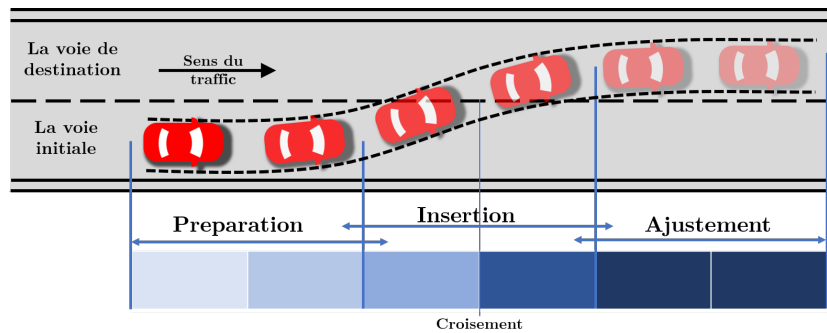


FIGURE 3.11 – Illustration des processus de changement de voie d'un véhicule : étapes de changement de voie et chronologie.

quent, l'état du véhicule est affecté à une classe d'intention, qui s'exprime comme suit :  $I \in \{\text{préparation CVG}^1, \text{insertion CVG}, \text{ajustement CVG}, \text{préparation CVD}^2, \text{insertion CVD}, \text{ajustement CVD}, \text{MV}^3\}$ .

### 3.6.1 Données d'apprentissage

#### 3.6.1.a Présentation de la base de données HighD

Toutes les données utilisées ont été extraites du jeu de données HighD [Krajewski et al., 2018]. Des algorithmes de suivi et un post-traitement ont été appliqués pour récupérer la position, les vitesses et les accélérations dans les axes  $x$  (longitudinal/ direction de conduite) et  $y$  (latéral/ perpendiculaire à la direction de conduite). La liste des attributs de l'ensemble de données est présentée dans le tableau 3.7. Au niveau du véhicule, les données présentent la trajectoire et les attributs physiques. La trajectoire du véhicule est caractérisée par sa position, sa vitesse et son accélération dans les directions longitudinale et latérale. Les attributs physiques des données sont la largeur et la longueur du véhicule. Les données comportent des attributs qui nous renseignent sur l'environnement du véhicule concerné.



FIGURE 3.12 – Photographie aérienne de la section de route pour l'enregistrement n° 1 dans le jeu de données HighD [Krajewski et al., 2018].

Il existe huit attributs pour l'identification des véhicules entourant le véhicule en question, à savoir : précédant à l'avant, suivant à l'arrière, à gauche (précédant, adjacent et suivant) et à droite (précédant, adjacent et suivant). Si aucun véhicule n'est présent pour une position particulière, une valeur de 0 lui est attribuée dans les données. La position de la voie du véhicule peut également être déterminée à partir de l'attribut "laneId" de l'ensemble de données. Certains attributs supplémentaires ont déjà été calculés dans les

1. CVG : Changement de Voie à Gauche
2. CVD : Changement de Voie à Droite
3. MV : Maintien dans la voie

Attribut	Description	Unité
frame	cadre actuel	[-]
id	l'identifiant du véhicule	[-]
x	position x du coin supérieur gauche de la boîte englobante du véhicule.	[m]
y	position en y du coin supérieur gauche de la boîte englobante du véhicule.	[m]
width	largeur de la boîte englobante du véhicule (longueur du véhicule)	[m]
height	hauteur de la boîte englobante du véhicule (largeur du véhicule)	[m]
xVelocity	vitesse longitudinale dans le système de coordonnées de l'image.	[m/s]
yVelocity	vitesse latérale dans le système de coordonnées de l'image	[m/s]
xAcceleration	accélération longitudinale dans le système de coordonnées de l'image	[m/s]
yAcceleration	accélération latérale dans le système de coordonnées de l'image	[m/s]
frontSightDistance	distance jusqu'à la fin de la section d'autoroute enregistrée dans le sens de la marche depuis le centre du véhicule	[m]
backSightDistance	distance jusqu'à l'extrémité du tronçon d'autoroute enregistré dans le sens inverse de la marche par rapport au centre du véhicule	[m]
dhw	distance Headway (= 0, s'il n'y a pas de véhicule précédent)	[m]
thw	time Headway (= 0, s'il n'y a pas de véhicule précédent)	[s]
ttc	temps jusqu'à la collision (= 0, s'il n'y a pas de véhicule précédent ou si un TTC valide existe)	[s]
precedingXVelocity	vitesse longitudinale du véhicule précédent dans le système de coordonnées de l'image	[-]
precedingId	identifiant du véhicule précédent sur la même voie	[-]
followingId	identifiant du véhicule suivant sur la même voie	[-]
leftPrecedingId	identifiant du véhicule précédent sur la voie adjacente gauche dans le sens de la marche	[-]
leftAlongsideId	identifiant du véhicule adjacent sur la voie adjacente gauche dans le sens de la marche	[-]
leftFollowingId	identifiant du véhicule suivant sur la voie adjacente gauche dans le sens de la marche	[-]
rightPrecedingId	identifiant du véhicule précédent sur la voie adjacente de droite dans le sens de la marche	[-]
rightAlongsideId	identifiant du véhicule adjacent sur la voie adjacente de droite dans le sens de la marche	[-]
rightFollowingId	identifiant du véhicule suivant sur la voie adjacente droite dans le sens de la marche	[-]
laneld	les identifiants commencent à 1 et sont attribués par ordre croissant	[-]
numLaneChanges	Le nombre de changement de voie	[-]

TABEAU 3.7 – Les attributs dans les données du jeux de données HighD [Krajewski et al., 2018]

données, à savoir la distance, le temps avant collision par rapport au véhicule précédent. Il existe une photographie aérienne de la section de la route pour chaque enregistrement dans les données. l'une de ces photographies est illustrée à la figure 3.12.

### 3.6.1.b Comparaison de HighD et de NGSIM

L'ensemble de données le plus similaire à l'ensemble de données HighD est NGSIM. Une comparaison de l'ensemble de données HighD avec l'ensemble de données NGSIM a déjà été effectuée et résumée dans le tableau 3.8. Le jeu de données HighD est plus grand que NGSIM en termes de nombre de véhicules observés et de durée des enregistrements. La longueur de la section de route observée et la durée moyenne par enregistrement sont plus courtes dans HighD que dans NGSIM. Le nombre moyen de changements de voie est supérieur dans HighD probablement en raison d'un nombre inférieur de voies par direction et d'une congestion moindre. Les vitesses moyennes dans HighD sont beaucoup plus élevées que dans NGSIM. L'ensemble de données HighD présente peu d'erreurs dans les trajectoires en raison de l'utilisation de caméras haute résolution et l'application d'algorithmes de post-traitement avancés [Krajewski et al., 2018].

### 3.6.2 Extraction et sélection des attributs

Notre étude vise à prédire trois types de manœuvres : *CVD*, *CVG*, *MV* et à détecter les phases du changement de voie : *préparation*, *insertion* et *ajustement*. Les changements de voie ont été extraits et étiquetés automatiquement selon plusieurs critères. L'attribut "numLaneChanges" est utilisé pour extraire tous les changements de voie disponibles dans le jeu de données HighD. Au total de 14 attributs d'entrée sont sélectionnés. Ils sont énumérées dans le tableau 3.9.

Caractéristique	Jeu de données NGSIM	Jeu de données HighD
Durée totale	1.5 heures	16.5 heures
Nombre des lieux	2	6
Durée moyenne d'enregistrement	45 minutes	15-20 minutes
Voies par direction	5-6	2-3
Total véhicules	9206	110,000
Pourcentage de camions	3%	23%
Vitesses moyennes	<75 km/h	voitures(120 km/h), camions(80 km/h)
Changements de voie par véhicule	0.45	0.10
Densité moyenne du trafic	Haute	Faible
Fréquence d'enregistrement (Hz)	10	25
Longueur de la section de route	0.50-1.00 km	0.42 km

TABLEAU 3.8 – Comparaison de HighD et de NGSIM [Krajewski et al., 2018]

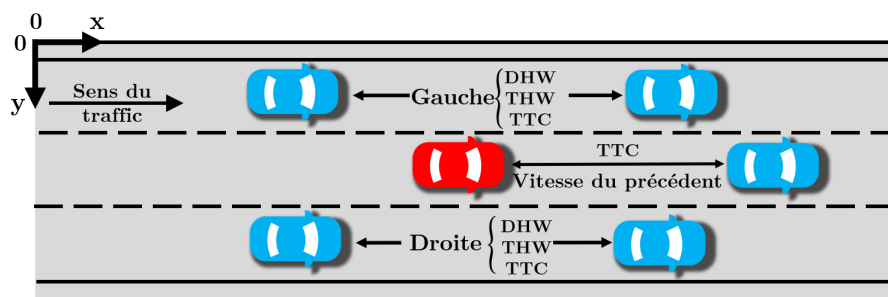


FIGURE 3.13 – Attributs sélectionnés des véhicules environnants

Comme mentionné précédemment, il y aura un maximum de cinq véhicules dans chaque vecteur d'entrée (figure 3.13). Si, toutefois, le véhicule précédent n'existe pas, nous attribuons sa vitesse et son TTC à un grand nombre. S'il n'y a pas de voie disponible d'un côté du véhicule cible, nous attribuons les caractéristiques des deux véhicules de cette voie inexistante à un grand nombre (10000). Un tel réglage peut garantir la faisabilité des résultats prévus.

Une manœuvre  $E$ , dans notre cas une étape d'un changement de voie ou d'un maintien de voie, est composé d'une séquence de mesures d'une durée de  $e_{size}$ . Pour changer de voie, le conducteur prend en moyenne 4 secondes avant de traverser la voie et 5 secondes pour ajuster la position du véhicule sur la nouvelle voie. Pour classer les manœuvres, la taille de la fenêtre temporelle  $w_{size}$  devrait se généraliser pour toutes les manœuvres d'intérêt. Celles que nous voulons classer se situent dans la fourchette de quelques secondes. Ainsi, nous avons essayé des tailles de fenêtre entre 3 et 5 secondes. Dans certains cas, la fenêtre de temps ne contient pas suffisamment d'informations pour assurer la continuité entre les étapes de changement de voie. Pour surmonter cette limitation, nous proposons de considérer les transitions entre les classes de conduite comme un chevauchement de 0.5 seconde (Figure 3.11) [Benterki et al., 2020].

### 3.6.3 La conception du modèle de classification

Pour l'analyse de séries temporelles ou d'autres données ayant une relation séquentielle, les réseaux de neurones profonds (DNN<sup>4</sup>) se sont avérés des modèles efficaces [Wang et al., 2017]. Dans cette partie, comme le montre le tableau 3.10, nous propo-

4. Deep Neural Network

	<i>Attributes</i>	<i>Description</i>
<b>Véhicule cible</b>	xVelocity	Vitesse longitudinale.
	yVelocity	Vitesse latérale.
	Yaw	Angle entre l'axe central du véhicule et la direction dans laquelle le véhicule se déplace.
	YawRate	Vitesse angulaire en lacet
	xAcceleration	Accélération longitudinale
	yAcceleration	Accélération latérale
	TTC	Le temps avant la collision
<b>Véhicule précédent</b>	PrecedingXVelocity	Vitesse longitudinale
	RightTTC	Temps de collision entre le véhicule de droite précédent et le véhicule de droite suivant
<b>Véhicules de la voie de droite</b>	RightTHW	Progression dans le temps entre les véhicules de droite qui précèdent et celles qui suivent
	RightDHW	Distance entre le véhicule de droite précédent et le véhicule de droite suivant
	LeftTTC	Temps de collision entre le véhicule de gauche précédent et le véhicule de gauche suivant
<b>Véhicules de la voie de gauche</b>	LeftTHW	Progression dans le temps entre les véhicules de gauche qui précèdent et ceux qui suivent
	LeftDHW	Distance entre le véhicule de gauche précédent et le véhicule de gauche suivant

TABEAU 3.9 – Attributs sélectionnés pour l'apprentissage

sons une architecture profonde pour extraire les informations les plus importantes des données. Le problème de prédiction des séries temporelles est reformulé de manière à pouvoir apprendre les relations temporelles entre les attributs et les classes. Le modèle prend la séquence d'entrée à chaque pas de temps et prédit l'intention du conducteur sous forme de classe. Les entrées de ce modèle sont les données de manœuvre continues du véhicule extraites du jeux de données HighD (Tableau 3.9).

Réseau de neurone profond	
Nombre des couches cachées	{4}
Nombre des noeuds dans chaque couche	{1024, 512, 512, 256}
La fonction d'activation dans chaque couche cachée	Tanh
La fonction d'activation dans la couche de sortie	Softmax
Fonction de coût	Categorical Cross-entropy
Optimisateur	Adam

TABLEAU 3.10 – Le paramétrage du réseau de neurone proposé

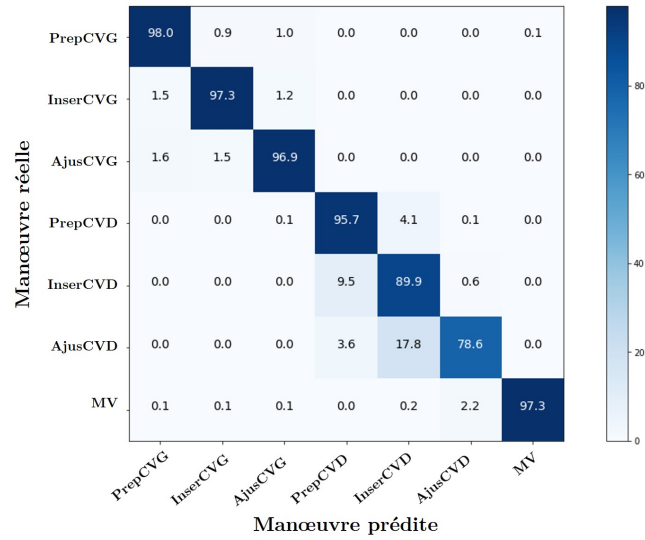
La structure du modèle utilisé est présentée dans le tableau 3.10, où le réseau se compose d'une couche d'entrée, de quatre couches cachées et d'une couche de sortie. Chaque nœud de la couche cachée a une fonction d'activation de Tanh, et la couche de sortie a une fonction d'activation de type Softmax, qui définit la sortie de ce nœud en fonction de sa propre entrée.

### 3.6.4 Implémentation

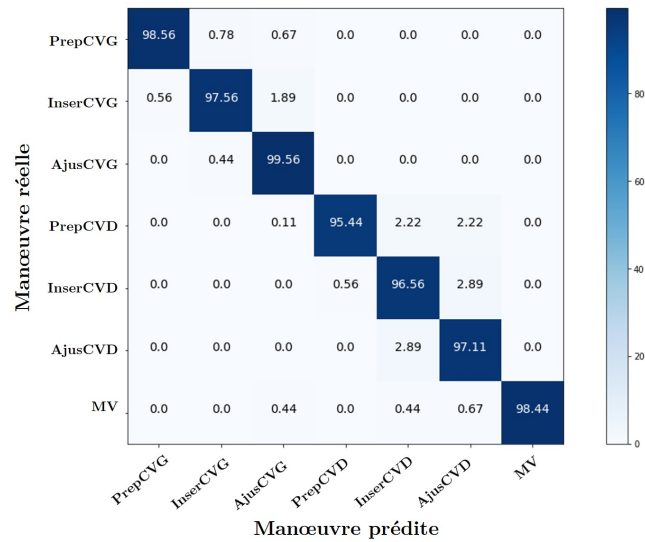
Le modèle de détection des intentions du conducteur a été entraîné en utilisant les données d'entrée de l'ensemble de données HighD. Le modèle DNN a été entraîné à classer les sept intentions du conducteur étiquetées comme suit : {PrepCVG, InserCVG, AdjusCVG, PrepCVD, InserCVD, AdjusCVD, MV} en utilisant 2000 trajectoires pour l'entraînement et 1000 trajectoires pour le test. La performance de classification du modèle peut être améliorée en sélectionnant la combinaison optimale des signaux d'entrée de Tableau 3.9, plutôt que d'utiliser toutes les données d'entrée disponibles.

#### 3.6.4.a Considérations relatives à l'entraînement

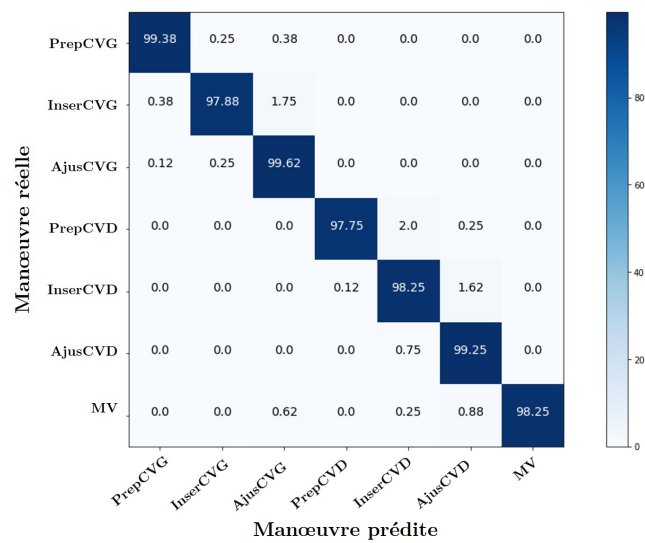
- Nous avons utilisé l'aire sous la courbe ROC (AUC) [Bradley, 1997] comme mesure d'évaluation pour les modèles entraînés, car il s'agit d'un compromis entre le taux de faux positifs et le taux de vrais positifs en considérant tous les seuils possibles pour la classification.
- Nous avons utilisé la cross-entropie catégorielle comme fonction de coût pour la classification des manœuvres.
- Pour utiliser les DNN, nous avons normalisé nos données en fonction des valeurs minimales et maximales des signaux. La couche d'entrée est alimentée par des séquences de 14 mesures.
- Trois tailles de fenêtre sont testées,  $w_{size} \in \{30, 40, 50\}$  avec un chevauchement de 0.5 secondes entre les fenêtres des phases de changement de voie. La meilleure taille de fenêtre est choisie en fonction des résultats de la matrice de confusion.



(a)  $w_{size} = 30$  seconds



(b)  $w_{size} = 40$  seconds



(c)  $w_{size} = 50$  seconds

FIGURE 3.14 – Les matrices de confusion de classification



### 3.6.5 Résultats et discussion

Dans cette partie, différentes techniques d'évaluation sont présentées pour évaluer la qualité du modèle et les résultats obtenus sont discutés.

L'évaluation comporte trois étapes :

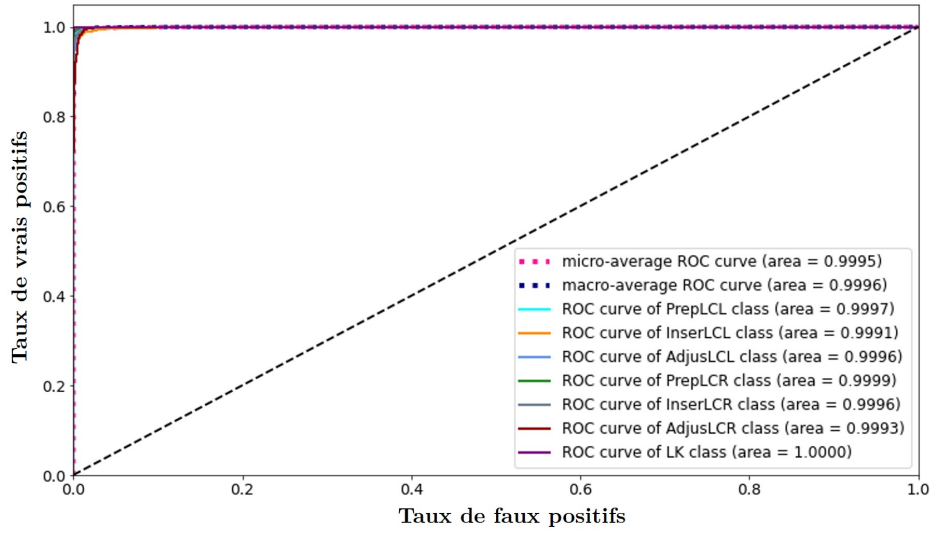
- Pour toutes les tailles de fenêtre proposées, nous créons la matrice de confusion de la classification (figure 3.14). Chaque matrice représente la performance de prédiction du classificateur. Selon les résultats de la matrice de confusion, nous choisissons la meilleure taille de fenêtre.
- Après avoir choisi la meilleure taille de fenêtre, nous traçons la courbe ROC (Receiver Operating Characteristic) pour toutes les classes afin de visualiser les performances de classification pour chaque classe.
- Nous utilisons SHAP (SHapley Additive exPlanations) [Lundberg, Lee, 2017] pour comprendre le modèle de prédiction et l'importance des attributs. SHAP attribue à chaque attributs une valeur d'importance pour une prédiction de manœuvre.

La figure 3.14 illustre la matrice de confusion des modèles entraînés pour trois tailles de fenêtre. Pour calculer les valeurs de la matrice de confusion, nous avons utilisé un jeu de tests de 1000 trajectoires. Il est clair que les trois modèles atteignent une précision de classification élevée, mais le troisième modèle ( $w_{size}= 50$  secondes) est plus performant que les autres, notamment pour la prédiction de l'étape de préparation du changement de voie. Nous avons donc adopté ce modèle pour le reste de l'évaluation.

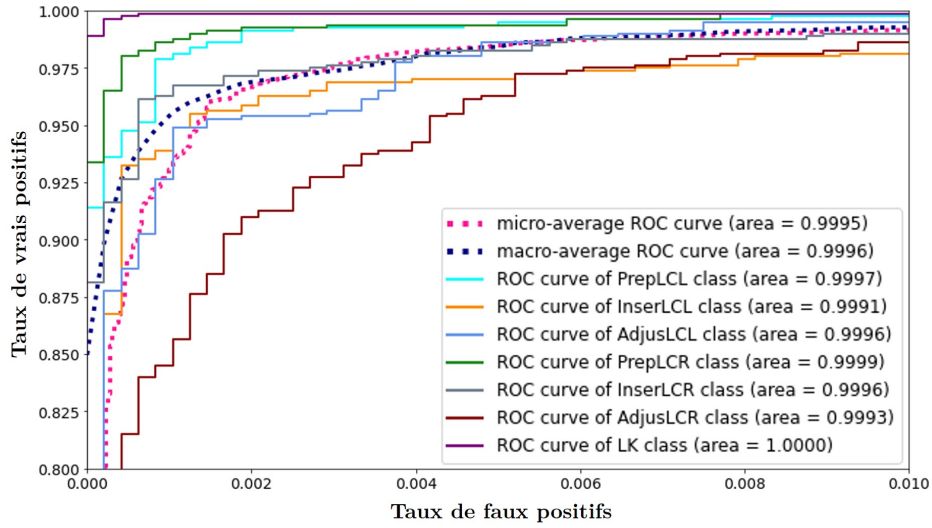
Les courbes ROC du modèle sélectionné sont visualisées sur la figure 3.15. Les courbes ont été créées en traçant le taux de vrais positifs contre le taux de faux positifs de toutes les classes d'intérêt. L'aire sous la courbe ROC (AUC) peut être utilisée comme une mesure agrégée de la performance du classificateur. Le vrai positif représente une prédiction correcte du maintien de la voie, du changement de voie à gauche ou à droite avec l'étape exacte, le faux positif indique une mauvaise prédiction de la direction du changement de voie et de l'étape. D'après la figure 3.15 et des valeurs AUC, nous observons que notre méthode a des performances de prédiction élevées pour les manœuvres de changement de voie.

SHAP est une méthode pour expliquer comment les modèles font des prédictions [Lundberg, Lee, 2017]. L'objectif de SHAP est d'expliquer la prédiction d'une instance en calculant la contribution de chaque caractéristique à la prédiction. La figure 3.16 montre le graphe récapitulatif du SHAP qui ordonne les attributs en fonction de leur importance pour détecter les manœuvres. Nous pouvons voir que l'accélération latérale et la vitesse longitudinale du véhicule cible sont les attributs les plus importants du modèle. Plus précisément, le DHW et le THD de la voie de gauche ont le plus grand impact sur le modèle. En terme de voie, on note que les attributs de la voie de gauche ont plus d'importance que celles de la voie de droite. Enfin, la TTC est l'attribut le moins significative.

Nous avons testé le meilleur modèle avec 2 trajectoires qui ne font pas partie du jeu d'apprentissage. Pour cela, nous avons utilisé deux manœuvres différentes (CVG, CVD) de deux véhicules différents. Pour chaque pas de temps, le classifieur prédit la classe des 50 échantillons précédents (5 secondes). Comme nous le voyons sur la figure 3.17, le modèle est capable de détecter les bonnes manœuvres sans fausse détection. De plus, le modèle est capable de détecter rapidement les changements de voie (1,8 seconde pour LCL et 1,5 seconde pour LCR) avant le franchissement de la nouvelle voie.



(a) Courbe ROC de classification ( $w_{size} = 50$  seconds)



(b) Courbe ROC de classification (Zoom sur le coin supérieur gauche)

FIGURE 3.15 – Évaluation de la classification des manœuvres à l’aide de la courbe ROC

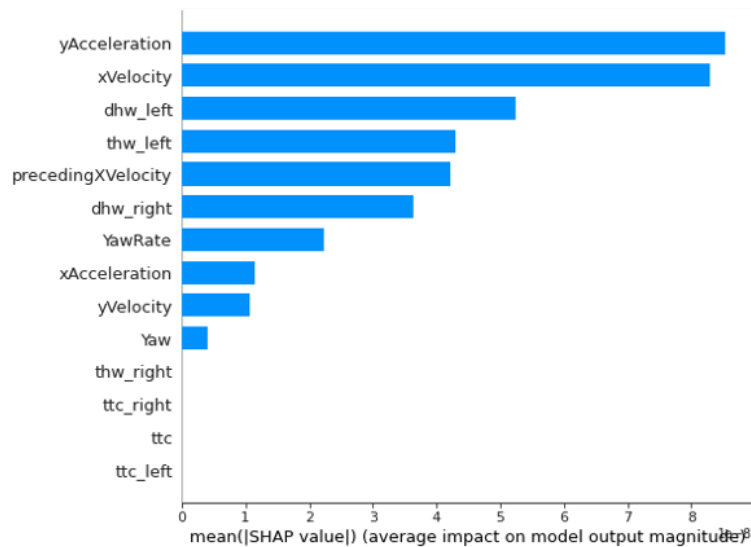
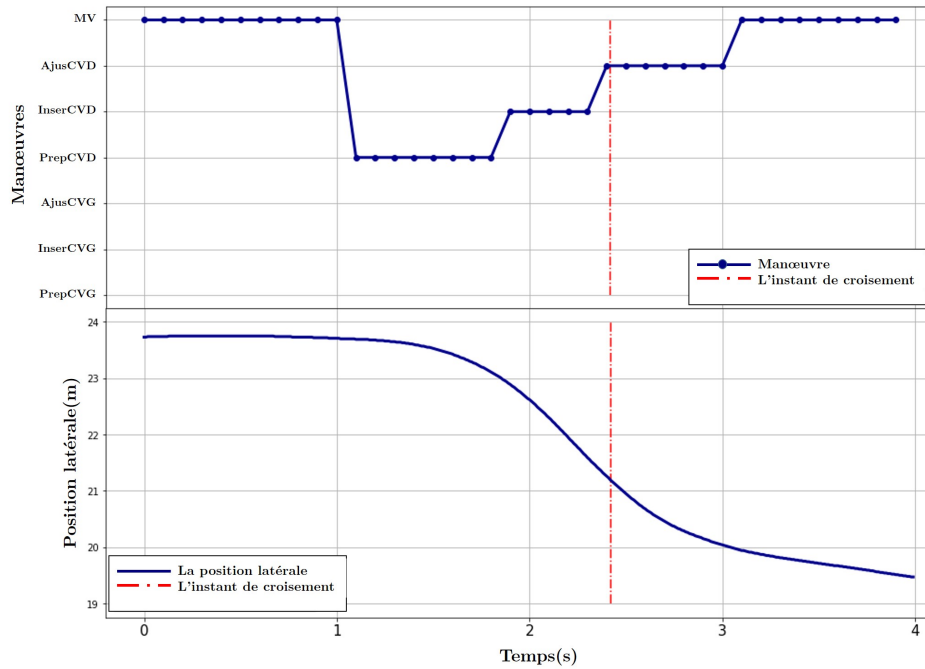
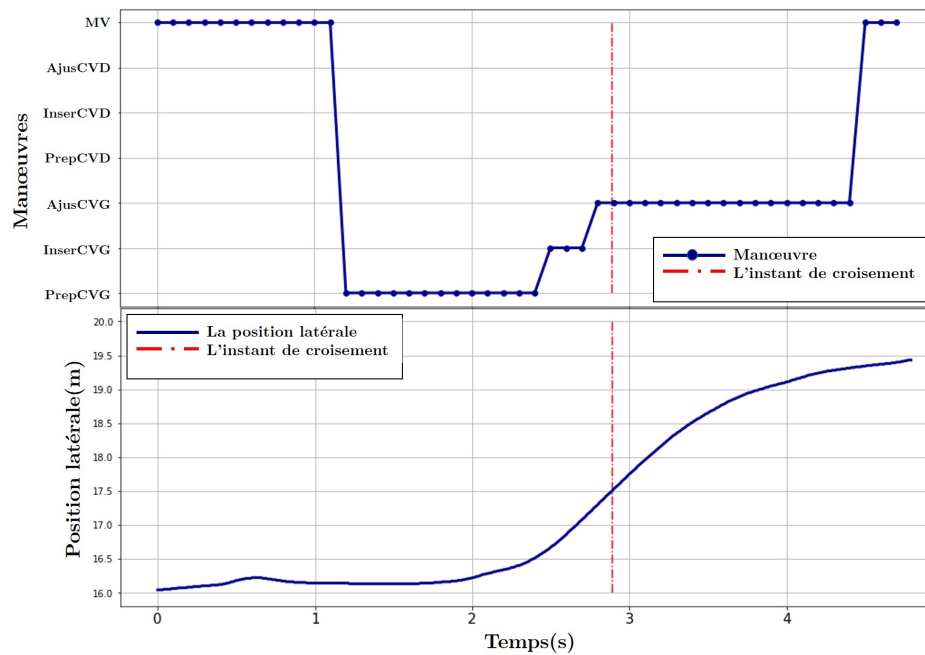


FIGURE 3.16 – Graphique de synthèse SHAP (analyse de l’importance des attributs)



(a) Trajectoire du changement de voie à gauche



(b) Trajectoire du changement de voie à droite

FIGURE 3.17 – Test du modèle en utilisant deux trajectoires ( $w_{size} = 50$  seconds)

## 3.7 Reconnaissance du style de conduite

Dans cette section, on se concentre sur la reconnaissance des styles de conduite. Avant d'aborder les approches, il est nécessaire de définir le style de conduite. Il n'y a toujours pas de définition unique du style de conduite dans la littérature. Le concept implique souvent d'autres constructions telles que l'état du conducteur, la condition du conducteur et le comportement du conducteur en général. Parmi les définitions les plus utilisées, citons celle de [Elander et al., 1993] : le style concerne la façon dont les individus choisissent de conduire, ou les habitudes de conduite qui se sont établies au fil des années. La définition adoptée par [Lundberg, Lee, 2017] est : le style de conduite concerne les habitudes de conduite individuelles, c'est-à-dire la manière dont un conducteur choisit de conduire. Ils sont très proches les uns des autres et s'accordent à dire que le style de conduite est la manière habituelle dont les conducteurs choisissent de conduire ou les compétences de conduite qu'ils ont développées au fil du temps.

Dans notre travail, nous adaptons ces définitions. Ainsi, un style de conduite doit être habituel et varier d'un individu à l'autre, et non les conditions ou les environnements de conduite. La motivation de ce travail est de déterminer s'il est possible de regrouper des conducteurs à partir d'un ensemble de données de conduite enregistrées. La majorité des jeux de données véhiculaires existants ne fournissent pas une information sur le style de conduite. En d'autre terme, les données de conduite ne sont pas labellisées pour pouvoir classer les différents types des conducteurs. Dans ce cas, nous faisons appel aux techniques d'apprentissage non supervisé (le clustering) afin de regrouper les conducteurs selon leurs données de conduite. Il est possible de classer le style de conduite en plusieurs catégories. Dans cette thèse et pour des raisons de simplification, on ne considère que deux styles de conduite : conduite calme et conduite agressive. Le meilleur scénario pour identifier le style de conduite d'une personne est la situation de changement de voie. D'après la manière de la réalisation d'un changement de voie, on peut conclure s'il s'agit d'un conducteur calme ou un conducteur agressif.

La question cruciale est la sélection des attributs à utiliser pour le clustering. Selon le contexte de conduite, différentes attributs peuvent s'avérer pertinents. Il s'agit des attributs qui sont largement et systématiquement utilisés dans les méthodes de reconnaissance de situation de conduite, plus particulièrement les changements de voie.

### 3.7.1 Le clustering des données de changement de voie

Pour identifier le style de conduite, les algorithmes de clustering K-means et le clustering spectral sont utilisés. Les algorithmes K-means et le clustering spectral fournissent un moyen simple et facile de classer un ensemble de données, où le nombre de clusters  $k$ , qui devrait être produit, est établi à l'avance. La principale caractéristique de ces algorithmes, par rapport aux autres méthodes de clustering, est que le nombre de clusters est fixé avant le clustering. Cela peut être considéré comme un avantage ou un inconvénient. La définition préalable du nombre de clusters à produire empêche la méthode K-means d'introduire de nouveaux clusters en cas de point de données anormale. Au lieu de cela, le point de données anormal est trié vers le cluster le plus proche. Le principal inconvénient d'avoir fixé le nombre de clusters à l'avance est qu'il peut être difficile de savoir combien de clusters un jeu de données peut contenir. L'utilisation d'un  $k$  inapproprié peut conduire à de mauvais résultats [ELIASSON, Rosen, 2013]. Dans notre cas, le but de cette étude est d'identifier les conducteurs agressifs et les conducteurs calmes, donc le nombre de clusters est fixé à l'avance ( $k = 2$ ). L'idée générale du clustering est de regrouper les

profils de conduite similaires au sein d'un même cluster. Dans la majorité des approches actuellement disponibles, deux individus sont considérés comme similaires lorsqu'ils ont des données de conduite proches à chaque point temporel.

Afin de bien affecter chaque profil de conducteur dans le groupe approprié, nous devons sélectionner soigneusement les attributs. Dans cette étude, les règles pour le choix des attributs sont énumérées ci-dessous :

- **Règle 1** : il n'y a pas de fortes corrélations entre deux attributs.
- **Règle 2** : la différence entre deux attributs ne doit pas être de l'ordre de magnitude.

La règle 1 est considérée pour éviter la considération répétée d'un attribut, et la règle 2 est de s'assurer que tous les attributs sélectionnés peuvent jouer des rôles appropriés sur le clustering.

Les données NGSIM fournissent divers attributs, notamment la vitesse, l'accélération, l'angle de lacet et la distance par rapport au marquage de la voie. Dans cette étude, la vitesse latérale et l'angle de lacet sont sélectionnés pour le clustering, car ils peuvent tous deux bien représenter le style de conduite pendant un changement de voie. L'accélération et la distance par rapport au marquage de la voie, qui sont fortement liées à la vitesse latérale, sont négligées conformément aux règles de choix des attributs présentées précédemment.

En outre, il faut noter que les unités de la vitesse latérale et l'angle de lacet sont différentes. Afin d'éliminer la différence de magnitude entre les variables, une méthode "élimination de la dimension" peut être utilisée pour les variables. Cependant, la méthode de normalisation min-max est adoptée pour rendre les variables sans dimension. Cette méthode est une méthode de régularisation linéaire qui a été largement appliquée dans le traitement de données. Ensuite, les données normalisées peuvent être calculées par la formule suivante [Joel, 2015] :

$$x'_{p_{ij}} = \frac{x_{p_{ij}} - \min\{x_{p_i}\}}{\max\{x_{p_i}\} - \min\{x_{p_i}\}} \quad (3.2)$$

où :  $x_{p_{ij}}$  est la donnée sélectionnée dans l'ensemble de données  $x_{p_i}$ ,  
 $j$  représente l'indice d'un point de données,  
 et  $\min\{x_{p_i}\}$  et  $\max\{x_{p_i}\}$ , respectivement, représentent la valeur minimale et maximale de l'ensemble de données  $x_{p_i}$ .

### 3.7.2 Évaluation des modèles de clustering

Dans cette partie, nous évaluons l'efficacité de nos algorithmes de clustering des styles de conduite. Nous adaptons deux méthodes pour évaluer la performance de chaque méthode (K-means et le clustering spectral) : évaluation par coefficient de silhouette et évaluation visuelle.

#### 3.7.2.a Le coefficient de silhouette

Le coefficient silhouette (CS) est obtenu en contrastant la distance moyenne des objets d'un même cluster avec la distance moyenne des objets des autres clusters. En d'autres termes, les clusters présentant une forte similarité au sein d'un même cluster et une faible similarité entre les clusters auront des valeurs de CS plus élevées [Rousseeuw, 1987].

Le CS pour la configuration de cluster est obtenu à partir des coefficients de silhouette individuels (CSI). L'CSI d'un point de données  $X_i$  est noté  $sil_i$  défini comme suit :

$$sil_i = \frac{b(X_i) - a(X_i)}{\max\{a(X_i), b(X_i)\}} \quad (3.3)$$

où  $a(X_i)$  est la distance moyenne entre le point  $X_i$  et toutes les autres données du même cluster

et  $b(X_i)$  est la plus petite distance moyenne de  $X_i$  par rapport aux points des autres clusters.

Le SC est alors donné par :

$$SC = \frac{\sum_{k=1}^K \frac{\sum_{i=1}^{|C_k|} sil_i}{|C_k|}}{K} \quad (3.4)$$

Selon la définition, la valeur de la silhouette est une mesure de la similarité d'un objet avec son propre cluster par rapport aux autres clusters. La valeur est comprise entre [-1,1]. Pour un profil de conducteur, une grande valeur de silhouette indique que le profil s'adapte bien à son propre cluster et mal aux autres clusters. Par conséquent, la configuration de clustering est appropriée si la plupart des profils ont une grande valeur de silhouette.

Le tableau 3.11 présente les scores moyens de la silhouette de chaque algorithme de clustering. Sur le jeu des données NGSIM, les conducteurs sont répartis en deux groupes à l'aide des algorithmes K-Means et clustering spectral. Le tableau 3.11 montre la valeur de silhouette en fonction de la direction de changement de voie (droite/gauche) et de l'attribut. On constate que les valeurs de silhouette de K-means sont élevées par rapport au clustering spectral, et que les valeurs de silhouette maximales peuvent atteindre 0,35. En particulier, dans le cas de changement de voie à droite, les valeurs de Silhouette sont supérieures à 0,3. Les résultats indiquent que la plupart des conducteurs des mêmes clusters ont un comportement de conduite similaire. Ils prouvent également que l'algorithme K-Means est approprié et que la classification des conducteurs est satisfaisante.

En outre, on peut voir que toutes les valeurs de silhouette sont supérieures à 0, ce qui indique que les résultats du clustering sont raisonnables. Néanmoins, certaines valeurs de silhouette sont inférieures à 0. Toutes les valeurs de silhouette de K-means sont supérieure à 0.2, ce qui montre que le K-means représente une meilleure performance de regroupement par rapport le clustering spectral.

		Clustering spectral	K-means
Changement de voie à gauche	Vitesse latérale	0.13	0.22
	Angle de lacet	0.01	0.20
Changement de voie à droite	Vitesse latérale	-0.008	0.32
	Angle de lacet	0.20	0.35

TABLEAU 3.11 – Scores moyens de la silhouette pour 3 clusters

### 3.7.2.b Évaluation visuelle

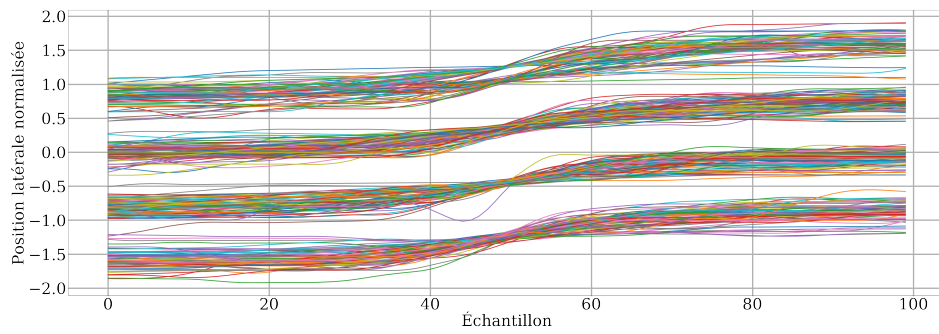
Pour mieux comprendre les résultats de clustering, on propose une représentation visuelle des changements de voie (figure 3.18) des profils de vitesse latérale (figure 3.19) et de l'angle de lacet (figure 3.20), ainsi que les profils regroupés selon les modèles de clustering (figures 3.21, 3.22,3.23,3.24). La figure 3.18 présente les trajectoires (positions

latérales normalisées) de 300 véhicules pendant des changements de voie dans les deux directions avant le clustering.

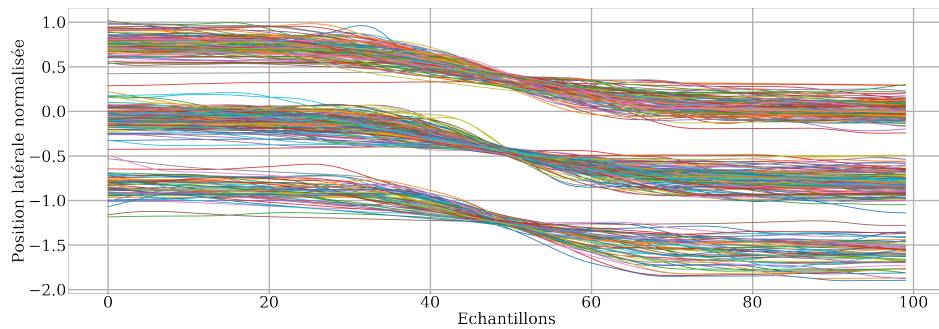
Les trajectoires de changement de voie peuvent être regroupé à l'aide des données de vitesse latérale et d'angle de lacet. La figure 3.18 indique que les trajectoires de changement de voie ainsi que les profils de vitesse latérale et d'angle de lacet sont différentes d'un conducteur à l'autre. Le résultat du clustering de la vitesse latérale en utilisant l'algorithme de clustering K-means et clustering spectral avec un nombre de cluster fixe  $k = 2$  sont illustrés à la figure 3.21 et figure 3.22. Les profils sont divisées en deux classes, où le rouge représente le groupe des conducteurs agressifs et le vert représentent le groupe des conducteurs calmes. Nous avons utilisé la représentation en boîte à moustaches pour figurer le profil essentiel de chaque cluster. Comme on peut le voir sur les deux figures 3.21 et 3.22, les vitesses latérales des conducteurs agressifs pendant un changement de voie sont plus importantes par rapport les conducteurs calme. On remarque aussi qu'un conducteur calme commence son changement de voie plus à l'avance qu'un conducteur agressif. L'algorithme de clustering K-means a les meilleures performances de clustering. Il est très clair sur la représentation par boîte à moustaches que les profils de vitesse latérale de chaque cluster sont bien séparé et la distance entre les deux groupes est plus importante par rapport à celle du clustering spectral.

Afin de vérifier davantage l'efficacité de l'algorithme K-means, nous avons également comparé le résultat du clustering des profils de l'angle de lacet. Les représentations visuelles des résultats de clustering de l'angle de lacet sont présentées dans les figures figures 3.23 et 3.24. Les profils en rouge représente les conducteurs agressifs, alors que les profils en bleu représente les conducteurs calmes. Comme on peut le voir clairement l'algorithme de clustering K-means a toujours de meilleures performances de clustering que le clustering spectral. On remarque que les profils d'angle de lacet sont bien séparés, aussi que les changements des conducteurs calmes sont moins importants et la manœuvre de changement de voie commence plus tôt que les conducteurs agressif.

En conclusion, il est évident que les performances de l'algorithme de clustering K-means sont meilleures que celles des autres algorithmes. Les résultats des deux attributs (vitesse latérale et ange de lacet) vérifient l'efficacité de la méthode.

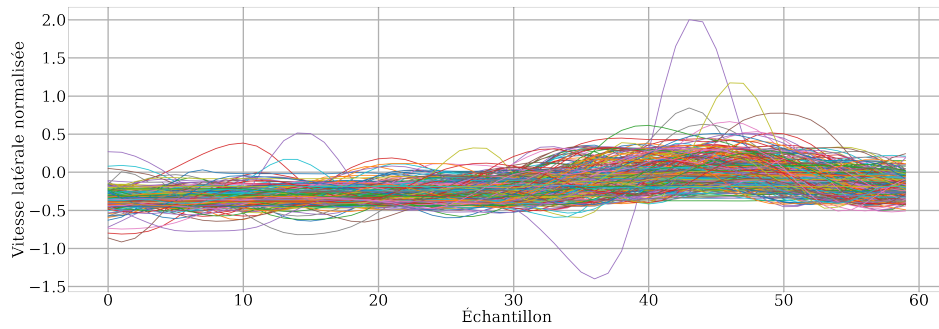


(a) Les trajectoires des changements de voie à gauche (position latérale)

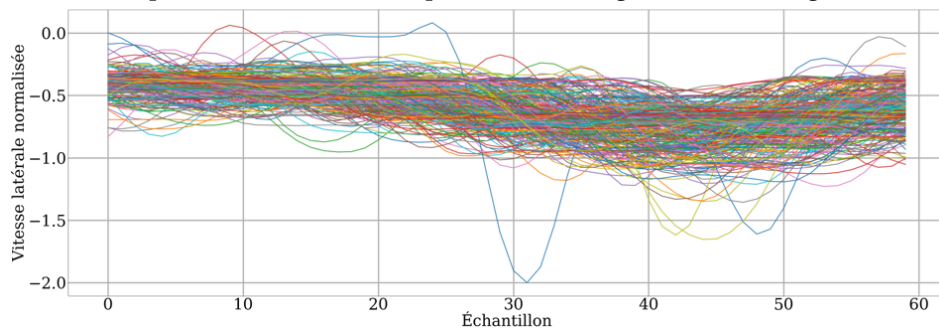


(b) Les trajectoires des changements de voie à droite (position latérale)

FIGURE 3.18 – Les positions latérales des changements de voie (jeu de données NGSIM)



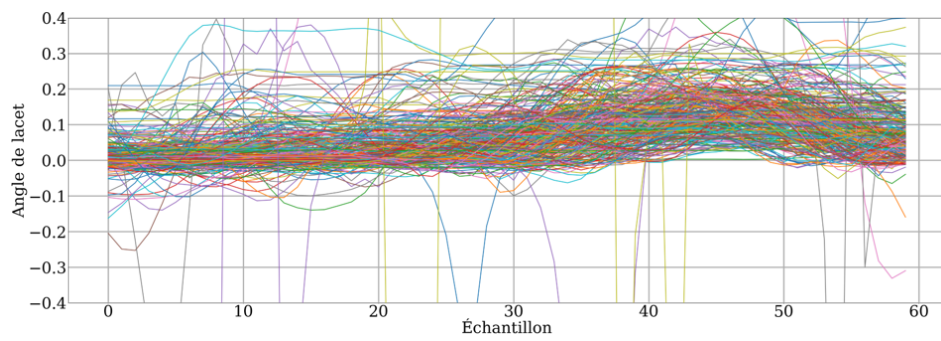
(a) Les profils de vitesse latérale pendant un changement de voie à gauche



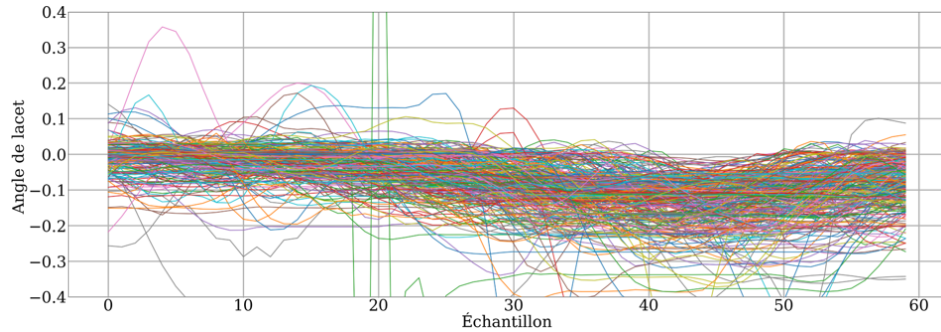
(b) Les profils de vitesse latérale pendant un changement de voie à droite

FIGURE 3.19 – Les profils de vitesse latérale pendant un changement de voie (jeu de données NGSIM)



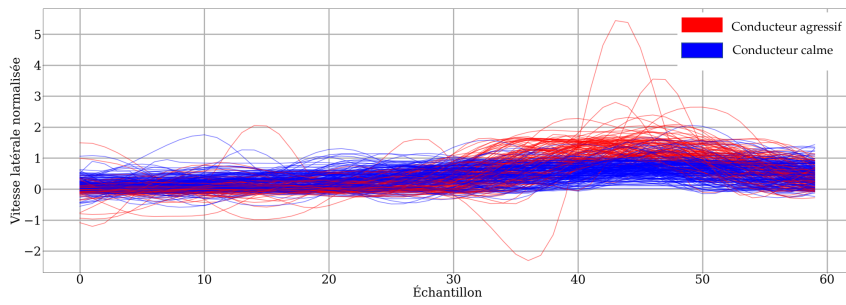


(a) Les profils d'angle de lacet pendant un changement de voie à gauche (Angle de lacet)

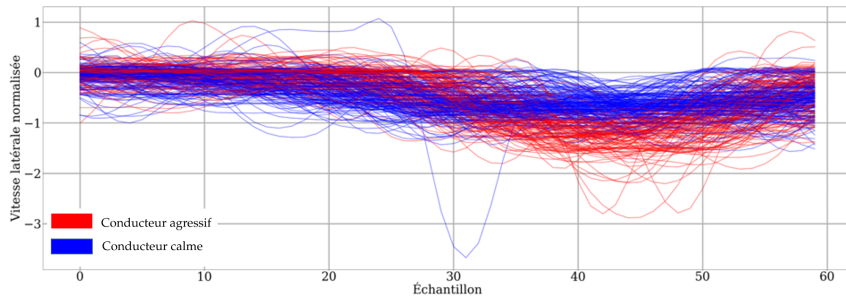


(b) Les profils d'angle de lacet pendant un changement de voie à droite (Angle de lacet)

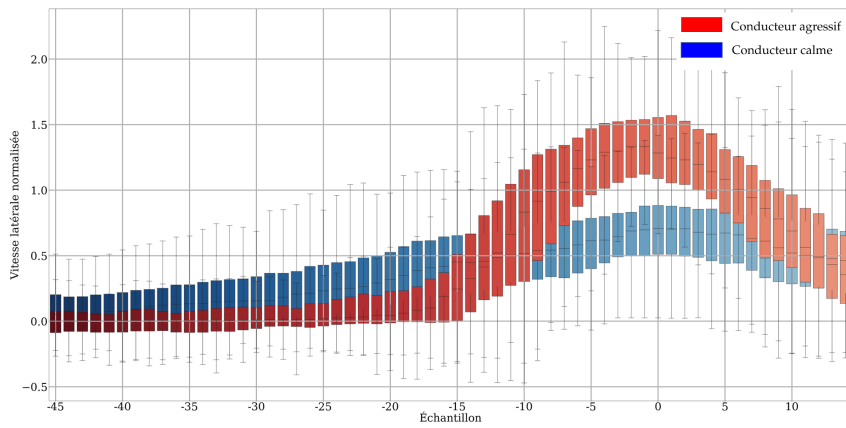
FIGURE 3.20 – Les profils d'angle de lacet pendant un changement de voie (jeu de données NGSIM)



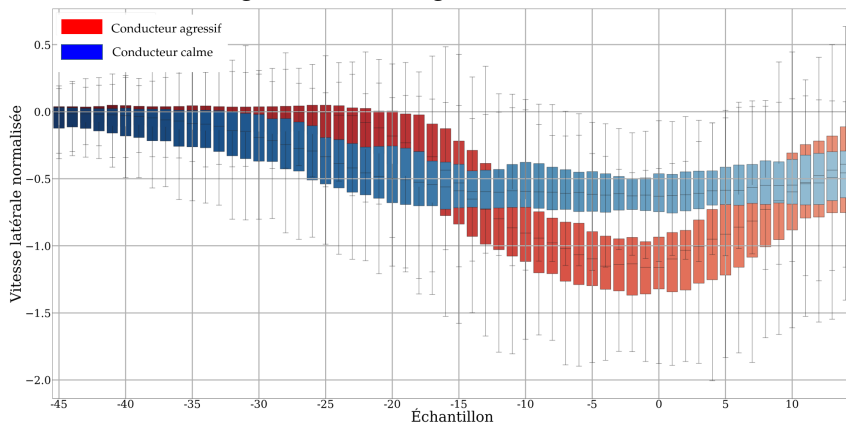
(a) Les résultats de clustering de la vitesse latérale en utilisant K-means (Changement de voie à gauche)



(b) Les résultats de clustering de la vitesse latérale en utilisant K-means (Changement de voie à droite)

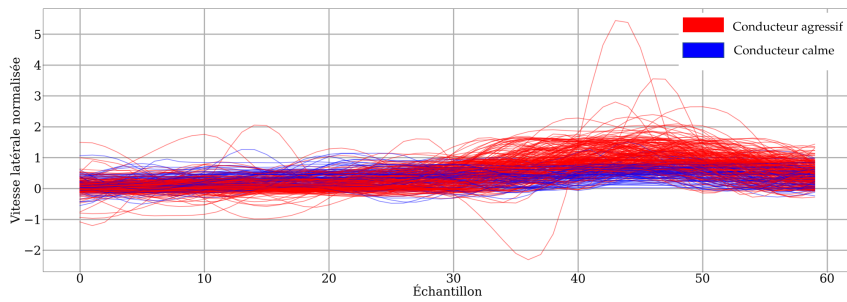


(c) Les résultats de clustering de la vitesse latérale en utilisant K-means représentés en boîte à moustache (Changement de voie à gauche)

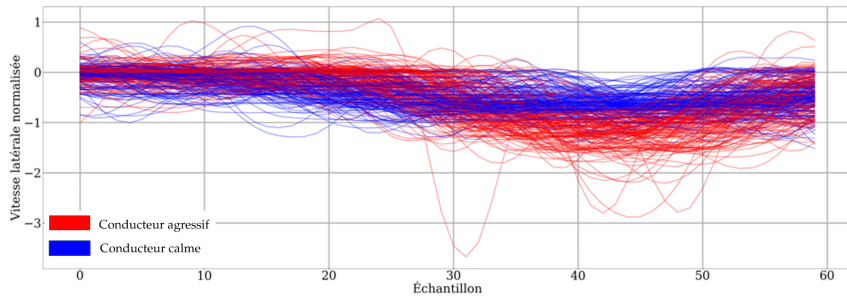


(d) Les résultats de clustering de la vitesse latérale en utilisant K-means représentés en boîte à moustache (Changement de voie à droite)

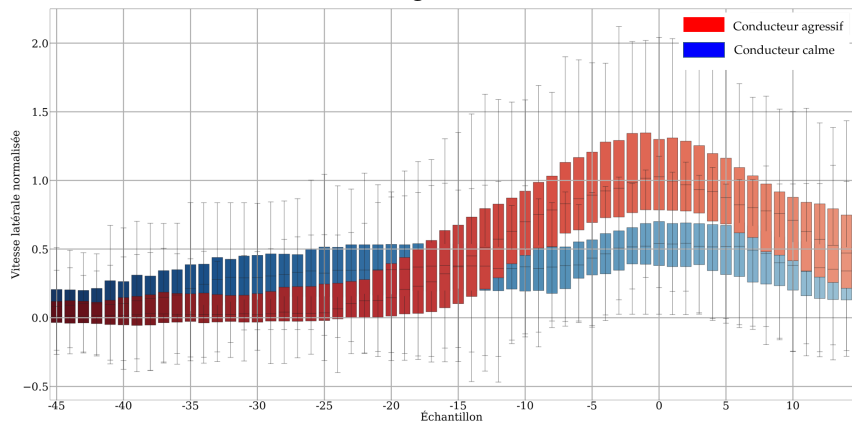
FIGURE 3.21 – Les résultats de clustering de la vitesse latérale en utilisant K-means



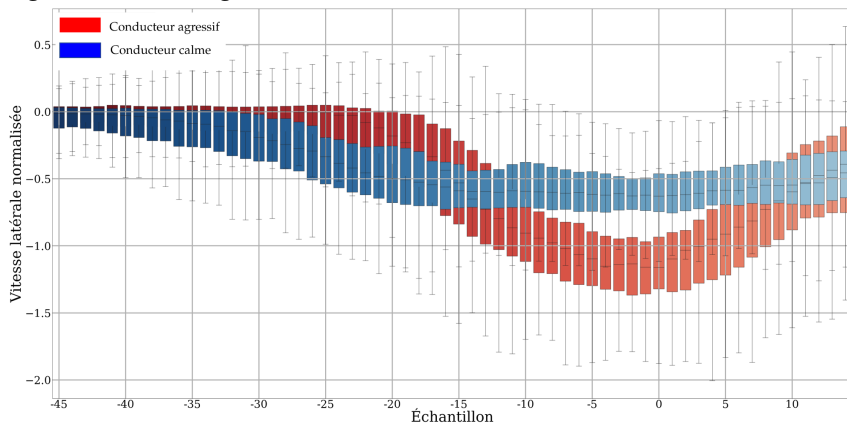
(a) Les résultats de clustering de la vitesse latérale en utilisant la clustering spectral (Changement de voie à gauche)



(b) Les résultats de clustering de la vitesse latérale en utilisant le clustering spectral représentés en boîte à moustache (Changement de voie à droite)

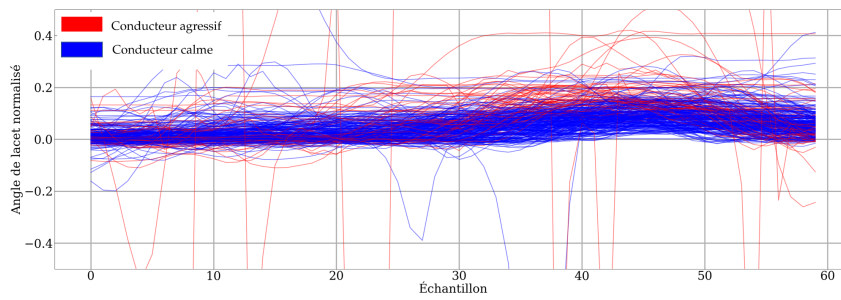


(c) Les résultats de clustering de la vitesse latérale en utilisant la clustering spectral (Changement de voie à gauche)

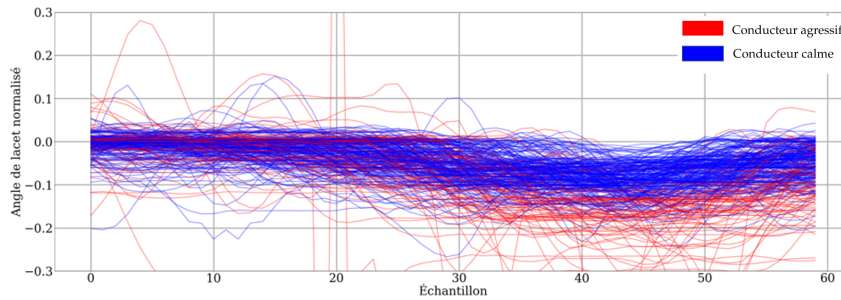


(d) Les résultats de clustering de la vitesse latérale en utilisant le clustering spectral représentés en boîte à moustache (Changement de voie à droite)

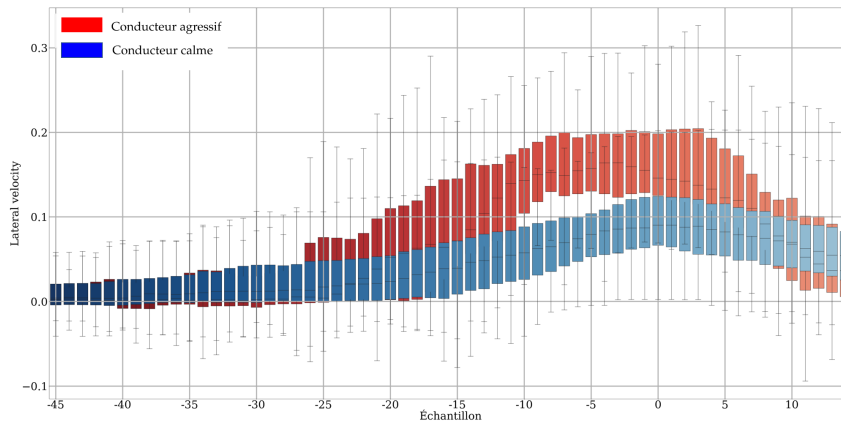
FIGURE 3.22 – Les résultats de clustering de la vitesse latérale en utilisant le clustering spectral



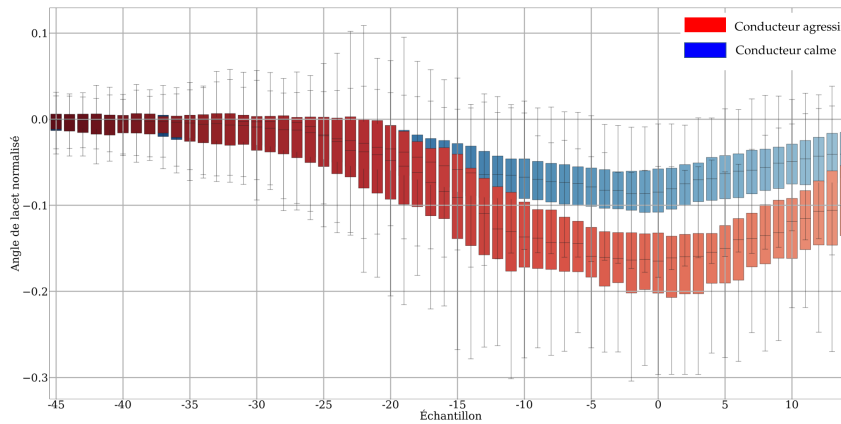
(a) Les résultats de clustering de l'angle de lacet en utilisant K-means (Changement de voie à gauche)



(b) Les résultats de clustering de l'angle de lacet en utilisant K-means (Changement de voie à droite)

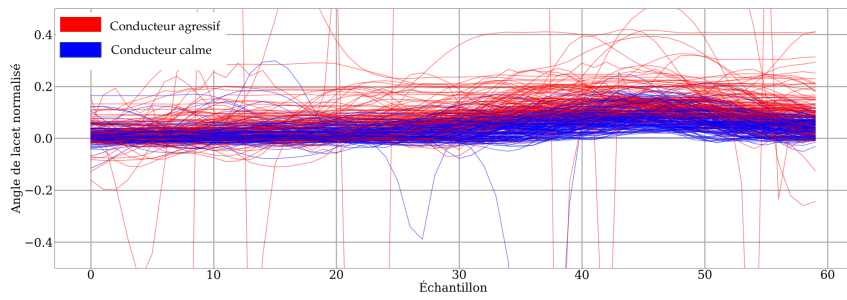


(c) Les résultats de clustering de l'angle de lacet en utilisant K-means représentés en boîte à moustache (Changement de voie à gauche)

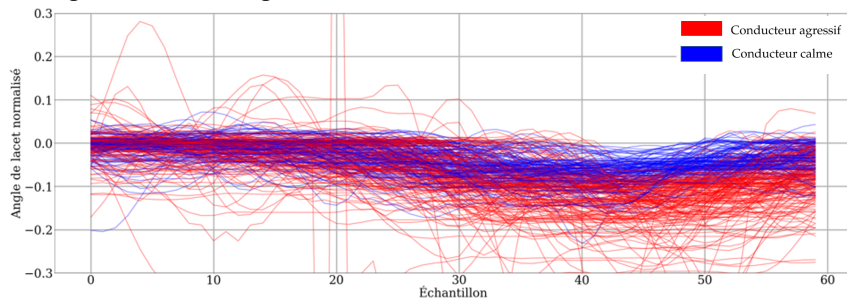


(d) Les résultats de clustering de l'angle de lacet en utilisant K-means représentés en boîte à moustache (Changement de voie à droite)

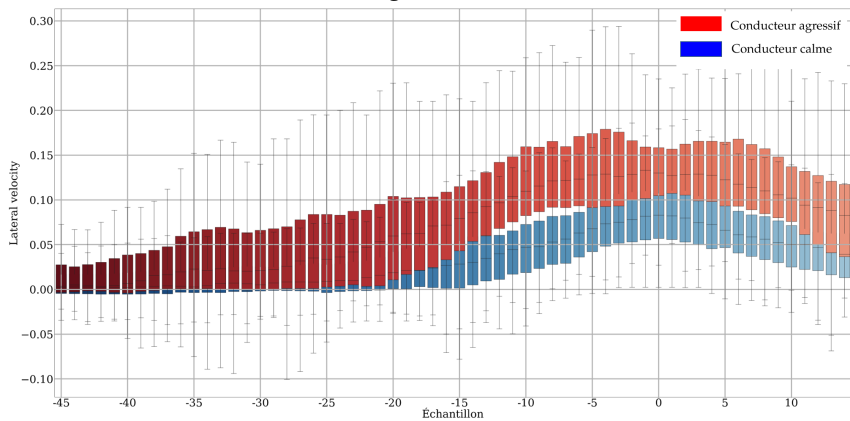
FIGURE 3.23 – Les résultats de clustering de l'angle de lacet en utilisant K-means



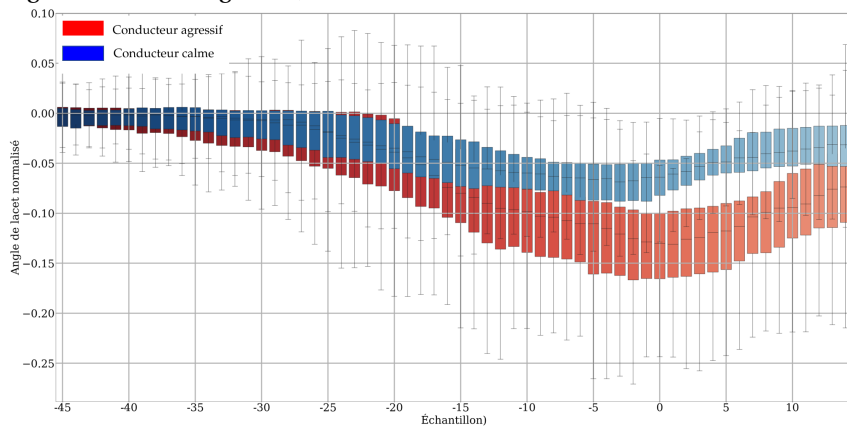
(a) Les résultats de clustering de l'angle de lacet en utilisant la clustering spectral (Changement de voie à gauche)



(b) Les résultats de clustering de l'angle de lacet en utilisant le clustering spectral représentés en boîte à moustache (Changement de voie à droite)



(c) Les résultats de clustering de l'angle de lacet en utilisant la clustering spectral (Changement de voie à gauche)



(d) Les résultats de clustering de l'angle de lacet en utilisant le clustering spectral représentés en boîte à moustache (Changement de voie à droite)

FIGURE 3.24 – Les résultats de clustering de l'angle de lacet en utilisant le clustering spectral

### 3.8 Conclusion

Dans ce chapitre, nous avons présenté les approches de reconnaissance de conduite. Nous avons divisé la reconnaissance de la situation de conduite en trois problèmes : la détection de la manœuvre, la reconnaissance de la phase de la manœuvre, et la reconnaissance du style de conduite.

La détection de la manœuvre consiste à anticiper la réalisation de manœuvre en se basant sur l'apprentissage automatique. Nous avons utilisé des approches de classification en les remettant dans un contexte de prédiction des manœuvres. Des approches basées sur une machine à vecteurs de support et des réseaux de neurones artificiel sont proposées. Nous avons ainsi comparé ces différentes approches de classification pour la reconnaissance de trois situations de conduite : changement de voie à droite, changement de voie à gauche et maintien de la voie. Les résultats obtenus montrent l'efficacité des deux approches avec une meilleure performance pour les réseaux de neurones.

La reconnaissance de la phase de la manœuvre est une approche que nous avons proposé pour suivre l'évolution de la réalisation d'un changement de voie. Contrairement aux autres approches de classification des manœuvres, l'approche proposée consiste à diviser la manœuvre de changement de voie en trois phases : préparation, insertion et ajustement, et à identifier chaque étape en utilisant un modèle basé sur des DNN. Le modèle est entraîné et testé en utilisant le jeu de données HighD, il montre des performances satisfaisantes dans la classification et l'anticipation des changements de voie.

Dans la dernière partie de ce chapitre, nous avons proposé une méthode pour la reconnaissance du style de conduite. Les algorithmes de clustering spectral et K-means sont utilisés à cette fin. Les résultats de clustering selon lesquels les conducteurs se distinguent en fonction de leurs vitesses latérales et d'angles de lacet montrent que K-means est capable de classifier les conducteurs agressifs et les conducteurs calmes avec une efficacité bien mieux que celle du clustering spectral.



# Prédiction de trajectoire et prise de décision

« Prédire l'avenir, ce n'est pas de la magie, c'est de l'intelligence artificielle. »

Dave Waters  
Professeur, université d'Oxford

## Sommaire

<b>3.1 Introduction</b> . . . . .	<b>67</b>
<b>3.2 Principe de la reconnaissance de la situation de conduite</b> . . . . .	<b>67</b>
<b>3.3 Préparation des données</b> . . . . .	<b>71</b>
3.3.1 Filtrage de données . . . . .	73
<b>3.4 Représentation de la manœuvre de changement de voie</b> . . . . .	<b>75</b>
<b>3.5 Détection de la manœuvre</b> . . . . .	<b>78</b>
3.5.1 Données d'apprentissage . . . . .	78
3.5.2 La conception du classifieur . . . . .	79
3.5.3 Résultats et discussions . . . . .	80
<b>3.6 Reconnaissance de la phase de la manœuvre</b> . . . . .	<b>85</b>
3.6.1 Données d'apprentissage . . . . .	86
3.6.2 Extraction et sélection des attributs . . . . .	87
3.6.3 La conception du modèle de classification . . . . .	88
3.6.4 Implémentation . . . . .	90
3.6.5 Résultats et discussion . . . . .	92
<b>3.7 Reconnaissance du style de conduite</b> . . . . .	<b>95</b>
3.7.1 Le clustering des données de changement de voie . . . . .	95
3.7.2 Évaluation des modèles de clustering . . . . .	96
<b>3.8 Conclusion</b> . . . . .	<b>105</b>



## 4.1 Introduction

Les véhicules autonomes entrent progressivement dans le réseau routier. Avant que les routes ne soient utilisées uniquement par des véhicules entièrement automatisés, il faut s'attendre à une période de transition au cours de laquelle des véhicules entièrement automatisés, des véhicules partiellement automatisés et des véhicules à conduite manuelle devront partager les mêmes routes. Pour interagir en toute sécurité avec ces participants au trafic, le véhicule autonome devra planifier sa trajectoire afin d'éviter toute collision potentielle. Il est donc essentiel que le véhicule autonome prédise la trajectoire future des véhicules circulant autour de lui. Les conducteurs humains font cette prédiction de manière intuitive en observant l'état des véhicules environnants ainsi que l'environnement. Les véhicules autonomes peuvent prédire la trajectoire future des véhicules environnants en utilisant des données sur leurs états passés pour connaître leur comportement, ainsi qu'en considérant les facteurs qui pourraient affecter leur trajectoire future, tels que les interactions entre les véhicules. La trajectoire passée est souvent considérée comme la principale source d'informations pour décrire le futur comportement d'un véhicule. Pour cela, il faut un modèle capable d'analyser la trajectoire passée d'un véhicule afin de prédire une trajectoire future probable.

Ce chapitre portera principalement sur la prédiction de trajectoire d'un véhicule. Dans un premier temps, nous fixons les critères de choix de notre modèle de prédiction en étudiant des différents modèles de prédiction. Ensuite, nous passerons en revue les principales notions théoriques de la prédiction de trajectoire basée sur les données. Dans la section, 4.2, nous mettons en œuvre une prédiction de trajectoire unique à partir des données extéroceptives, puis dans 4.3, on inclut une classification de la manœuvre pour essayer d'améliorer la prédiction pendant la phase de changement de voies. Nous proposons des approches basées sur les réseaux de neurones récurrents et profonds pour estimer les futures trajectoires des véhicules, entraînées et testées en utilisant des jeux de données véhiculaires. Enfin, des tests sur des données réelles collectées depuis le démonstrateur VEDECOM seront mis en œuvre pour valider ces différentes approches.

Dans la seconde partie de ce chapitre, nous proposons un algorithme de prise de décision pour un véhicule capable d'effectuer des changements de voie sûrs. Un logiciel de simulation d'environnement est nécessaire pour simuler et visualiser l'environnement et le trafic. Un modèle de réseau de neurones semble adopté pour concevoir un agent qui interagit avec des conditions de circulation complexes. La technique d'apprentissage par renforcement est utilisée pour entraîner le modèle à s'adapter à l'environnement dynamique avec des séquences infinies d'états d'action. L'agent doit apprendre la politique optimale pour choisir d'effectuer le meilleur changement de voie.

### 4.1.1 Critères de sélection du modèle de prédiction

Cette section répertorie et détaille les critères de sélection d'un modèle de prédiction. Les critères sont résumés comme suit :

- **Le modèle peut fonctionner sur des données séquentielles**

Pour la tâche de prédiction de trajectoire, l'entrée du modèle et la sortie attendue sont une séquence de données qui sont liées dans le temps. Il est donc essentiel qu'un modèle puisse établir des relations temporelles non seulement au sein des données d'entrée et des données de sortie, mais entre les deux.

- **Le modèle peut gérer les non-linéarités**

Le mouvement du véhicule peut être non linéaire lorsqu'il se déplace le long de la route, par exemple l'accélération est quadratique en position. Pour prédire un tel mouvement, les modèles doivent être capables de modéliser un comportement non linéaire.

— **Le modèle peut fonctionner sur des entités cinématiques et non cinématiques**

Le modèle ne doit pas se limiter à la prédiction de trajectoires uniquement sur la base des états cinématiques passés (position, vitesse et accélération). Au contraire, le modèle devrait également être capable d'associer la variation des états du véhicule à des caractéristiques non cinématiques telles que la courbure de la route et de les utiliser pour améliorer les performances de prédiction.

— **Le modèle peut être facilement étendu pour intégrer de nouvelles fonctionnalités**

Le modèle doit pouvoir ajouter une nouvelle fonction (dimension) aux données d'entrée pour permettre une flexibilité et une adaptation facile pour prédire les trajectoires dans des scénarios plus complexes.

#### 4.1.2 Quel modèle à utiliser pour prédire la trajectoire ?

Nous résumons les trois approches de modélisation que nous avons citées dans l'état de l'art du chapitre I :

— **Modèles déterministes**

Un modèle déterministe [Miller, Huang, 2002; Hillenbrand et al., 2006] est basé sur des équations différentielles qui décrivent la cinématique (ou la dynamique) d'un véhicule. Le rendement de ces modèles est entièrement déterminé par les équations cinématiques et les conditions initiales. Les modèles déterministes supposent que les états d'un véhicule sont parfaitement connus et que les équations différentielles représentent parfaitement le mouvement du véhicule. Ainsi, ces modèles ne tiennent pas compte les incertitudes sur l'état du véhicule, par exemple en raison du bruit du capteur. Ceci, limite l'utilité des modèles déterministes de manière significative.

— **Modèles stochastiques**

Les modèles stochastiques ne supposent pas que les états des véhicules sont déterministes, mais plutôt prendre en considération les incertitudes sur l'état du véhicule et son évolution. La simulation du bruit gaussien et la simulation de Monte Carlo sont deux modèles stochastiques courants utilisées pour la prédiction de la trajectoire [Lefèvre et al., 2014]. Ces deux approches utilisent un modèle cinématique ou dynamique pour représenter le mouvement attendu du véhicule.

Cependant, comme les modèles déterministes, les approches stochastiques supposent de proposer un modèle capable de représenter le comportement d'un véhicule. Cependant, il peut être difficile de décrire le mouvement du véhicule dans des situations complexes avec des équations de mouvement. Cela limite la complexité des modèles stochastiques à la compréhension du mouvement du véhicule par le concepteur. De plus, l'intégration de nouvelles caractéristiques telles que l'effet de la courbure de la route sur la vitesse du véhicule peut nécessiter une reconception coûteuse du modèle de mouvement.

— **Modèle basé sur les données**

Contrairement aux modèles déterministes et stochastiques, les modèles pilotés par les données [Kim et al., 2017; Laugier et al., 2011] ne sont pas basés sur des équations cinématiques. Ils apprennent plutôt à mettre en correspondance un ensemble d'entrées avec des sorties attendues, sur la base de données d'entraînement. Selon leur complexité, ils peuvent être assez puissants pour généraliser à partir de données d'entraînement limitées et apprendre des modèles difficiles à modéliser avec des équations cinématiques. En outre, ils sont plus faciles à recycler avec de nouvelles données chaque fois qu'une dimension supplémentaire doit être ajoutée, pour modéliser des scénarios de conduite plus complexes. Toutefois, ces modèles présentent deux limites : d'une part, les modèles basés sur des données ne sont pas basés sur des relations cinématiques qui peuvent produire des résultats inattendus. Deuxièmement, l'entraînement de ces modèles nécessite une quantité considérable de données, qui peuvent ne pas être facilement disponibles.

En comparant les différents modèles et en se basant sur les critères cités dans la partie précédente, dans cette thèse, nous avons sélectionné les modèles basés sur les données pour la prédiction de la trajectoire des véhicules.

### 4.1.3 Le principe de la prédiction de trajectoire basée sur les données

Avant de développer une technique de prédiction de la trajectoire, il faut supposer avoir accès aux flux de données en temps réel provenant de capteurs extéroceptifs classiques pour un véhicule autonome tels que le lidar, le radar ou la caméra, installés à bord du véhicule autonome et qu'il existe déjà un système de suivi opérationnel qui permet de détecter et de suivre les véhicules environnants en temps réel. Voici des exemples d'informations qui décrivent un véhicule : position, vitesse, accélération, angle de lacet et la vitesse angulaire. Il peut également être nécessaire de disposer de données cartographiques de la zone où l'ego-véhicule roule, c'est-à-dire la position des voies et des passages de piétons, le sens des voies et d'autres informations cartographiques pertinentes. Les trajectoires passées et futures peuvent être représentées dans une carte de système de coordonnées relative à l'ego-véhicule roule.

Le but de cette partie est de développer un système capable d'apprendre la relation entre une scène actuelle et une future trajectoire sur une horizon de prédiction fixe. Le système de prédiction effectue une correspondance entre la représentation d'état d'un véhicule et des véhicules adjacents dans un intervalle de temps entre  $t - T_k$  et  $t$ , et la future trajectoire, prédite sur une horizon de temps  $T_i$ . Cela implique l'entraînement d'un modèle d'apprentissage, qui prends comme entrée l'historique de la scène. Après une phase d'apprentissage, le modèle est utilisé pour prédire la trajectoire jusqu'à l'instant  $t + T_i$ . Ce processus est illustré dans la figure 4.1.

L'apprentissage et la prédiction sont effectuées de manière autonome en utilisant des données provenant d'enregistrements de situations de conduite réelles. Une fois que les correspondances entre la représentation de la scène et la trajectoire ont été apprises, la sortie de prédiction est un vecteur formé de l'ensemble des positions produites par modèle. Afin d'évaluer la performance du modèle, nous calculons les distances entre les positions prédites et les positions futures mesurées dans les données. Les trajectoires prédites et la vérité de terrain sont comparés.

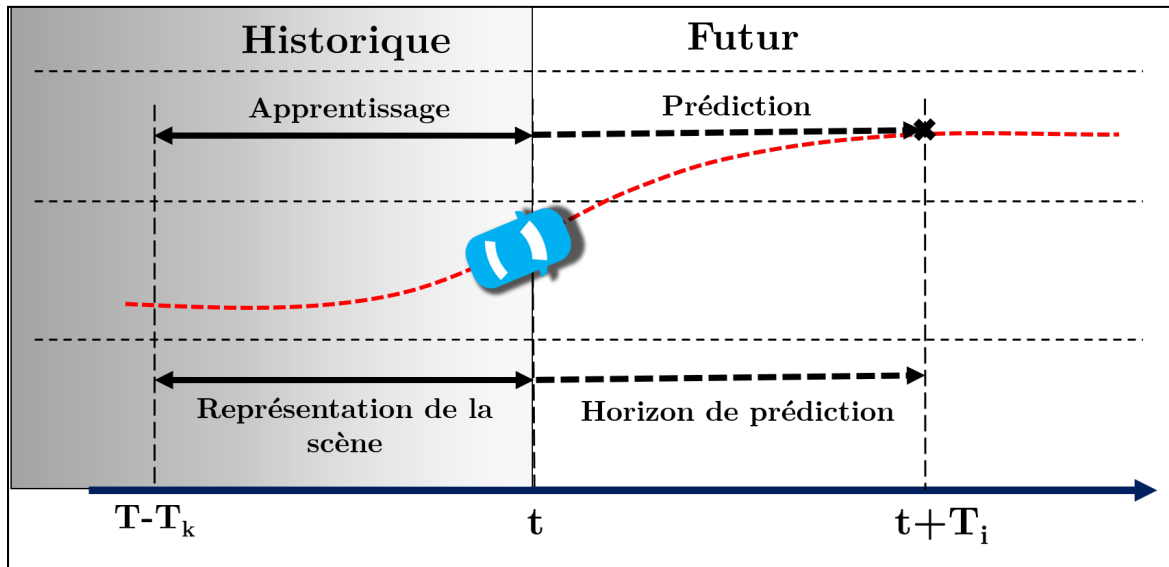


FIGURE 4.1 – Principe de prédiction de trajectoire d'un véhicule

## 4.2 Approches de prédiction de trajectoire

Le but est de développer un algorithme qui prédit la trajectoire future des usagers de la route à partir de données d'un véhicule cible ainsi que la prédiction de la manœuvre et les données des véhicules adjacents. La manœuvre estimée du véhicule est utilisée pour améliorer la précision des trajectoires prédites. L'algorithme doit également être évolutif de manière à pouvoir être utilisé pour plusieurs scénarios. Dans cette étude, nous mettons l'accent sur les scénarios de changement de voies sur autoroute.

Le changement de voie est une action fréquente faite par les véhicules sur la route et se traduit généralement par un gain certain pour le véhicule en termes de confort de conduite ou de réduction du temps de trajet. Le changement de voie peut être défini [Ghumman et al., 2008] comme un déplacement intentionnel et significatif de la position latérale d'un véhicule d'une voie à l'autre (Figure 4.2). Un changement continu de position latérale à l'intérieur d'une voie ou une incursion partielle et involontaire dans une voie adjacente par un véhicule ne constitue pas un changement de voie. Il peut s'agir d'une action volontaire délibérée, d'une procédure obligatoire dans un tronçon de route en fusion/fractionnement ou d'un acte forcé par un véhicule menant plus lentement ou un véhicule suivant plus rapidement. Le processus lui-même est peut représenter un risque et n'est effectué volontairement que lorsque le gain compense le risque.

Le processus de changement de voie peut être considéré comme une série de multiples phases distinctes : La phase de planification, la phase de préparation, la phase d'insertion et la phase d'ajustement.

Chaque fois le conducteur qui a l'intention de changer de voie trouve un écart suffisant entre les véhicules de la voie cible, la phase de planification du changement de voie commence. La phase de planification est dépourvue de toute intervention physique. Au cours de cette phase, le conducteur évalue la faisabilité du changement de voie et en évalue également les avantages par rapport aux risques encourus [Kanaris et al., 2001]. Cette évaluation est subjective et dépend de la nature du conducteur, de l'évaluation du comportement des conducteurs sur la voie cible, de la position relative et les estimations de la vitesse et sa vitesse prédite, ainsi que la vitesse longitudinale et latérale l'accélé-

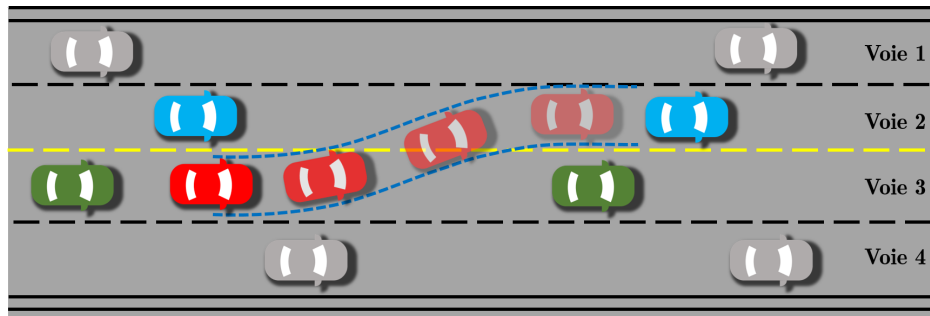


FIGURE 4.2 – Une manœuvre de changement de voie : véhicule cible et les véhicules environnants

ration/décélération requise pour un changement de voie en toute sécurité [Chaurasia, Verma, 2009].

Si le conducteur décide de changer de voie, La phase de préparation commence. Dans cette phase, le conducteur poursuit sa trajectoire dans la voie initiale mais accélère ou décélère par rapport à la position prévue des véhicules dans la voie de destination.

Après cet ajustement, le véhicule accélère latéralement pour se déplacer dans la voie de destination choisie. La phase d'insertion est caractérisée par une accélération initiale importante, puis une décélération après le véhicule est passé sur la voie de destination. Cette accélération latérale doit être inférieure au seuil d'accélération confortable ( $4 \text{ m/s}^2$ ) et la vitesse latérale est supérieure à zéro et inférieure à la vitesse latérale maximale autorisée [Turner, Griffin, 1999]. La vitesse longitudinale est généralement supposé être constante. Cependant, les données réelles sur les changements de voie montrent que cette vitesse varie pendant la phase de croisement. Environ 80% des conducteurs accélèrent/décélèrent pendant la phase de croisement. Sur ce total, 60% accélèrent tandis que les autres (20% à 40% décélèrent pour atteindre la voie cible [Caveney, 2010].

Dans la dernière phase, le véhicule accélère/ décélère pour ajuster sa vitesse pour maintenir un écart de sécurité en fonction de l'état du véhicule de devant. Ainsi, le processus de changement de voie peut être caractérisé par des vitesses variables avec une accélération et une décélération constantes dans les deux directions longitudinale et latérale.

La manœuvre de changement de voie est une source majeure d'accidents sur la route. Selon différentes études, plus de 10% des accidents de la route sont des collisions lors d'un changement de voie et entraînent également des accidents à la fois dans la voie d'origine et la voie cible [Thiemann et al., 2008]. Même lorsque la collision est évitée par une action d'urgence du véhicule de changement de voie ou d'autres véhicules, une mauvaise décision de changement de voie induit une forte perturbation dans le trafic. Plus de 75% des collisions lors d'un changement de voie sont associées à une erreur de reconnaissance lors de la phase de planification [Kesting et al., 2007]. Il a été constaté que la cause principale est l'absence d'évaluation de la faisabilité du changement de voie dans la phase de planification. Si les trajectoires des véhicules concernés par le changement de voie peuvent être prédites, les risques de collision peuvent être estimés avec une précision suffisante. Des messages d'alerte à long terme appropriés peuvent être envoyés au conducteur pour une action corrective ou préventive. Il est intéressant de noter qu'environ 78% de ces accidents se produisent à faible vitesse ( $6-7 \text{ m/s}$ ). Il faut noter que les mauvaises décisions prises au cours des phases précédentes ne sont pas corrigées dans les phases ultérieures. En outre, près de 95% des accidents sont des collisions latérales qui se produisent pendant la phase d'insertion au milieu d'un changement de voie réel. Les collisions latérales à faible vitesse pendant la phase de croisement indiquent que la

prédiction à court terme des trajectoires des véhicules peut aider à éviter la collision.

Une collision lors d'un changement de voie est indiquée par une intersection des trajectoires du véhicule qui change de voie avec les véhicules qui suivent ou qui roule dans la voie d'origine ou la voie cible dans le domaine spatio-temporel. Cela nécessite la connaissance des trajectoires des véhicules impliqués dans la manœuvre avec des valeurs de position absolues [Sun, Ioannou, 1995].

L'objectif de ce travail est d'étudier la faisabilité d'une prédiction précise de la trajectoire d'un véhicule changeant de voie dans un système de conduite autonome afin d'obtenir une estimation satisfaisante de la sécurité de la manœuvre. Le système autonome utilise uniquement des capteurs embarqués pour trouver les valeurs de position passées sans aucune mesure ou communication inter-véhicules des états relatifs des autres véhicules. Une prédiction de trajectoire à longue distance aide à déterminer la faisabilité d'un changement de voie. Lorsque le processus de changement de voie est en cours, la prédiction à court terme de la position future du véhicule permet de prédire avec précision le risque de collision et le délai avant la collision. Le système de planification peut interrompre le processus et éviter une collision imminente.

#### 4.2.1 Sélection des attributs pour la représentation de la situation

Afin de valider une approche de prédiction de trajectoire dans un véhicule autonome, les données d'entrée utilisées doivent être celles fournies par les capteurs embarqués tels que Radar, Lidar et Caméra. En outre, pour permettre au modèle de prédiction de détecter les corrélations et les attributs les plus pertinents, le modèle de prédiction doit prendre en compte autant d'informations que possible.

Cette section décrit comment une situation sur la route est représentée par le vecteur d'attributs. Le vecteur d'attributs est composé de deux parties, une description du véhicule cible et une représentation des véhicules adjacents au véhicule cible.

Comme le comportement du conducteur est influencé par les autres véhicules sur la route, la performance de la prédiction peut être améliorée si on représente également les véhicules environnants. On prend en considération 6 véhicules autour du véhicule cible, comme le montre la scène de conduite illustrée à la figure 4.3. S'il n'y a pas de véhicule dans l'une des positions spécifiées, les entrées correspondantes seront remplies avec des données artificielles (grandes valeurs). Les données utilisées sont supposées obtenus par une étape de poste-traitement des données de perception. Les attributs choisis sont les suivants :

- Pour le véhicule cible :
  - La position locale (coordonnée latérale et longitudinale) par rapport à un repère fixe
  - La vitesse absolues (latérale et longitudinale)
  - L'accélération (latérales et longitudinales)
  - Le temps de collision (entre le véhicule cible et le véhicule suivant)
- Pour les véhicules adjacents :
  - Les distances entre la cible et les six véhicules adjacents (Figure 4.3)
  - Les vitesses absolues (latérales et longitudinale) de chacun de ces véhicules
  - Les accélérations (latérales et longitudinales) de chacun de ces véhicules

Ce choix d'attributs a été fait pour reproduire les informations sur lesquelles un conducteur humain est susceptible de fonder ses décisions : les données des véhicules environnants sont toutes relatives au véhicule cible, car les conducteurs prennent généralement

leurs décisions en fonction des estimations des distances. Les données de vitesse du véhicule cible sont dans un cadre absolu (relatif à la route), car les conducteurs sont généralement au courant des informations fournies par le compteur de vitesse; de même, nous utilisons des positions relatives à la route, car le conducteur est généralement capable de mesurer visuellement les distances latérales par rapport au bord de la route et connaître sa position longitudinale. Le choix d'inclure explicitement le temps de collision comme un attribut vient de la grande importance de cette mesure dans les décisions de changement de voie [Schlechtriemen et al., 2015]. En outre, les neurosciences indiquent que les cerveaux humains se basent fortement des estimations du temps de collision pour effectuer des tâches de prise de planification de mouvement [Field, Wann, 2005].

La normalisation est un processus qui consiste à redimensionner les données de l'échelle initiale à une échelle spécifique [Brownlee, 2016]. La normalisation est utilisée pour transformer les attributs à une échelle fixe afin d'éliminer l'influence d'une caractéristique sur une autre. Les attributs sont normalisés pour rester dans une plage acceptable en ce qui concerne les fonctions d'activation. Nos données sont normalisées sur une échelle de  $[-2, 2]$ .

Le but de la prédiction est de déterminer les trajectoires futures à partir d'un jeu de données passées de séries temporelles. La prédiction est basée sur l'hypothèse que le futur d'une série temporelle est basé sur son passé. Les données d'entraînement et de test sont organisées sous forme de séries temporelles d'une taille fixe (plusieurs tailles sont testées)

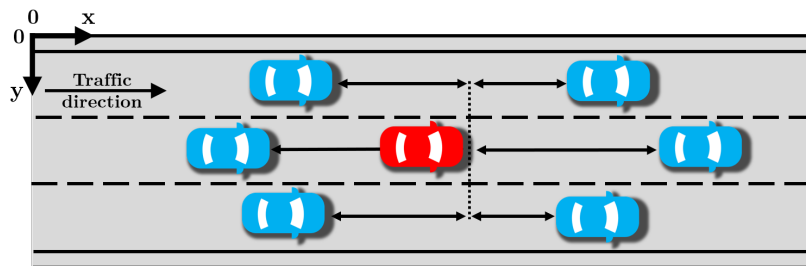
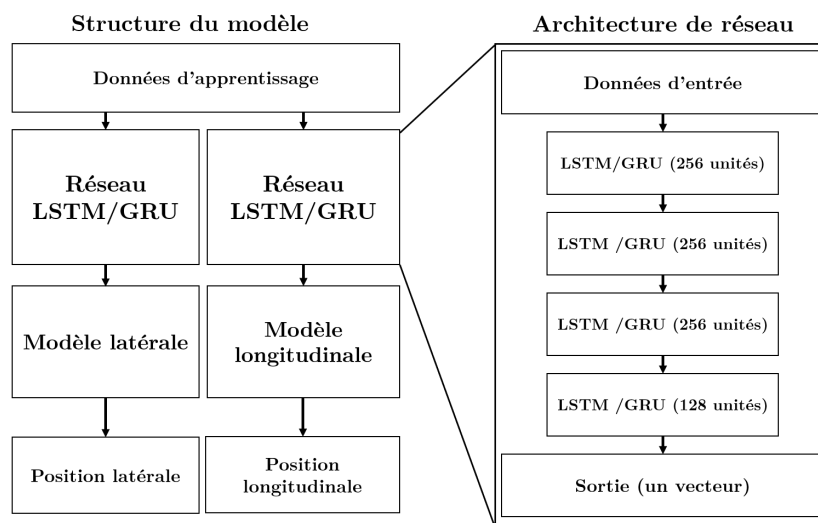


FIGURE 4.3 – Représentation de la scène de conduite avec six véhicules autour du véhicule cible.

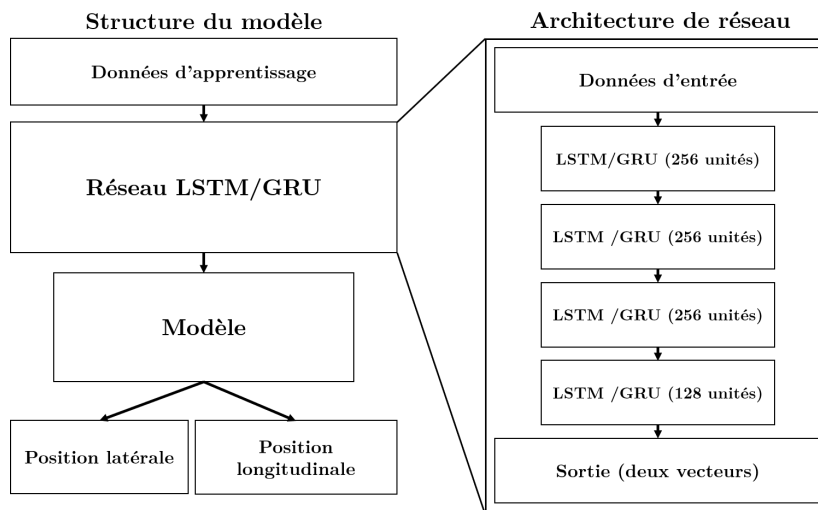
### 4.2.2 Architecture du modèle

Dans cette partie, nous présentons les architectures de nos modèles LSTM et GRU utilisés pour la prédiction de trajectoire. Comme mentionné précédemment, deux variants de RNN sont utilisées, nous avons donc deux types de modèles, le premier est un modèle basé sur LSTM et le second est un modèle basé sur GRU. L'objectif est de prédire les coordonnées longitudinales et latérales de la trajectoire du véhicule. Dans la littérature, de nombreuses méthodes d'apprentissage ont été proposées pour prédire la trajectoire du véhicule [Phillips et al., 2017; Altché, La Fortelle de, 2017; Wang et al., 2018]. Cependant, la plupart de ces méthodes proposent des modèles à deux sorties. Pour ce travail, à côté des modèles traditionnels, nous proposons un modèle séparé pour chaque sortie afin de générer un modèle indépendant pour chacune. Les modèles proposés sont résumé comme suit : modèle LSTM à deux sorties avec un modèle LSTM pour prédire la position latérale, un modèle LSTM pour prédire la position longitudinale; Un modèle GRU à deux sorties composé d'un modèle GRU pour prédire la position latérale et un modèle GRU pour prédire la position longitudinale.

Après avoir défini la structure générale des modèles, nous présentons maintenant l'architecture du réseau. Les modèles proposés sont basés sur des réseaux de neurones récurrents profonds, avec deux types d'unités récurrentes (LSTM et GRU), nous avons choisi une architecture plus profonde pour assurer une meilleure précision des prédictions [Phillips et al., 2017; Altché, La Fortelle de, 2017; Wang et al., 2018]. L'architecture de réseau proposée consiste en une couche d'entrée, de forme K (le nombre d'attributs) et une couche de sortie de forme L (le nombre de sorties, cela dépend du modèle comme expliqué précédemment). Les couches cachées sont soit des couches LSTM soit des couches GRU. Le cadre général de la prédiction de trajectoire à l'aide de LSTM et GRU est présenté dans la figure 4.4.



(a) Modèles à une sortie



(b) Modèle à deux sorties

FIGURE 4.4 – Structure et architecture de réseau des modèles proposés.

Une fois l'architecture de réseau concrète à mettre en œuvre choisie, on commence l'entraînement de réseau en utilisant les attributs sélectionnés. Selon des études précédentes [Phillips et al., 2017; Altché, La Fortelle de, 2017; Wang et al., 2018] et nos expériences, le réseau à 4 couches cachées est choisi pour obtenir le meilleur compromis entre les performances d'apprentissage et la consommation de calcul.



Les réseaux sont entraînés en utilisant des lots de taille 500 échantillons, avec 4 couches LSTM ou GRU de taille 256 pour les 3 premières couches et 128 pour la dernière couche, la couche d'entrée et la couche de sortie sont des couches denses (entièrement connectées). Dans ce travail, nous avons utilisé la bibliothèque de Keras [Chollet, others, 2018] avec les réseaux de neurones récurrents optimisés par Nvidia cuDNN [Chetlur et al., 2014] qui offrent une vitesse jusqu'à 6 fois supérieure à celle des RNN traditionnels. La fonction de perte utilisée est l'erreur quadratique moyenne, l'optimiseur utilisé est ADAM, avec un taux d'apprentissage décroissant, à partir de 0,001. De plus, chaque modèle est entraîné pendant 25 époques, la perte par chaque époque est présentée dans la figure 4.5. Tous les modèles concernés sont entraînés sur une seule carte graphique GPU Nvidia GeForce GTX Titan X 12 Go.

		RMSE de la position latérale(m)					
Séquence d'entrée	Horizon de prédiction	LSTM (2 sorties)	LSTM Latérale	GRU (2 sorties)	GRU Latérale	Altché, La Fortelle de	Xin et al.
10s	5s	0.45	<b>0.41</b>	0.42	<b>0.41</b>	0.47	0.49
10s	10s	0.73	0.79	0.69	0.77	<b>0.65</b>	-

TABLEAU 4.1 – L'erreur de la position latérale (m)

		RMSE de la position longitudinale (m)					
Séquence d'entrée	Horizon de prédiction	LSTM (2 sorties)	LSTM Longitudinale	GRU (2 sorties)	GRU Longitudinale	Altché, La Fortelle de	Xin et al.
10s	5s	4.02	<b>3.54</b>	3.77	4.46	9.00	5.77
10s	10s	12.91	12.77	<b>11.28</b>	12.01	-	-

TABLEAU 4.2 – L'erreur de la position longitudinale (m)

	Temps de prédiction		RMSE Position longitudinale		RMSE Position latérale	
	GRU (s)	LSTM (s)	GRU (m)	LSTM (m)	GRU (m)	LSTM (m)
Trajectoire 1	<b>0.228</b>	0.262	<b>6.34</b>	7.14	<b>0.85</b>	0.89
Trajectoire 2	<b>0.320</b>	0.362	<b>3.43</b>	3.88	<b>0.23</b>	0.28
Trajectoire 3	<b>0.669</b>	0.725	<b>4.11</b>	4.20	<b>0.19</b>	0.24

TABLEAU 4.3 – Évaluation des modèles : Le temps de prédiction et l'erreur de la position latérale et longitudinale de trois véhicules de test

### 4.2.3 Résultats et discussions

Ce travail a permis d'explorer la prédiction à long terme (plus de 2 secondes) des trajectoires futures des véhicules pour des situations de changement de voies. Les résultats obtenus par les algorithmes de prédiction des trajectoires sont présentés dans le tableau 4.1 et le tableau 4.2 pour la prédiction des positions longitudinale et latérale sur différents horizons temporels (5s et 10s). Les données utilisées proviennent des données préparées de NGSIM (70% pour l'entraînement, 30% pour la validation).

Les entrées du modèle sont des séries temporelles des attributs choisis, la sortie est une série des positions futures. Les données d'apprentissage et de test servent à tester les performances du modèle en permanence pendant l'entraînement basé sur la validation croisée des séries temporelles. Pour chaque itération, une fois que le réseau est entraîné, les poids sont fixés, les données de test sont introduites dans le réseau et la sortie du réseau est comparée à la sortie souhaitée. Les poids sont modifiés sur la base de leur valeur précédente et d'une correction des termes et des règles. Une fois la règle sélectionnée, la correction doit être appliquée aux poids, en fonction du taux d'apprentissage. Si le taux d'apprentissage est trop faible, alors l'apprentissage prend plus de temps. S'il est trop élevé, alors l'adaptation diverge et les poids sont inutilisables. En utilisant la méthode de

recherche en grille, nous avons déterminé les meilleures valeurs des paramètres (taille de lot, taux d'apprentissage) pour optimiser le modèle, il faut que le réseau soit entraîné plusieurs fois afin de produire l'erreur la plus faible possible (figure 4.5). Le modèle met fin à l'apprentissage sur la base de l'erreur quadratique moyenne.

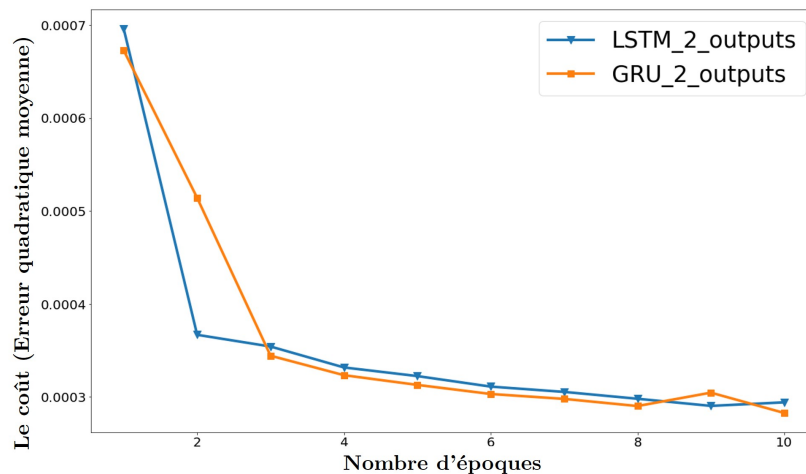
Dans cette étude, la métrique d'évaluation RMSE (l'erreur quadratique moyenne) est utilisée pour évaluer la performance des modèles, nous comparons nos résultats avec [Altché, La Fortelle de, 2017] et [Xin et al., 2018] pour deux raisons : la première, ils ont les meilleurs résultats dans l'état de l'art, la seconde, qu'ils utilisent le même jeu de données (NGSIM). Le tableau 4.1 et le tableau 4.2 font état des performances des modèles à deux sorties, des modèles latéraux et des modèles longitudinaux. Nos conclusions sont résumées comme suit :

- **Prédiction de la position latérale :** pour des horizons de prédiction de 5 secondes, on remarque que les modèles latéraux atteignent la meilleure précision de prédiction avec 0,41m d'erreur, le modèle GRU à deux sorties n'est pas si loin de ce résultat avec 0,42m d'erreur. En général, les modèles proposés sont meilleurs que les approches comparées [Altché, La Fortelle de, 2017] et [Xin et al., 2018] qui ont obtenu des erreurs de 0.47m et 0.49m respectivement. Pour l'horizon de 10 secondes, [Altché, La Fortelle de, 2017] ont toujours la meilleure précision, mais le modèle GRU à deux sorties présente également une précision de prédiction significative.
- **Prédiction de la position longitudinale :** pour un horizon de prédiction de 5 secondes, les modèles à deux sorties surpassent à la fois les modèles longitudinaux et les approches de pointe, avec une RMSE qui ne dépasse pas 4m d'erreur. Pour un horizon de 10 secondes, nous n'avons pas pu comparer nos résultats avec ceux de [Altché, La Fortelle de, 2017] et [Xin et al., 2018], car [Altché, La Fortelle de, 2017] prédit la vitesse longitudinale et [Xin et al., 2018] ne prédit pas la position à l'horizon de 10 secondes. Par conséquent, le modèle GRU à deux sorties surpasse significativement tous les autres modèles.
- La performance de nos modèles à deux sorties dépasse celle des sorties latérales et longitudinales, et le modèle GRU présente une meilleure précision que le modèle LSTM.

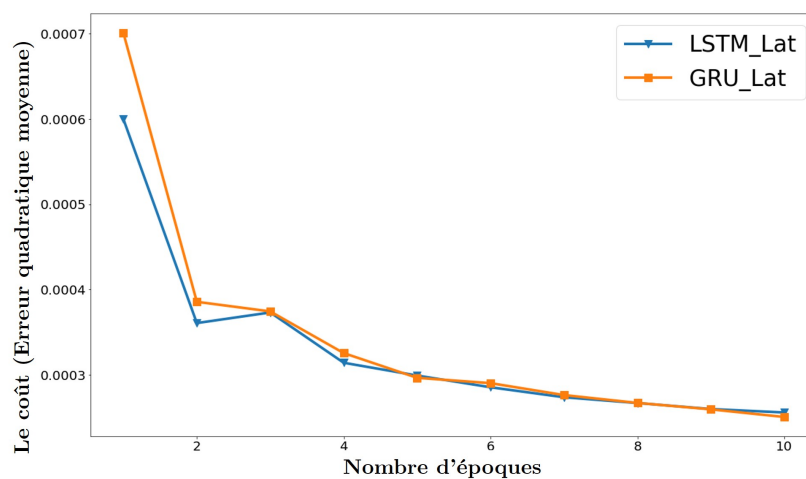
Après avoir conclu que le modèle à deux sorties est plus efficace pour prédire la trajectoire pendant le changement et le maintien de la voie que celui à une seule sortie et afin d'entrer dans les détails du fonctionnement de chaque modèle (modèles à deux sorties), nous avons choisi trois véhicules au hasard pour valider les performances de prédiction de véhicules séparés. En plus d'évaluer la performance prédictive des modèles, il est important d'avoir une idée du temps de réponse du modèle lors de la prédiction. Nous incluons donc dans ce travail une comparaison du temps de prédiction (le temps entre la réception de la donnée d'entrée et la génération de la trajectoire prédite) des modèles GRU et LSTM avec deux sorties. La trajectoire prédite par chaque modèle ainsi que la trajectoire réelle de chaque véhicule sont présentées dans la figure 4.6. Les deux modèles ont donné des prédictions très proches des trajectoires réelles. La figure 4.6(a), montre que, même avec un retard de prédictions (en moyenne 0.4s) dans les trajectoires latérales, les modèles ont réalisé une prédiction précise pour un véhicule en situation de changement de voie. Il a été remarqué que ce retard se produit principalement pour des changements de voie rapides. Dans des situations de maintien de voie, comme le montre les figures 4.6(b), 4.6(c), 4.6(e) et 4.6(f) les deux modèles sont capables de prédire avec précision et sans retard la position latérale et longitudinale. Le tableau 4.3 présente les

résultats de l'évaluation en termes de temps de perdition et de précision de prédiction de la position. Les résultats (Tableau 4.3) valide l'importance du modèle basé sur le GRU. En raison de la simplicité de la structure des unités GRU, le temps de traitement est très réduit par rapport le modèle basé sur la LSTM. Par conséquent, les méthodes basées sur GRU sont très adaptées aux applications de prédiction en temps réel.

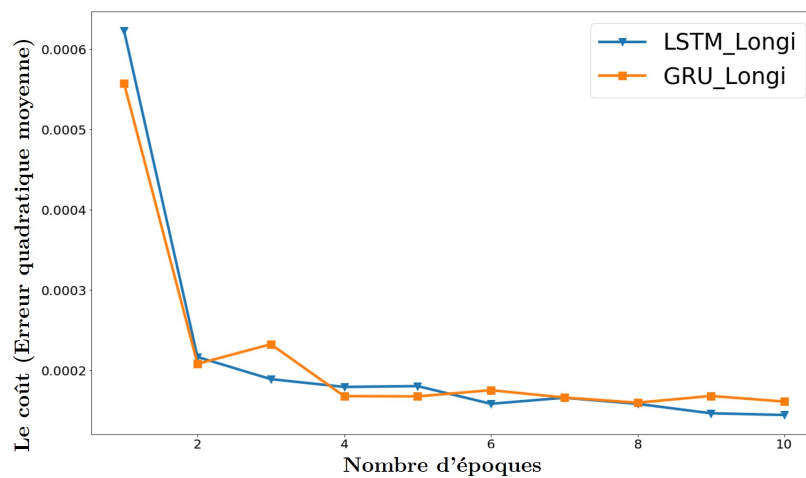
Cette étude a démontré la capacité des réseaux de neurones récurrents profonds DRNN (LSTM et GRU) à prédire efficacement la trajectoire du véhicule avec une précision intéressante. Nous avons testé quatre modèles différents avec différentes structures et unités RNN. Les modèles à deux sorties ont montré des performances significatives en terme de précision de prédiction de la position longitudinale et latérale. Nous avons constaté que le modèle basé sur GRU fonctionne très bien, il a démontré un avantage à tout égards.



(a) Modèle à deux sorties



(b) Modèle latérale



(c) Modèle longitudinale

FIGURE 4.5 – Courbes d'entraînement du jeu de donnée d'apprentissage des différents modèles.

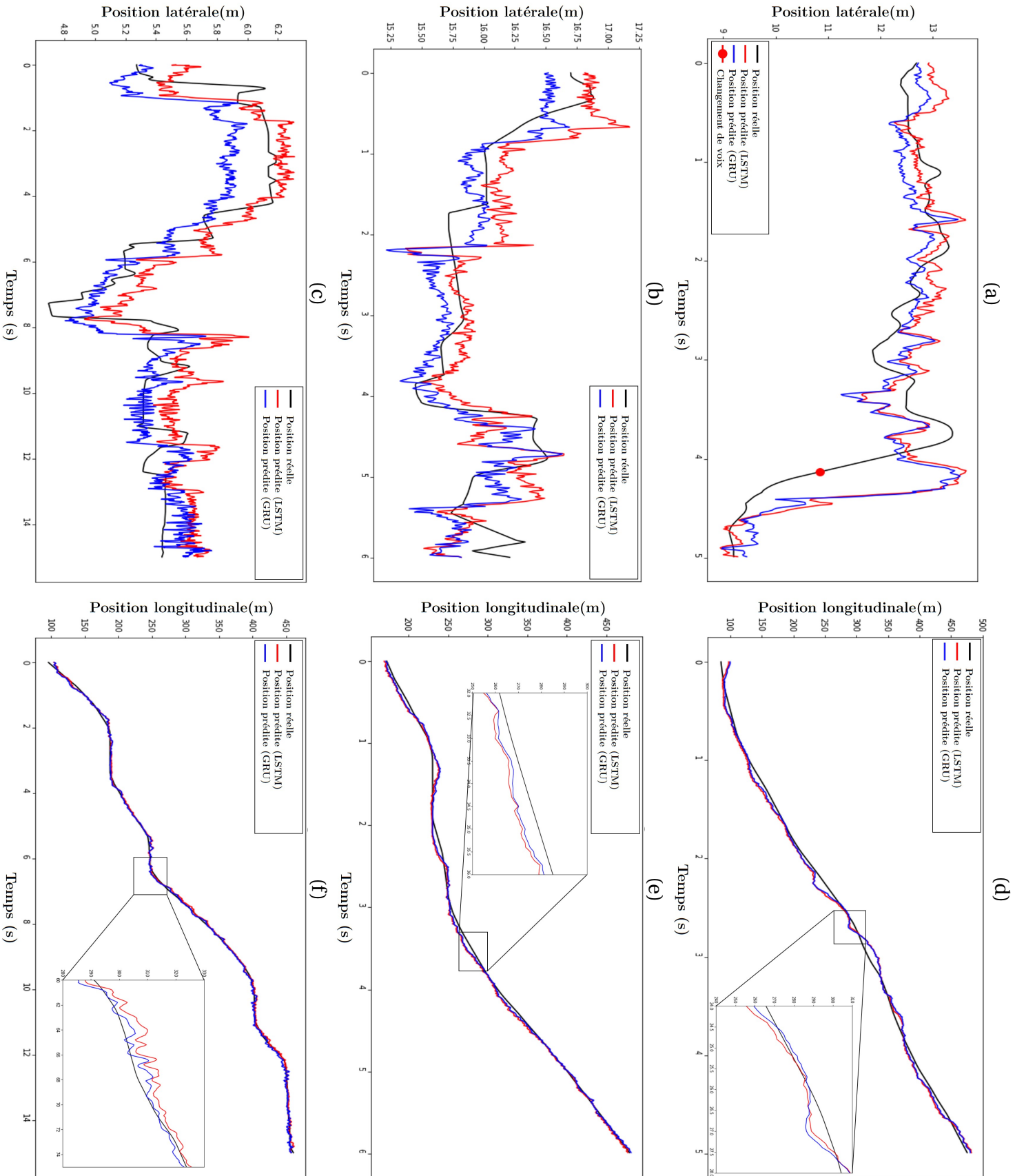


FIGURE 4.6 – La prédiction de la trajectoire de trois véhicules de test : (a), (b) et (c) représentent les positions latérales et (d), (e) et (f) représentent respectivement les positions longitudinales (noir : position réelle, rouge : prédiction LSTM, bleu : prédiction GRU).

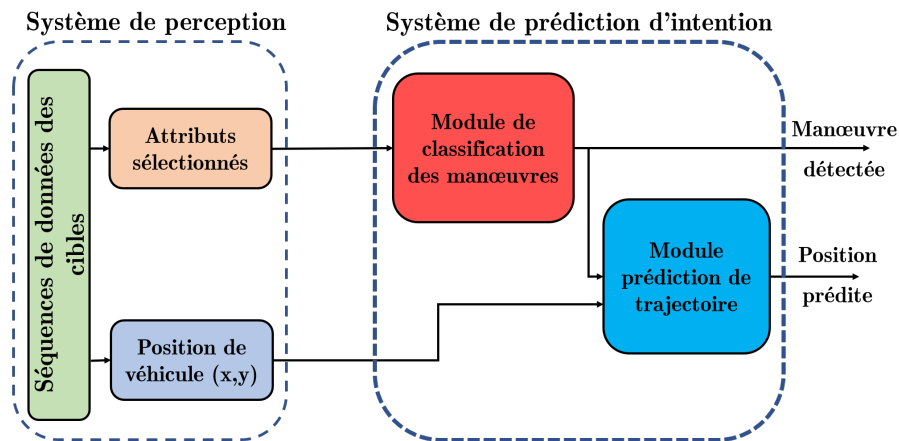


FIGURE 4.7 – Présentation du système : Intégration de la classification des manœuvres et de l’approche basée sur la prédiction de la trajectoire

## 4.3 Approche hybride pour la prédiction de trajectoire

### 4.3.1 Présentation du système de prédiction

Afin d’améliorer la précision de la prédiction en phase de changement de voie, nous proposons dans cette partie d’inclure un module de classification des manœuvres. Ce travail vise à développer un modèle robuste pour la prédiction de trajectoire basé sur le réseau de neurones artificiel (ANN) et les réseaux de neurones récurrents LSTM profonds à partir d’un véhicule autonome (figure 4.7). Nous proposons un nouveau modèle hybride qui combine l’approche basée sur les manœuvres et la prédiction de trajectoire. Nous nous concentrons sur la prédiction de l’intention de changement de voie pour des scénarios d’autoroute. Dans cette partie, le véhicule cible est considéré comme un agent indépendant, de sorte que l’interaction avec les autres véhicules n’est pas prise en compte pour des considérations de validation expérimentale. L’événement de changement de voie se produit lorsque le segment de trajectoire du véhicule et le marquage de la voie se croisent. Dans le Chapitre III, les résultats démontrent que l’intention du conducteur de changer de voie peut être détectée avec plus de précision en utilisant des modèles basés sur l’ANN. L’ANN utilise des séquences de conduite comme entrée pour classer les manœuvres. Nous définissons la tâche de classification comme la reconnaissance de trois classes de manœuvres : changement de voie à gauche, changement de voie à droite et maintien de la voie. Le module de prédiction de trajectoire LSTM utilise la sortie de la classification avec une séquence de positions passées du véhicule cible pour prédire les positions futures de ce véhicule.

Le système de prédiction d’intention est subdivisé en deux parties principales : la classification des manœuvres et la prédiction de la trajectoire, comme indiqué dans la figure 4.7. Dans la partie de classification des manœuvres, la future manœuvre de conduite du véhicule cible est estimée par le biais de réseaux de neurones artificiels. Pour cela, un ANN est alimenté par les données mesurées du véhicule, qui sont disponibles via le système de perception qui traite les données des capteurs extéroceptif. Le module de prédiction combine les résultats de la classification avec les l’historique des positions pour prédire la nouvelle position du véhicule. Par conséquent, s’il connaît à l’avance la manœuvre qui sera effectuée par le conducteur, le modèle de prédiction peut générer la trajectoire qui correspond à cette manœuvre.

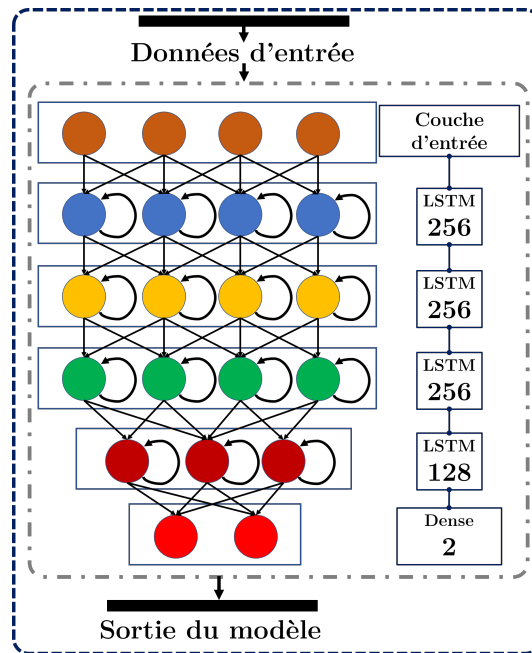


FIGURE 4.8 – Architecture du réseau LSTM de prédiction de trajectoire.

## 4.3.2 Expérimentation et évaluation

### 4.3.2.a Architecture du modèle

Dans cette partie, nous présentons les architectures des réseaux ANN et LSTM utilisés pour construire le modèle hybride. Comme mentionné précédemment, l'ANN est utilisé pour classer les manœuvres et le LSTM est utilisé pour prédire la position future. Le but est de prédire les coordonnées longitudinales et latérales de la trajectoire du véhicule en utilisant les résultats de classification et la séquence des positions passés.

Le réseau à 4 couches cachées est choisi pour obtenir le meilleur compromis entre performances d'apprentissage et consommation de calcul. Les réseaux sont entraînés à l'aide de lots de 500 échantillons, avec 4 couches LSTM de taille 256 pour les 3 premières couches et 128 pour la dernière couche, la couche de sortie est une couche dense (entièrement connectée) (figure 4.8).

### 4.3.2.b Protocole d'évaluation

Cette partie est consacrée à l'analyse et à la comparaison des résultats de prédiction et de classification. L'analyse de deux cas indépendants est présentée pour visualiser et comparer la prédiction et la mesure de la vérité terrain. Ce travail explore la prédiction à long terme (plus de 2 s) des trajectoires futures des véhicules. L'erreur quadratique moyenne est utilisée pour évaluer les performances des modèles. Selon la prédiction, la distance euclidienne entre la position réelle du véhicule et la position future correspondante est rapportée. Les résultats sont classés en trois catégories en fonction de la longueur correspondante de la séquence d'entrée. Les résultats obtenus par les algorithmes de détection de manœuvre et de prédiction de trajectoire sont rapportés dans le tableau 4.4. Ce dernier présente les erreurs de prédiction de la position longitudinale et latérale ainsi que la précision de la classification sur différents horizons temporels (1 s, 3 s et 5 s). Les résultats obtenus sont résumés ci-dessous.

#### 4.3.2.c Évaluation sur deux exemples

Nous présentons dans cette partie les résultats de notre approche sur deux échantillons de trajectoires choisis aléatoirement. Le premier contient deux changements de voie vers la droite et le second est une succession de trois changements de voie gauche. Comme le montrent les figures 4.9 et 4.10, les positions réelles et prédites sont affichées, ainsi que les résultats de la classification. Dans chaque cas, on trouve deux types de trajectoires : les trajectoires de changement de voie et les trajectoires de maintien de voie. La vitesse moyenne est de 7.3 *m/s* pour le premier véhicule et de 8.0 *m/s* pour le deuxième véhicule. Les séquences utilisées ont une longueur de 6 s et l'horizon de prédiction est de 5 s.

Le résultat de la prédiction de la trajectoire du véhicule basée sur le LSTM et la classification des manœuvres dans le scénario de changement de voie à droite sont présentés dans la figure 4.9. Le scénario de changement de voie à gauche est présenté dans la figure 4.10. Sur les deux figures, on note que le modèle de classification a détecté les changements de voie très tôt avant le franchissement de voie, ce qui permet plus de précision à la position prédite. Les résultats indiquent que le modèle proposé fonctionne bien dans la prédiction à long terme. L'approche proposée peut intégrer les avantages des deux modèles (classification des manœuvres et prédiction de trajectoire). En d'autres termes, le modèle de classification assure la reconnaissance de la situation et le modèle de prédiction génère la trajectoire précise qui correspond à la situation détectée. Les deux figures (4.9, 4.10) montrent clairement que de bons résultats de prédiction peuvent être obtenus même dans des scénarios de succession de plusieurs changements de voie.

#### 4.3.2.d Résultats d'évaluation

**Résultats de la classification des manœuvres :** Le but ici est de vérifier si la bonne manœuvre est détectée avant que le véhicule rentre dans la voie de destination. La vérité terrain est ainsi déterminée en analysant le changement de l'identifiant de voie du véhicule. Comme indiqué dans le tableau 4.4, la précision de la classification varie de selon la taille de séquence d'entrée. On voit clairement que les longues séquences donnent une meilleure précision de classification. Pour les séquences de 6 secondes, on a 97,49 % de manœuvres correctement prédites. Le but de la classification n'est pas seulement de comprendre la situation de conduite, mais aussi de fournir une entrée supplémentaire pour le modèle de prédiction. Les résultats de test démontrent l'efficacité de l'ANN pour la détection et le classement des manœuvres.

**Résultats de la prédiction de trajectoire :** Dans cette expérience, 80 trajectoires de véhicule sont utilisées pour chaque cas direction de changement de voie pour calculer l'erreur moyenne de la position prédite. À l'aide des résultats de classification des manœuvres, la trajectoire du véhicule peut être prédite avec plus de précision. Un aspect important des modèles de prédiction est leur évolution en fonction de l'horizon de prédiction. L'interprétation du tableau 4.4 indique que l'incertitude de prédiction du modèle augmente largement avec l'augmentation de l'horizon de prédiction, en particulier dans la prédiction à long terme. Les erreurs des positions latérales et longitudinales sont très faibles pour les prédictions à court terme (moins de 2 s), mais ces erreurs augmentent rapidement à mesure que l'horizon temporel s'agrandit. Nous voyons que la précision de la prédiction pour tous les horizons de prédiction est proportionnelle à la longueur de la séquence d'entrée. Une séquence d'entrée de 6 s montre une grande précision pour un horizon de prédiction de 5 s. Les erreurs de prédiction sont d'environ 0,092 m pour la position latérale et 0,112 m pour la position longitudinale. Il est donc important d'ac-



TABEAU 4.4 – Évaluation des modèles : précision de la classification et RMSE de la position latérale et longitudinale.

Les paramètres du système			RMSE de la position prédite (m)	
Séquences d'entrée	Horizon de prédiction	Précision de classification	Prédiction de la position	
			Position latérale	Position longitudinale
3 s	1 s	86.19 %	0.043	0.122
	3 s		0.125	0.235
	5 s		0.235	0.264
5 s	1 s	96.65 %	0.042	0.071
	3 s		0.072	0.100
	5 s		0.126	0.142
6 s	1 s	97.49 %	0.040	0.062
	3 s		0.063	0.082
	5 s		0.092	0.112

quérir plus d'informations sur l'historique des positions d'un véhicule pour obtenir des prédictions plus précises des positions futures. LSTM est un modèle complexe comparé à d'autres types de modèles de réseaux de neurones (équations [2.17]) [Chung et al. \[2014\]](#). Cette complexité est un point positif qui rend le modèle plus efficace pour apprendre la dynamique complexe du mouvement du véhicule et pour prédire ses positions dans le futur lors des changements de voie. Contrairement à [\[Altché, La Fortelle de, 2017\]](#), la trajectoire latérale prédite ne présente pas de retard par rapport la trajectoire réelle. L'ajout d'une information sur la future manœuvre au modèle de prédiction résout la limitation du retard et améliore la précision de la prédiction.

#### 4.3.2.e Résultats sur données réelles

Cette partie présente les résultats de l'évaluation de l'approche de prédiction de trajectoire à l'aide de données réelles de changement de voie. Le modèle utilisé est le même modèle évalué dans la partie précédente sans passer par une étape de réapprentissage sur les nouvelles données. Dans notre expérience, nous avons utilisé le démonstrateur VEDECOM comme véhicule d'essai (figure (4.11)). Ce démonstrateur est un véhicule automatisé basé sur la plate-forme Renault ZOE équipé d'un radar longue portée (Continental ARS 408), d'un Lidar (Velodyne VLP-16), d'un GPS RTK et d'une caméra de Mobyleye. Nous avons collecté les données de test de conduite sur les pistes d'essai de Satory à Versailles (figure 4.12). Pendant la conduite, le véhicule d'essai enregistre les mesures des capteurs pour les scénarios de changement de voie avec une vitesse moyenne de 22 m/s. Nous nous sommes concentrés sur les comportements d'un seul conducteur avec l'espoir d'étendre le test à plus de conducteurs à l'avenir. Pour ce test, on adopte un scénario de conduite avec deux véhicules, un ego-véhicule et un véhicule cible. Il a été demandé du conducteur du véhicule cible d'effectuer des changements de voie. L'objectif est de suivre et d'enregistrer les données de ce véhicule à partir de l'ego-véhicule. Les données sont poste-traitées pour avoir les mêmes attributs utilisés comme entrées pour le modèle entraîné à l'aide des données NGSIM. La validation est faite en comparant les trajectoires prédites avec la vérité de terrain des positions obtenues avec un GPS RTK.

Pour évaluer le modèle entraîné et procéder aux étapes de prédiction, la trajectoire du scénario de test montré dans la figure 4.12 a été réalisée. Avec ce scénario, deux changements de voie de droite et un changement de voie de gauche ont été enregistrés pour évaluer le système de prédiction décrit dans la dernière section. Pour prédire les positions futures du véhicule d'essai, quelques étapes ont été suivies. Initialement, l'approche détecte la manœuvre en utilisant la vitesse latérale, l'accélération latérale, le lacet et la vitesse an-

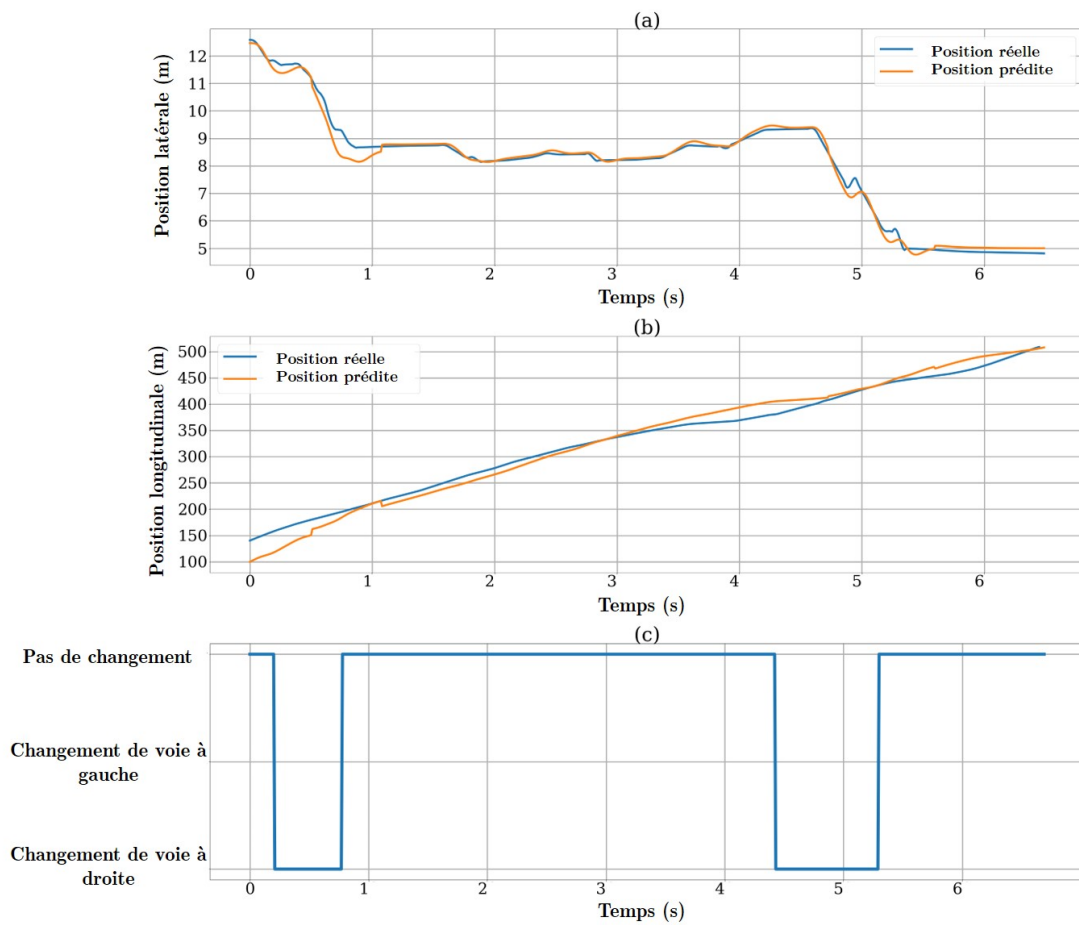


FIGURE 4.9 – Prédiction de la trajectoire d'un véhicule avec deux changements de voie à droite : (a) et (b) représentent respectivement la trajectoire longitudinale et latérale prévue et réelle. (c) représente les résultats de la classification.

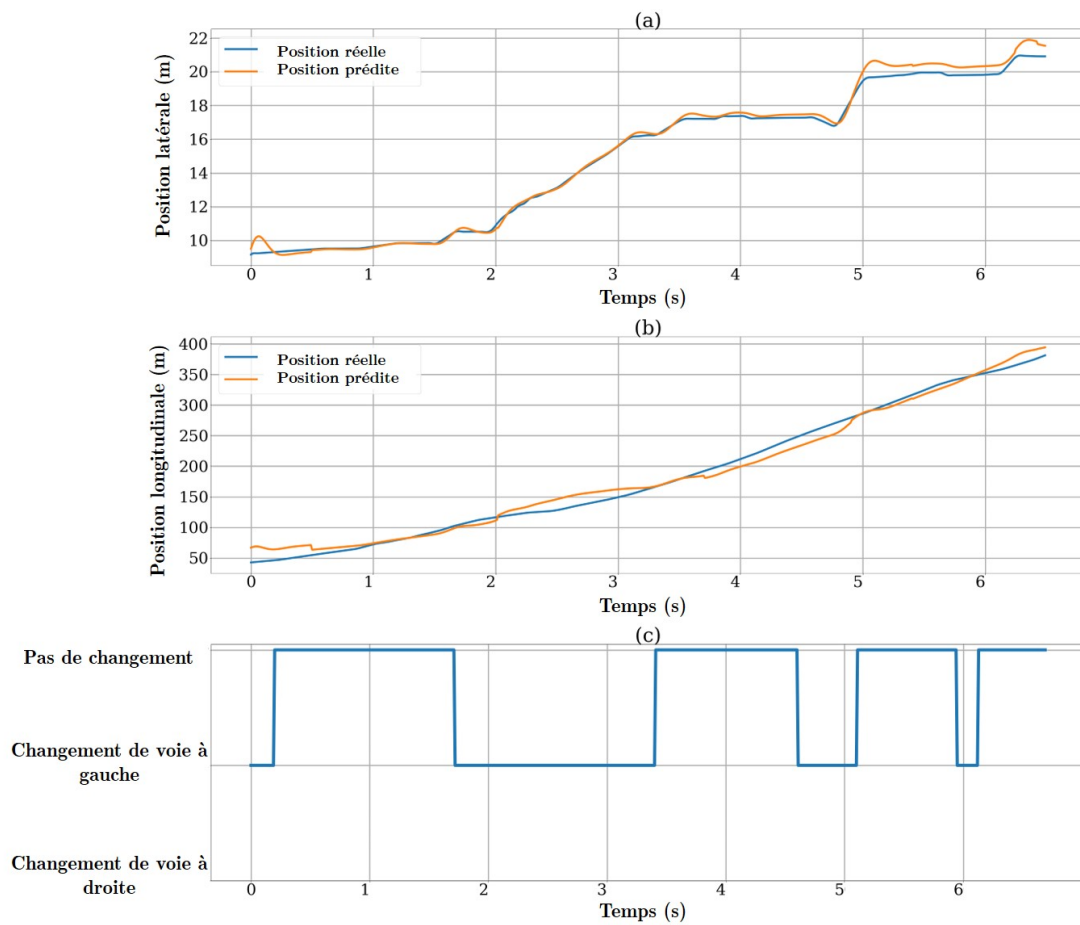


FIGURE 4.10 – Prédiction de la trajectoire d'un véhicule avec trois changements de voie à gauche : (a) et (b) représentent respectivement la trajectoire longitudinale et latérale prévue et réelle. (c) représente les résultats de la classification.

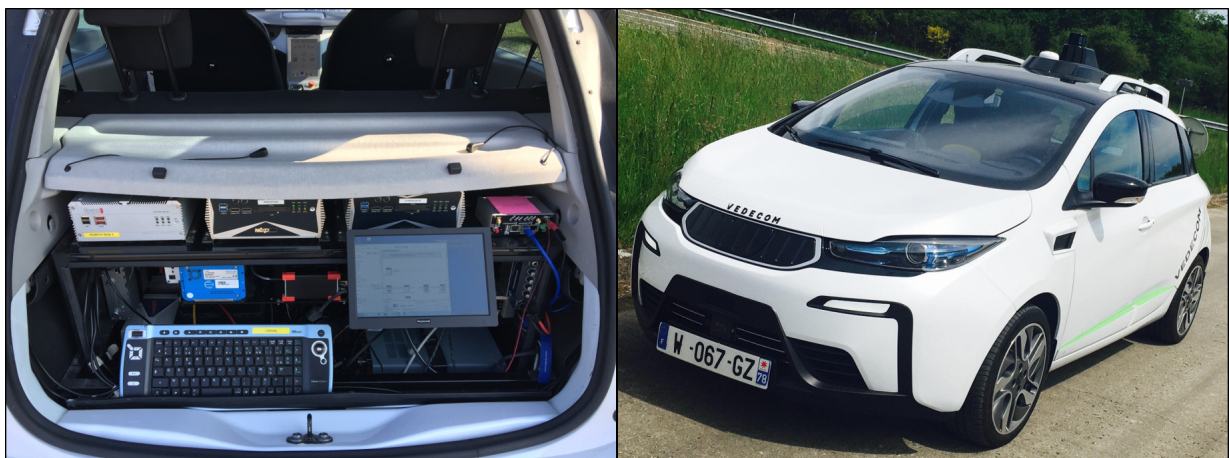


FIGURE 4.11 – Démonstrateur VEDECOM : capteurs installés et plateforme hardware interne.

gulaire. Ensuite, il utilise la manœuvre détectée et la séquence des positions passées pour générer une prédiction de la future trajectoire.



FIGURE 4.12 – Pistes d’essai Satory (a) et trajectoire du véhicule d’essai(b).

L’évaluation s’est déroulée en deux étapes. Tout d’abord, nous évaluons le modèle de classification pour tester les performances de détection de manœuvre. Deuxièmement, nous évaluons le modèle de prédiction de trajectoire pour un horizon de 5 s, puis nous comparons les trajectoires prévues et réelles. Les résultats des tests de prédiction de trajectoire et de classification des manœuvres sont présentés dans la figure 4.13. Comme le montre cette figure, le modèle de classification des manœuvres est capable de détecter avec précision les changements de voie (rappel = 1) en moyenne 2,2 s avant l’instant de croisement. En ce qui concerne la trajectoire, les résultats de la prédiction sont très similaires à la vérité terrain. Les erreurs quadratiques moyennes des positions latérales et longitudinales sont respectivement de 0,30 m et 3,1 m sur une horizon de 5s. Les résultats de la validation démontrent l’efficacité du modèle hybride proposé. Cette validation sur des données expérimentales, alors que l’apprentissage a été réalisé sur une base de données très différentes (NGSIM), donne des résultats très proches que ceux obtenus lors de la validation sur les données NGSIM, sans avoir nécessité de réapprentissage. Notre approche semble donc peu sensible au type de données d’entrée. Malgré que la validation a été faite sur des données qui sont totalement différentes des données d’apprentissage (style de conduite, roue, pays.. etc), le modèle hybride a montré une grande robustesse de changement de donnée et une bonne précision de prédiction.

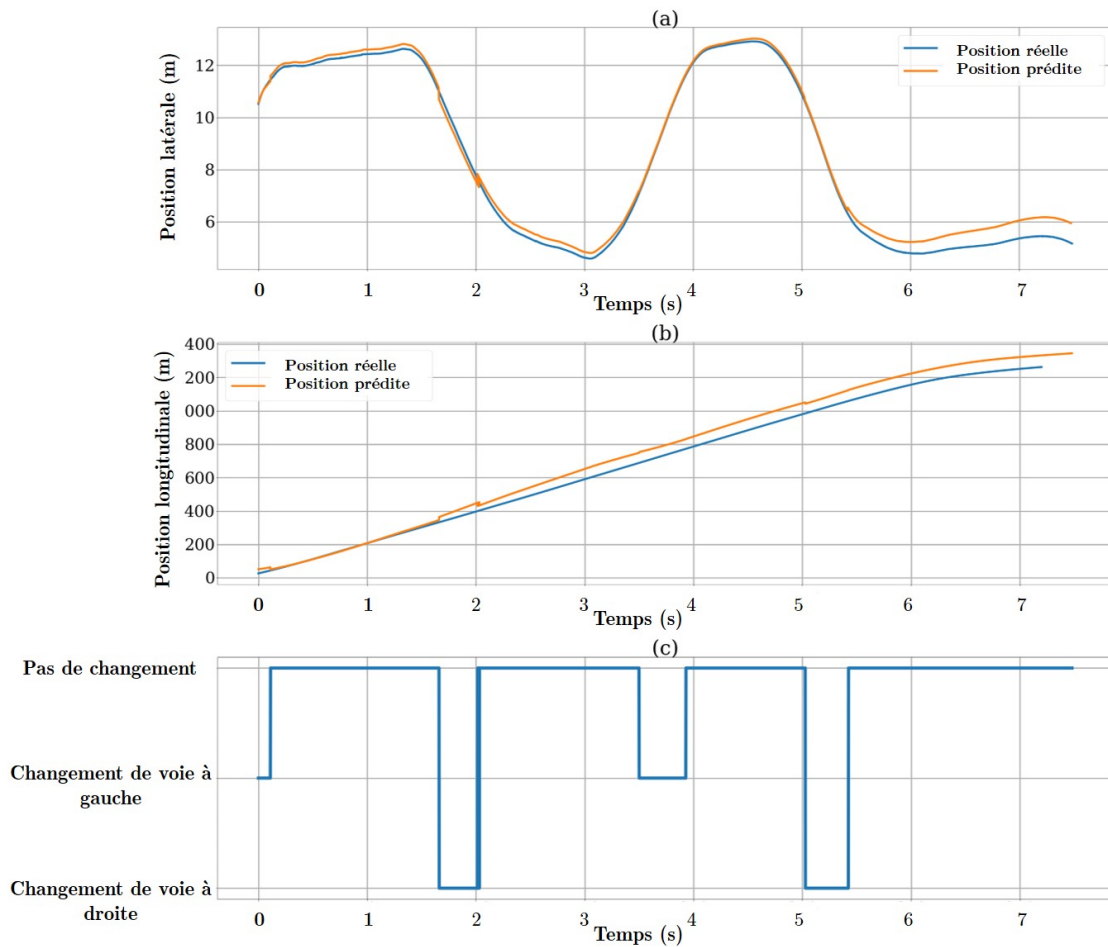


FIGURE 4.13 – Prédiction de la trajectoire du démonstrateur VEDECOM : (a) et (b) représentent respectivement la trajectoire longitudinale et latérale prédite et réelle. (c) représente les résultats de la classification.

## 4.4 Prise de décision de changement de voie

En général, les algorithmes de conduite autonome se compose de trois modules : un module de perception de l'environnement, un module de prise de décision, un module de contrôle. La tâche du module de prise de décision est de prendre une décision de conduite appropriée en fonction des informations des capteurs du module de perception, puis de planifier une trajectoire de conduite vers le module de contrôle. La prise de décision raisonnable dans une variété d'environnements complexes est un grand défi, et il est impossible d'énumérer les stratégies d'adaptation dans diverses situations. Par conséquent, une méthode capable d'apprendre un comportement approprié à partir de ses propres expériences serait souhaitable. Au cours des dernières années, l'apprentissage par renforcement profond (DRL<sup>1</sup>) a fait des percées dans de nombreux domaines de l'intelligence artificielle, tels que les jeux [Mnih et al., 2013, 2015; Silver et al., 2017] et la conduite autonome [Kiran et al., 2021; Sallab et al., 2017]. L'apprentissage profond a une forte capacité d'apprentissage de représentations et de généralisation de modèles de correspondance à partir de données brutes comme les images et les vidéos [LeCun et al., 2015], tandis que l'apprentissage par renforcement a une bonne capacité de prise de décision basée sur des attributs de faible dimension [Zhao et al., 2016]. Par conséquent, les algorithmes de DRL sont très efficaces dans les tâches nécessitant à la fois la représentation des attributs et l'apprentissage de politiques, par exemple, la prise de décision de changement de voie en conduite autonome.

Dans cette partie, on propose une approche d'apprentissage par renforcement basée sur le DRL pour l'apprentissage de la prise de décision pour des manœuvres de changement de voie pour la conduite autonome sur les autoroutes. Plus précisément, nous présentons une nouvelle approche qui est capable de :

1. Apprendre la prise de décision de haut niveau pour effectuer un changement ou un maintien de voie sur des autoroutes pendant qu'un système de bas niveau effectue la manœuvre choisie,
2. Utiliser une représentation minimale de l'environnement pour réduire la dimensionnalité de l'espace d'état pour un apprentissage accéléré,
3. Assurer l'efficacité et la sécurité dans le système pour garantir que l'agent n'exécute que des actions sûres.

### 4.4.1 Prise de décision basée sur l'apprentissage par renforcement

Un changement de voie est généralement effectué lorsque le véhicule doit sortir de l'autoroute, dépasser un véhicule plus lent, s'adapter à la voie de circulation qui le précède, etc. Les manœuvres de changement de voie peuvent être classées en deux grandes catégories : obligatoires et discrétionnaires [Vechione et al., 2018]. Par rapport à un changement de voie discrétionnaire qui a pour but de dépasser un véhicule, un changement de voie obligatoire se produit généralement lorsque l'ego véhicule est obligé de changer de voie en raison l'apparition ou la disparition d'une voie, d'une entrée ou d'une sortie d'autoroute.

Dans notre formulation, nous nous concentrons sur les situations de changement de voie discrétionnaires. Notre tâche consiste à décider quand et comment effectuer la manœuvre de changement de voie en fonction des états des véhicules environnants et de l'ego véhicule lui-même. Une fois qu'une décision est prise par le modèle, un contrôleur

---

1. En anglais Deep Reinforcement Learning

de bas niveau responsable des mouvements longitudinaux et latéraux est utilisé pour générer une commande correspondante pour exécuter la décision. La politique de changement de voie doit respecter les trois objectifs suivantes :

- Éviter les collisions avec les véhicules environnants
- Obtenir une efficacité de conduite élevée
- Exécuter des manœuvres en douceur

L'apprentissage par renforcement est un algorithme sans modèle permettant de contrôler un agent ou un processus afin d'atteindre l'objectif souhaité tout en interagissant dans un environnement stochastique. L'agent reçoit les états d'observation et les récompenses de l'environnement et essaie de maximiser le rendement à long terme accumulé au cours de l'interaction avec l'environnement. Notre objectif est d'apprendre à un agent RL à prendre des décisions de haut niveau dans des scénarios d'autoroute en traitant un espace d'état continu et un espace d'action discret, dans le but de conduire aussi proche que possible de la vitesse désirée.

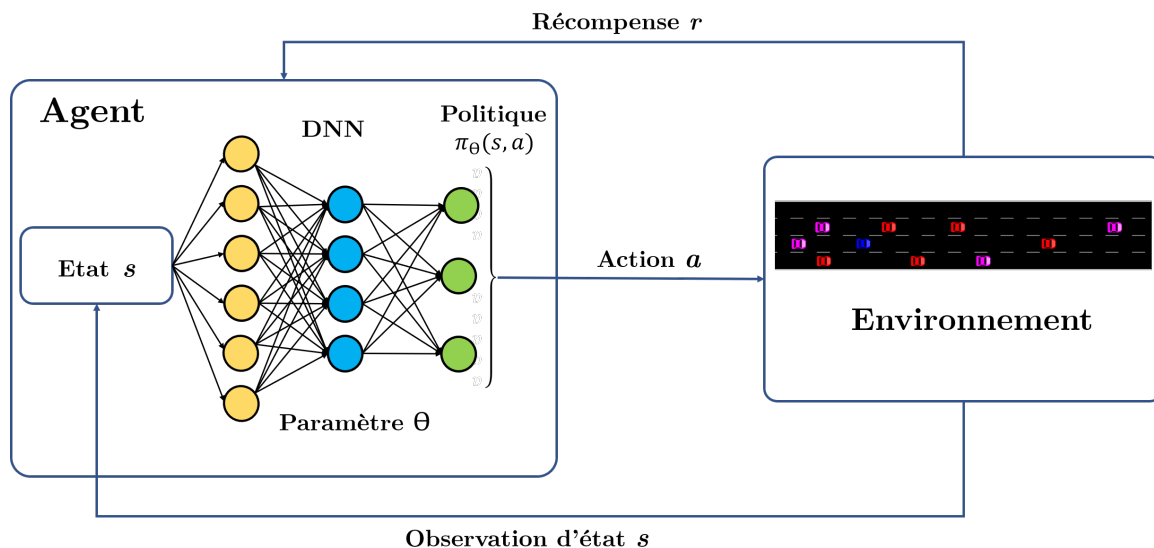


FIGURE 4.14 – Architecture du système de prise de décision proposé.

L'architecture globale du système permettant le changement de voie automatisé est présentée par la figure 4.14. Le système comprend deux composants principaux : un modèle d'apprentissage et un environnement de simulation. Plus précisément, le modèle d'apprentissage utilise le réseau de neurones pour entraîner l'ego-véhicule (agent) à apprendre une politique de haut niveau pour les tâches de prise de décision tout en interagissant avec le trafic environnant. L'environnement de simulation, qui comprend le réseau routier, le trafic et différents scénarios de tâches, est généré à l'aide d'une plateforme de simulation de trafic microscopique, SUMO (Simulation of Urban Mobility) [Behrisch et al., 2011], et il est utilisé pour interagir avec l'agent d'apprentissage. Grâce à SUMO et à son interface de contrôle du trafic (TraCI) (figure 4.15), nous pouvons accéder aux informations sur les véhicules dans le réseau routier, exécuter des décisions de haut niveau dans le modèle d'apprentissage et prendre en compte la dynamique des véhicules générée dans le modèle de simulation. Pour permettre des comportements de conduite sûrs, fluides et efficaces sur les autoroutes, l'ego véhicule reçoit d'abord son état actuel et celui des véhicules environnants de l'environnement SUMO par l'intermédiaire de TraCI, et ces états sont transmis par le réseau de politiques. Ensuite, l'ego véhicule détermine les

actions longitudinales et latérales de haut niveau sur la base du réseau de politiques développé, qui renvoie ensuite l'action à SUMO pour modéliser le mouvement du véhicule au cours de la prochaine étape temporelle et calculer la récompense correspondante.

#### 4.4.2 L'environnement de simulation

Comme nous l'avons mentionné dans la section précédente, le logiciel SUMO est utilisé pour simuler l'environnement de trafic. SUMO est choisi parce qu'il peut représenter la situation de trafic d'intérêt et il peut produire directement des mesures qui sont utilisables pour l'entraînement de notre agent (véhicule autonome).

SUMO est un logiciel libre de simulation de trafic microscopique et continu, principalement développé par l'Institut des systèmes de transport du Centre aérospatial allemand. Il est disponible gratuitement et publié sous la licence publique Eclipse V2, et a été largement utilisé ces dernières années [Lopez et al., 2018]. Le terme "microscopique" signifie ici que chaque véhicule et sa dynamique sont représentés individuellement. Dans la simulation du trafic, ses avantages comprennent des entrées simples, des sorties utiles et la possibilité de modifier facilement les modèles. En outre, son interface graphique claire permet d'observer la situation des véhicules pendant la simulation.

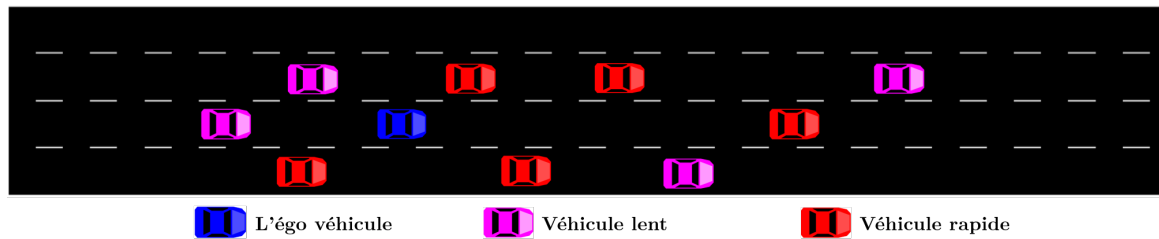


FIGURE 4.15 – Interface de la plateforme de simulation SUMO

Une autoroute à  $n$  voies est utilisée comme plate-forme de simulation pour développer nos agents d'apprentissage par renforcement. Un nombre  $m$  de points initiaux est choisi au hasard dans les limites de l'écart initial maximale d'écartement des véhicules,  $E_{max}$ , afin de déterminer les positions longitudinale  $x_0$  et latérale  $y_0$  initiales de chaque véhicule, avec une distance minimale entre les véhicules  $d_{max}$ . Chaque véhicule a une dimension de  $4,5 \times 2,5$  mètres. L'angle de cap initial  $\psi_0$  est choisi égal à 0 pour tous les véhicules au départ. Les véhicules sont classés selon leur position longitudinale, et le véhicule du milieu est choisi comme l'ego véhicule. Dans notre environnement, nous avons choisi deux types de véhicules : véhicules lents et véhicules rapides. Pour les véhicules lents, les vitesses initiales aléatoires  $v_0$  sont définies dans la plage  $[v_{min}^{lent}, v_{max}^{lent}]$ , la même logique est appliquée pour les véhicules rapides et le véhicule ego lui-même avec les plages  $[v_{min}^{rap}, v_{max}^{rap}]$  et  $[v_{min}^{ego}, v_{max}^{ego}]$ . La section de la route a une longueur de  $L_{route}$ . Ces paramètres sont pour la plupart tirés de [Krajewski et al., 2018]. Lors de la création de notre environnement, nous avons essayé de reproduire au mieux le trafic du jeu de données HighD (figure 4.16). Les valeurs des paramètres sont explicitement décrites dans le tableau 4.5. Ces paramètres sont choisis à partir de la description de HighD [Krajewski et al., 2018].

#### 4.4.3 Espace d'état et d'action

Nous considérons l'état de 6 véhicules autour de l'ego véhicule impliqués dans la phase de décision et d'exécution du changement de voie, comme le montre la figure 4.17.



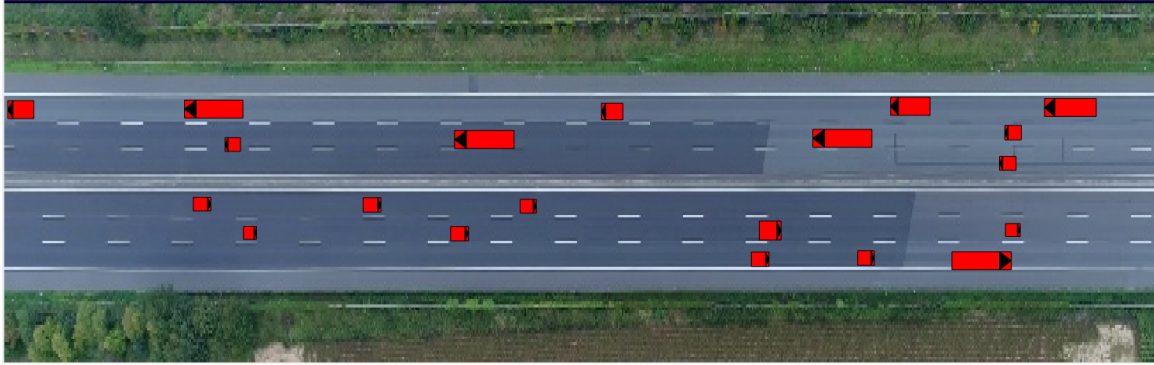


FIGURE 4.16 – Exemple d’une scène de conduite du jeux de donnée HighD

Nombre de voie, $n$	3
Nombre de véhicules, $m$	{20, 30, 40}
Écart maximal initial entre les véhicules, $E_{max}$	200 m
Distance minimale entre véhicules, $d_{max}$	2.5 m
Plage de vitesse des véhicules rapides, $[v_{min}^{rap}, v_{max}^{rap}]$	[20, 28] m/s
Plage de vitesse des véhicules lents, $[v_{min}^{lent}, v_{max}^{lent}]$	[28, 38] m/s
Plage de vitesse pour l’ego véhicule, $[v_{min}^{ego}, v_{max}^{ego}]$	[25, 30] m/s
Longueur de la section de la route, $L_{route}$	4000 m

TABLEAU 4.5 – Paramètres de l’environnement de simulation

L’espace d’état est composé d’un total de 15 variables d’état continues provenant à la fois du véhicule ego et des véhicules qui l’entourent. Plus précisément, l’ego véhicule possède 3 variables d’état; sa vitesse, son accélération, ainsi que le numéro de sa voie. De plus, chaque véhicule environnant possède 2 variables d’état : la distance relative par rapport à l’ego-véhicule et sa vitesse absolue.

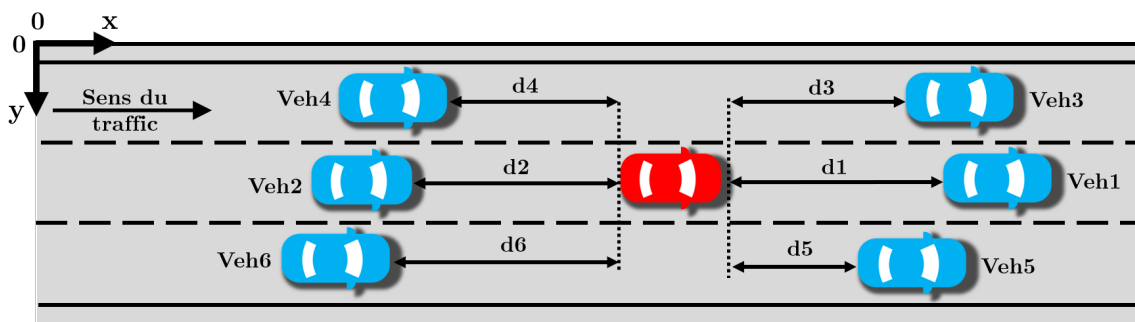


FIGURE 4.17 – Six véhicules sont considérés autour de l’ego véhicule

Dans cette étude, nous concevons l’espace d’action dans la direction latérale, afin qu’un agent puisse apprendre quand et comment effectuer un changement de voie. Pour la commande latérale, nous avons trois actions différentes 0, 1, 2. Le "0" indique le maintien de la voie, "1" représente le changement de voie à droite, et "2" indique la manœuvre de changement de voie à gauche. La tâche consiste à apprendre à sélectionner l’une des actions dans l’espace d’action discret. Pour s’assurer que seule une action sûre est exécutée, la vérification de la sécurité est effectuée avant que l’action discrète ne soit réalisée.

#### 4.4.4 La fonction de récompense

La fonction de récompense est un attribut important de l'apprentissage par renforcement. La fonction de récompense est conçue pour intégrer les objectifs clés de cette étude, qui est de développer une stratégie de changement de voie automatisée centrée sur la sécurité, l'efficacité et le confort. La fonction de récompense prendra donc en compte des critères liés à trois objectifs. La fonctions de récompense  $R_T$  totale sera la somme des sous-fonctions décrites ci-après :

##### 4.4.4.a Sécurité

Dans le problème de la prise de décision de changement de voie, la sécurité est l'une des questions les plus cruciales pour les véhicules autonomes. Ces dernier sont capables de faire le choix approprié pour éviter la collision. Pour éviter les collisions, il faut que les interdistances restent supérieures à un seuil minimal. De plus, le problème de l'apprentissage par renforcement avec récompense clairsemée est toujours un problème difficile, ce qui signifie que l'environnement avec récompense clairsemée est difficile à entraîner un agent DRL. Afin d'éviter les collisions, la récompense  $R_d$  sur la distance est conçue comme suit :

$$R_d = -(25 - d_{\text{proche}}) \times 0.3 \quad (4.1)$$

Où  $d_{\text{proche}}$  est la distance minimale en mètre entre l'ego véhicule et les autres véhicules avant dans la même voie. Différentes fonctions de récompense non linéaires de pénalité de distance ont été essayées dans l'expérience (polynomiale, exponentielle) mais nous trouvons que la fonction de récompense linéaire est suffisante pour que les agents apprennent la politique.

Et voici la fonction de pénalité de collision  $R_p$  qui est choisie constante comme suit :

$$R_p = -10 \quad (4.2)$$

Ainsi, la fonction de pénalité de la distance et celle de la collision ci-dessus peuvent garantir toutes les conditions. De plus, ces fonctions de récompense indiquent la priorité de ces tâches : l'évitement de la collision est plus important que tout.

##### 4.4.4.b Temps de changement de voie

Afin d'éviter que le véhicule ne change constamment de voie, la fonction de pénalité est conçue pour le changement de voie  $R_{\text{change}}$ . Elle vise à apprendre à l'agent à rester sur sa voie jusqu'à ce qu'il doive effectuer un changement pour éviter une collision. De plus, compte tenu de certaines actions de violation comme le fait que le véhicule ego ne peut pas changer de voie à gauche s'il se trouve sur la voie la plus à droite, nous définissons une fonction de récompense de pénalité  $R_{\text{vio}}$ .

$$\begin{cases} R_{\text{change}} = -2 \\ R_{\text{vio}} = -5 \end{cases} \quad (4.3)$$

##### 4.4.4.c Le confort

Pour éviter un changement significatif pendant les changements de voie du véhicules autonomes, la dérivée de l'accélération est utilisée comme indice pour évaluer le confort. la dérivée de l'accélération reflète le taux de changement de l'accélération. Plus cet indice

est faible dans le processus de changement de voie, plus le processus d'accélération du véhicule est doux, assurant ainsi un meilleur confort de conduite. La fonction de récompense pour prendre en compte le confort de l'agent est la suivante :

$$R_{\text{confort}} = -0.2 \times \dot{a} \quad (4.4)$$

Où  $\dot{a}$  est le jerk ou la dérivée de l'accélération en  $m/s^3$

La fonction de récompense totale est présentée par la somme de toutes les fonctions de récompense :

$$R_T = R_d + R_p + R_{\text{change}} + R_{\text{vio}} + R_{\text{confort}} \quad (4.5)$$

#### 4.4.5 Résultats et discussion

Après quelques essais et une simple optimisation par grille de recherche, nous sommes arrivés à un perceptron multicouche (MLP) dense et simple avec {64, 128, 128, 64} comme nombre de neurones dans quatre couches cachées, tous activés avec Relu comme fonction d'activation. La simplicité du réseau permet non seulement d'accélérer le temps d'apprentissage, mais aussi d'éviter le sur-apprentissage. Le réseau de neurones reçoit un vecteur d'observation et génère une action à exécuter.

Après avoir mis en place tous les composants de notre système, nous avons entraîné un agent RL nommé agent I pendant 2000 pas de temps en utilisant l'algorithme DRL.

Le résultat de la fonction de coût du modèle pour chaque entraînement est l'une des principales mesures d'évaluation. Il permet de s'assurer que le modèle atteint la convergence tout au long de l'entraînement. En outre, le nombre des changements de voie de l'agent ainsi que la moyenne des actions sans collisions pendant l'entraînement sont des mesures pour évaluer le taux d'apprentissage optimal pour atteindre la convergence. La récompense de chaque épisode est une autre métrique cruciale pour mesurer la capacité du modèle à apprendre la politique optimale.

Type de trafic	Nombre de véhicules $n$
Trafic fluide	20
Trafic moyen	30
Trafic dense	40

TABLEAU 4.6 – Le nombre de véhicule pour chaque type de trafic

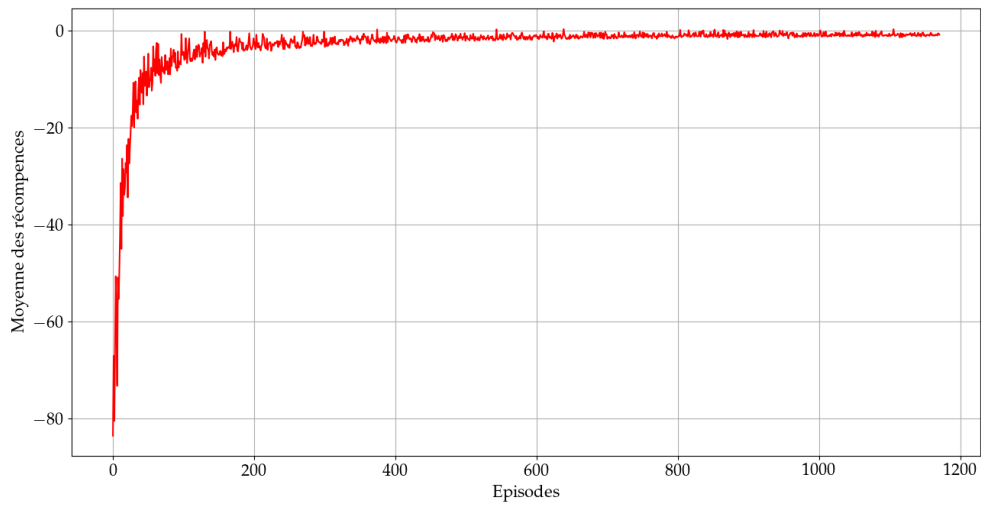
Nous avons fait varier le nombre de voitures simulées dans la scène pour évaluer la capacité de l'agent à apprendre le changement de voie dans toutes les situations (trafic fluide, trafic moyen et trafic dense). Le nombre de véhicules simulés pour chaque situation est présenté par le tableau 4.6.

La récompense de chaque épisode est le cumul des récompenses reçues. L'agent reçoit une récompense lorsqu'il fait un bon changement de voie et reçoit une pénalité dans le cas d'une collision. D'autres pénalités, telles que le non respect de la distance minimale ou le manque du confort pendant la manoeuvre sont appliquées. D'après la figure 4.18, la moyenne des récompenses du modèle dans chaque expérience augmente au fil des épisodes d'entraînement. Le score passe d'une échelle négative pour rapprocher de zéro lorsque le modèle a appris la stratégie permettant de maximiser les récompenses et d'éviter les pénalités. Le modèle a réussi à maximiser la récompense dans toutes les situations de trafic malgré l'augmentation du nombre de véhicules.

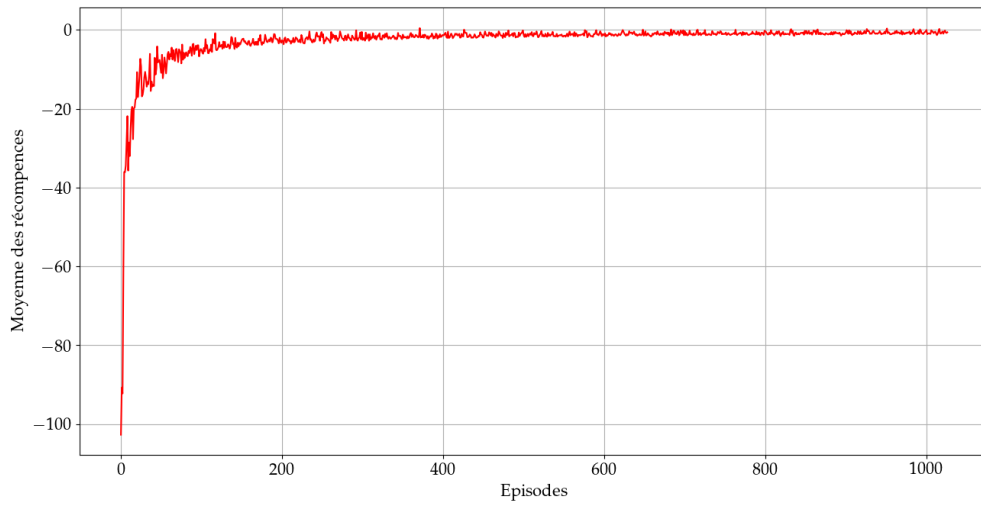
Les résultats de la fonction de coût au cours de l'entraînement sont un autre indicateur permettant de montrer à quel point le comportement d'un modèle est proche de l'environnement. La figure 4.19 montre que le coût d'entraînement diminue dans l'ensemble. Les pics de pertes d'entraînement sont dus à la mise à jour régulière du réseau de neurones. On peut voir clairement que la courbe de chaque cas converge pendant l'entraînement, ce qui montre l'efficacité du système proposé.

Afin de suivre l'évolution des actions de l'agent, nous avons tracé le nombre de changements de voie (figure 4.20) et des actions sans collisions (figure 4.21) à chaque épisode. On remarque que l'agent optimise ses actions à chaque épisode, le nombre de changement de voie converge vers un intervalle fixe après plusieurs itérations d'entraînement. On peut également voir que le nombre des actions sans collision converge pour arriver en moyenne à 97% d'actions sans collision. A la fin de l'entraînement, l'agent apprend à réaliser des changements de voie nécessaire avec un minimum de collision.

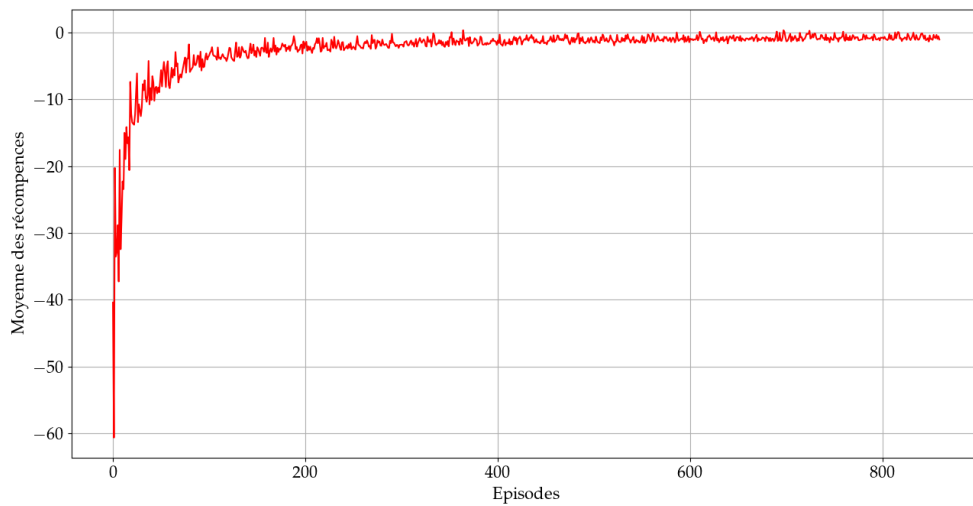
Sur la base des données d'entraînement présentées, il est très clair que le modèle dans chaque expérience a réussi à apprendre la politique optimale pour effectuer des bonnes manoeuvres de changement de voie et minimiser le nombre de collisions. Afin de maximiser sa récompense, l'agent a appris à exécuter l'action optimale à un état donné de l'environnement pour chaque expérience. Les métriques mesurées pendant la phase d'apprentissage sont utilisées pour suivre le processus d'apprentissage des modèles jusqu'à la convergence. Les résultats montrent que chaque modèle converge pour générer la politique optimale pour l'agent-voiture autonome.



(a)  $n=20$

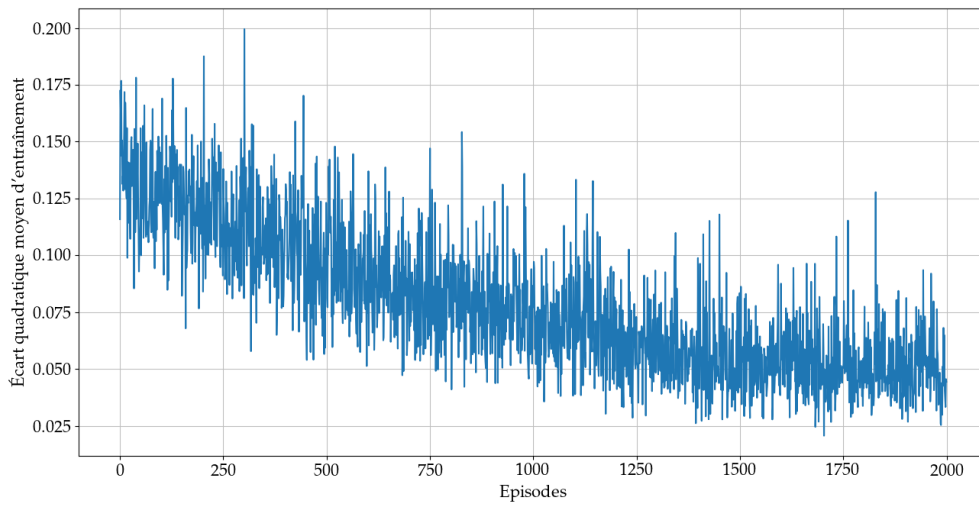


(b)  $n=30$

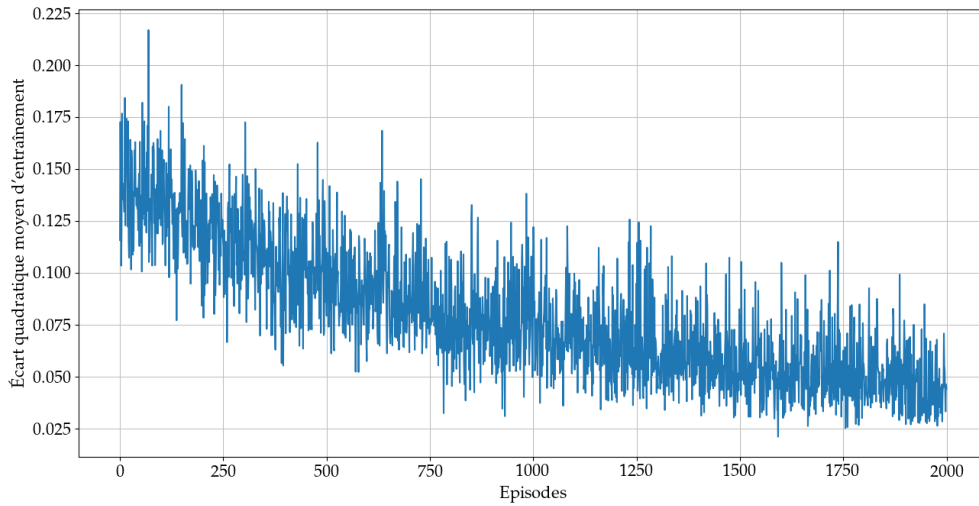


(c)  $n=40$

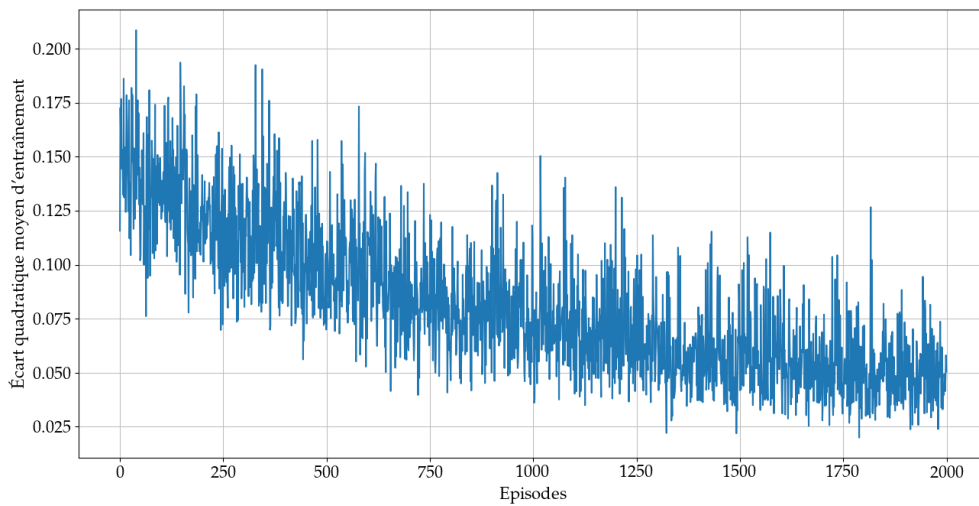
FIGURE 4.18 – Moyennes des récompenses de l'agent pendant l'entraînement)



(a)  $n=20$

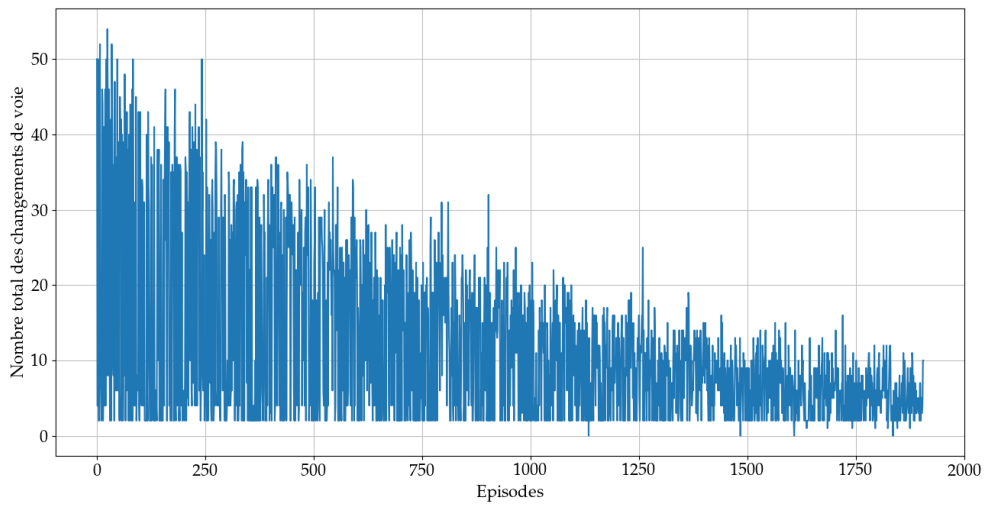


(b)  $n=30$

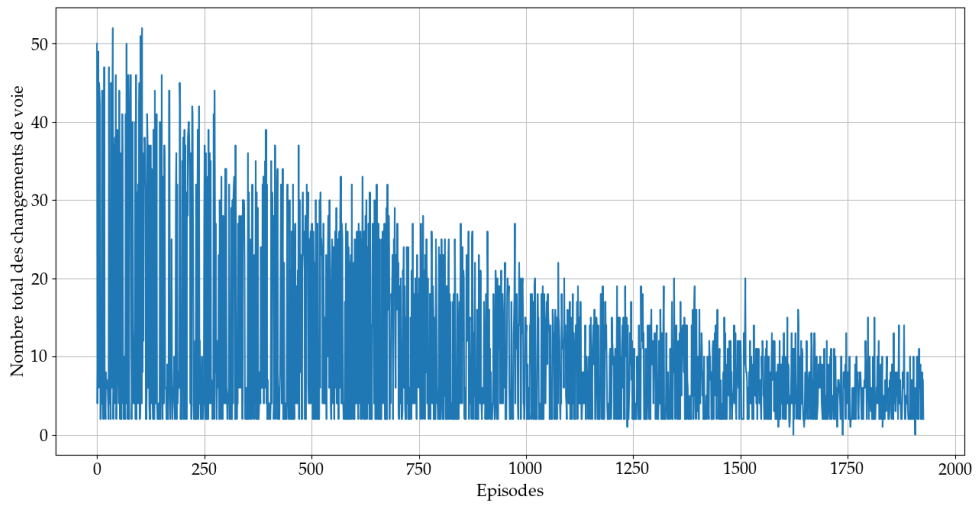


(c)  $n=40$

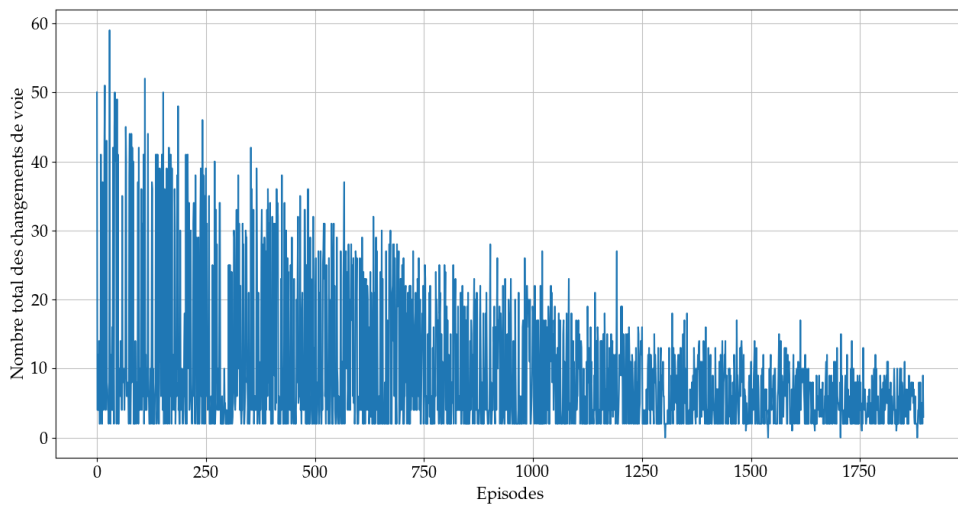
FIGURE 4.19 – Résultats d'entraînement - La fonction de coût(écart quadratique moyen)



(a)  $n=20$

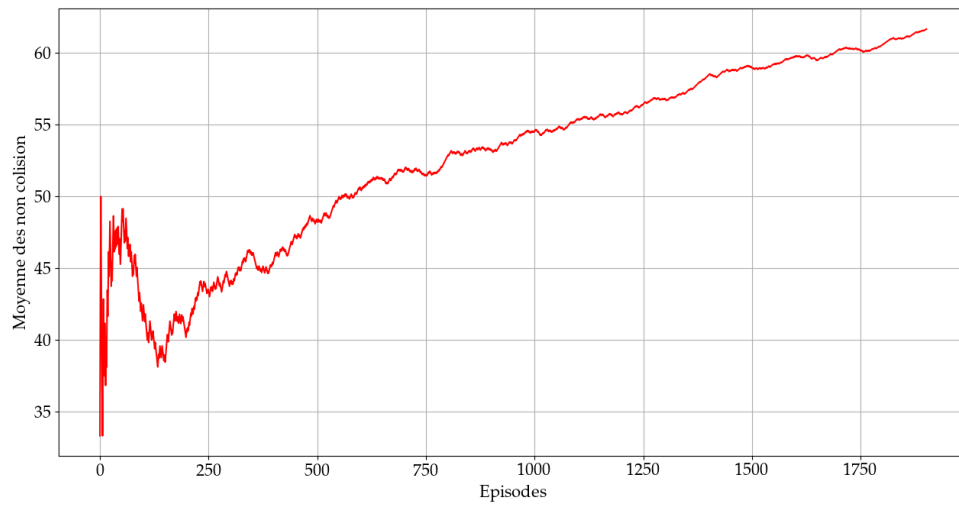


(b)  $n=30$

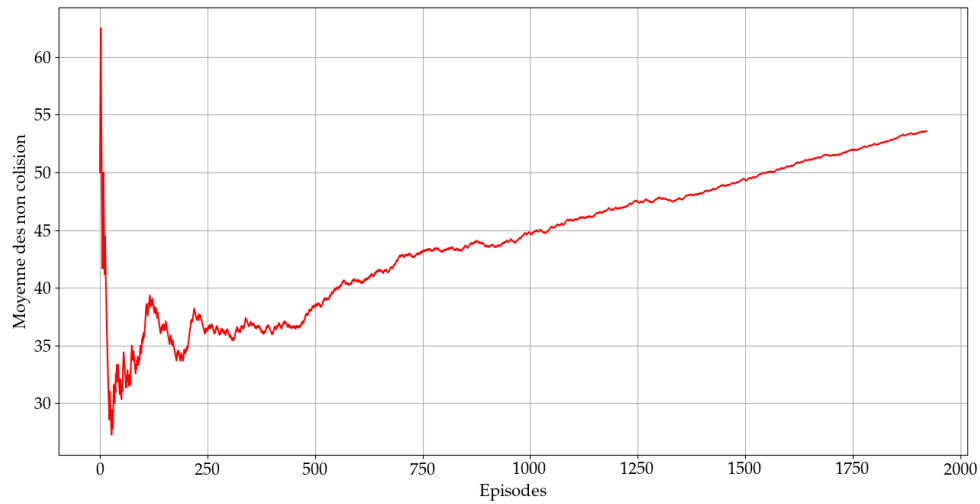


(c)  $n=40$

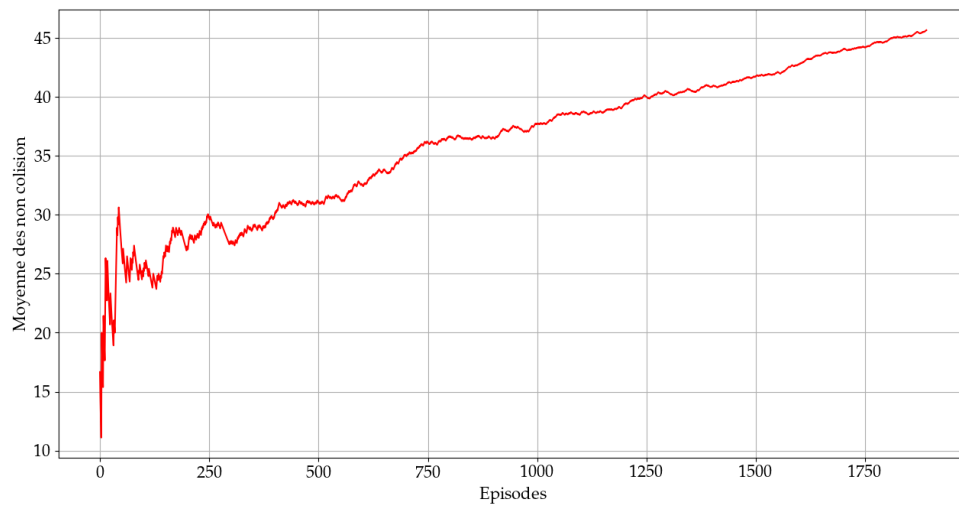
FIGURE 4.20 – Nombre de changements de voie effectués par l'agent pendant l'entraînement



(a)  $n=20$



(b)  $n=30$



(c)  $n=40$

FIGURE 4.21 – Moyenne des non collisions de l'agent pendant l'entraînement



## 4.5 Conclusion

Dans ce chapitre, nous avons présenté des solutions pour combler le fossé entre la perception et la planification de trajectoire en proposant des approches de prédiction de trajectoire des véhicules adjacents. Nous avons proposé deux techniques de prédiction, basées sur le deep learning, et en particulier les réseaux de neurones récurrents. Dans la première approche, des réseaux de neurones récurrents de type LSTM et GRU sont entraînés pour construire des modèles de conduite à partir des données d'apprentissage, afin de prédire les états futurs d'un véhicule cible. Il a été montré que ces techniques fonctionnent de manière fiable pour les prédictions à moyen terme (jusqu'à 5 s). Les résultats montrent une performance élevée dans des situations de changement de voie par rapport aux différentes méthodes existantes. Par conséquent, nous avons proposé une approche hybride qui intègre deux modules de prédiction des intentions : la classification des manœuvres basée sur les réseaux de neurones et la prédiction de trajectoires basée sur les réseaux de mémoire à long court terme (LSTM) pour obtenir les futures positions d'un véhicule cible. L'approche proposée est validée expérimentalement en utilisant des données du démonstrateur VEDECOM. La validation sur des données totalement différentes des données utilisées pour l'apprentissage fait preuve de la robustesse de la technique proposée. Les résultats démontrent que l'approche permet de prédire l'intention du conducteur de changer la voie en moyenne 2,2 secondes à l'avance. Les erreurs de la moyenne quadratique (RMS) des positions latérales et longitudinales sont de 0,30 m et 3,1 m respectivement.

Dans le deuxième volet de ce chapitre, nous avons présenté également la mise en œuvre d'un modèle d'apprentissage profond entraîné par apprentissage par renforcement pour la prise de décision de changement de voie pour un véhicule autonome sur une voie rapide à plusieurs voies. Cette mise en œuvre propose l'utilisation d'une technique d'apprentissage profond de pointe pour aborder des tâches complexes avec un objectif clair. Le modèle généré à partir de l'expérience est doté de hautes performances dans la réalisation de son objectif d'apprendre le changement de voie en toute sécurité. Le modèle peut être appliqué avec des agents intelligents artificiels pour contrôler une voiture autonome dans un environnement dynamique. Les modèles générés ont été entraînés dans l'environnement de simulation dans lequel le véhicule autonome reçoit les données des véhicules adjacents et apprend à prendre la meilleure décision. En outre, le modèle proposé a placé la minimisation des collisions comme objectif principal de l'apprentissage par renforcement. Cependant, le confort de l'utilisateur de la voiture autonome est un autre aspect important à privilégier comme l'un des objectifs de l'apprentissage par renforcement.

## Conclusion générale et perspectives

Ces travaux de thèse avaient pour objectif principal la mise au point des approches de prédiction des intentions des véhicules pour la conduite autonome en se basant sur l'intelligence artificielle. Ces travaux ont été réalisés dans le cadre d'une collaboration entre l'équipe du projet "Véhicule Autonome et Connecté" de l'Institut VEDECOM, le laboratoire ESTACA-LAB de l'école supérieure des techniques aéronautiques et de construction automobile, et le laboratoire LCOMS de l'université de Lorraine et visent à maximiser la fiabilité des véhicules autonomes. Les véhicules autonomes ne sont pas encore disponibles. Cela s'explique par le fait qu'un certain nombre de défis n'ont pas encore été relevés pour garantir que les véhicules autonomes puissent rouler en toute sécurité et efficacement sur les voies publiques. Cette thèse explore les raisons pour lesquelles ce problème n'est toujours pas résolu, et présente des méthodes de prédiction de l'intention du véhicule. La thèse s'est concentrée sur la conduite sur autoroutes, car il s'agit d'un scénario très complexe dans lequel il est important de prédire les actions des conducteurs humains.

La démarche que nous avons menée a conduit au développement de deux types d'approches : la reconnaissance de la situation de conduite et la prédiction de trajectoire. Pour ce qui concerne la reconnaissance de la situation de conduite, nous avons divisé ce problème en trois parties. La première partie vise à détecter et anticiper la réalisation de la manœuvre en se basant sur l'apprentissage automatique. Des approches basées sur la machine à vecteurs de support et le réseau de neurones artificiels sont proposées afin de classer trois types de situation de conduite : changement de voie à droite, changement de voie à gauche et maintien de la voie. Les résultats obtenus montrent l'efficacité des deux approches avec une meilleure performance pour les réseaux de neurones. Dans la deuxième partie, nous avons proposé une approche de reconnaissance de la phase de la manœuvre afin de détailler l'évolution de la réalisation d'un changement de voie. Contrairement aux autres approches de classification des manœuvres, l'approche proposée consiste à diviser la manœuvre de changement de voie en trois phases (préparation, insertion et ajustement) et à identifier chaque étape en utilisant un modèle basé sur les réseaux de neurones profonds. Les modèles sont entraînés et testés en utilisant des données de conduite réelle. Ils ont montré des performances satisfaisantes dans la classification et l'anticipation des changements de voie.

La dernière partie présente une méthode de reconnaissance du style de conduite. Les algorithmes de clustering spectral et K-means sont utilisés à cette fin. Les résultats de clustering, selon lesquels les conducteurs se distinguent en fonction de leurs vitesses la-

térales et d'angles de lacet, montrent que K-means est capable de classer les conducteurs agressifs et les conducteurs calmes avec une efficacité meilleure à celle du clustering spectral.

La seconde problématique abordée dans ces travaux de thèse porte sur la prédiction de trajectoire. Dans ce manuscrit, nous avons présenté des solutions capables d'analyser la trajectoire passée d'un véhicule afin de prédire une trajectoire future en proposant des approches de prédiction de trajectoire des véhicules adjacents. Nous avons mis en oeuvre des modèles basés sur les réseaux de neurones récurrents. Afin d'améliorer la précision de prédiction en phase de changement de voie, nous avons inclus un module de classification des manœuvres. En combinant la classification des manœuvres et la prédiction de trajectoire, le modèle a été capable de générer une trajectoire très proche de la vérité de terrain et sans retard de prédiction. Pour mettre en pratique le modèle hybride proposé, ce modèle a été validé en utilisant des données réelles issues du démonstrateur VEDECOM. Les résultats obtenus ont montré l'efficacité du modèle hybride. La dernière partie de cette thèse a été consacrée à la mise en oeuvre d'un modèle d'apprentissage profond basé sur l'apprentissage par renforcement pour la prise de décision de changement de voie pour le véhicule autonome sur autoroute à plusieurs voies. Un agent autonome a été entraîné dans un environnement de simulation pour gérer les décisions de changement de voie. Dans un cas de conduite sur autoroute, il est montré que la méthode est capable d'effectuer des changements de voie en toute sécurité en minimisant le nombre de collisions et garantir plus de confort aux passagers.

Par ailleurs, à l'issue de ces travaux de thèse, même si de nombreux résultats ont été exposés, des pistes restent ouvertes et plusieurs perspectives sont envisagées afin d'améliorer les approches proposées dans cette thèse. Nous avons identifié des axes de recherche qui sont formalisés de la manière suivante :

- Comme mentionné dans ce manuscrit, les approches développées sont testées sur des scénarios d'autoroute. Dans la suite de ces travaux, nous proposons d'étendre ces approches pour traiter la problématique de la prédiction des intentions dans des milieux urbains et pour d'autres types d'usagers de la route (vélo, camion ...)
- La communication de véhicule à véhicule (V2V) peut être considérée comme une solution pour améliorer l'efficacité des véhicules autonomes pendant les interactions avec les autres véhicules. En exigeant aux véhicules qu'ils communiquent leurs intentions les uns avec les autres, ces données de communication sont une source supplémentaire d'informations sur les autres véhicules. Dans ce cadre, il est possible de prévoir une étude comparative entre les résultats de prédiction des méthodes proposées et les données communiquées par les autres véhicules via la communication V2V.
- Dans le cadre de cette thèse, nous avons exploité les données actuelles des véhicules pour développer une méthode de prise de décision basée sur l'apprentissage par renforcement. Afin de garantir plus d'efficacité et de sécurité de conduite, nous envisageons d'intégrer les données de prédiction dans la politique de prise de décision.
- Une autre méthode hybride de prédiction de trajectoire des véhicules peut être proposée en combinant les modèles basés sur la physique et les modèles basés sur les données. La prédiction de trajectoire basée sur la physique peut garantir une précision à court terme en tenant compte des changements de la dynamique du véhicule, tandis que les approches de prédiction basées sur les données ont une vision à long terme des trajectoires.

- Sur le plan expérimental, une connaissance des conditions météorologiques de la route reste une tâche réalisable. L'exploitation de ces informations peut augmenter la précision des prédictions. En effet, l'idée est d'établir une corrélation entre les phénomènes physiques et le comportement d'un conducteur dans chaque situation météorologique.
- Pour atteindre des niveaux acceptables de sécurité et de fiabilité, il sera nécessaire de procéder à des travaux de vérification et de validation considérables. A moyen et voire à long terme, une validation en ligne dans des scénarios de conduite réelle (route ouverte) est à prévoir pour tester l'efficacité des approches proposées en temps réel.



# Bibliographie

- Aeberhard MICHAEL, Kühbeck T, Seidl B, Friedl M, Thomas J, Scheickl O.* Automated Driving with ROS at BMW // ROSCon 2015 Hamburg, Germany. 2015. [12](#)
- Aeberhard Michael, Kaempchen Nico.* High-level sensor data fusion architecture for vehicle surround environment perception // Proc. 8th Int. Workshop Intell. Transp. 665. 2011. [8](#)
- Agroverse .* Overview of Agroverse dataset. 2021. [Online; accessed 12-mai-2021]. [vii](#), [38](#)
- Altché Florent, La Fortelle Arnaud de.* An LSTM network for highway trajectory prediction // 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC). 2017. 353–359. [114](#), [115](#), [116](#), [117](#), [124](#)
- Althoff Matthias, Mergel Alexander.* Comparison of Markov chain abstraction and Monte Carlo simulation for the safety assessment of autonomous cars // IEEE Transactions on Intelligent Transportation Systems. 2011. 12, 4. 1237–1247. [17](#), [21](#)
- Ammoun Samer, Nashashibi Fawzi.* Real time trajectory prediction for collision risk estimation between vehicles // 2009 IEEE 5th International Conference on Intelligent Computer Communication and Processing. 2009. 417–422. [21](#)
- Aoude Georges, Joseph Joshua, Roy Nicholas, How Jonathan.* Mobile agent trajectory prediction using Bayesian nonparametric reachability trees // Infotech@ Aerospace 2011. 2011. 1512. [24](#)
- Arulkumaran Kai, Deisenroth Marc Peter, Brundage Miles, Bharath Anil Anthony.* Deep reinforcement learning : A brief survey // IEEE Signal Processing Magazine. 2017. 34, 6. 26–38. [62](#)
- Atiya Amir F.* Bankruptcy prediction for credit risk using neural networks : A survey and new results // IEEE Transactions on neural networks. 2001. 12, 4. 929–935. [42](#)
- Balaprakash Prasanna, Salim Michael, Uram Thomas D, Vishwanath Venkat, Wild Stefan M.* DeepHyper : Asynchronous hyperparameter search for deep neural networks // 2018 IEEE 25th international conference on high performance computing (HiPC). 2018. 42–51. [48](#)

- Bayes FRS*. An essay towards solving a problem in the doctrine of chances // *Biometrika*. 1958. 45, 3-4. 296–315. [44](#)
- Behrisch Michael, Bieker Laura, Erdmann Jakob, Krajzewicz Daniel*. SUMO–simulation of urban mobility : an overview // *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*. 2011. [130](#)
- Benterki Abdelmoudjib, Boukhnifer Moussa, Judalet Vincent, Choubeila Maaoui*. Prediction of surrounding vehicles lane change intention using machine learning // *2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems : Technology and Applications (IDAACS)*. 2. 2019. 839–843. [81](#)
- Benterki Abdelmoudjib, Boukhnifer Moussa, Judalet Vincent, Maaoui Choubeila*. Artificial intelligence for vehicle behavior anticipation : Hybrid approach based on maneuver classification and trajectory prediction // *IEEE Access*. 2020. 8. 56992–57002. [88](#)
- Berndt Holger, Emmert Jorg, Dietmayer Klaus*. Continuous driver intention recognition with hidden markov models // *2008 11th International IEEE Conference on Intelligent Transportation Systems*. 2008. 1189–1194. [15](#), [24](#)
- Bhardwaj Nitin, Langlois Robert E, Zhao Guijun, Lu Hui*. Kernel-based machine learning protocol for predicting DNA-binding proteins // *Nucleic Acids Research*. 2005. 33, 20. 6486–6493. [42](#)
- Bishop Christopher M*. *Pattern recognition and machine learning*. 2006. [42](#)
- Blanco Jose Luis, Fuchs Steffen, Parsons Matthew, Ribeirinho Maria João, others*. Artificial intelligence : Construction technology's next frontier // *Building Economist, The*. 2018. Sep 2018. 7. [53](#)
- Bradley Andrew P*. The use of the area under the ROC curve in the evaluation of machine learning algorithms // *Pattern recognition*. 1997. 30, 7. 1145–1159. [57](#), [90](#)
- Brand Matthew, Oliver Nuria, Pentland Alex*. Coupled hidden Markov models for complex action recognition // *Proceedings of IEEE computer society conference on computer vision and pattern recognition*. 1997. 994–999. [15](#), [25](#)
- Brownlee Jason*. *Time Series Forecasting as Supervised Learning*. 2016. [114](#)
- Brownlee Jason*. *Introduction to time series forecasting with python : how to prepare data and develop models to predict the future*. 2017. [54](#)
- Buzan Dan, Sclaroff Stan, Kollios George*. Extraction and clustering of motion trajectories in video // *Proceedings of the 17th International Conference on Pattern Recognition*, 2004. ICPR 2004. 2. 2004. 521–524. [23](#)
- Campbell Kenneth L*. The SHRP 2 naturalistic driving study : Addressing driver performance and behavior in traffic safety // *Tr News*. 2012. 282. [15](#)
- Castignani German, Frank Raphaël, Engel Thomas*. Driver behavior profiling using smartphones // *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. 2013. 552–557. [16](#)
- Caveney Derek*. Cooperative vehicular safety applications // *IEEE Control Systems Magazine*. 2010. 30, 4. 38–53. [112](#)

- Chang Chih-Chung, Lin Chih-Jen.* LIBSVM : a library for support vector machines // ACM transactions on intelligent systems and technology (TIST). 2011. 2, 3. 1–27. [80](#)
- Chang Ming-Fang, Lambert John, Sangkloy Patsorn, Singh Jagjeet, Bak Slawomir, Hartnett Andrew, Wang De, Carr Peter, Lucey Simon, Ramanan Deva, others .* Argoverse : 3d tracking and forecasting with rich maps // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019. 8748–8757. [38](#)
- Chaurasia Brijesh Kumar, Verma Shekhar.* Haste induced behavior and VANET communication // 2009 IEEE International Conference on Vehicular Electronics and Safety (ICVES). 2009. 19–24. [112](#)
- Chetlur Sharan, Woolley Cliff, Vandermersch Philippe, Cohen Jonathan, Tran John, Catanzaro Bryan, Shelhamer Evan.* cudnn : Efficient primitives for deep learning // arXiv preprint arXiv :1410.0759. 2014. [116](#)
- Cho Kyunghyun, Van Merriënboer Bart, Bahdanau Dzmitry, Bengio Yoshua.* On the properties of neural machine translation : Encoder-decoder approaches // arXiv preprint arXiv :1409.1259. 2014. [52](#), [53](#)
- Cho Sung-Bae, Won Hong-Hee.* Machine learning in DNA microarray analysis for cancer classification // Proceedings of the First Asia-Pacific Bioinformatics Conference on Bioinformatics 2003-Volume 19. 2003. 189–198. [42](#)
- Chollet François, others .* Keras : The python deep learning library // Astrophysics Source Code Library. 2018. ascl–1806. [116](#)
- Chung Junyoung, Gulcehre Caglar, Cho KyungHyun, Bengio Yoshua.* Empirical evaluation of gated recurrent neural networks on sequence modeling // arXiv preprint arXiv :1412.3555. 2014. [51](#), [52](#), [124](#)
- Coifman Benjamin, Li Lizhe.* A critical evaluation of the Next Generation Simulation (NG-SIM) vehicle trajectory dataset // Transportation Research Part B : Methodological. 2017. 105. 362–377. [18](#)
- Boosting Car Safety in the EU. // . 2016 (accessed December 12, 2016). <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52016DC0787>. [13](#)
- Cooijmans Tim, Ballas Nicolas, Laurent César, Gülçehre Çağlar, Courville Aaron.* Recurrent batch normalization // arXiv preprint arXiv :1603.09025. 2016. [51](#)
- Corazza Marco, Sangalli Andrea.* Q-Learning and SARSA : a comparison between two intelligent stochastic control approaches for financial trading // University Ca'Foscari of Venice, Dept. of Economics Research Paper Series No. 2015. 15. [62](#)
- Cristianini Nello, Shawe-Taylor John, others .* An introduction to support vector machines and other kernel-based learning methods. 2000. [79](#)
- Dayan Peter, Sahani Maneesh, Deback Grégoire.* Unsupervised learning // The MIT encyclopedia of the cognitive sciences. 1999. 857–859. [58](#)
- Deo Nachiket, Rangesh Akshay, Trivedi Mohan M.* How would surround vehicles move? a unified framework for maneuver classification and motion prediction // IEEE Transactions on Intelligent Vehicles. 2018. 3, 2. 129–140. [18](#), [26](#), [67](#)



- Deo Nachiket, Trivedi Mohan M.* Multi-modal trajectory prediction of surrounding vehicles with maneuver based lstms // 2018 IEEE Intelligent Vehicles Symposium (IV). 2018. 1179–1184. [18](#)
- Dogan Ürün, Edelbrunner Johann, Iossifidis Ioannis.* Autonomous driving : A comparison of machine learning techniques by means of the prediction of lane change behavior // 2011 IEEE International Conference on Robotics and Biomimetics. 2011. 1837–1843. [16](#)
- Doshi Anup, Trivedi Mohan Manubhai.* On the roles of eye gaze and head dynamics in predicting driver's intent to change lanes // IEEE Transactions on Intelligent Transportation Systems. 2009. 10, 3. 453–462. [15](#)
- Dosovitskiy Alexey, Ros German, Codevilla Felipe, Lopez Antonio, Koltun Vladlen.* CARLA : An open urban driving simulator // arXiv preprint arXiv :1711.03938. 2017. [19](#)
- Du Yong, Wang Wei, Wang Liang.* Hierarchical recurrent neural network for skeleton based action recognition // Proceedings of the IEEE conference on computer vision and pattern recognition. 2015. 1110–1118. [48](#)
- ELIASSON PHILIP, Rosen Niklas.* Efficient K-means clustering and the importance of seeding. 2013. [95](#)
- Elander James, West Robert, French Davina.* Behavioral correlates of individual differences in road-traffic crash risk : an examination of methods and findings. // Psychological bulletin. 1993. 113, 2. 279. [95](#)
- Elbanhawi Mohamed, Simic Milan, Jazar Reza.* In the passenger seat : investigating ride comfort measures in autonomous cars // IEEE Intelligent transportation systems magazine. 2015. 7, 3. 4–17. [1](#)
- Elzen Christopher L Van Dan, Cheok Ka Chai, Radovnikovich Micho.* Lane keeping system and lane centering system. XI 10 2015. US Patent 9,180,908. [6](#)
- Engström Johan, Johansson Emma, Östlund Joakim.* Effects of visual and cognitive load in real and simulated motorway driving // Transportation research part F : traffic psychology and behaviour. 2005. 8, 2. 97–120. [67](#)
- Eskandarian Azim.* Handbook of intelligent vehicles. 2. 2012. [75](#)
- FHWA .* US Department of Transportation – NGSIM – Next Generation SIMulation. 2008. [Online; accessed 31.10.08]. [vii](#), [viii](#), [17](#), [36](#), [37](#), [71](#)
- Field David T, Wann John P.* Perceiving time to collision activates the sensorimotor cortex // Current Biology. 2005. 15, 5. 453–458. [114](#)
- Fitch GM, Lee SE, Klauer S, Hankey J, Sudweeks J, Dingus T.* Analysis of lane-change crashes and near-crashes // US Department of Transportation, National Highway Traffic Safety Administration. 2009. [2](#)
- Flanagan J Randall, Vetter Philipp, Johansson Roland S, Wolpert Daniel M.* Prediction precedes control in motor learning // Current biology. 2003. 13, 2. 146–150. [14](#)
- Frazzoli Emilio, Dahleh Munther A, Feron Eric.* Real-time motion planning for agile autonomous vehicles // Journal of guidance, control, and dynamics. 2002. 25, 1. 116–129. [8](#)

- Galceran Enric, Cunningham Alexander G, Eustice Ryan M, Olson Edwin.* Multipolicy decision-making for autonomous driving via changepoint-based behavior prediction : Theory and experiment // *Autonomous Robots*. 2017. 41, 6. 1367–1382. [19](#)
- Ghumman Usman Shaukat, Kunwar Faraz, Benhabib Beno, others .* Guidance-based on-line motion planning for autonomous highway overtaking. 2008. [111](#)
- Gosavi Abhijit.* Reinforcement learning : A tutorial survey and recent advances // *INFORMS Journal on Computing*. 2009. 21, 2. 178–192. [62](#)
- Gurghian Alexandru, Koduri Tejaswi, Bailur Smita V, Carey Kyle J, Murali Vidya N.* Deep-planes : End-to-end lane position estimation using deep neural networks // *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2016. 38–45. [17](#)
- Gurney Kevin.* An introduction to neural networks. 1997. [80](#)
- Ha David, Dai Andrew, Le Quoc V.* Hypernetworks // arXiv preprint arXiv :1609.09106. 2016. [51](#)
- Ha David, Eck Douglas.* A neural representation of sketch drawings // arXiv preprint arXiv :1704.03477. 2017. [19](#)
- Herrmann Andreas, Brenner Walter, Stadler Rupert.* Autonomous driving : how the driverless revolution will change the world. 2018. [7](#)
- Hillenbrand Jrg, Spieker Andreas M, Kroschel Kristian.* A multilevel collision mitigation approach—Its situation assessment, decision making, and performance tradeoffs // *IEEE Transactions on intelligent transportation systems*. 2006. 7, 4. 528–540. [109](#)
- Hochreiter Sepp, Schmidhuber Jürgen.* Long short-term memory // *Neural computation*. 1997. 9, 8. 1735–1780. [51](#)
- Houenou Adam, Bonnifait Philippe, Cherfaoui Véronique, Yao Wen.* Vehicle trajectory prediction based on motion model and maneuver recognition // 2013 IEEE/RSJ international conference on intelligent robots and systems. 2013. 4363–4369. [15](#)
- Hsu Chih-Wei, Chang Chih-Chung, Lin Chih-Jen, others .* A practical guide to support vector classification. 2003. [79](#)
- Hyndman Rob J, Athanasopoulos George.* Forecasting : principles and practice. 2018. [55](#)
- Joel Grus.* Data Science from Scratch. 2015. [96](#)
- Julier Simon J, Uhlmann Jeffrey K.* New extension of the Kalman filter to nonlinear systems // *Signal processing, sensor fusion, and target recognition VI*. 3068. 1997. 182–193. [28](#)
- Käfer Eugen, Hermes Christoph, Wöhler Christian, Ritter Helge, Kummert Franz.* Recognition of situation classes at road intersections // 2010 IEEE International Conference on Robotics and Automation. 2010. 3960–3965. [25](#)
- Kala Rahul.* On-road intelligent vehicles : Motion planning for intelligent transportation systems. 2016. [6](#)

- Kanaris Alexander, Kosmatopoulos Elias B, Loannou Petros A.* Strategies and spacing requirements for lane changing and merging in automated highway systems // IEEE transactions on vehicular technology. 2001. 50, 6. 1568–1581. [111](#)
- Kasper Dietmar, Weidl Galia, Dang Thao, Breuel Gabi, Tamke Andreas, Wedel Andreas, Rosenstiel Wolfgang.* Object-oriented Bayesian networks for detection of lane change maneuvers // IEEE Intelligent Transportation Systems Magazine. 2012. 4, 3. 19–31. [16](#)
- Kesting Arne, Treiber Martin, Helbing Dirk.* General lane-changing model MOBIL for car-following models // Transportation Research Record. 2007. 1999, 1. 86–94. [112](#)
- Khodayari Alireza, Ghaffari Ali, Ameli Sina, Flahatgar Jamal.* A historical review on lateral and longitudinal control of autonomous vehicle motions // 2010 International Conference on Mechanical and Electrical Technology. 2010. 421–429. [10](#)
- Kim ByeoungDo, Kang Chang Mook, Kim Jaekyum, Lee Seung Hi, Chung Chung Choo, Choi Jun Won.* Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network // 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC). 2017. 399–404. [18](#), [26](#), [110](#)
- Kim Jihun, Lee Minho.* Robust lane detection based on convolutional neural network and random sample consensus // International conference on neural information processing. 2014. 454–461. [17](#)
- Kiran B Ravi, Sobh Ibrahim, Talpaert Victor, Mannion Patrick, Al Sallab Ahmad A, Yogamani Senthil, Pérez Patrick.* Deep reinforcement learning for autonomous driving : A survey // IEEE Transactions on Intelligent Transportation Systems. 2021. [129](#)
- Krajewski Robert, Bock Julian, Kloeker Laurent, Eckstein Lutz.* The highd dataset : A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems // 2018 21st International Conference on Intelligent Transportation Systems (ITSC). 2018. 2118–2125. [viii](#), [xi](#), [36](#), [86](#), [87](#), [88](#), [131](#)
- Kuefler Alex, Morton Jeremy, Wheeler Tim, Kochenderfer Mykel.* Imitating driver behavior with generative adversarial networks // 2017 IEEE Intelligent Vehicles Symposium (IV). 2017. 204–211. [17](#), [18](#)
- Kumar Puneet, Perrollaz Mathias, Lefevre Stéphanie, Laugier Christian.* Learning-based approach for online lane change intention prediction // 2013 IEEE Intelligent Vehicles Symposium (IV). 2013. 797–802. [16](#)
- Lapedes A, Barnes Christopher, Burks Christian, Farber R, Sirotkin K.* Application of neural networks and other machine learning algorithms to DNA sequence analysis. 1988. [42](#)
- Laugier Christian, Paromtchik Igor E, Perrollaz Mathias, Yong Mao, Yoder John-David, Tay Christopher, Mekhnacha Kamel, Nègre Amaury.* Probabilistic analysis of dynamic scenes and collision risks assessment to improve driving safety // IEEE Intelligent Transportation Systems Magazine. 2011. 3, 4. 4–19. [110](#)
- LeCun Yann, Bengio Yoshua, Hinton Geoffrey.* Deep learning // nature. 2015. 521, 7553. 436–444. [129](#)

- Lee Donghan, Kwon Youngwook Paul, McMains Sara, Hedrick J Karl.* Convolution neural network-based lane change intention prediction of surrounding vehicles for ACC // 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC). 2017a. 1–6. [17](#)
- Lee John A, Verleysen Michel.* Nonlinear dimensionality reduction. 2007. [59](#)
- Lee Namhoon, Choi Wongun, Vernaza Paul, Choy Christopher B, Torr Philip HS, Chandraker Manmohan.* Desire : Distant future prediction in dynamic scenes with interacting agents // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017b. 336–345. [19](#)
- Lee Yi-Ching, Lee John D, Ng Boyle Linda.* Visual attention in driving : The effects of cognitive load and visual disruption // Human Factors. 2007. 49, 4. 721–733. [14](#)
- Lefèvre Stéphanie, Vasquez Dizan, Laugier Christian.* A survey on motion prediction and risk assessment for intelligent vehicles // ROBOMECH journal. 2014. 1, 1. 1–14. [9](#), [13](#), [15](#), [20](#), [67](#), [75](#), [109](#)
- Leonhardt Veit, Wanielik Gerd.* Feature evaluation for lane change prediction based on driving situation and driver behavior // 2017 20th International Conference on Information Fusion (Fusion). 2017. 1–7. [16](#)
- Lin Chiu-Feng, Ulsoy A Galip, LeBlanc David J.* Vehicle dynamics and external disturbance estimation for vehicle path prediction // IEEE Transactions on Control Systems Technology. 2000. 8, 3. 508–518. [20](#)
- Lipton Zachary C, Berkowitz John, Elkan Charles.* A critical review of recurrent neural networks for sequence learning // arXiv preprint arXiv :1506.00019. 2015. [48](#), [49](#)
- Liu Xiao, Liu Yuanwei, Chen Yue, Hanzo Lajos.* Enhancing the fuel-economy of V2I-assisted autonomous driving : A reinforcement learning approach // IEEE Transactions on Vehicular Technology. 2020. 69, 8. 8329–8342. [1](#)
- Lopez Pablo Alvarez, Behrisch Michael, Bieker-Walz Laura, Erdmann Jakob, Flötteröd Yun-Pang, Hilbrich Robert, Lücken Leonhard, Rummel Johannes, Wagner Peter, Wießner Eva-marie.* Microscopic traffic simulation using sumo // 2018 21st International Conference on Intelligent Transportation Systems (ITSC). 2018. 2575–2582. [131](#)
- Lundberg Scott, Lee Su-In.* A unified approach to interpreting model predictions // arXiv preprint arXiv :1705.07874. 2017. [92](#), [95](#)
- Maaoui Choubeila, Bousefsaf Frédéric, Pruski Alain.* Automatic human stress detection based on webcam photoplethysmographic signals // Journal of Mechanics in Medicine and Biology. 2016. 16, 04. 1650039. [44](#)
- MacQueen James, others .* Some methods for classification and analysis of multivariate observations // Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. 1, 14. 1967. 281–297. [58](#), [59](#)
- Makhoul John, Kubala Francis, Schwartz Richard, Weischedel Ralph, others .* Performance measures for information extraction // Proceedings of DARPA broadcast news workshop. 1999. 249–252. [56](#), [57](#)

- Mandalia Hiren M, Salvucci Mandalia Dario D.* Using support vector machines for lane-change detection // Proceedings of the human factors and ergonomics society annual meeting. 49, 22. 2005. 1965–1969. [16](#), [44](#)
- Mänttari Joonatan, Folkesson John, Ward Erik.* Learning to predict lane changes in highway scenarios using dynamic filters on a generic traffic representation // 2018 IEEE Intelligent Vehicles Symposium (IV). 2018. 1385–1392. [17](#)
- Maurer Markus, Christian Gerdes J, Lenz Barbara, Winner Hermann.* Autonomous driving : technical, legal and social aspects. 2016. [1](#)
- McLachlan Geoffrey J, Lee Sharon X, Rathnayake Suren I.* Finite mixture models // Annual review of statistics and its application. 2019. 6. 355–378. [58](#)
- Meyer-Delius Daniel, Plagemann Christian, Burgard Wolfram.* Probabilistic situation recognition for vehicular traffic scenarios // 2009 IEEE International Conference on Robotics and Automation. 2009. 459–464. [16](#)
- Milan Anton, Rezatofighi S Hamid, Dick Anthony, Reid Ian, Schindler Konrad.* Online multi-target tracking using recurrent neural networks // Proceedings of the AAAI Conference on Artificial Intelligence. 31, 1. 2017. [48](#)
- Miller Ronald, Huang Qingfeng.* An adaptive peer-to-peer collision warning system // Vehicular Technology Conference. IEEE 55th Vehicular Technology Conference. VTC Spring 2002 (Cat. No. 02CH37367). 1. 2002. 317–321. [109](#)
- Mitchell Tom M.* Does machine learning really work? // AI magazine. 1997. 18, 3. 11–11. [42](#)
- Learning driving patterns to support navigation. // . 2004. [42](#)
- Mnih Volodymyr, Kavukcuoglu Koray, Silver David, Graves Alex, Antonoglou Ioannis, Wierstra Daan, Riedmiller Martin.* Playing atari with deep reinforcement learning // arXiv preprint arXiv :1312.5602. 2013. [62](#), [129](#)
- Mnih Volodymyr, Kavukcuoglu Koray, Silver David, Rusu Andrei A, Veness Joel, Bellemare Marc G, Graves Alex, Riedmiller Martin, Fidjeland Andreas K, Ostrovski Georg, others .* Human-level control through deep reinforcement learning // nature. 2015. 518, 7540. 529–533. [64](#), [129](#)
- Montanino Marcello, Punzo Vincenzo.* Making NGSIM data usable for studies on traffic flow theory : Multistep method for vehicle trajectory reconstruction // Transportation Research Record. 2013. 2390, 1. 99–111. [73](#)
- Mood AM, Graybill FA, Boes DC.* Introduction to the Theory of Statistics. Mc-Graw Hill Book Company // Inc., New York. 1963. [56](#)
- Morton Jeremy, Wheeler Tim A, Kochenderfer Mykel J.* Analysis of recurrent neural networks for probabilistic modeling of driver behavior // IEEE Transactions on Intelligent Transportation Systems. 2016. 18, 5. 1289–1298. [18](#)
- Neubig Graham.* Neural machine translation and sequence-to-sequence models : A tutorial // arXiv preprint arXiv :1703.01619. 2017. [18](#), [26](#)

ONISR . L'accidentalité routière en 2019. 2020. [13](#)

*Olabiyi Oluwatobi, Martinson Eric, Chintalapudi Vijay, Guo Rui.* Driver action prediction using deep (bidirectional) recurrent neural network // arXiv preprint arXiv :1706.02257. 2017. [17](#)

*Organisation mondiale de la santé .* Global status report on road safety 2018 : Summary. 2018. [13](#)

*Ortiz Michaël Garcia, Fritsch Jannik, Kummert Franz, Gepperth Alexander.* Behavior prediction at multiple time-scales in inner-city scenarios // 2011 IEEE Intelligent Vehicles Symposium (IV). 2011. 1068–1073. [24](#)

*Pananurak Worrawut, Thanok Somphong, Parnichkun Manukid.* Adaptive cruise control for an intelligent vehicle // 2008 IEEE International Conference on Robotics and Biomimetics. 2009. 1794–1799. [6](#)

*Park Seong Hyeon, Kim ByeongDo, Kang Chang Mook, Chung Chung Choo, Choi Jun Won.* Sequence-to-sequence prediction of vehicle trajectory via LSTM encoder-decoder architecture // 2018 IEEE Intelligent Vehicles Symposium (IV). 2018. 1672–1678. [18](#), [26](#)

*Pendleton Scott Drew, Andersen Hans, Du Xinxin, Shen Xiaotong, Meghjani Malika, Eng You Hong, Rus Daniela, Ang Marcelo H.* Perception, planning, control, and coordination for autonomous vehicles // *Machines*. 2017. 5, 1. 6. [vii](#), [9](#)

*Peng Jinshuan, Guo Yingshi, Fu Rui, Yuan Wei, Wang Chang.* Multi-parameter prediction of drivers' lane-changing behaviour with neural network model // *Applied ergonomics*. 2015. 50. 207–217. [16](#)

*Phillips Derek J, Wheeler Tim A, Kochenderfer Mykel J.* Generalizable intention prediction of human drivers at intersections // 2017 IEEE intelligent vehicles symposium (IV). 2017. 1665–1670. [114](#), [115](#)

*Ponziani Richard.* Turn signal usage rate results : A comprehensive field study of 12,000 observed turning vehicles. 2012. [2](#)

*Ren Kui, Wang Qian, Wang Cong, Qin Zhan, Lin Xiaodong.* The security of autonomous driving : Threats, defenses, and future directions // *Proceedings of the IEEE*. 2019. 108, 2. 357–372. [1](#)

*Ross Catherine, Guhathakurta Subhrajit.* Autonomous vehicles and energy impacts : a scenario analysis // *Energy Procedia*. 2017. 143. 47–52. [1](#)

*Rousseeuw Peter J.* Silhouettes : a graphical aid to the interpretation and validation of cluster analysis // *Journal of computational and applied mathematics*. 1987. 20. 53–65. [96](#)

*Rouzikhah Hossein, King Mark, Rakotonirainy Andry.* The validity of simulators in studying driving behaviours // *Proc. Australas. Road Saf. Res., Policing Educ. Conf.* 2010. 1. [19](#)

*Rumelhart David E, Hinton Geoffrey E, Williams Ronald J.* Learning internal representations by error propagation. 1985. [49](#)

- Sallab Ahmad EL, Abdou Mohammed, Perot Etienne, Yogamani Senthil.* Deep reinforcement learning framework for autonomous driving // *Electronic Imaging*. 2017. 2017, 19. 70–76. [129](#)
- Savitzky Abraham, Golay Marcel JE.* Smoothing and differentiation of data by simplified least squares procedures. // *Analytical chemistry*. 1964. 36, 8. 1627–1639. [73](#)
- Schlechtriemen Julian, Wedel Andreas, Hillenbrand Joerg, Breuel Gabi, Kuhnert Klaus-Dieter.* A lane change detection approach using feature ranking with maximized predictive power // *2014 IEEE Intelligent Vehicles Symposium Proceedings*. 2014. 108–114. [77](#)
- Schlechtriemen Julian, Wirthmueller Florian, Wedel Andreas, Breuel Gabi, Kuhnert Klaus-Dieter.* When will it change the lane? A probabilistic regression approach for rarely occurring events // *2015 IEEE Intelligent Vehicles Symposium (IV)*. 2015. 1373–1379. [16](#), [114](#)
- Schmerling Edward, Leung Karen, Vollprecht Wolf, Pavone Marco.* Multimodal probabilistic model-based planning for human-robot interaction // *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 2018. 1–9. [19](#)
- Schoettle Brandon, Sivak Michael.* Potential Improvements in Safety and Efficiency with Autonomous Trucking. 2017. [vii](#), [10](#)
- Schubert Robin, Richter Eric, Wanielik Gerd.* Comparison and evaluation of advanced motion models for vehicle tracking // *2008 11th international conference on information fusion*. 2008. 1–6. [20](#)
- Schulman John, Levine Sergey, Abbeel Pieter, Jordan Michael, Moritz Philipp.* Trust region policy optimization // *International conference on machine learning*. 2015. 1889–1897. [62](#)
- Schulman John, Wolski Filip, Dhariwal Prafulla, Radford Alec, Klimov Oleg.* Proximal policy optimization algorithms // *arXiv preprint arXiv :1707.06347*. 2017. [62](#)
- Schwarting Wilko, Alonso-Mora Javier, Rus Daniela.* Planning and decision-making for autonomous vehicles // *Annual Review of Control, Robotics, and Autonomous Systems*. 2018. [9](#)
- Shalev-Shwartz Shai, Ben-Zrihem Nir, Cohen Aviad, Shashua Amnon.* Long-term planning by short-term prediction // *arXiv preprint arXiv :1602.01580*. 2016. [19](#)
- Silver David, Schrittwieser Julian, Simonyan Karen, Antonoglou Ioannis, Huang Aja, Guez Arthur, Hubert Thomas, Baker Lucas, Lai Matthew, Bolton Adrian, others .* Mastering the game of go without human knowledge // *nature*. 2017. 550, 7676. 354–359. [129](#)
- Singh Santokh.* Critical reasons for crashes investigated in the national motor vehicle crash causation survey. 2018. [6](#)
- Sohn Kihyuk, Lee Honglak, Yan Xinchun.* Learning structured output representation using deep conditional generative models // *Advances in neural information processing systems*. 2015. 3483–3491. [19](#)

- Sun Pei, Kretschmar Henrik, Dotiwalla Xerxes, Chouard Aurelien, Patnaik Vijaysai, Tsui Paul, Guo James, Zhou Yin, Chai Yuning, Caine Benjamin, others*. Scalability in perception for autonomous driving : Waymo open dataset // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020. 2446–2454. [38](#)
- Sun Yixiang, Ioannou Petros A.* A Handbook for Inter-Vehicle Spacing in Vehicle Following. 1995. [113](#)
- Sutskever Ilya, Vinyals Oriol, Le Quoc V.* Sequence to sequence learning with neural networks // arXiv preprint arXiv :1409.3215. 2014. [48](#)
- Sutton Richard S, Barto Andrew G.* Reinforcement learning : An introduction. 2018. [viii](#), [62](#), [63](#)
- Tamke Andreas, Dang Thao, Breuel Gabi.* A flexible method for criticality assessment in driver assistance systems // 2011 IEEE Intelligent Vehicles Symposium (IV). 2011. 697–702. [24](#)
- Tay Christopher.* Analysis of dynamic scenes : application to driving assistance // Theses. Institut National Polytechnique de Grenoble-INPG (cit. on pp. 7, 9, 44–46). 2009. [23](#), [25](#)
- Tealab Ahmed.* Time series forecasting using artificial neural networks methodologies : A systematic review // Future Computing and Informatics Journal. 2018. 3, 2. 334–340. [54](#)
- Thiemann Christian, Treiber Martin, Kesting Arne.* Estimating acceleration and lane-changing dynamics from next generation simulation trajectory data // Transportation Research Record. 2008. 2088, 1. 90–101. [18](#), [112](#)
- Toledo-Moreo Rafael, Zamora-Izquierdo Miguel A.* Collision avoidance support in roads with lateral and longitudinal maneuver prediction by fusing GPS/IMU and digital maps // Transportation Research part C : emerging technologies. 2010. 18, 4. 611–625. [15](#)
- Tomar Ranjeet Singh, Verma Shekhar.* Lane change trajectory prediction using artificial neural network // International Journal of Vehicle Safety. 2013. 6, 3. 213–234. [16](#)
- Torkkola Kari, Venkatesan Srihari, Liu Huan.* Sensor selection for maneuver classification // Proceedings. The 7th International IEEE Conference on Intelligent Transportation Systems (IEEE Cat. No. 04TH8749). 2004. 636–641. [42](#)
- Tran Cuong, Doshi Anup, Trivedi Mohan Manubhai.* Modeling and prediction of driver behavior by foot gesture analysis // Computer Vision and Image Understanding. 2012. 116, 3. 435–445. [14](#)
- Tran Quan, Firl Jonas.* Online maneuver recognition and multimodal trajectory prediction for intersection assistance using non-parametric regression // 2014 IEEE Intelligent Vehicles Symposium Proceedings. 2014. 918–923. [17](#), [23](#)
- Turner Mark, Griffin Michael J.* Motion sickness in public road transport : the effect of driver, route and vehicle // Ergonomics. 1999. 42, 12. 1646–1664. [112](#)
- Van Brummelen Jessica, O'Brien Marie, Gruyer Dominique, Najjaran Homayoun.* Autonomous vehicle perception : The technology of today and tomorrow // Transportation research part C : emerging technologies. 2018. 89. 384–406. [7](#), [12](#)



- Vapnik Vladimir*. The nature of statistical learning theory. 2013. [44](#), [45](#), [46](#)
- Vechione Matthew, Balal Esmaeil, Cheu Ruey Long*. Comparisons of mandatory and discretionary lane changing behavior on freeways // International journal of transportation science and technology. 2018. 7, 2. 124–136. [129](#)
- Veeraraghavan Harini, Papanikolopoulos Nikos, Schrater Paul*. Deterministic sampling-based switching Kalman filtering for vehicle tracking // 2006 IEEE Intelligent Transportation Systems Conference. 2006. 1340–1345. [17](#)
- Veres Sandor M, Molnar Levente, Lincoln Nick K, Morice Colin P*. Autonomous vehicle control systems—a review of decision making // Proceedings of the Institution of Mechanical Engineers, Part I : Journal of Systems and Control Engineering. 2011. 225, 2. 155–195. [10](#)
- Vlacic Ljubo, Parent Michel, Harashima Fumio*. Intelligent vehicle technologies. 2001. [12](#)
- Vu Trung-Dung, Aycard Olivier, Tango Fabio*. Object perception for intelligent vehicle applications : A multi-sensor fusion approach // 2014 IEEE Intelligent Vehicles Symposium Proceedings. 2014. 774–780. [8](#)
- Wang Xiao, Wu Jinqiang, Gu Yanlei, Sun Hongbin, Xu Linhai, Kamijo Shunsuke, Zheng Nanning*. Human-like maneuver decision using lstm-crf model for on-road self-driving // 2018 21st International Conference on Intelligent Transportation Systems (ITSC). 2018. 210–216. [114](#), [115](#)
- Wang Zhiguang, Yan Weizhong, Oates Tim*. Time series classification from scratch with deep neural networks : A strong baseline // 2017 International joint conference on neural networks (IJCNN). 2017. 1578–1585. [88](#)
- Ward Jr Joe H*. Hierarchical grouping to optimize an objective function // Journal of the American statistical association. 1963. 58, 301. 236–244. [58](#)
- Waymo* . Overview of Waymo dataset. 2021. [Online; accessed 12-mai-2021]. [vii](#), [39](#)
- Wei Junqing, Snider Jarrod M, Kim Junsung, Dolan John M, Rajkumar Raj, Litkouhi Bakh-tiar*. Towards a viable autonomous driving research platform // 2013 IEEE Intelligent Vehicles Symposium (IV). 2013. 763–770. [10](#)
- Wiest Jürgen, Höffken Matthias, Krefsel Ulrich, Dietmayer Klaus*. Probabilistic trajectory prediction with gaussian mixture models // 2012 IEEE Intelligent Vehicles Symposium. 2012. 141–146. [17](#)
- Willmott Cort J, Matsuura Kenji*. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance // Climate research. 2005. 30, 1. 79–82. [56](#)
- Woo Hanwool, Ji Yonghoon, Kono Hitoshi, Tamura Yusuke, Kuroda Yasuhide, Sugano Takashi, Yamamoto Yasunori, Yamashita Atsushi, Asama Hajime*. Dynamic potential-model-based feature for lane change prediction // 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC). 2016. 000838–000843. [16](#)

*Xin Long, Wang Pin, Chan Ching-Yao, Chen Jianyu, Li Shengbo Eben, Cheng Bo.* Intention-aware long horizon trajectory prediction of surrounding vehicles using dual lstm networks // 2018 21st International Conference on Intelligent Transportation Systems (ITSC). 2018. 1441–1446. [116](#), [117](#)

*Yue-Hei Ng Joe, Hausknecht Matthew, Vijayanarasimhan Sudheendra, Vinyals Oriol, Monga Rajat, Toderici George.* Beyond short snippets : Deep networks for video classification // Proceedings of the IEEE conference on computer vision and pattern recognition. 2015. 4694–4702. [48](#)

*Zhang Guoqiang, Hu Michael Y, Patuwo B Eddy, Indro Daniel C.* Artificial neural networks in bankruptcy prediction : General framework and cross-validation analysis // European journal of operational research. 1999. 116, 1. 16–32. [42](#)

*Zhao Dongbin, Shao Kun, Zhu Yuanheng, Li Dong, Chen Y, Wang H, Liu De-Rong, Zhou Tong, Wang Cheng-Hong.* Review of deep reinforcement learning and discussions on the development of computer Go // Control Theory & Applications. 2016. 33, 6. 701–717. [129](#)

**Titre:** Prédiction des intentions des véhicules pour la conduite autonome en utilisant l'intelligence artificielle

**Mots clés:** Véhicule Autonome, Intelligence Artificielle, Prédiction des Intentions, Prédiction de Trajectoire, Classification des Manoeuvres, Prise de Décision.

**Résumé:** Le défi actuel des constructeurs automobiles est de permettre au véhicule autonome de rouler sur la route d'une manière fiable, dans des conditions de circulation réelles et sur des infrastructures non spécifiques, sans l'intervention du conducteur. Son bon fonctionnement dépendra fortement du niveau de sécurité de l'équipement, de la qualité du logiciel et de la fiabilité des informations utilisées par l'intelligence intégrée. La véhicule autonome roule sur la route avec différents types de véhicules tels que les camions, les voitures et les motos ; ce qui l'entoure d'un environnement incertain et en plein de changement. Pour garantir la sécurité de la conduite autonome dans un tel environnement, le véhicule autonome doit pouvoir anticiper le mouvement futur des véhicules adjacents. La prédiction des intentions est la solution pour anticiper les mouvements de ces usagers de la route. Dans ce contexte, plusieurs difficultés doivent être

surmontées pour assurer la cohabitation entre véhicules autonomes et véhicules traditionnels. L'adoption d'une méthode intelligente pour la prédiction des intentions est primordiale. Dans le cadre de cette thèse, des modèles de classification de manœuvres, prédiction de trajectoire et prise de décision sont mises en oeuvre. En outre, les approches proposées sont basées sur l'apprentissage automatique. Ce type d'approche nécessite une large quantité de données incluant une variété de scènes, plusieurs jeux de données véhiculaires sont utilisés dans le cadre de cette thèse. L'objectif est d'avoir une prédiction à long-terme avec une grande précision de la trajectoire prédite et en anticipant la réalisation de la manœuvre en avance. Ce travail comporte une forte composante expérimentale avec des campagnes de mesure sur les véhicules d'essai VEDECOM. Ces essais ont permis la validation des méthodes d'apprentissage proposées avec des données réelles.

**Title:** Predicting vehicle intentions for autonomous driving using artificial intelligence

**Keywords:** Autonomous Vehicle, Artificial Intelligence, Prediction of Intentions, Prediction of Trajectory, Classification of Maneuvers, Decision Making.

**Abstract:** The current challenge for car manufacturers is to allow to the autonomous vehicle to run reliably on the road, in real traffic conditions and on non-specific infrastructures, without driver intervention. Its proper functioning will strongly depend on the level of security of the equipment, the quality of the software and the reliability of the information used by the embedded intelligence. The autonomous car shares the road with different types of vehicles such as trucks, cars and motorcycles, which surrounds it with an uncertain and changing environment. To ensure safe driving in such environment, the system must be able to anticipate the future motion of surrounding vehicles. The prediction of intentions is the best solution for anticipating the motions of other road users. In this context, several difficulties must be overcome to ensure

coexistence between autonomous vehicles and traditional vehicles. The adoption of an intelligent method for the prediction of intentions is essential. As part of this thesis, maneuver classification models, trajectory prediction and decision making are implemented. In addition, the proposed approaches are based on machine learning. This type of approach requires a large amount of data including a variety of scenes, several vehicular datasets are used in this work. The objective is to have a long-term prediction by ensuring the best precision of the predicted trajectory and by anticipating the completion of the maneuver in advance. This work therefore has a strong experimental component with measurement campaigns on Vedecom test vehicles. These tests allowed the validation of the proposed learning methods with real data.