



**HAL**  
open science

# Algorithmes d'apprentissage pour le cyclisme urbain : Modèles implicites et infrastructure dynamique

Lucas Magnana

## ► To cite this version:

Lucas Magnana. Algorithmes d'apprentissage pour le cyclisme urbain : Modèles implicites et infrastructure dynamique. Apprentissage [cs.LG]. INSA Lyon, 2024. Français. NNT : 2024ISAL0019 . tel-04487508v2

**HAL Id: tel-04487508**

**<https://hal.science/tel-04487508v2>**

Submitted on 24 Oct 2024

**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.



Distributed under a Creative Commons Attribution 4.0 International License



N°d'ordre NNT : 2024ISAL0019

**THESE de DOCTORAT DE L'INSA LYON,  
membre de l'Université de Lyon**

**Ecole Doctorale EDA 512  
InfoMaths**

**Spécialité/ discipline de doctorat :**  
Informatique

Soutenue publiquement le 14/02/2024, par :  
**Lucas Magnana**

---

**Algorithmes d'apprentissage pour le  
cyclisme urbain : modèles implicites et  
infrastructure dynamique**

---

Devant le jury composé de :

DUCOURTHIAL Bertrand	Professeur des Universités, Heudiasyc	Président
MITTON Nathalie BILLOT Romain	Directrice de Recherche, INRIA Lille-Nord Europe Professeur des Universités, IMT Atlantique	Rapportrice Rapporteur
GARIN Federica GARCIA Christophe ROCHE Stéphane	Chargée de Recherche, INRIA Grenoble Professeur des Universités, INSA Lyon Professeur des Universités, Université Laval	Examinatrice Examineur Examineur
RIVANO Hervé CHIABAUT Nicolas	Professeur des Universités, INSA Lyon Directeur des Nouvelles Mobilités, Département de la Haute-Savoie	Directeur de thèse Directeur de thèse

Référence : TH1079\_MAGNANA Lucas

L'INSA Lyon a mis en place une procédure de contrôle systématique via un outil de détection de similitudes (logiciel Compilatio). Après le dépôt du manuscrit de thèse, celui-ci est analysé par l'outil. Pour tout taux de similarité supérieur à 10%, le manuscrit est vérifié par l'équipe de FEDORA. Il s'agit notamment d'exclure les auto-citations, à condition qu'elles soient correctement référencées avec citation expresse dans le manuscrit.

Par ce document, il est attesté que ce manuscrit, dans la forme communiquée par la personne doctorante à l'INSA Lyon, satisfait aux exigences de l'Établissement concernant le taux maximal de similitude admissible.

**INSA LYON**

Campus LyonTech La Doua  
20, avenue Albert Einstein - 69621 Villeurbanne cedex - France  
Tél. +33 (0)4 72 43 83 83 - Fax +33 (0)4 72 43 85 00  
[www.insa-lyon.fr](http://www.insa-lyon.fr)



## Département FEDORA – INSA Lyon - Ecoles Doctorales

SIGLE	ECOLE DOCTORALE	NOM ET COORDONNEES DU RESPONSABLE
ED 206 CHIMIE	<b>CHIMIE DE LYON</b> <a href="https://www.edchimie-lyon.fr">https://www.edchimie-lyon.fr</a> Sec. : Renée EL MELHEM Bât. Blaise PASCAL, 3e étage <a href="mailto:secretariat@edchimie-lyon.fr">secretariat@edchimie-lyon.fr</a>	<b>M. Stéphane DANIELE</b> C2P2-CPE LYON-UMR 5265 Bâtiment F308, BP 2077 43 Boulevard du 11 novembre 1918 69616 Villeurbanne <a href="mailto:directeur@edchimie-lyon.fr">directeur@edchimie-lyon.fr</a>
ED 341 E2M2	<b>ÉVOLUTION, ÉCOSYSTÈME, MICROBIOLOGIE, MODÉLISATION</b> <a href="http://e2m2.universite-lyon.fr">http://e2m2.universite-lyon.fr</a> Sec. : Bénédicte LANZA Bât. Atrium, UCB Lyon 1 Tél : 04.72.44.83.62 <a href="mailto:secretariat.e2m2@univ-lyon1.fr">secretariat.e2m2@univ-lyon1.fr</a>	<b>Mme Sandrine CHARLES</b> Université Claude Bernard Lyon 1 UFR Biosciences Bâtiment Mendel 43, boulevard du 11 Novembre 1918 69622 Villeurbanne CEDEX <a href="mailto:e2m2.codir@listes.univ-lyon1.fr">e2m2.codir@listes.univ-lyon1.fr</a>
ED 205 EDISS	<b>INTERDISCIPLINAIRE SCIENCES-SANTÉ</b> <a href="http://ediss.universite-lyon.fr">http://ediss.universite-lyon.fr</a> Sec. : Bénédicte LANZA Bât. Atrium, UCB Lyon 1 Tél : 04.72.44.83.62 <a href="mailto:secretariat.ediss@univ-lyon1.fr">secretariat.ediss@univ-lyon1.fr</a>	<b>Mme Sylvie RICARD-BLUM</b> Laboratoire ICBMS - UMR 5246 CNRS - Université Lyon 1 Bâtiment Raulin - 2ème étage Nord 43 Boulevard du 11 novembre 1918 69622 Villeurbanne Cedex Tél : +33(0)4 72 44 82 32 <a href="mailto:sylvie.ricard-blum@univ-lyon1.fr">sylvie.ricard-blum@univ-lyon1.fr</a>
ED 34 EDML	<b>MATÉRIAUX DE LYON</b> <a href="http://ed34.universite-lyon.fr">http://ed34.universite-lyon.fr</a> Sec. : Yann DE ORDENANA Tél : 04.72.18.62.44 <a href="mailto:yann.de-ordenana@ec-lyon.fr">yann.de-ordenana@ec-lyon.fr</a>	<b>M. Stéphane BENAYOUN</b> Ecole Centrale de Lyon Laboratoire LTDS 36 avenue Guy de Collongue 69134 Ecully CEDEX Tél : 04.72.18.64.37 <a href="mailto:stephane.benayoun@ec-lyon.fr">stephane.benayoun@ec-lyon.fr</a>
ED 160 EEA	<b>ÉLECTRONIQUE, ÉLECTROTECHNIQUE, AUTOMATIQUE</b> <a href="https://edeea.universite-lyon.fr">https://edeea.universite-lyon.fr</a> Sec. : Philomène TRECOURT Bâtiment Direction INSA Lyon Tél : 04.72.43.71.70 <a href="mailto:secretariat.edeea@insa-lyon.fr">secretariat.edeea@insa-lyon.fr</a>	<b>M. Philippe DELACHARTRE</b> INSA LYON Laboratoire CREATIS Bâtiment Blaise Pascal, 7 avenue Jean Capelle 69621 Villeurbanne CEDEX Tél : 04.72.43.88.63 <a href="mailto:philippe.delachartre@insa-lyon.fr">philippe.delachartre@insa-lyon.fr</a>
ED 512 INFOMATHS	<b>INFORMATIQUE ET MATHÉMATIQUES</b> <a href="http://edinfomaths.universite-lyon.fr">http://edinfomaths.universite-lyon.fr</a> Sec. : Renée EL MELHEM Bât. Blaise PASCAL, 3e étage Tél : 04.72.43.80.46 <a href="mailto:infomaths@univ-lyon1.fr">infomaths@univ-lyon1.fr</a>	<b>M. Hamamache KHEDDOUCI</b> Université Claude Bernard Lyon 1 Bât. Nautibus 43, Boulevard du 11 novembre 1918 69 622 Villeurbanne Cedex France Tél : 04.72.44.83.69 <a href="mailto:direction.infomaths@listes.univ-lyon1.fr">direction.infomaths@listes.univ-lyon1.fr</a>
ED 162 MEGA	<b>MÉCANIQUE, ÉNERGÉTIQUE, GÉNIE CIVIL, ACOUSTIQUE</b> <a href="http://edmega.universite-lyon.fr">http://edmega.universite-lyon.fr</a> Sec. : Philomène TRECOURT Tél : 04.72.43.71.70 Bâtiment Direction INSA Lyon <a href="mailto:mega@insa-lyon.fr">mega@insa-lyon.fr</a>	<b>M. Etienne PARIZET</b> INSA Lyon Laboratoire LVA Bâtiment St. Exupéry 25 bis av. Jean Capelle 69621 Villeurbanne CEDEX <a href="mailto:etienne.parizet@insa-lyon.fr">etienne.parizet@insa-lyon.fr</a>
ED 483 ScSo	<b>ScSo<sup>1</sup></b> <a href="https://edsciencesociales.universite-lyon.fr">https://edsciencesociales.universite-lyon.fr</a> Sec. : Mélina FAVETON Tél : 04.78.69.77.79 <a href="mailto:melina.faveton@univ-lyon2.fr">melina.faveton@univ-lyon2.fr</a>	<b>M. Bruno MILLY</b> (INSA : J.Y. TOUSSAINT) Univ. Lyon 2 Campus Berges du Rhône 18, quai Claude Bernard 69365 LYON CEDEX 07 Bureau BEL 319 <a href="mailto:bruno.milly@univ-lyon2.fr">bruno.milly@univ-lyon2.fr</a>

# Résumé

L'augmentation de l'utilisation du vélo comme mode de transport est essentielle pour diminuer la congestion ainsi que la pollution dans les villes. Le développement de nouvelles technologies ainsi qu'une prise de conscience des dangers du dérèglement climatique ont permis entre autres une augmentation du nombre de cyclistes dans les villes au cours des 20 dernières années. Cela s'est naturellement accompagné de nouveaux moyens de récupérer des données comportementales de cyclistes, diversifiant et multipliant les jeux de données existants. Ces données peuvent être utilisées pour affiner les politiques urbaines visant à augmenter la part modale du vélo dans les transports urbains. L'objectif de cette thèse est d'utiliser certaines données comportementales de cyclistes ainsi que des algorithmes récents d'intelligence artificielle pour créer des outils innovants d'aide aux politiques urbaines d'incitation au cyclisme. Dans un premier temps, un état de l'art des différentes sources de données de comportements de cyclistes est dressé. Cet état de l'art permet de se rendre compte de la diversité ainsi que du nombre de jeux de données, mais aussi des difficultés d'utilisation accompagnant chaque source. Cet état de l'art est aussi l'occasion de justifier les sources de données utilisées ensuite. Dans un second temps, une méthode de création de modèles de choix d'itinéraire implicites est développée à partir de traces GPS. Les modèles créés avec cette méthode permettent de générer un itinéraire cyclable à partir d'une origine et d'une destination. Le développement d'un modèle commence par l'identification des segments de route préférés par les cyclistes ayant généré les traces GPS. Un réseau de neurones artificiels choisit ensuite un groupe de segments de route préférés pertinents à partir d'une origine et d'une destination. Une pondération de graphe routier est ensuite effectuée à l'aide des segments de route sélectionnés pour générer un itinéraire cyclable. Les modèles créés permettent de créer des itinéraires plus proches de comportements réels de cyclistes lorsque comparés à d'autres méthodes de génération d'itinéraires. Enfin, un feu de signalisation intelligent permettant de sécuriser le passage des cyclistes à l'intersection qu'il régule est développé. L'utilisation de l'intelligence artificielle permet de séparer les flux de vélos et de véhicules motorisés en limitant l'impact sur le trafic à l'intersection. Les simulations dans lesquelles ce feu est testé ont un volume de trafic de voitures et de vélos réaliste grâce à l'utilisation de données de compteurs. La méthode d'apprentissage utilisée pour la régulation surpasse les performances d'autres méthodes de gestion de feu utilisées en comparaison.

# Table des matières

## Résumé

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Les sources de données</b>	<b>5</b>
2.1	Les questionnaires de préférence	6
2.2	Les systèmes de vélo en libre service	9
2.3	Les capteurs GPS	12
2.3.1	Strava Metro	14
2.3.2	Véléval	15
2.3.3	MonRésoVélo	16
2.4	Les comptages de vélos	17
2.5	Positionnement	19
2.6	Résumé	22
<b>3</b>	<b>Modèles de choix d'itinéraire implicites</b>	<b>24</b>
3.1	Introduction	24
3.2	Détail des données	26
3.2.1	Traces expérimentées	26
3.2.2	Traces générées	27
3.2.3	Comparaisons des traces expérimentées et des traces générées	27
	Cartes de chaleurs	29
	CDF	31
	Suppression des traces faisant de trop gros détours	32
3.2.4	Résumé de la section	33
3.3	Identification des segments de route préférés	33
3.3.1	Métriques de distance	33
	Distance personnalisée	34
	Distance géodésique	35
3.3.2	Algorithmes de <i>clustering</i>	35
	DBSCAN	35
	k-médoïdes	35
3.3.3	Paramétrage	37
	DBSCAN	39
	k-médoïdes	40
	Algorithmes et paramètres retenus	43
3.3.4	Analyse visuelle des plus grands <i>clusters</i>	44
3.3.5	Résumé de la section	48
3.4	Générateur de traces basées sur l'expérience	49
3.4.1	Coefficient de cyclabilité	49
3.4.2	Modification du graphe routier	49
3.4.3	Qualité des traces basées sur l'expérience	50
3.4.4	Traces de comparaison	50

3.4.5	Résultats	51
	Veival	51
	MonRésoVélo	53
3.4.6	Résumé de section	53
3.5	Classifieur	54
3.5.1	Réseau de neurones LSTM	55
3.5.2	Prétraitement des traces	56
3.5.3	Structure du classifieur et hyperparamétrage	57
3.5.4	Performances	57
3.5.5	Résumé de la section	58
3.6	Modèles finaux	59
3.6.1	Analyse quantitative	59
3.6.2	Analyse visuelle de traces basées sur l'expérience	60
3.6.3	Résumé de la section	62
3.7	Conclusion et discussion	63
<b>4</b>	<b>Feu de circulation intelligent pour cyclistes</b>	<b>66</b>
4.1	Introduction	66
4.1.1	Les cyclistes et feux de signalisation	66
4.1.2	DRL pour le contrôle de feux de signalisation en simulation	67
4.1.3	Positionnement	68
4.2	Apprentissage par renforcement profond	68
4.2.1	Double Dueling Deep Q-Network (3DQN)	69
	Deep Q-Network	69
	Double Deep Q-Network	70
	Dueling Deep Q-Network	70
	Algorithme final	70
4.2.2	Proximal Policy Optimization	71
4.3	Approche DRL	72
4.3.1	Environnement	72
4.3.2	États	72
4.3.3	Actions	73
4.3.4	Récompenses	73
	Fonction de récompenses	74
	Facteur d'étalement	74
4.4	Simulation de l'environnement	74
4.4.1	Environnement simulé	74
4.4.2	Données de comptage et demande de trafic	75
4.5	Entraînements	76
4.5.1	Architecture des réseaux de neurones	76
4.5.2	Hyperparamétrage	78
4.5.3	Performances	79
	Agent 3DQN	79
	Agent PPO	80
4.6	Résultats	83
4.6.1	Approches de comparaison	83
4.6.2	Résultats horaires	84
4.6.3	Robustesse aux changements du trafic vélos	85
4.7	Analyse du comportement du feu	89
4.7.1	Journée entière	90
4.7.2	Heure de pointe	91

4.8 Conclusion et discussion . . . . .	93
<b>5 Conclusion</b>	<b>94</b>
5.1 Résumé . . . . .	94
5.2 Discussion et perspectives . . . . .	95
<b>A Distribution des véhicules comptés à Paris</b>	<b>98</b>
<b>Bibliographie</b>	<b>99</b>



# Table des figures

1.1	Diagramme résumant cette thèse. . . . .	4
2.1	Exemple de question posée par Tilahun et al.[10] . . . . .	8
2.2	Carte contenant une partie des données ouvertes de Velo'v, le BSS de Lyon. . . . .	10
2.3	Carte affichant certaines données Strava ajoutées au graphe routier de la ville de Glasgow. . . . .	15
2.4	Carte affichant certaines traces GPS du jeu de donnée Véléval. . . . .	16
2.5	Visuel de l'application MonRésoVélo. . . . .	17
2.6	Carte montrant les compteurs automatiques de vélos dans la métropole de Lyon. . . . .	18
3.1	Diagramme résumant la méthode de création de modèles de choix d'itinéraire implicites présentée dans ce chapitre. . . . .	25
3.2	CDF des différences de distances entre les traces expérimentées et les traces générées des jeux de données. . . . .	28
3.3	Cartes de chaleur des traces expérimentées (à gauche) et des traces générées (à droite) du jeu de données Véléval (Lyon). . . . .	30
3.4	Cartes de chaleur des traces expérimentées (à gauche) et des traces générées (à droite) du jeu de données MonRésoVélo (Montréal). . . . .	30
3.5	CDF des différences de distances entre les traces expérimentées et les traces générées des jeux de données après la suppression des traces faisant de trop gros détours. . . . .	32
3.6	Exemple graphique de l'intersection entre les ensembles étendus de 2 traces. . . . .	34
3.7	Évolution du nombre de grands <i>clusters</i> (courbe noire), du score de silhouette moyen (courbe bleue), du nombre de <i>clusters</i> (courbe rouge) et du bruit (courbe verte) des <i>clusters</i> créés par DBSCAN selon $\epsilon$ pour le jeu de données Véléval. . . . .	41
3.8	Évolution du nombre de grands <i>clusters</i> (courbe noire), du score de silhouette moyen (courbe bleue), du nombre de <i>clusters</i> (courbe rouge) et du bruit (courbe verte) des <i>clusters</i> créés par DBSCAN selon $\epsilon$ pour le jeu de données MonRésoVélo. . . . .	42
3.9	Évolution du nombre de grands <i>clusters</i> (courbe noire) et du score de silhouette moyen (courbe bleue) des <i>clusters</i> créés par k-médoïdes selon $k$ pour le jeu de données Véléval. . . . .	43
3.10	Évolution du nombre de grands <i>clusters</i> (courbe noire) et du score de silhouette moyen (courbe bleue) des <i>clusters</i> créés par k-médoïdes selon $k$ pour le jeu de données MonRésoVélo. . . . .	44
3.11	Cartes de chaleurs des plus grands <i>clusters</i> pour les trois paramétrages retenus avec le jeu de données Véléval . . . . .	46
3.12	Cartes de chaleurs des plus grands <i>clusters</i> pour les trois paramétrages retenus avec le jeu de données MonRésoVélo. . . . .	47

3.13	Boîtes à moustaches des distances entre les traces expérimentées de Véléval et les quatre autres types de traces. . . . .	52
3.14	Boîtes à moustaches des distances entre les traces expérimentées de MonRésoVélo et les quatre autres types de traces. . . . .	54
3.15	Diagramme montrant la structure d'une cellule LSTM. . . . .	55
3.16	Prédictions des classifieurs avec les sous-jeux de test. . . . .	58
3.17	Boîtes à moustaches des distances entre les traces expérimentées de Véléval et les trois types de traces basées sur l'expérience. . . . .	60
3.18	Boîtes à moustaches des distances entre les traces expérimentées de MonRésoVélo et les trois types de traces basées sur l'expérience. . . . .	60
3.19	Cartes montrant 2 traces expérimentées (en orange), les traces générées correspondantes (en bleu) et les traces calculées par le modèle de choix d'itinéraire entraîné avec le jeu Véléval (en rouge). . . . .	62
3.20	Cartes montrant 2 traces expérimentées (en orange), les traces générées correspondantes (en bleu) et les traces calculées par le modèle de choix d'itinéraire entraîné avec le jeu MonRésoVélo (en rouge). . . . .	63
4.1	Diagramme de la construction d'une matrice de position à partir du schéma d'une intersection. . . . .	73
4.2	Capture d'écran de l'environnement simulé par SUMO. . . . .	75
4.3	Somme des véhicules comptés par heure dans les 2 directions sur le boulevard Montparnasse le 20 juin 2023. . . . .	77
4.4	Diagramme de la structure du Q-réseau. . . . .	78
4.5	Évolution du temps d'attente moyen des véhicules pendant l'entraînement de l'agent 3DQN. . . . .	81
4.6	Évolution du temps d'attente moyen des véhicules pendant l'entraînement de l'agent PPO. . . . .	82
4.7	Résultats d'une simulation d'un jour. . . . .	86
4.8	Nombre de véhicules et sommes des temps d'attentes entre selon le coefficient multiplicateur du trafic vélo (de 6h à 20h). . . . .	88
4.9	Temps d'attente moyen des véhicules selon le coefficient multiplicateur du trafic vélo (de 6h à 20h). . . . .	89
4.10	Nombre d'étapes passées en phase verte et orange par voie durant une simulation d'un jour. . . . .	91
4.11	Chronologie des phases par voie de 19h à 20h. . . . .	92
A.1	Somme des véhicules comptés dans la ville de Paris le 20 juin 2023. . . . .	98

# Liste des tableaux

2.1	Tableau résumé des quatre sources de données et des caractéristiques les différenciant. . . . .	7
4.1	Tableau des hyperparamètres. . . . .	79

*À mon père, puisse-t-il avoir enfin trouvé la paix...*

# Chapitre 1

## Introduction

La mobilité urbaine est en train de vivre un changement significatif avec le retour de l'utilisation du vélo pour les trajets quotidiens. Les motivations d'un tel changement modal sont nombreuses. Individuellement, cela permet d'être en meilleure santé [1, 2], d'économiser du temps et de l'argent. À l'échelle de la ville, cela permet une réduction de la congestion des véhicules motorisés ainsi que de la pollution de l'air [3]. Pour ces raisons, les décideurs politiques lancent de nombreux projets pour promouvoir l'utilisation du vélo en ville. Classiquement, ces projets visent à redistribuer l'espace urbain plus équitablement entre les véhicules motorisés et les autres modes de transport. Les villes sont en effet dans l'immense majorité des cas structurées autour de la voiture individuelle. En conséquence, la part des trajets quotidiens effectués en voiture dans les pays de l'OCDE varie de 46% à 86% selon les études [4].

La pandémie de COVID-19 a temporairement augmenté l'attrait des modes de transports individuels au détriment des transports publics pour des raisons de distanciations sociales. Pour éviter un report modal massif vers la voiture, les villes ont cherché à rapidement augmenter l'attrait de l'utilisation du vélo en leur sein. Pour y parvenir, celles-ci ont beaucoup utilisé l'urbanisme tactique pour augmenter la part d'infrastructures cyclables dans leurs réseaux routiers [5]. L'urbanisme tactique consiste à fermer temporairement certaines voies aux véhicules motorisés pour les réserver aux modes de transport actifs. Les modes de transport actifs comprennent tous les moyens de se déplacer utilisant l'activité physique humaine comme principale source d'énergie. L'urbanisme tactique s'effectue la plupart du temps par l'ajout d'un marquage au sol sur les voies ciblées.

Cette stratégie n'est cependant pas durable, car elle se concentre surtout sur la quantité des infrastructures sans intégrer la dimension qualitative de celles-ci dans la planification. Le nombre des aménagements cyclables étant positivement liés au nombre de cyclistes [6, 7], les décideurs politiques estiment que la présence d'infrastructures cyclables sur les voies principales suffit pour une pratique du vélo rapide et sécurisée [8]. Les cyclistes, bien que d'accord avec cette affirmation, évoquent également les effets positifs d'autres facteurs, comme la présence d'espaces verts, le sentiment de sécurité ou des infrastructures séparées des véhicules motorisés [8, 9, 10]. L'installation d'une bande cyclable, un type d'infrastructure séparée de la chaussée par un marquage au sol, n'aura pas le même impact que l'installation d'une piste cyclable, un type d'infrastructure séparé lui par un terre-plein non franchissable. À noter qu'un cycliste désigne un individu pratiquant le cyclisme. Le cyclisme se réfère tout au long de cette thèse à l'utilisation du vélo, quelle qu'en soit la raison. Une compréhension précise des comportements de cyclistes est cruciale pour mettre en place des politiques urbaines efficaces d'incitation au cyclisme. L'étude des comportements de cyclistes dans le but de créer une structure cyclable optimale est devenue

un sujet de recherche récurrent pour cette raison. Une partie de cette compréhension doit se faire à l'échelle des trajets individuels, car être capable de reproduire des itinéraires de cyclistes est décisif pour anticiper les performances de futurs aménagements. Un verrou important de ce travail scientifique a longtemps été l'absence de données quantitatives de trajets effectués par des cyclistes.

Malgré les limites évoquées de certaines politiques, le nombre de cyclistes augmente dans les villes d'Europe [11]. Ce regain d'intérêt pour le cyclisme, accompagné de l'essor technologique démarré au début du siècle, a permis de transformer l'utilisation du vélo en ville [12]. On peut citer parmi les innovations technologiques l'automatisation de la location de vélos individuels à la demande, ou la création de vélos avec assistance électrique utilisable sur de plus longues distances et sur des topographies qui demandent des efforts importants. Ces transformations se sont naturellement accompagnées de nouvelles possibilités de récupération de données comportementales de cyclistes. Ces nouvelles données sont de différentes natures, et sont regroupées en jeux de différentes tailles n'ayant pas tous le même niveau d'accessibilité. Ces données sont aussi multidimensionnelles, avec des propriétés spatiales mais aussi temporelles, pouvant rendre difficile leur exploitation. Néanmoins, analyser, comprendre et même prédire les comportements de cyclistes est possible lorsque cette exploitation est effectuée correctement.

Parallèlement à l'émergence de ces nouvelles données, l'apprentissage automatique, une branche de l'intelligence artificielle, a fait d'énormes progrès. L'apprentissage automatique consiste à développer et à analyser des algorithmes pouvant généraliser certaines tâches sans instructions spécifiques. L'apprentissage automatique a été appliqué à de nombreux domaines comme l'agriculture [13, 14], la médecine [15, 16] ou encore le transport [17, 18]. Il est possible d'identifier 4 types d'apprentissage automatique, chacun regroupant une multitude d'algorithmes différents. L'apprentissage supervisé consiste à construire un modèle mathématique à partir d'un jeu de données contenant des entrées et des sorties désirées [19]. L'apprentissage non supervisé consiste à prendre un jeu de données ne possédant que des entrées et à regrouper celles-ci en paquets (appelés *clusters*) d'entrées qu'une métrique estime similaires. L'apprentissage semi-supervisé est un mixte entre les deux précédents apprentissages. Certaines données utilisées possèdent une sortie désirée et d'autres non [20]. Enfin, l'apprentissage par renforcement consiste à apprendre à un agent à effectuer des actions dans son environnement pour maximiser un signal de récompense numérique. Le but de l'agent est de découvrir par lui-même quelles actions sont les plus profitables dans les différentes situations dans lesquelles il peut se trouver [21].

L'apprentissage supervisé est le type d'apprentissage le plus utilisé. Deux types de problèmes sont traitables par de l'apprentissage supervisé : la régression et la classification. La régression consiste à approcher une variable de sortie à partir de variables explicatives corrélées à celle-ci. Dans le cas du cyclisme, la régression peut par exemple permettre de prédire si un individu est propice à utiliser son vélo à partir de variables comme la météo, l'heure de la journée, le genre ou encore la situation financière de celui-ci. La classification consiste à identifier à quelle catégorie une observation appartient parmi un ensemble prédéfini de catégories. Pour le cyclisme, cela peut servir par exemple à trouver le niveau d'expérience d'un individu à partir des distributions de vitesse de ses trajets. Dans le cadre de cette

thèse, les algorithmes d'apprentissage supervisé sont divisés en 2 catégories : implicites et explicites. Durant ce travail de recherche, les données utilisées en entrée des algorithmes d'apprentissage supervisé sont des données comportementales de cyclistes. Les algorithmes dits explicites comprennent les algorithmes nécessitant la définition explicite de variables descriptives représentant certaines caractéristiques de ces comportements. On peut citer entre autres les algorithmes de régression ou les arbres de décision. Un algorithme de régression attribue un poids à chacune des variables explicatives dont le but est de prédire la variable de sortie. Les algorithmes de régression étant fréquemment utilisés sur des données de comportements individuels de cyclistes, plus de détails sont présentés plus tard. Les algorithmes dits implicites ne nécessitent eux pas de variables descriptives, car ils apprennent des représentations implicites des données d'entrée dans le but de renvoyer la sortie désirée. C'est par exemple le cas des réseaux de neurones artificiels qui sont inspirés du cerveau des êtres vivants. Il existe différents types de réseaux de neurones permettant d'apprendre différents types de représentations. Certains réseaux de neurones sont capables de traiter des données multidimensionnelles, comme celles représentant les comportements de cyclistes. Les réseaux de neurones convolutifs permettent par exemple de faire de l'apprentissage sur des données spatiales. Les réseaux de neurones récurrents sont eux adaptés aux données temporelles. Ces dernières années, les réseaux de neurones ont permis de faire avancer de nombreux domaines scientifiques, comme la reconnaissance d'image [22], la reconnaissance vocale [23] ou encore la modélisation du langage [16, 24].

L'objectif de cette thèse est de développer des outils innovants ayant comme objectif d'inciter au cyclisme. Ces outils utilisent des algorithmes d'apprentissage automatique sur des données comportementales de cyclistes en ville. Deux approches différentes sont testées. La première est construite autour de la politique urbaine la plus répandue pour inciter au cyclisme : la création d'infrastructures cyclables pertinentes. La pertinence de ces infrastructures est fortement dépendante de la capacité à prédire le comportement des cyclistes. Une méthode de création de modèles de choix d'itinéraire permettant de générer des trajets individuels de cyclistes est développée dans ce but. La deuxième, plus innovante, vise à développer un nouveau type d'infrastructure dynamique permettant de sécuriser le passage des cyclistes à un feu de circulation en limitant l'impact sur le trafic. Les trois contributions de cette thèse peuvent être résumées comme suit :

1. **Identification de segments de routes préférés de cyclistes.** Cette méthode utilise des traces GPS et un algorithme d'apprentissage non supervisé (*clustering*) pour identifier les segments de route que les cyclistes utilisent le plus.
2. **Développement d'une méthode de génération d'itinéraires cyclables.** Cette méthode est capable de générer des trajets individuels entre n'importe quelle origine et destination dans une ville en utilisant des segments de routes préférés identifiés au préalable. Celle-ci utilise un algorithme d'apprentissage supervisé (apprentissage profond) et une pondération de graphe routier.
3. **Création d'un feu de circulation intelligent sécurisé pour les cyclistes.** Ce feu est contrôlé par un algorithme d'apprentissage par renforcement, utilisant lui-même des réseaux de neurones. Le développement de ce feu se fait en simulations, et des données de comptage vélos et voitures sont utilisées pour paramétrer celles-ci.

Ces contributions ont pu être valorisées par l'écriture de 2 articles, un premier publié dans la revue *PLOS ONE* [25] et un second soumis dans la revue *Journal of Intelligent Transportation* [26]. La Figure 1.1 résume cette thèse, les données et les types d'apprentissage utilisés, ainsi que les contributions qui ont été permises par leurs combinaisons. Ce manuscrit est divisé en 5 chapitres, le premier étant cette introduction.

Le Chapitre 2 résume les différentes sources de données décrivant le comportement des cyclistes, et argumente le choix des 2 types de données utilisés ensuite.

Le Chapitre 3 décrit une méthode de création de modèles de choix d'itinéraire implicites. Les modèles implicites créés avec cette méthode diffèrent des modèles existants tant par leur objectif que par la manière de l'atteindre. L'objectif des modèles implicites est de prédire le comportement des cyclistes à partir de données comportementales. Celui des modèles plus classiques est de quantifier l'impact de certaines variables sur le choix des cyclistes. L'identification par les modèles implicites de *patterns* (c.-à-d. de schémas) dans les données comportementales remplace le choix explicite de variables descriptives effectué lors du développement de modèles plus classiques. Cela permet, entre autres, d'outrepasser certaines limites amenées par cette sélection. Au cours du développement de cette méthode, l'existence de segments de route préférés par les cyclistes est démontrée.

L'idée derrière le Chapitre 4 est de développer un nouveau type d'infrastructure, pour inciter l'utilisation de certains segments de route. Cette infrastructure est un feu de circulation intelligent sécurisé pour les cyclistes. Les infrastructures séparant les flux de vélos et de voitures sont particulièrement plébiscitées par les cyclistes. Les approches classiques sont cependant très coûteuses en place, car la séparation est effectuée physiquement grâce à la construction de pistes cyclables. Ajouter des phases vertes pour cyclistes à un feu de circulation permet d'effectuer cette séparation virtuellement. Cette solution permet de ne pas modifier la voirie, mais augmente significativement le temps d'attente des véhicules à l'intersection si implémentée de manière naïve. Un algorithme d'apprentissage par renforcement est proposé pour contrôler le feu de manière à limiter le coût en temps d'attente d'une telle installation.

Enfin, le Chapitre 5 conclut le travail effectué durant cette thèse et met en lumière les perspectives de celui-ci.

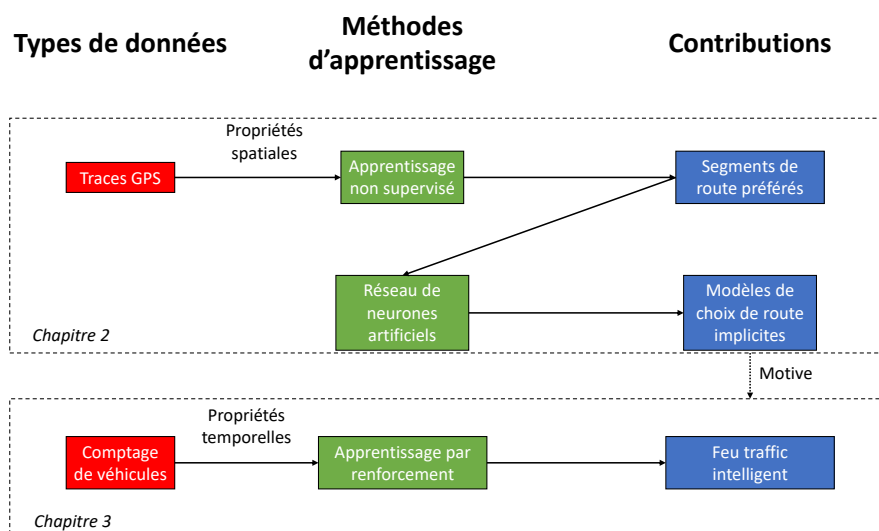


FIGURE 1.1 – Diagramme résumant cette thèse.



## Chapitre 2

# Les sources de données

Les données décrivant le comportement des cyclistes peuvent venir de différentes sources. La description et la modélisation du comportement des cyclistes est une problématique ancienne, et les moyens technologiques ont grandement évolué depuis la naissance de cette discipline scientifique. Aujourd’hui, il existe 4 sources de données majeures utilisées dans la littérature scientifique :

- *Les réponses à des questionnaires de préférences* : plus ancienne manière de récupérer des données comportementales, les questionnaires ont tout de même évolué au fil des avancées technologiques, notamment celles relatives à l’enregistrement vidéo ainsi qu’à la transmission d’informations.
- *Les systèmes de location de vélos en libre-service* : le premier système totalement informatisé est déployé à Rennes en 1998<sup>1</sup>. Depuis, la majorité des grandes villes sont dotées de ce type de systèmes et partagent certaines données les concernant.
- *Les capteurs GPS* : d’abord accrochés aux vélos ou portés par les cyclistes, la miniaturisation et l’inclusion de capteurs GPS dans les smartphones a permis la multiplication des jeux de données de traces GPS de cyclistes.
- *Les compteurs vélos* : installés sur certaines pistes cyclables, ils permettent de quantifier leur utilisation à différentes échelles temporelles. Les villes rendent publiques les données agrégées qu’ils enregistrent.

Ces différents types de données sont complémentaires et peuvent être utilisés ensemble [27], mais peu de travaux ont été effectués pour les fusionner entre eux [28]. Ce travail de fusion est difficile car il demande la récupération de jeux de données de différentes sources comparables temporellement et géographiquement. De plus, il est souvent nécessaire d’émettre des hypothèses fortes demandant des connaissances préalables sur le comportement des cyclistes et pouvant compromettre la représentativité des jeux de données initiaux. Il a donc été décidé d’utiliser individuellement les sources de données choisies. Pour choisir les sources les plus pertinentes, quatre grandes caractéristiques permettant de discriminer celles-ci sont utilisées :

- *La disponibilité* qui prend en compte le nombre de jeux de données existants ainsi que leurs modalités de mise à disposition.
- *La compatibilité avec les méthodes d’apprentissage* qui décrit les prétraitements qui doivent être faits sur les données avant de pouvoir utiliser un algorithme d’apprentissage sur celles-ci.
- *Le degré d’information du comportement des cyclistes* qui représente à quel point des trajets individuels peuvent être retracés à partir des données.

---

1. [https://fr.wikipedia.org/wiki/Vélos\\_en\\_libre-service](https://fr.wikipedia.org/wiki/Vélos_en_libre-service)

- *Les problèmes principaux auxquels les données répondent* qui listent les problématiques les plus communément trouvées dans la littérature utilisant la source de donnée pour être étudiées.

Les sources de données ainsi que ces caractéristiques sont résumées dans le tableau 2.1. Les sections 2.1 à 2.4 sont structurées de la même façon. Premièrement, les données récupérables à partir de la source ainsi que la méthode de collecte sont décrites. Deuxièmement, un état de l'art de l'utilisation des données provenant de la source est effectué. La Section 2.5 discute ensuite des différentes sources de données au vu des contributions de cette thèse, et argumente le choix des sources de données utilisées. Enfin, la Section 2.6 est un résumé du chapitre.

## 2.1 Les questionnaires de préférence

L'analyse du comportement des cyclistes est un sujet présent dans la littérature scientifique depuis de nombreuses années. Au départ, l'un des moyens principaux de récupérer des données comportementales de cyclistes était de les interroger grâce à un questionnaire de préférences. Cela permettait de récupérer des données tant qualitatives que quantitatives. L'avènement des systèmes de vélos en libre-service, des puces GPS ou autres compteurs ont amené de nouvelles manières de récupérer des données quantitatives relatives au comportement des cyclistes. Cependant, lorsqu'il s'agit de récupérer des données qualitatives, ce sont ces questionnaires qui sont encore aujourd'hui privilégiés. Par conséquent, de nombreux questionnaires ont été développés [29, 30, 31, 37, 32, 33, 10, 34, 35, 36]. Les études utilisant un questionnaire de préférences proposent à des personnes interrogées différents itinéraires ou parties d'itinéraires représentés par certaines variables descriptives. Ces variables sont dans la majorité des cas similaires entre les études. Elles peuvent être relatives au type de voie (vitesse maximum des voitures, fréquentation...), aux infrastructures cyclables (présence ou non de pistes cyclables, continuité de celles-ci...), ou encore à l'itinéraire en lui-même (nombre de feux rouges, temps de trajet...). Additionnellement, ces questionnaires demandent diverses informations personnelles aux personnes interrogées, ce qui permet d'éviter à l'une d'entre elles de répondre plusieurs fois, mais surtout de vérifier si certaines variables sociales comme le genre, la fourchette de revenu ou la possession d'un ou plusieurs véhicules motorisés influent sur le comportement des cyclistes.

Une fois le questionnaire créé, celui-ci est diffusé que ce soit par voie électronique (chaîne de mail, sites web...) ou directement dans des lieux propices à la création de mobilités (universités, arrêts de bus ou de métro, hôpitaux...). Les réponses sont ensuite triées et analysées. Dans la plupart des cas, l'analyse porte sur la représentativité du profil des répondants, mais celle-ci peut être plus poussée. En 2007, Tilahun et al. [10] ont par exemple cherché à quantifier le temps que les cyclistes sont prêts à ajouter à leurs trajets pour utiliser des infrastructures cyclables de meilleures qualités. Ils ont pour cela identifié 5 niveaux de qualités d'infrastructures cyclables allant de la piste cyclable à la voie sans bande cyclable avec voitures garées le long du trottoir. Durant l'expérience, les répondants étaient invités à choisir entre deux itinéraires utilisant des infrastructures de niveaux de qualités différentes, comme montré dans la Figure 2.1. Les itinéraires sont hypothétiquement composés du même type

Source de données	Disponibilité	Compatibilité avec l'apprentissage automatique	Degré d'information du comportement individuel des cyclistes	Problèmes principaux solvables avec les données
Questionnaires de préférences	Réponses contenant des données personnelles, donc non publiées	Utilisable avec des variables décrivant les réponses	Choix des cyclistes dans des situations précises en situation contrôlée	<ul style="list-style-type: none"> <li>— Analyse de choix d'itinéraire [29, 30, 31, 32, 33, 34, 35, 36]</li> <li>— Classement de perception de types d'infrastructures [37]</li> <li>— Quantification de détours [10]</li> </ul>
Systèmes de vélo en libre-service	Données publiques et à accès restreint, plusieurs jeux de données disponibles	Utilisable avec peu ou pas de prétraitements	Coordonnées géographiques des origines et des destinations, horodatage ou temps de trajet	<ul style="list-style-type: none"> <li>— Optimisation du remplissage du système [38, 39, 40, 41, 42, 43, 44, 45, 46]</li> <li>— Analyses diverses à l'échelle de la ville [47, 48, 3, 49, 50, 51, 52, 53]</li> <li>— Prédiction de trajets individuels [52, 54]</li> </ul>
Capteurs GPS	Jeux de données majoritairement privés, mais certains publics avec prétraitement	Utilisable avec ou sans variables décrivant les traces	Détail des trajets individuels, sauf si application d'un prétraitement d'anonymisation spécifique	<ul style="list-style-type: none"> <li>— Prétraitement des traces [55, 56, 57, 58, 59, 60]</li> <li>— Analyse d'infrastructures cyclables [61, 62]</li> <li>— Modèles de choix d'itinéraire [63, 64, 65, 66, 67]</li> </ul>
Comptage de vélos	Données publiques	Pas d'études utilisant des méthodes d'apprentissage trouvées	Nombre de vélos à certains points dans la ville, données assez représentatives théoriquement	<ul style="list-style-type: none"> <li>— Prédiction du nombre moyen annuel de cyclistes à certains points [68, 69, 70, 71]</li> <li>— Quantification/Amélioration de la représentativité de jeux de données de traces GPS [62, 72, 73, 74, 75, 76]</li> </ul>

TABLE 2.1 – Tableau résumé des quatre sources de données et des caractéristiques les différenciant.

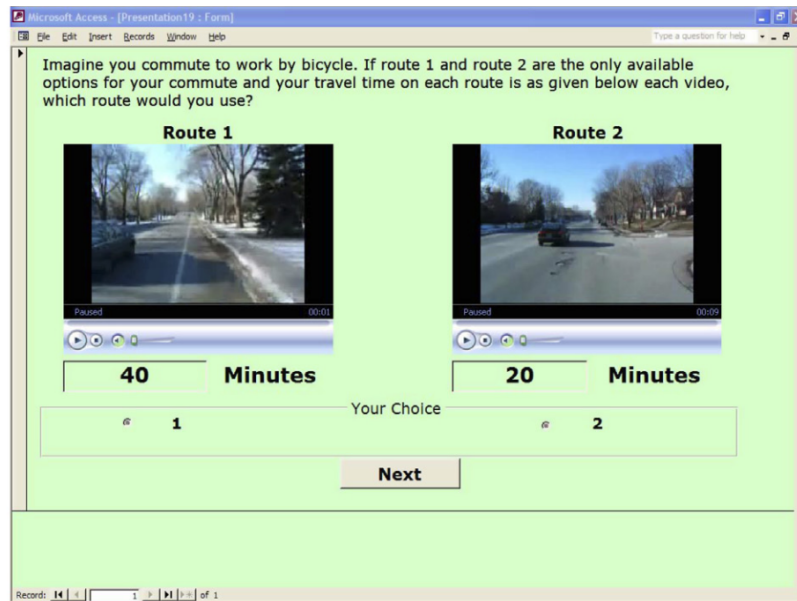


FIGURE 2.1 – Exemple de question posée par Tilahun et al.[10]

d'infrastructure en continu. Chaque itinéraire a un temps de trajet affilié, celui utilisant l'infrastructure de qualité supérieure ayant un temps de trajet initial de 40 minutes et celui utilisant l'infrastructure de niveau inférieur un temps de trajet de 20 minutes. Si le répondant choisit l'infrastructure de qualité supérieure, il est de nouveau confronté au même choix, mais le temps de trajet de l'itinéraire de meilleure qualité est augmenté. À l'inverse, s'il choisit l'infrastructure de plus basse qualité, le temps de trajet de l'itinéraire de meilleure qualité est diminué. Les expériences étaient menées durant 4 itérations, et les auteurs concluent entre autres que les cyclistes interrogés sont prêts à rajouter 23 minutes environ pour utiliser une piste cyclable à la place d'une voie sans bande cyclable et voitures garées le long du trottoir.

Après la première analyse des réponses, un ou plusieurs algorithmes de régression logistique sont utilisés pour affilier un poids à chacune des variables descriptives des trajets étudiés. Ces poids permettent de quantifier à quel point une variable influence le choix des personnes interrogées, et si cette influence est positive ou négative. En 2009, Sener et al. [35] ont étudié l'influence de variables démographiques, d'accessibilité aux parkings, d'infrastructure cyclable et de chaussée sur le choix d'itinéraire. Les personnes interrogées étaient séparées en 2 groupes : celles répondant dans le cadre d'un trajet domicile-travail et celles répondant dans le cadre d'un trajet loisir. Les personnes interrogées devaient choisir entre 3 trajets, chacun décrits avec 5 variables descriptives prises sur les 11 identifiées au total (10 seulement pour les trajets loisir, le temps de trajet n'étant pas inclus). Un modèle logistique multinomial mixte est ensuite appliqué sur les réponses. Les auteurs concluent que les cyclistes préfèrent la non-présence de parking le long de leurs itinéraires, et que si un parking doit être présent, ils préfèrent un stationnement en biais à un stationnement parallèle. Les résultats montrent aussi une préférence pour des infrastructures cyclables en continu, ainsi qu'un trafic automobile, une vitesse maximum, un nombre de panneaux stop, de feux de circulation et de croisements faible, ce qui est cohérent avec le résultat de précédentes études. Seul point surprenant, les cyclistes interrogés

préfèrent les faibles pentes aux trajets plats. Ce résultat est influencé par les cyclistes faisant des trajets pour le loisir, car l'effort physique induit par le passage sur une faible pente peut être recherché par ceux-ci. Les auteurs argumentent que leur étude, et ainsi toutes celles du même type, permet de trouver le coût maximum (que ce soit en temps de trajet ou en argent) que les cyclistes interrogés sont prêts à payer pour que leurs trajets possèdent ou ne possèdent pas une caractéristique.

L'évolution des technologies et particulièrement la démocratisation des puces GPS a permis la création de nouveaux moyens de récupération de données de cyclistes. Ces méthodes ne nécessitent pas la création d'un questionnaire et sont capables de récupérer les données de manière passive, limitant ainsi les biais psychologiques, ainsi que de manière massive, limitant les biais démographiques des échantillons étudiés. Cependant, cette évolution technologique a aussi permis le développement de nouveaux moyens de décrire les choix auxquels les auteurs veulent confronter les cyclistes. En 2020, Monsere et al. [37] se sont ainsi filmés à vélo, prenant différentes intersections sans véhicule tournant, avec véhicule tournant au loin et avec véhicule tournant proche, obligeant le cycliste à changer de trajectoire. Ils ont ensuite demandé à 275 personnes interrogées de noter le niveau de confort de chacun des extraits vidéo. Sans surprise, la piste cyclable reçoit la meilleure note de confort, et la bande cyclable reçoit la moins bonne note de confort. Les auteurs remarquent aussi que le niveau de confort perçu baisse quasiment systématiquement lorsqu'un véhicule tourne près du cycliste, à l'exception de la bande cyclable. Les personnes interrogées étaient ensuite invitées à choisir parmi 4 types d'intersection et d'expliquer leur choix. Le média vidéo permet dans cette étude de montrer différentes situations plus ou moins dangereuses à différents types d'intersections sans avoir à décrire celles-ci textuellement. Cela permet aux personnes interrogées d'expliquer leurs choix sans être biaisés par la description des situations et/ou des intersections faites par les auteurs.

## 2.2 Les systèmes de vélo en libre service

Les systèmes de vélo en libre-service (BSSs pour *Bike-Sharing Systems*) se démocratisent depuis une vingtaine d'années partout dans le monde<sup>2</sup>. Les plus anciens sont constitués de stations disséminées dans la ville sur lesquelles sont attachés les vélos. Les utilisateurs empruntent et rendent les vélos à ces stations. Il existe deux types de données issues de ces systèmes. Les premières sont les données dites "ouvertes", car disponibles en ligne dans la plupart des cas. Ces données sont composées des localisations et tailles de toutes les stations ainsi que du nombre de vélos accrochés à celles-ci en temps réel. Un exemple graphique des données ouvertes du système de BSS de Lyon est montré dans la Figure 2.2. D'autres données sont à accès restreint. Celles-ci sont composées des stations de départ et d'arrivée de chacun des trajets effectués sur une période donnée. À cela s'ajoute un temps de trajet ou un horodatage de départ et d'arrivée. On trouve aussi souvent un identifiant d'utilisateur permettant probablement de retrouver un individu, posant donc des problèmes de respect de la vie privée. Ces données ne sont donc accessibles que dans certaines situations, chaque ville étant libre de les distribuer selon ses propres modalités. Plus récemment, un nouveau type de BSS se développe dans différentes villes. Ces nouveaux BSSs fonctionnent de manière très similaire aux anciens, mais n'ont plus de stations.

2. [https://en.wikipedia.org/wiki/List\\_of\\_bicycle-sharing\\_systems](https://en.wikipedia.org/wiki/List_of_bicycle-sharing_systems)

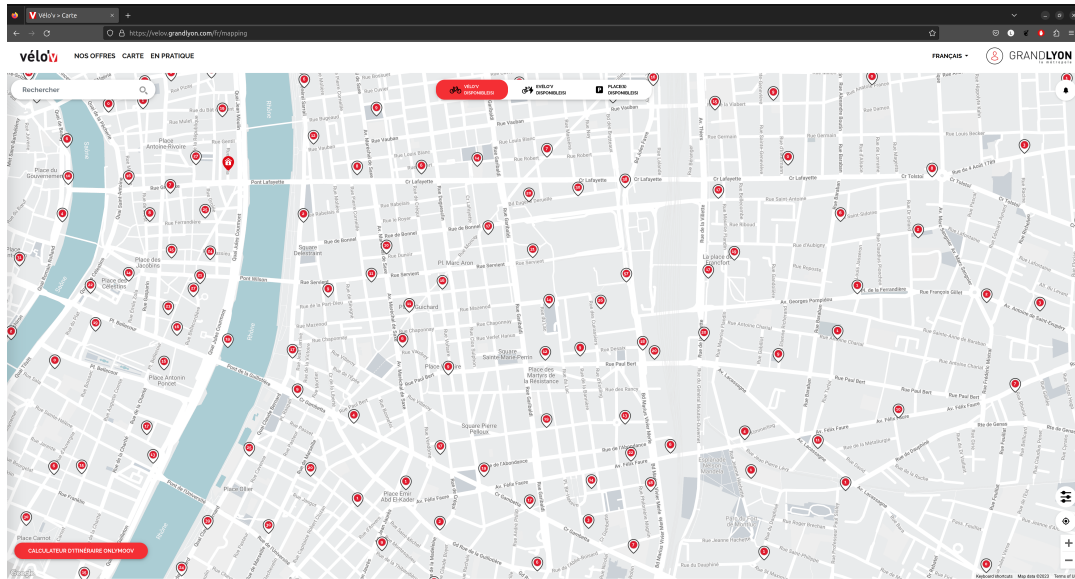


FIGURE 2.2 – Carte contenant une partie des données ouvertes de Velo’v, le BSS de Lyon.

Les utilisateurs peuvent emprunter et rendre les vélos partout dans la zone d’opération du système (souvent une ville entière), à condition de respecter certaines règles dictées par la ville. Ces nouveaux systèmes n’ont pas de données ouvertes. Cela s’explique par l’absence de stations, mais aussi par la nature privée des exploitants de ces systèmes, là où les systèmes à stations sont dans la plupart des cas des délégations de service public. Cependant, les BSSs sans stations peuvent fournir des données de trajets ayant la même structure que celles des anciens systèmes. À noter que certains BSSs, à stations ou non, mettent à disposition des vélos équipés de capteurs pouvant être de différentes natures [77]. Les plus couramment utilisés sont les capteurs GPS qui permettent de récupérer les traces des trajets. Celles-ci ne sont cependant pas souvent mises à la disposition du public. Le cas de figure des traces GPS n’est pas évoqué ici, mais dans la Section 2.3.

Baucoup d’études utilisent des données issues de BSSs [48, 51, 47, 50, 40, 43, 53, 38, 39, 46, 52, 44, 3, 54, 41, 49, 42, 45]. Cela s’explique d’abord par leur facilité d’accès. Ces données ne sont pas considérées comme privées, car rien ne permet de relier un individu à un trajet si l’identifiant utilisateur est supprimé. Certaines villes sensibles aux améliorations rendues possibles grâce aux travaux de recherche n’hésitent pas à fournir leurs données. Il existe plusieurs problématiques pouvant être étudiées à l’aide des données provenant d’un ou plusieurs BSSs. Les impacts produits par l’installation d’un de ces systèmes à l’échelle de la ville en font partie. Les impacts étudiés sont de différentes natures : ils peuvent être d’ordre environnemental [47, 48, 3], relatif au trafic routier [47, 48], au trafic cyclable [49, 50], à la santé [47, 48], et même à l’utilisation du système en lui-même [51, 49, 52, 53]. Ces analyses sont intéressantes pour améliorer les BSSs existants, mais aussi le déploiement des futurs systèmes. Cependant, elles n’essaient pas ni de modéliser ni de prédire les comportements individuels des utilisateurs de ces systèmes.

La structure des données issues des BSSs à stations est assez simple, l'espace des coordonnées géographiques possibles étant limité par le nombre de stations existantes. Cette particularité permet de facilement utiliser des algorithmes d'apprentissage supervisé explicites [38, 42] ou implicites [39, 40] sur ce type de données. Cela permet de faire de la prédiction en plus de l'analyse simple. Beaucoup d'études tentent de prédire l'état futur du système. Ce type de prédiction est primordial au bon fonctionnement des BSSs car il permet d'optimiser le processus de rééquilibrage. En effet, certaines stations voient leurs vélos plus souvent empruntés que rendus, et inversement. Pour assurer une offre cohérente avec la demande, des transferts de vélos doivent être effectués entre certaines stations. Pour optimiser le rééquilibrage, certains auteurs croisent les données de trajets avec des données de trafic, de météo et/ou événementielles. Ces croisements sont utilisés en entrée d'algorithmes d'apprentissage supervisé implicites et/ou explicites dans le but de prédire le nombre d'emprunts et/ou de rendus de vélos à différentes échelles [38, 39, 40, 41, 42, 43, 44, 45, 46]. En 2019, Li et al. [42] ont regroupé les stations proches géographiquement et ayant des matrices de transitions similaires à l'aide d'un algorithme d'apprentissage non supervisé. Ils ont ensuite utilisé un algorithme d'apprentissage supervisé et un outil d'inférence ainsi que des données temporelles (heure de la journée et jour de la semaine), météorologiques et événementielles pour prédire le nombre de départs de trajets à trois échelles différentes (ville entière, clusters de stations et stations individuelles). Enfin, ils ont conçu une méthode utilisant la matrice de transition entre *clusters* et une approximation du nombre total de départs de trajets pour déduire le nombre d'arrivées à chacune des échelles. Leurs travaux surpassent les capacités de prédiction des méthodes précédentes à l'échelle de la ville et des clusters, mais échouent à prédire correctement l'utilisation des stations individuellement, celle-ci étant trop variable.

Ces dernières années, les réseaux de neurones profonds ont démontré leurs capacités prédictives dans de nombreux domaines. Les réseaux de neurones récurrents sont particulièrement performants face à des problèmes utilisant des données temporelles, telles que celui de la prédiction de l'état futur d'un BSS. En 2019, Pan et al. [40] ont utilisé un réseau LSTM (*Long Short Term Memory* : un type de réseau de neurones récurrent adapté aux longues séquences détaillé dans le Chapitre 3) sur les données du BSS de New-York pour prédire le nombre de vélos loués et rendus dans le système. Leur réseau de neurones est capable d'effectuer des prédictions sur des stations individuelles à l'échelle horaire. La structure du LSTM permettant de prendre en compte les aspects temporels des données, les capacités de prédiction qui en résultent surpassent celles des réseaux de neurones dits *fully-connected* (c.-à-d. les réseaux de neurones profonds ayant la structure la plus simple). Les réseaux LSTM peuvent également être utilisés sur les systèmes de BSS sans station. Les données issues de ce type de BSS sont plus compliquées à traiter, car les points de départ et d'arrivée sont dans un domaine pouvant être considéré comme continu contrairement aux BSSs plus anciens. Des études récentes divisent la ville en plusieurs parties et utilisent un réseau LSTM pour prédire la distribution des vélos [45] ou les départs et arrivées de trajets [46] dans chacune de ces parties. Toutes ces études sont intéressantes, car elles démontrent que l'utilisation d'outils de prédiction sur les données BSSs permet de modéliser une partie du comportement des cyclistes à différentes échelles. Cependant, même l'échelle la plus précise (les stations) ne permet pas de faire de la prédiction de comportements individuels, le but de ces études étant uniquement de prédire l'état futur du BSS.

Malgré la prédominance des travaux de prédictions concernant le rééquilibrage du système, des auteurs ont tenté d'utiliser les données BSS à d'autres fins. En 2016, Zhang et al. [52] se sont intéressés à la prédiction de trajets. Cette problématique consiste à prédire les caractéristiques d'un trajet individuel. Ils ont utilisé un premier algorithme de régression pour prédire la station d'arrivée et un second pour prédire le temps de trajet. Leur étude est centrée sur les variables descriptives à fournir aux algorithmes pour obtenir la plus grande précision de prédiction possible. Ces variables sont propres aux stations, à l'utilisateur et au moment de départ. Les auteurs concluent que c'est la combinaison de toutes ces variables qui permet de faire les meilleures prédictions. Cette étude prouve la possibilité de prévoir un trajet individuel à partir des données BSSs. Cependant, les données prédites comprennent uniquement l'origine, la destination et le temps du trajet. La prédiction finale n'est que partielle, car les données ne contiennent pas assez d'informations sur les trajets effectués. Une reconstitution plus complète est néanmoins possible à condition d'ajouter de fortes hypothèses. C'est ce qu'ont fait en 2017 Caggiani et al. [54]. Ces auteurs ont développé un modèle de choix d'itinéraire en utilisant les données ouvertes de deux BSSs ainsi que les conclusions de différents modèles de choix d'itinéraire publiés au préalable. Ce modèle de choix d'itinéraire est implémenté dans une application de test et fonctionne comme suit. Premièrement, l'utilisateur choisit un point de départ et un point d'arrivée. À partir des données BSSs, le système calcule le chemin le plus court jusqu'au vélo en libre-service le plus proche. Le modèle calcule ensuite un chemin optimal pour atteindre le point d'arrivée dans le cas d'un vélo sans station, ou de la station la plus proche du point d'arrivée sinon. Le chemin calculé est optimal au regard des hypothèses formulées par les auteurs, provenant des conclusions de précédentes études. Il est obtenu en pondérant le graphe routier avec des données structurelles, de trafic et de pollution. Caggiani et al. réussissent à prédire le comportement d'un cycliste individuel de manière détaillée. Cependant, ils ne prennent en considération que la perception de la pollution de l'air et la perception de sécurité des itinéraires possibles, toutes les deux calculées de manière assez arbitraire. De plus, leur modèle de choix d'itinéraire n'est jamais confronté à des données réelles.

## 2.3 Les capteurs GPS

La démocratisation des capteurs GPS depuis une vingtaine d'années a permis le développement d'une nouvelle manière de collecter des données de trajets de cyclistes. Cette méthode de récupération a l'avantage d'être passive et mise en place dans des conditions réelles. Les traces récupérées sont constituées de points GPS que le capteur enregistre à intervalles réguliers. Chaque point est constitué à minima d'une latitude, d'une longitude et d'un horodatage. À cela peuvent s'ajouter des informations supplémentaires comme l'altitude ou une estimation de la vitesse par exemple. Les traces GPS de cyclistes présentent cependant deux inconvénients qui demandent chacun un prétraitement distinct avant leur utilisation. Le premier est un problème de précision inhérent aux capteurs GPS utilisés. En effet, il arrive à une fréquence non négligeable qu'un ou plusieurs points GPS soient imprécis, c'est-à-dire éloignés de l'endroit où se trouvait réellement le capteur lorsqu'ils ont été enregistrés [78]. Pour pouvoir utiliser un jeu de données de traces GPS, les traces contenant ces imprécisions doivent être *mapmatchées* lorsque cela est possible, ou



supprimées sinon. Les algorithmes de *mapmatching* ont pour objectif de transformer une trace composée d'une succession de points géographiques en une succession d'arêtes d'un graphe routier. Chaque point géographique doit être ramené sur l'arête sur laquelle se trouvait le capteur GPS lors de l'enregistrement de celui-ci. L'ensemble des arêtes ou l'ensemble des points ramenés sur les arêtes peuvent ensuite être utilisés, suivant les besoins. Le deuxième problème est spécifique à l'étude de traces GPS cyclables. Dans de nombreuses études, les capteurs sont portés par les candidats durant toute la journée. Il faut donc diviser l'unique trace récupérée au cours de la journée en différentes activités, puis les labelliser dans le but de retrouver les sous-traces capturées durant un trajet à vélo. Les problématiques liées au prétraitement des traces GPS forment une discipline scientifique qui mérite que l'on s'y attarde un instant.

De nombreuses études se concentrent sur ces problématiques [55, 56, 57, 58, 59, 60]. En 2021, Lisner et al. [60] ont développé une méthode complète de prétraitement orientée sur le cyclisme. Les auteurs ont commencé par calculer la vitesse du capteur à chaque point GPS, puis ont supprimé les points ayant une vitesse supérieure à 90 km/h, les considérant comme aberrants. Ils ont ensuite appliqué un algorithme de lissage de gradient dans le but de lisser les vitesses pour que celles-ci soient plus réalistes. Pour la segmentation en différentes activités, les auteurs ont remarqué que lorsque le porteur du capteur n'est pas en train d'effectuer un trajet, les variations de vitesse et d'orientation entre les points GPS sont plus élevées. Ils ont donc développé une méthode de calcul utilisant les variations de vitesse et d'orientation dans une fenêtre de temps pour identifier si un point GPS fait partie d'un trajet ou non. Un algorithme d'apprentissage supervisé implicite est ensuite utilisé sur des traces GPS de test pour la détection du mode de transport. Des variables descriptives du trajet telles que la distance, la distribution des vitesses ou encore le détour par rapport au trajet à vol d'oiseau sont utilisées. Le résultat de cette étude est une méthode flexible, transparente et interprétable de prétraitement de traces GPS, avec une bonne précision pour la reconnaissance de trajets cyclables.

Une fois les traces GPS nettoyées, le jeu de données peut être utilisé à différentes fins. Même si certaines études se concentrent sur d'autres problématiques comme l'analyse des infrastructures cyclables [61, 62] ou la représentativité de jeux de données [79], la grande majorité des études utilisant des traces GPS cyclables font de la modélisation de choix d'itinéraire [63, 64, 65, 66, 67]. Les traces GPS de cyclistes ne suffisent pas à modéliser leurs choix d'itinéraire. Celles-ci décrivent les choix effectués, mais ne donnent aucune information sur les autres possibilités qui s'offraient aux cyclistes lorsqu'ils les effectuaient, ni sur les motivations de leurs choix. Il faut donc au préalable construire un jeu de trajets possibles artificiellement. Une version naïve de construction de ce jeu de trajets possibles est d'utiliser le chemin le plus court entre les origines et destinations des traces, mais des méthodes plus complexes utilisant notamment le *clustering* des origines et des destinations [64], l'optimisation d'attribut [65] ou encore la pénalité de lien [63] existent. La construction de trajets de choix est une question scientifique à part entière à laquelle certaines études sont entièrement dédiées [80]. Une fois le jeu de trajets possibles généré, des variables descriptives de différentes natures sont choisies pour décrire chaque trajet. Ces variables sont du même type que celles choisies dans les études utilisant les questionnaires de préférences, et tout comme dans celles-ci, un ou plusieurs algorithmes de

régression logistique sont utilisés pour affilier un poids à chaque variable. La prédiction est aussi faisable à partir des modèles finaux. L'utilité d'un trajet est calculable à partir du modèle, et prédire quel trajet est le plus proche du comportement d'un cycliste est donc possible en choisissant celui ayant l'utilité la plus grande à partir de différents trajets possibles. Une description plus détaillée des modèles de choix d'itinéraire discrets est faite dans la Section 3.1.

Les traces GPS générées par des cyclistes permettent donc de décrire leurs comportements individuels en situation réelle. À partir de celles-ci, il est possible d'analyser, de modéliser et même de prédire ces comportements. De plus, les traces GPS sont compatibles avec certaines méthodes d'apprentissage profond, moyennant des prétraitements spécifiques [81]. Néanmoins, les traces GPS individuelles sont considérées comme données privées, car elles peuvent donner des informations sur le lieu de vie et/ou de travail des individus. Il n'est donc pas possible de les publier sans utiliser d'algorithmes d'anonymisation au préalable. Après des recherches tant internes qu'externes au laboratoire, 3 jeux de données de traces GPS ont été identifiés. Deux sont publics et le dernier est privé. Les Sections 2.3.1, 2.3.3 et 2.3.2 présentent ces jeux, et décrivent les travaux de recherches effectués avec ceux-ci jusqu'à présent.

### 2.3.1 Strava Metro

Strava est une solution commerciale permettant d'enregistrer des activités physiques. Conçue à l'origine par des cyclistes et principalement utilisée par ceux-ci, Strava détient un jeu de données de traces GPS cyclables *crowdsourcé* mondial mis à disposition de la recherche et des politiques urbaines par le biais du projet Strava Metro<sup>3</sup>. La nature privée des traces GPS empêche la distribution du jeu de données brut. Strava agrège donc celles-ci en premier lieu. Le jeu de données mis à disposition est un graphe routier avec les données de cyclistes ajoutées sur chaque segment de route et sur chaque intersection, ainsi que des polygones décrivant les origines et destinations. Un exemple de graphe routier contenant les données Strava de la ville de Glasgow est montré en Figure 2.3. Les données ajoutées au graphe sont le nombre de cyclistes qui sont passés sur le segment ou à l'intersection sur une période donnée (heure, année, jour de semaine ou weekend), ainsi que le temps d'attente moyen (uniquement aux intersections). Strava met à disposition leurs données par zones géographiques. Les équipes de recherche travaillant avec les responsables de la politique urbaine de leurs villes y ont accès gratuitement. Ces données sont donc beaucoup utilisées dans la littérature [82]. Celles-ci permettent, tout comme les données BSSs, de faire différentes analyses à l'échelle de la ville. La nature plus détaillée des données Strava permet cependant d'effectuer des analyses à plus fine échelle. Il est par exemple possible d'évaluer l'impact de l'installation d'une nouvelle infrastructure cyclable en comparant les données Strava dans la zone concernée avant et après l'installation [73, 83]. Certains auteurs se sont aussi intéressés à l'exposition des cyclistes à la pollution [84, 85] ou aux risques d'accidents [75]. Pour ce qui est du comportement des cyclistes, les données Strava permettent tout d'abord d'identifier les caractéristiques de leurs trajets, qu'elles soient géographiques [86], temporelles [87] ou selon le type de trajets (domicile-travail ou autre) [85]. Associées à des données infrastructurelles et/ou de compteurs de vélos, les données Strava peuvent aussi augmenter la précision de modèle de prédiction de volume de cyclistes, que ce soit au niveau de compteurs [76, 68] ou sur le réseau entier [88]. Enfin,

3. <https://metro.strava.com/>

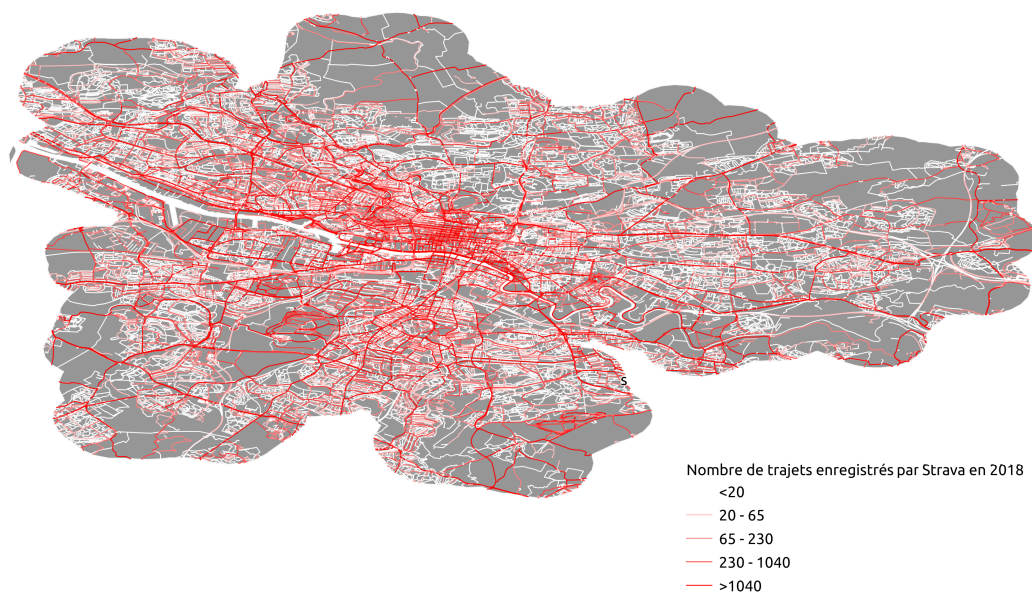


FIGURE 2.3 – Carte affichant certaines données Strava ajoutées au graphe routier de la ville de Glasgow.

les données Strava permettent de faire de l'analyse de choix d'itinéraire. Une étude s'intéressant aux comportements individuels des utilisateurs de l'application Strava a été conduite par Huber et al. en 2019 [89]. Les auteurs ont réussi à reconstruire un trajet à partir d'une origine, d'une destination et des données Strava. Ils ont pour cela créé des zones de transit et ont utilisé un algorithme de routage associé à deux contraintes. Les trajets créés ont été validés par un modèle de choix d'itinéraire existant.

### 2.3.2 Véléal

Véléal est un projet lancé en 2016, et financé par le LabEx IMU<sup>4</sup>. Son but est d'évaluer la cyclabilité en milieu urbain, en intégrant dans l'analyse les conditions matérielles, techniques, mais aussi sociales. Durant ce projet, 40 cyclistes volontaires ont été invités à effectuer des trajets domicile-travail à vélo de mai 2016 à octobre 2019 à Lyon et à Saint-Étienne. Chaque cycliste était équipé d'un capteur GPS ainsi que d'une caméra. Les cyclistes participaient ensuite à un entretien qui leur permettaient, avec l'appui des vidéos de leurs trajets, d'expliquer les choix fait durant ceux-ci. Une collecte participative a parallèlement été menée. Le résultat de ce projet est, entre autres, un jeu de données privé de traces GPS de cyclistes. Certaines traces de ce jeu de données sont montrées en Figure 2.4.

À cause de son statut de jeu de données privé, Véléal n'a pas été beaucoup utilisé dans la littérature. Les porteurs du projet ont sorti un article en 2020 [27] ainsi qu'un chapitre de livre en 2022 [90]. Cependant, ceux-ci se concentrent plus sur ce

4. <https://imu.universite-lyon.fr/>



FIGURE 2.4 – Carte affichant certaines traces GPS du jeu de donnée Véloval.

qu’apporte la capture de vidéos en parallèle de la capture de traces GPS pour des entretiens sociologiques que sur l’analyse des traces GPS seules. L’objectif du projet était de comprendre plus que de modéliser le comportement des cyclistes.

### 2.3.3 MonRésoVélo

MonRésoVélo est une initiative de la ville de Montréal. En 2013, l’application mobile MonRésoVélo, dont un aperçu de l’interface est montré en Figure 2.5, est mise à la disposition du public. Cette application permet à ses utilisateurs d’afficher les infrastructures cyclables de la ville de Montréal, de sauvegarder et de consulter leurs trajets à la manière d’un tableau de bord, et d’afficher la vitesse moyenne et une estimation des calories dépensées lors du trajet en cours. Pour la ville, c’est un moyen de récupérer les traces GPS des cyclistes volontaires pour mieux comprendre leurs comportements, dans le but d’améliorer les aménagements cyclables. Les traces GPS sont anonymisées en supprimant les premiers et derniers points GPS ainsi que l’horodatage, et mises à la disposition de tout le monde<sup>5</sup>. Le jeu de données ainsi constitué contient 4881 traces GPS, générées exclusivement par des cyclistes entre juin 2013 et octobre 2015. D’autres informations sont disponibles, comme le motif du trajet ou les heures de départ et d’arrivée.

Malgré son statut de jeu de données public, MonRésoVélo n’est que peu utilisé dans la littérature. Cela est probablement dû à une méconnaissance de son existence, ainsi qu’à sa taille pouvant être considérée comme insuffisante pour certains travaux. Le peu d’études effectuées avec ces données, écrites par Strauss et al. entre

5. <https://donnees.montreal.ca/ville-de-montreal/trajets-individuels-velo-enregistre-mon-resovelo>

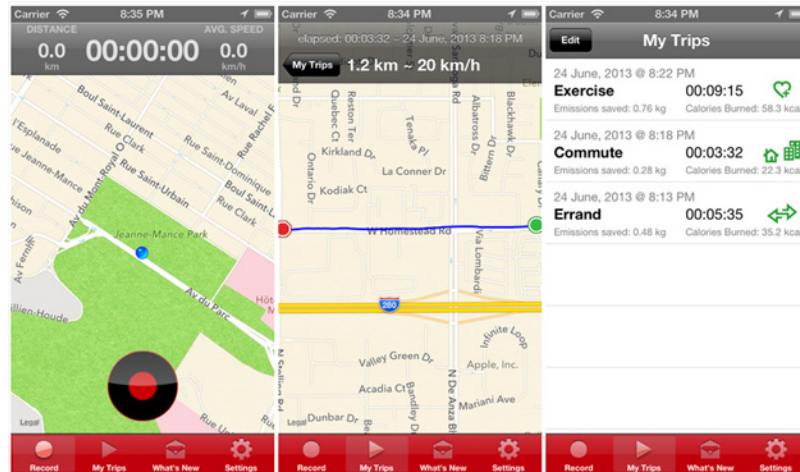


FIGURE 2.5 – Visuel de l’application MonR&eacute;V&eacute;lo.

2015 et 2017 [72, 74, 91], utilisent d’ailleurs directement les donn&eacute;es g&eacute;n&eacute;r&eacute;es pendant le projet sans avoir &agrave; passer par le portail de donn&eacute;es ouvertes de la ville de Montr&eacute;al. Leur jeu de donn&eacute;es est donc plus grand, contenant 10000 traces GPS g&eacute;n&eacute;r&eacute;es par environ 1000 cyclistes diff&eacute;rents. Ces &eacute;tudes s’int&eacute;ressent &agrave; la vitesse des cyclistes aux intersections, ainsi qu’aux risques d’accidents. Elles d&eacute;montrent entre autres une forte corr&e9;lation entre les endroits o&ugrave; les utilisateurs de l’application effectuent de fortes d&eacute;cel&eacute;rations et les accidents impliquant un cycliste rapport&eacute;s aux autorit&eacute;s.

## 2.4 Les comptages de v&eacute;los

Compter le nombre de v&eacute;los passant &agrave; un endroit pr&eacute;cis dans la ville ne permet que de quantifier l’utilisation d’une voie ou d’un type de voie. Cette quantification est plut&otilde;t repr&e9;sentative en th&e9;orie, car elle englobe tous les cyclistes passant par un point donn&eacute;. Lorsque les comptages sont diss&e9;min&e9;s dans une zone g&e9;ographique donn&eacute;, le comportement des cyclistes peut &eatre; &eacute;tudi&e9; de mani&eere; plus globale. Il existe diff&e9;rentes mani&eeres; de compter les v&eacute;los, class&e9;es en deux types : les mani&eeres; temporaires et les mani&eeres; durables. Les comptages temporaires peuvent &eatre; faits manuellement en positionnant une personne &agrave; une intersection par exemple, et en lui demandant de compter les cyclistes qu’il voit. Ils peuvent aussi &eatre; automatis&e9;s en positionnant un capteur pneumatique sur la voie que l’on souhaite &eacute;tudier. Les comptages durables sont eux effectu&e9;s via l’installation de compteurs automatiques fixes. Ces compteurs peuvent prendre la forme de d&e9;tecteur/cam&e9;ra infrarouge, mais la grande majorit&e9; sont des boucles d’induction magn&e9;tique. Ce type de capteur utilise un aimant pour induire un courant &e9;lectrique lors du passage d’un v&e9;hicule. La reconnaissance du type de v&e9;hicule d&e9;tect&e9; est difficile [92], obligeant ces capteurs &agrave; &eatre; d&e9;ploy&e9;s sur des pistes cyclables s&e9;par&e9;es. Une fois install&e9;s, les compteurs enregistrent des donn&e9;es sur des p&e9;riodes relativement longues, plusieurs ann&e9;es dans la plupart des cas. Les comptages durables remontent donc des donn&e9;es plus repr&e9;sentatives que les comptages temporaires, qui sont eux effectu&e9;s sur des p&e9;riodes beaucoup plus courtes allant de quelques heures &agrave; quelques jours. Installer un compteur de v&e9;los automatique peut &eatre; fait pour diff&e9;rentes raisons. Premièrement, certains de ces compteurs affichent en temps r&e9;el le nombre de cyclistes compt&e9;s le jour et/ou l’ann&e9;e courant(e). Cela a pour but d’informer les

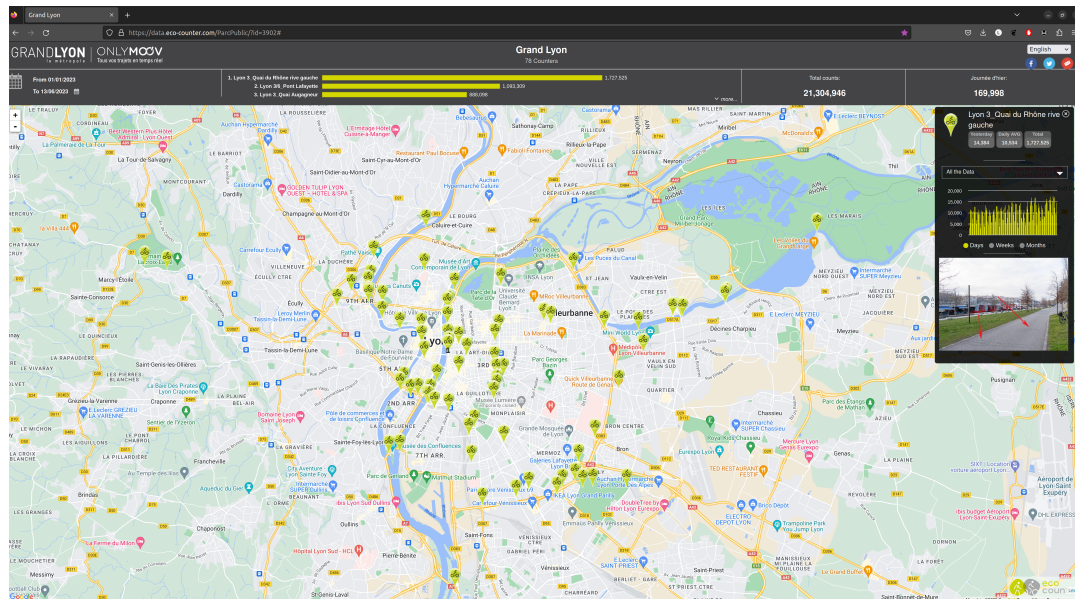


FIGURE 2.6 – Carte montrant les compteurs automatiques de vélos dans la métropole de Lyon.

habitants et de les inciter à se déplacer à vélo. Deuxièmement, la plupart des villes publient gratuitement les données de ces compteurs, ce qui permet d'étudier les flux de cyclistes à leurs emplacements. Une carte interactive présentant les compteurs de vélos de la ville de Lyon est montrée en Figure 2.6. Les données publiées contiennent la position des compteurs ainsi que le nombre de cyclistes détectés par ceux-ci pendant une période donnée. Les agrégations temporelles des mesures proposées diffèrent selon les villes, pouvant aller de l'heure jusqu'à l'année entière.

Beaucoup de villes mettent les données de leurs capteurs de vélos permanents à disposition du public, permettant la publication de nombreuses études utilisant celles-ci [69, 71, 68, 11, 70]. En 2021, Nordengen et al. [11] ont utilisé les données de 89 compteurs de vélos en Norvège pour étudier l'évolution du nombre de cyclistes. Ils ont pour cela calculé l'indice de trafic cyclable à 3 échelles (nationale, régionale et locale). Cet indice est le nombre de cyclistes comptés pendant une année divisé par le nombre de cyclistes comptés pendant une année de référence. Les 89 compteurs étant tous opérationnels depuis le 1er janvier 2018 minimum, c'est naturellement cette année qui a été choisie comme année de référence. Les auteurs observent une augmentation de 11% des cyclistes comptés sur la période 2018-2020 au niveau national, mais avec des différences au niveau local et régional. Ils concluent que leurs observations au niveau national sont en accord avec la littérature au regard de l'impact de la Covid-19 sur la pratique du cyclisme, et que les différences locales et régionales mettent en évidence la nécessité de privilégier les politiques cyclables à ces échelles plutôt qu'à l'échelle nationale. Certaines études font plus que de l'analyse en tentant de prédire le nombre moyen de cyclistes moyen annuel à différentes échelles. En 2016, Schmiedeskamp et al. [71] ont construit un modèle estimant la moyenne annuelle de cyclistes comptés par jour de la semaine au pont de Fremont à Seattle, où un compteur automatique est installé. Ils ont pour cela entraîné un modèle binomial négatif en y intégrant des variables descriptives saisonnières (la durée du jour et la fermeture ou non de l'université proche du point étudié) et météorologiques (température et précipitations entre autres). À celles-ci s'ajoute le jour de la semaine ainsi

que le numéro du jour depuis le début de la période étudiée, car la tendance générale du nombre de cyclistes comptés par jour au point étudié est à la hausse durant les 2 ans pendant lesquels l'étude s'est déroulée. Les auteurs concluent que le modèle binomial négatif est performant pour la prédiction du nombre de cyclistes par jour et conseillent de réfléchir à une manière de limiter l'impact négatif de la météo sur le nombre de cyclistes, notamment durant l'hiver.

L'accessibilité des données de compteurs de vélos les rend attractives. Cependant, leur principal problème est leur non-continuité spatiale. En effet, les compteurs sont positionnés à différentes localisations dans les villes et ne donnent aucune information sur la situation entre ces points. C'est pour cela que la majorité des études utilisent d'autres données en complément, et notamment des données GPS [62, 72, 73, 74, 75, 76]. Celles-ci utilisent les données de compteurs vélos pour quantifier voire améliorer la représentativité des données GPS utilisées. En 2019, Rupi et al. [62] ont utilisé des données de compteurs de vélos et des traces GPS récupérées suite à un appel public pour étudier l'utilisation par les cyclistes des liens du graphe routier de la ville de Bologne en Italie. Les données de compteurs vélos ont permis d'augmenter la représentativité des traces GPS récupérées : une régression linéaire entre le nombre de traces GPS passant par les liens possédant des compteurs et les mesures de ceux-ci permet de trouver un coefficient multiplicateur  $c$ . Il suffit ensuite de multiplier  $c$  par le nombre de traces GPS passant par un lien pour estimer le nombre de cyclistes passant par celui-ci. Une comparaison de la carte de chaleur obtenue à partir de cette estimation et de celle fournie par Strava permet de confirmer la représentativité des données. Les auteurs génèrent ensuite le chemin le plus court entre les origines et les destinations de chaque trace GPS pour étudier les détours faits par les cyclistes. Ils concluent entre autres que le chemin emprunté par un cycliste est en moyenne 20% plus long que le chemin le plus court entre son origine et sa destination. Cette notion de détours effectués par les cyclistes est utilisée comme hypothèse pour la méthode de création de modèles de choix d'itinéraires implicites décrites dans le Chapitre 3.

## 2.5 Positionnement

Comme expliqué en introduction de ce chapitre, les sources de données choisies sont utilisées individuellement, car la fusion entre les données de deux sources différentes est un travail nécessitant des connaissances préalables sur le comportement des cyclistes et constituant un domaine de recherche à lui tout seul. Dans cette section, les possibilités des 4 sources de données étudiées sont discutées au vu des contributions de cette thèse, et le choix d'utiliser ou non chacune d'entre elle est argumenté.

Les questionnaires de préférence permettent de récolter des données sur les choix faits par les cyclistes. Le niveau de détails de ces données est assez important, car elles contiennent certaines étapes intermédiaires des trajets étudiés. Cela permet une analyse fine du comportement individuel des cyclistes dans une situation donnée, ce qui est notamment utile lors de la création et/ou de l'amélioration d'infrastructures cyclables. Cependant, les situations auxquelles sont confrontés les cyclistes sont choisies à l'avance, et leurs choix sont effectués en situation contrôlée (le plus souvent en laboratoire). Ces deux caractéristiques de la méthode de récupération des données sont sources de biais pouvant amener les cyclistes interrogés à prendre

des décisions non réalistes, c'est-à-dire des décisions qu'ils n'auraient pas pris en situation réelle. Cette particularité rend l'utilisation de données provenant de cette source problématique pour la génération de trajets individuels de cyclistes, car celle-ci nécessite un jeu de données contenant des informations réalistes. Il est donc préférable que les données utilisées soient générées en situation réelle, et non dans un laboratoire face à des choix restreints. Ensuite, les réponses aux questionnaires comprennent quasiment systématiquement des données personnelles. Celles-ci ne sont donc pas publiées par les auteurs. Chaque étude doit donc développer son propre questionnaire, ce qui requiert des connaissances préalables dans le domaine du transport. Enfin, le développement d'un questionnaire de préférences demande de choisir des variables descriptives dans la plupart des cas. En effet, les situations proposées aux cyclistes doivent être décrites, et certaines caractéristiques sont donc préalablement sélectionnées. C'est pour cette raison que les seuls algorithmes d'apprentissage utilisés sur celles-ci sont des régressions. Une régression permet en effet d'assigner un poids à chaque caractéristique choisie. Les études utilisant des questionnaires de préférences se concentrent sur ces poids et analysent quelles caractéristiques influent le plus le choix des cyclistes. L'utilisation d'algorithmes d'apprentissage ne nécessitant pas de variables descriptives serait inadaptée, car ils rendent l'analyse de l'importance des différentes caractéristiques impossibles. À noter que le développement de nouvelles technologies, notamment dans le domaine de la vidéo, a permis la création de questionnaires ne décrivant pas explicitement les situations présentées. Cependant, les études trouvées utilisant ce type de questionnaires ne font que de l'analyse de choix sans utiliser d'algorithmes d'apprentissage. Toutes ces remarques rendent l'utilisation de questionnaires de préférences incompatibles avec les travaux de cette thèse.

Les données issues de systèmes de vélos en libre-service offrent 2 avantages. Premièrement, elles sont facilement accessibles. Une partie est ouverte à tous mais l'accès aux données les plus intéressantes demandent tout de même souvent l'accord de la ville ou de l'opérateur privé du système. Deuxièmement, leur structure est relativement simple. Un point d'origine, un point d'arrivée et une information temporelle (horodatage ou temps de trajet) par trajet. Ces caractéristiques ont permis à de nombreux auteurs d'étudier l'impact de ces systèmes sur les villes dans lesquelles ils sont déployés, mais surtout de faire de la prédiction de comportement des utilisateurs de ces systèmes grâce à des algorithmes d'apprentissage. Cependant, ces prédictions sont dans la plupart des cas à l'échelle du système et non pas à celle de l'individu. Les rares études tentant de prédire des trajets individuels le font soit de manière partielle, soit en formulant plusieurs hypothèses fortes. Cela s'explique par la structure même des données, qui ne contiennent pas d'informations suffisantes pour étudier les trajets individuels des utilisateurs de ces systèmes. Il n'est pas non plus possible de connaître la quantité de ces utilisateurs traversant une intersection donnée pour la même raison. Il est enfin important de noter que ces utilisateurs ne représentent qu'une partie biaisée de la totalité des cyclistes, avec une faible utilisation par les minorités et les personnes à faible revenu aux États-Unis notamment [93]. Il a donc été décidé de ne pas utiliser les données provenant de BSSs.

Les données récoltées par des capteurs GPS n'ont pas les inconvénients de celles récoltées par les deux sources de données précédentes. Les traces GPS décrivent en effet les trajets dans leur entièreté, et sont récoltées de manière passive, ce qui permet



de décrire des trajets effectués en situation réelle. Elles sont de plus utilisables relativement facilement avec des algorithmes d'apprentissage sans utiliser de variables descriptives. Ces caractéristiques font de cette source de données une candidate intéressante pour produire des trajets individuels de cyclistes. Le problème principal de ces données est cependant leur accessibilité limitée. Les traces GPS étant considérées comme des données privées, peu de jeux sont disponibles. Trois ont tout de même été trouvés, présentant chacun leurs propres avantages et inconvénients :

- *Strava* : Cette thèse n'a pas été effectuée en collaboration avec des responsables de politiques urbaines. Strava a pour cette raison refusé la demande d'accès aux données de Lyon qui leur a été adressée. Il a tout de même été possible de récupérer des données en provenance de Grande-Bretagne. Ces données ont été mises à disposition par l'université de Glasgow, qui se propose de faire l'intermédiaire entre Strava et d'autres acteurs de la recherche en transport cyclable. Celles-ci, bien qu'elles reflètent le comportement des cyclistes, décrivent difficilement les comportements individuels. Les études autour de ce jeu de données ont principalement pour but d'analyser et de prédire le comportement des cyclistes à une échelle plus globale, bien que la granularité des analyses soit plus fine que la granularité de celles utilisant les données de BSSs. La seule étude trouvée s'intéressant à la reconstruction de trajets individuels à partir de données Strava n'utilise comme validation qu'un modèle de choix d'itinéraire simple. Les auteurs parlent d'ailleurs eux-mêmes de chemins plausibles, et leur méthode ne peut donc pas être utilisée pour générer un chemin décrivant précisément un comportement individuel. Ces caractéristiques du jeu de données Strava ne lui permettent pas d'être utilisé dans ce travail comme données de base pour génération de trajets individuels. Le jeu Strava a donc été mis de côté au profit de traces GPS non agrégées.
- *Véléval* : Le laboratoire au sein duquel ce travail a vu le jour ayant participé au projet Véléval, il a été possible d'accéder à ce jeu de données privé. Celui-ci présente un biais certain de représentativité, les nombres de traces et de cyclistes les ayant générées étant peu élevés. Ceux-ci ont cependant été jugés suffisants pour que le jeu de données Véléval soit utilisé dans le Chapitre 3. En effet, la représentativité du jeu de données n'est pas un point indispensable au développement de modèles de choix d'itinéraire capable de générer des trajets individuels, tant que les résultats de ceux-ci sont interprétés en gardant à l'esprit ce biais.
- *MonRésoVélo* : Le jeu de données MonRésoVélo est plus représentatif que le jeu de données Véléval car celui-ci est *crowdsourcé* et contient plus de traces générées par de plus nombreux cyclistes. Cependant, il ne peut pas non plus être considéré comme représentatif. De plus, les cyclistes ayant généré les traces de ce jeu de données n'ont pas majoritairement effectué des trajets domicile-travail, contrairement à ceux ayant généré le jeu Véléval. Les différences entre les deux jeux de données sont tout de même intéressantes, car elles permettent une analyse plus détaillée des avantages et des inconvénients des modèles de choix d'itinéraire créés par la méthode détaillée dans le Chapitre 3. Le jeu de données MonRésoVélo est donc utilisé pour créer un modèle de choix d'itinéraire implicite. Celui-ci est comparé au modèle développé avec le jeu Véléval à toutes les étapes de leurs créations respectives.

Les données GPS permettent de connaître le nombre de cyclistes passant à une intersection donnée. Cependant, le manque de représentativité est plus problématique.

En effet, une intersection intelligente doit être capable de fonctionner avec une quantité de trafic réaliste, pouvant être élevée en heure de pointe. Les jeux de traces GPS trouvés ne peuvent que sous-estimer le nombre de vélos passant à une intersection, car ils sont générés par un sous-ensemble de cyclistes. Par conséquent, des données plus représentatives ont été privilégiées.

Les jeux de données de comptages vélos sont nombreux et mis à la disposition du public. Ils sont de plus assez représentatifs du nombre de cyclistes passant aux points où sont installés les capteurs. Ces deux caractéristiques font des données de compteurs de vélos un choix facile pour l'étude du comportement des cyclistes. Cependant, les capteurs sont souvent peu nombreux et disséminés dans la ville. Les données qu'ils remontent sont donc partielles, et la faible densité spatiale de celles-ci empêchent l'inférence des données manquantes. Cette partialité limite les possibilités qu'offre ce type de données. Pour palier ce problème, certains auteurs utilisent les données de compteurs vélos en complément d'autres types de données, des traces GPS principalement. Dans la plupart des cas, les données de compteurs sont utilisées pour améliorer la représentativité des données GPS utilisées. Comme expliqué précédemment, les jeux de données de traces GPS choisis dans le cadre de ce travail de recherche n'ont pas vocation à être représentatifs. Utiliser des données de compteurs vélo pour augmenter leur représentativité aurait été un travail intéressant, mais en dehors du cadre de cette thèse. Cette partialité n'est cependant pas un problème lorsque l'on cherche à quantifier le nombre de cyclistes traversant une intersection. Des hypothèses fortes, notamment relatives au comportement des cyclistes lorsqu'ils traversent l'intersection, doivent tout de même être émises. Cela est moins problématique que lors de la construction de trajets individuels, car la quantité de véhicules est plus importante que leurs comportements lors de la simulation d'une intersection. Des données de compteurs de vélos sont donc utilisées dans le Chapitre 4.

## 2.6 Résumé

Cette section synthétise les différentes sources de données qu'il est possible d'utiliser pour décrire/analyser/prédire le comportement des cyclistes, leurs avantages ainsi que leurs inconvénients, et le choix de les utiliser ou non dans le cadre de ce travail de recherche.

**Les questionnaires de préférences :** Ils permettent d'avoir des informations détaillées sur les choix faits par les cyclistes. Mis devant plusieurs choix en situation contrôlée, les cyclistes peuvent effectuer et expliquer leurs choix. Cela permet de mieux comprendre la perception qu'ont les cyclistes de différentes situations, et donc d'adapter les politiques cyclables. Les situations sont présentées selon certaines variables descriptives. L'utilisation d'algorithmes d'apprentissage sur ces données a pour but d'expliquer l'influence de chacune de ces variables sur le choix des cyclistes. Utiliser les données comportementales issues de questionnaires de préférences sans variables descriptives n'a donc pas beaucoup de sens. Ces questionnaires posent de plus un problème de fiabilité, les choix théoriques faits par les cyclistes en situation contrôlée pouvant être différents de ceux qu'ils effectueraient en situation réelle. À cela s'ajoute le besoin de connaissances dans le domaine du transport pour la conception d'un questionnaire, les réponses des questionnaires précédents n'étant

pas publique. Ces 3 limitations majeures font des questionnaires de préférences un choix peu pertinent pour les travaux de cette thèse.

**Les systèmes de vélos en libre-service :** Les données sont abondantes et accessibles relativement facilement, car elles ne présentent que peu de caractéristiques relatives à la vie privée des cyclistes étudiés une fois un prétraitement basique effectué. Elles permettent d'étudier les flux de cyclistes à l'échelle d'une ville ou d'un quartier. Leur structure simple les rend utilisables par des algorithmes d'apprentissage sans utiliser de variables descriptives, pour faire de la prédiction de flux notamment. Cependant, les systèmes de vélo en libre-service ne récoltent dans la grande majorité des cas que des informations au niveau des stations dont ils sont constitués. Par conséquent, les analyses et les prédictions effectuées avec ces données sont majoritairement faites à l'échelle de la ville, et non à celle de trajets individuels. Cette limitation fait des données de systèmes de vélos libre-service un candidat non compatible avec les travaux de cette thèse.

**Les capteurs GPS :** Les traces GPS générées à l'aide de capteurs GPS outrepassent les limites de celles récoltées à partir des deux sources précédentes. Contrairement aux données des systèmes de vélos en libre-service, les traces GPS permettent d'étudier le comportement individuel des cyclistes. De plus, les informations sont passivement récoltées pendant des trajets réels et, moyennant un prétraitement adapté, sont utilisables par des algorithmes d'apprentissage sans nécessiter la définition de variables descriptives. Néanmoins, les traces GPS sont considérées comme données privées. Un prétraitement est donc nécessaire avant leur publication. Cet inconvénient limite le nombre de jeux de données disponibles publiquement. De plus, le prétraitement effectué avant publication peut faire perdre l'aspect "comportement individuel" des données. C'est notamment le cas des jeux de données Strava. Deux jeux de données de traces GPS sans prétraitements supprimant les comportements individuels ont cependant été trouvés : Véléval et MonRésoVélo. Le Chapitre 3 de cette thèse utilise ces jeux comme base d'une méthode de développement de modèle de choix d'itinéraire implicites. Le premier est le résultat d'un précédent projet auquel le laboratoire a participé. Il n'est pas mis à la disposition du public et n'a donc pas subi de prétraitement. Le deuxième est un jeu de données public dont le prétraitement (la suppression des premier et dernier points GPS de chaque trace) n'a pas supprimé les informations relatives aux comportements individuels des cyclistes.

**Les comptages de vélos :** Les données issues de comptages présentent les mêmes avantages et inconvénients que les systèmes de vélos en libre-service. Les jeux de données sont nombreux et accessibles publiquement, mais ceux-ci ne permettent pas d'étudier les trajets individuels des cyclistes lorsqu'ils sont utilisés seuls. Ils sont toutefois régulièrement utilisés en complément d'autres données, majoritairement des traces GPS. Les données de compteurs de vélos permettent de quantifier et/ou d'augmenter la représentativité des données utilisées. La représentativité n'étant pas un élément nécessaire à l'atteinte des objectifs du Chapitre 3, les données de capteurs de vélos n'ont pas été utilisées en complément des jeux de traces GPS choisis. Cependant, les données de compteurs de vélos sont assez représentatives du passage de cyclistes en un point donné. Celles-ci sont donc utilisées pour paramétrer les simulations d'un feu de circulation intelligent dans le Chapitre 4.

## Chapitre 3

# Modèles de choix d'itinéraire implicites

### 3.1 Introduction

L'objectif de ce chapitre est de générer des trajets individuels de cyclistes à partir de données comportementales. Ces trajets sont appelés des itinéraires cyclables dans la suite. Les études trouvées dans la littérature capables de produire des itinéraires cyclables utilisent majoritairement des modèles de choix d'itinéraire discrets. Les modèles de choix d'itinéraire discrets ont pour but l'analyse des comportements individuels parmi un ensemble limité d'options [94]. Une mesure de préférence, appelée utilité, est calculée pour chacune des options. L'utilisateur, considéré comme un être rationnel, est supposé choisir l'option ayant l'utilité la plus importante. Chaque option est décrite par des caractéristiques sélectionnées en amont. Un poids est affilié à chaque caractéristique. Ces poids sont le plus souvent calculés à l'aide d'une régression logistique. La somme des caractéristiques pondérées par leurs poids permet ensuite de calculer l'utilité de chaque option. Les modèles de choix d'itinéraire discrets créés spécifiquement pour les cyclistes [35, 34, 66, 65, 67, 36, 64, 63, 95] utilisent des itinéraires possibles comme options. Ces modèles sont donc capables de produire des trajets individuels, en sélectionnant l'itinéraire ayant l'utilité la plus importante. Les itinéraires choisis par les cyclistes peuvent être collectés par le biais de questionnaires de préférences [35, 34] ou bien de capteurs GPS [66, 65, 67, 36, 64, 63]. Dans le cas des questionnaires, les cyclistes choisissent leurs itinéraires parmi un ensemble d'itinéraires possibles. Dans le cas de capteurs GPS, les cyclistes choisissent leurs itinéraires eux-mêmes, et les itinéraires possibles doivent être générés à posteriori. Les variables utilisées pour décrire les caractéristiques des itinéraires choisis et possibles peuvent être de différentes natures. Elles sont cependant principalement d'ordre infrastructurel, comme le nombre de virages ou la proportion de pistes cyclables, global, comme la longueur du trajet ou la différence avec le chemin le plus court, dynamique, comme les conditions météorologiques ou le volume du trafic, ou encore social, comme l'âge, le sexe ou bien la possession d'une voiture.

Un problème bien connu associé aux modèles de choix d'itinéraire discrets est la taille très importante de l'ensemble des chemins possibles entre une origine et une destination. L'ensemble des itinéraires envisagés par un cycliste est bien évidemment un sous-ensemble de celui-ci. Ce sous-ensemble est néanmoins inconnu. Comme le modèle doit choisir un itinéraire parmi un ensemble de trajets possibles, la précision de la prédiction du modèle est étroitement liée à la qualité de cet ensemble. De plus, la sélection explicite de variables décrivant certaines caractéristiques des itinéraires ne permet qu'une analyse partielle des choix des cyclistes, car

toutes les caractéristiques ne peuvent être analysées en même temps. Cette limitation est d'autant plus présente que les caractéristiques sélectionnées sont pratiquement les mêmes dans toutes les études. La conception de nouvelles caractéristiques susceptibles d'affecter le comportement d'individus est une tâche difficile et multidisciplinaire. Enfin, les études développant des modèles de choix d'itinéraire discrets se concentrent principalement sur les caractéristiques influençant le plus les choix des cyclistes. En effet, les analyses s'intéressent très souvent plus à la compréhension des poids trouvés par le modèle qu'aux capacités de celui-ci à produire des trajets individuels.

Dans ce chapitre, une méthode de création de modèle de choix d'itinéraire est développée. L'objectif des modèles créés est de générer des itinéraires cyclables à partir d'une paire origine/destination (O/D). Les modèles créés sont dits implicites, car des représentations implicites remplacent les variables descriptives choisies explicitement dans les méthodes plus classiques. Les modèles développés avec cette méthode apprennent des représentations implicites de trajets effectués à l'aide d'algorithmes d'apprentissage automatique. Cette méthode a de plus l'avantage de ne pas nécessiter de jeu de trajets possibles. Les caractéristiques de celle-ci lui permettent donc de limiter les biais relatifs à la sélection des facteurs externes, ainsi qu'à la génération d'un jeu de trajets possibles. Les 2 contributions de ce chapitre peuvent être résumées comme suit :

- Les segments de route préférés des cyclistes d'un jeu de données sont identifiés à l'aide de la création de *clusters*.
- Une méthode de génération de traces basées sur l'expérience utilisant ces segments de route préférés ainsi qu'un réseau de neurones profond est développée.

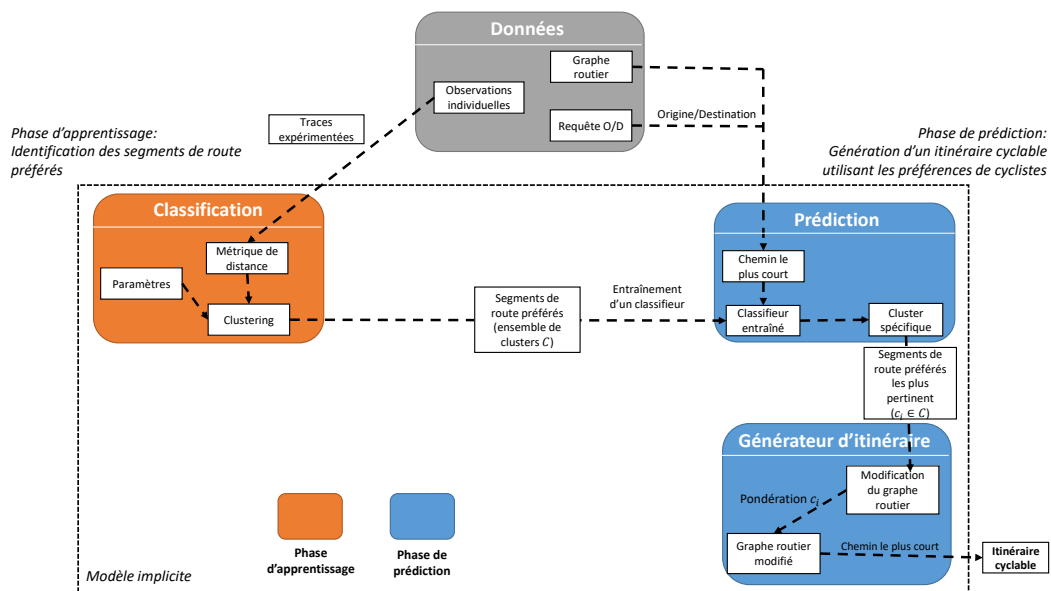


FIGURE 3.1 – Diagramme résumant la méthode de création de modèles de choix d'itinéraire implicites présentée dans ce chapitre.

## 3.2 Détail des données

Le temps de trajet (ou son coût) est le facteur principal pris en compte par les automobilistes lors du choix de leurs itinéraires [96, 97]. Si ce facteur reste crucial pour les cyclistes, les modèles de choix d'itinéraire existants montrent qu'ils sont également influencés par d'autres caractéristiques des trajets considérés. On peut citer par exemple le sentiment de sécurité sur les routes empruntées, le nombre d'intersections, la minimisation des pentes/tournants ou la maximisation de l'utilisation d'infrastructures cyclables [9]. Par conséquent, une première hypothèse est émise :

**Hypothèse 1.** Les cyclistes ont tendance à utiliser des segments de route qu'ils préfèrent, quitte à dériver du chemin le plus court jusqu'à leurs destinations.

Une comparaison qualitative et quantitative entre les itinéraires choisis par des cyclistes et les plus courts chemins entre leurs origines et destinations permet d'obtenir une première intuition sur la forme et la quantité de détours que les cyclistes sont prêts à faire. Dans cette section, les 2 jeux de données utilisés sont d'abord formellement présentés. La manière dont les plus courts chemins sont générés est détaillée et une comparaison quantitative puis qualitative entre ceux-ci et les itinéraires des cyclistes est effectuée.

### 3.2.1 Traces expérimentées

Le choix des jeux de données n'est pas discuté ici, car celui-ci est détaillé en dans le Chapitre 2. Deux jeux de données sont utilisés :

- Un ensemble privé de traces GPS provenant du projet Véléal (voir Section 2.3.2). Le résultat de ce projet est entre autre 2527 traces GPS contenues dans des fichiers GPX. Chaque trace GPS  $t$  est une séquence de  $N$  points ordonnés  $t = \{p_1, \dots, p_N\}$  avec chaque point  $p_t = (lat_t, lon_t, time_t)$  composé d'une latitude, d'une longitude et d'un horodatage.
- L'ensemble de données fourni par la ville de Montréal au Canada, collecté par leur application MonRésoVélo (voir Section 2.3.3). 4881 traces GPS issues de cette application sont publiques. Elles ont été générées par des cyclistes entre juin 2013 et octobre 2015 et sont contenues dans un fichier JSON. Chaque trace  $t$  est une séquence de  $N$  points ordonnés  $t = \{p_1, \dots, p_N\}$  avec chaque point  $p_t = (lat_t, lon_t)$  contenant une latitude et une longitude. D'autres informations telles que l'heure de départ et d'arrivée sont disponibles, seuls les points GPS sont utilisés. Les traces sont *mapmatchées* (voir Section 2.3) et anonymisées.

Les propriétés temporelles des traces sont ignorées, au profit d'une exploitation des propriétés spatiales uniquement. Ce choix a été fait dans le but de faciliter le paramétrage des algorithmes d'apprentissage utilisés plus tard. Les horodatages des traces du projet Véléal sont donc ignorés. L'algorithme Ramer-Douglas-Peucker [98] est utilisé sur les 2 jeux de données afin de réduire le nombre de points GPS des traces en minimisant la perte d'information. Le nombre de points GPS des jeux de données est environ divisé par 26 après utilisation de cet algorithme. Moins de points GPS signifie un temps de traitement plus court. Les différences de performances entre les 2 jeux de données sont discutées tout au long de ce travail. À noter tout de même que les trajets de Véléal sont quasiment tous des trajets domicile/travail là où les trajets MonRésoVélo peuvent avoir été effectués dans d'autres

buts comme la pratique sportive par exemple. Il est important de le préciser dans la mesure où l'hypothèse 1 n'a de sens que si le cycliste utilise le vélo dans le but d'atteindre une destination, et non pour une autre raison. Une méthode permettant de supprimer une partie des traces n'ayant probablement pas été générées dans ce but est explicitée en Section 3.2.3. Les traces GPS des cyclistes sont appelées "traces expérimentées" dans la suite de ce chapitre.

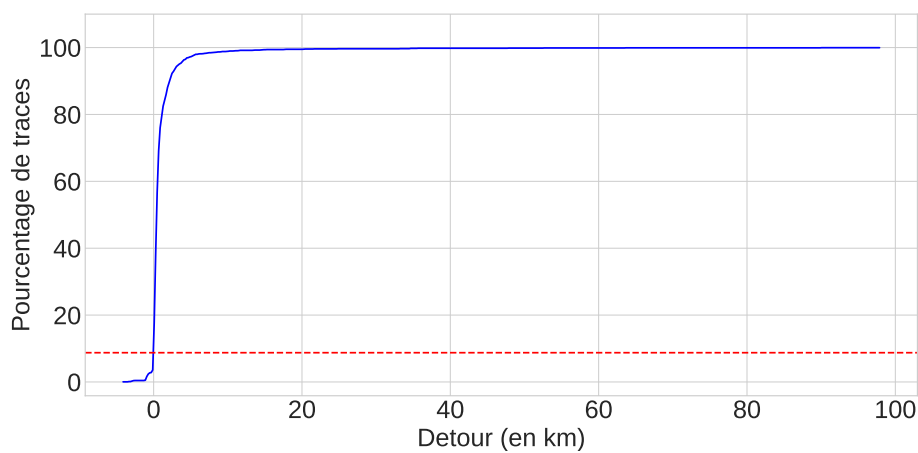
### 3.2.2 Traces générées

Dans le but de valider la première hypothèse émise, le chemin le plus court reliant l'origine et la destination de chaque trace expérimentée est généré en utilisant l'algorithme de Dijkstra. Celui-ci est exécuté sur un graphe routier  $G$  mis à disposition par OpenStreetMap. Un graphe routier est un graphe orienté pondéré qui est une paire  $G = (V, E)$  avec  $V$  un ensemble de sommets et  $E \subset \{(x, y) | (x, y) \in V^2 \wedge x \neq y\}$  un ensemble d'arêtes qui sont des paires ordonnées de sommets. Chaque arête possède un poids. Dans un graphe routier, les sommets représentent les intersections et les arêtes représentent les routes reliant 2 intersections. Les graphes routiers des villes de Lyon et Saint-Etienne pour les traces de Véléal et de Montréal pour les traces de MonRésoVélo sont générés. Leurs tailles sont respectivement de 451344 arêtes reliant 167243 nœuds, 91896 arêtes reliant 34138 nœuds et 1058253 arêtes reliant 381824 nœuds. Les poids attribués aux arêtes sont les longueurs des routes qu'elles représentent. L'algorithme de Dijkstra utilise les poids des arêtes pour calculer le chemin le plus court entre 2 points.

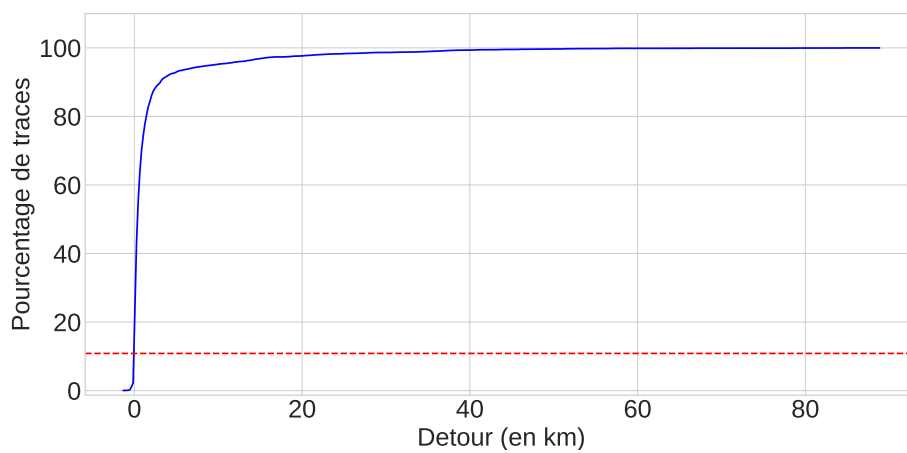
Une fonction de distribution cumulative (CDF pour *cumulative distribution function*) des différences de distance entre les traces expérimentées et les plus courts chemins est tracée par jeu de données. Elles sont montrées en Figure 3.2a et Figure 3.2b. Les détours en dessous des lignes rouges en pointillé sont négatifs. 8,5% des traces expérimentées de Véléal et 10,8% de celles de MonRésoVélo sont donc plus courtes que leurs plus courts chemins. Respectivement 4,5% et 7,7% de ces traces sont plus courtes de moins de 100 mètres. Cela est principalement dû à l'imprécision des capteurs GPS ayant capturé les traces : si une trace expérimentée et son chemin le plus court correspondant passent exactement par la même suite de segments de route, leurs distances ne seront pas les mêmes. Parfois, la trace expérimentée sera légèrement plus courte que le chemin le plus court, même si cela paraît paradoxal à première vue. Pour les traces plus courtes que leurs chemins le plus court restantes (4% des traces Véléal et 3,1% des traces MonRésoVélo), les différences de distance se situent entre 100 m et 4,1 km. Ce phénomène est dû au graphe routier mis à disposition par OpenStreetMap. Certaines routes existantes ne figurent pas dans le graphe. Si une trace expérimentée emprunte l'une d'entre elles, Dijkstra est obligé de faire un détour par rapport à la trace originale. Les traces dans ce cas de figure sont cependant minoritaires, et la différence de distance est dans la plupart des cas peu élevée. Leurs présences dans les jeux de données n'a donc pas d'influence significative sur les performances des modèles créés. Dans la suite de ce travail, les chemins le plus court sont appelés "traces générées".

### 3.2.3 Comparaisons des traces expérimentées et des traces générées

Pour vérifier que les trajets effectués par les cyclistes sont bien différents des chemins le plus court, 2 cartes de chaleurs sont tracées par jeu de données. La première contient toutes les traces expérimentées et la deuxième toutes les traces générées.



(A) Jeu de données Véléal.



(B) Jeu de données MonRésoVélo.

FIGURE 3.2 – CDF des différences de distances entre les traces expérimentées et les traces générées des jeux de données.



Une analyse qualitative des différences permet de confronter l'hypothèse 1. Il faut cependant noter qu'effectuer un détour à vélo entraîne une augmentation de l'effort physique fourni ainsi que du temps de trajet. Ces coûts ne doivent pas annuler les avantages offerts par l'utilisation de routes préférentielles. Dès lors, une seconde hypothèse peut être formulée :

**Hypothèse 2.** Il existe un seuil d'acceptation au-delà duquel les cyclistes refusent de faire un détour plus important.

Une analyse détaillée des CDFs précédemment tracées permet de confronter cette deuxième hypothèse.

### Cartes de chaleurs

Les cartes de chaleurs des trajets Véléval effectués à Lyon sont montrées en figure 3.3. Quel que soit le type de traces, il est possible de séparer la ville en trois : la partie Ouest, la partie centrale et la partie Est. La quasi-totalité des segments de route affichés en rouge se trouve dans la partie centrale, quel que soit le type de traces. Cela démontre une grande densité de traces, signe que le centre est très utilisé par les cyclistes, mais aussi par l'algorithme de chemin le plus court. La partie Est est elle aussi utilisée, mais contient moins de segments de route rouges et plus de segments de route bleus. L'utilisation des segments de cette partie de la ville est plus homogène, quel que soit le type de traces. Enfin, la partie Ouest est en grande partie délaissée par les traces expérimentées ainsi que par les traces générées. Même si ces trois parties sont discernables sur les 2 cartes de chaleur, il existe des différences notables entre celles-ci. Les plus marquantes se situent au centre. En effet, les traces générées forment un amas en plein centre de la ville, empêchant la distinction de plusieurs segments tant tous sont très utilisés. Les traces expérimentées quant à elles utilisent quelques segments discernables, notamment plusieurs successifs le long du Rhône traversant la partie centrale du Nord au Sud. À mesure que l'on s'éloigne de la partie centrale pour rentrer dans la partie Est, les segments affichés en rouge deviennent plus rares. Certains deviennent vert-jaune et d'autres deviennent bleus. Cependant, la carte des traces expérimentées contient plus de segments vert-jaune et ceux-ci vont plus loin, atteignant les bordures Sud et Est dans certains cas. Enfin, peu de différences sont notables entre les 2 cartes de chaleurs dans la partie Ouest.

Les cartes de chaleur du jeu de données MonRésoVélo sont montrées en figure 3.4. Les 2 cartes sont séparables en trois parties : l'hypercentre, le centre et le reste de la ville. Dans l'hypercentre, on observe une forte utilisation d'un nombre important de segments de route. À mesure que l'on s'éloigne de celui-ci, l'utilisation s'amenuise de manière assez hétérogène : certains segments restent très utilisés là où d'autres le sont beaucoup moins. Enfin, les segments de route éloignés du centre de la ville sont dans la plupart des cas utilisés de manière homogène. Il existe tout de même des différences entre traces expérimentées et traces générées dans chacune de ces trois parties. Dans l'hypercentre, les traces générées sont disséminées de manière à pouvoir discerner quelques routes particulièrement utilisées. Les traces expérimentées, elles, montrent une forte utilisation d'une grande partie des routes, créant ainsi un amas rendant la distinction de segments de route plus utilisés que les autres difficile. Lorsque l'on s'éloigne de l'hypercentre, les segments de route affichés en rouge sont moins nombreux pour les traces expérimentées et disparaissent quasi totalement pour les traces générées. La partie de la ville la plus éloignée du centre présente

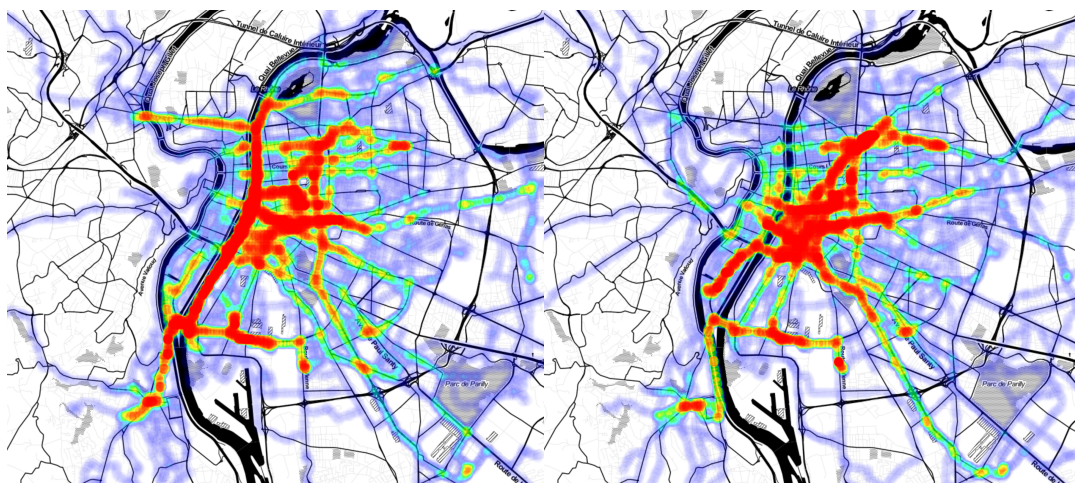


FIGURE 3.3 – Cartes de chaleur des traces expérimentées (à gauche) et des traces générées (à droite) du jeu de données Véléval (Lyon).

le plus de différences entre les 2 types de traces. Les traces générées utilisent cette partie de manière quasi homogène, avec une majorité de segments de route affichés en bleu. À l'inverse, on peut observer au moins 4 routes affichées en rouge qui atteignent quasiment la bordure de la carte de chaleur des traces expérimentées, signe que ces voies sont utilisées par une majorité des cyclistes. À ces voies rouges s'ajoute une multitude de segments de route affichés en jaune ou en vert, montrant une certaine hétérogénéité d'utilisation parmi les segments qui ne sont pas très fortement utilisés.

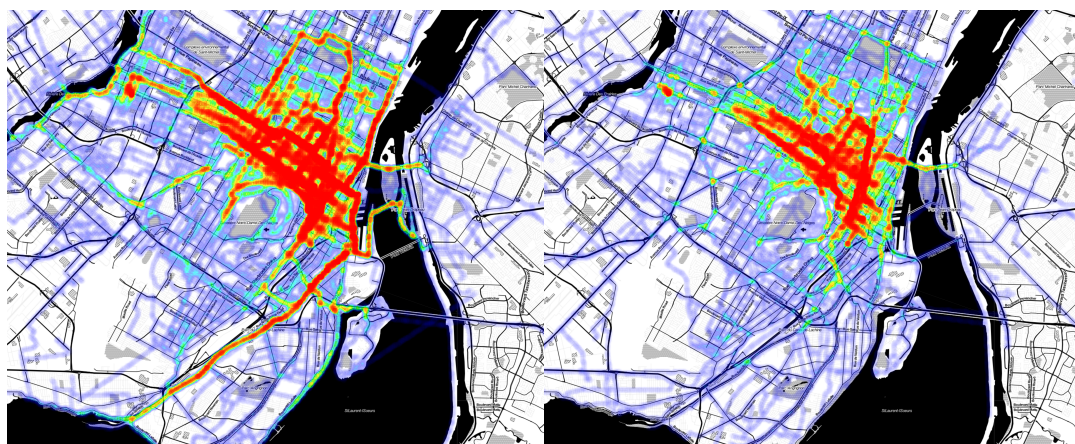


FIGURE 3.4 – Cartes de chaleur des traces expérimentées (à gauche) et des traces générées (à droite) du jeu de données MonRésoVélo (Montreal).

Les observations faites sur les Figures 3.3 et 3.4 montrent des points communs mais aussi des différences dans le comportement des cyclistes ayant généré les 2 jeux de données. Tout d'abord, on observe une forte concentration de traces dans le centre des 2 villes. Que ce soit sur la carte de chaleur des traces expérimentées ou celles des traces générées, une grande partie des voies affichées en rouge s'y trouve. Il est plus facile de discerner certaines routes dans la partie centrale de Lyon sur la carte des traces expérimentées que sur celle des traces générées. Cela indique que

les cyclistes ont tendance à utiliser certaines voies plus que d'autres, quitte à ne pas utiliser le chemin le plus court jusqu'à leurs destinations. Le constat semble être opposé sur les cartes de chaleurs de MonRésoVélo. Cela est dû à la différence de taille entre Montréal et Lyon. Lyon étant environ 10 fois plus petite que Montréal, il est nécessaire de dézoomer les cartes de chaleurs de MonRésoVélo pour englober une majorité des traces du jeu de données. Si l'on zoome sur l'hypercentre de Montréal, on observe que les traces expérimentées se concentrent principalement sur trois axes, là où les traces générées utilisent l'espace de manière plus homogène. Lorsque l'on s'éloigne du centre de la ville en direction de l'Ouest, une zone très peu utilisée par les cyclistes Véléval est discernable. Cela s'explique par la topologie de la ville de Lyon. En effet, celle-ci est constituée de 2 collines, toutes 2 situées dans sa partie Ouest. Ces collines sont généralement évitées par les cyclistes pour des raisons évidentes de dénivelé. La ville de Montréal est aussi constituée d'une colline ainsi que d'un mont. Au Sud de l'hypercentre, ils sont tous 2 évités par les cyclistes ainsi que par l'algorithme de chemin le plus court, bien que ce dernier fasse quelques exceptions. Le comportement des cyclistes face aux dénivelés semblent là encore identiques au sein des 2 jeux de données, malgré des différences évidentes visibles entre les 2 figures. Enfin, lorsque l'on s'éloigne du centre sur une partie relativement plane, les observations sont identiques quel que soit le jeu de données : les segments de route affichés en rouge sont moins nombreux au profit de segments vert-jaune et bleus. Les segments rouges et vert-jaune sont tout de même plus longs et plus nombreux à l'extérieur du centre sur les cartes de chaleurs des traces expérimentées. Cela montre que les cyclistes utilisent certains segments de route spécifiques quitte à s'éloigner du chemin le plus court jusqu'à leurs destinations dans ces zones aussi.

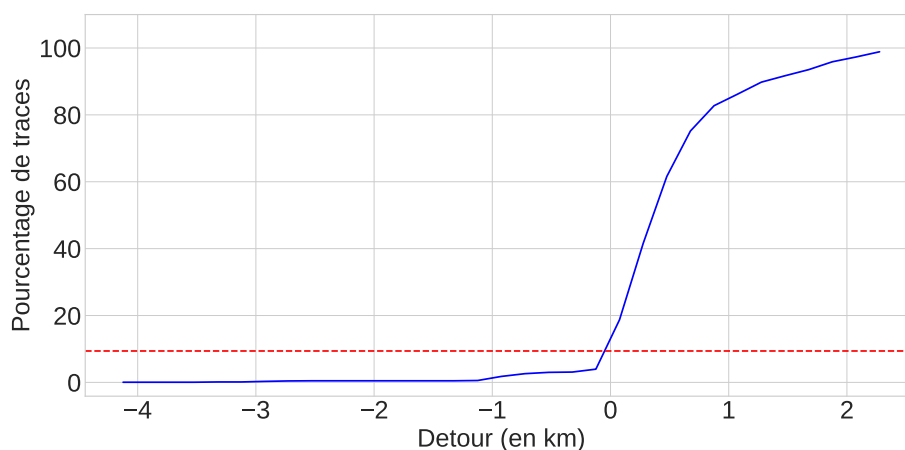
Toutes ces observations confirment la première hypothèse. Les cyclistes sont prêts à emprunter des segments de route préférés, même si cela signifie faire un détour par rapport au chemin le plus court jusqu'à leurs destinations. Cependant, ils semblent vouloir minimiser la distance parcourue pour accéder à ces routes préférées. C'est pourquoi de moins en moins de segments routes spécifiques sont visibles à mesure que l'on s'éloigne du centre-ville, au profit d'une utilisation de l'espace de plus en plus homogène. Les itinéraires des cyclistes ayant généré les 2 jeux de données seraient donc un compromis entre l'utilisation de segments de route préférés, et de chemins le plus court pour y accéder.

## CDF

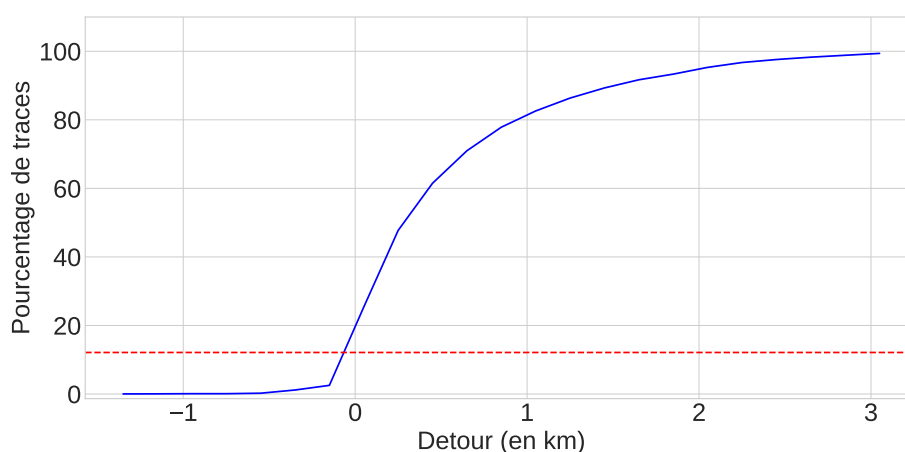
L'hypothèse 1 semble être valide pour les 2 jeux de données étudiés. Une analyse des CDFs des différences de distances entre les traces expérimentées et les traces générées (Figure 3.2) peut être effectuée pour confronter la deuxième. Les 2 CDFs sont très similaires, avec une grande majorité des détours compris entre 0 et 3 km. Une augmentation assez importante de la taille des détours est visible lorsque l'on regarde les 10 derniers pourcents de traces. En effet, 90 % des détours faits par les cyclistes sont inférieurs à 2,4 km pour Véléval et inférieurs à 3,2 km pour MonRésoVélo, là où le détour maximal atteint presque 100 km pour les 2 jeux de données. Cela est cohérent avec la deuxième hypothèse. La plupart des cyclistes ne souhaitent pas faire de détours au-delà d'une certaine distance, car cela signifierait pour eux que le coût de celui-ci devient plus important que les bénéfices qu'il apporte.

### Suppression des traces faisant de trop gros détours

Même si une majorité des trajets des 2 jeux de données font moins de 3 km de détour, certains en effectuent des beaucoup plus grands. Il est probable que les cyclistes acceptant d'effectuer des détours de quasiment 100 km n'utilisent pas le vélo comme un moyen de transport, mais comme une activité sportive ou pour le loisir. Le comportement des cyclistes ayant pour but d'arriver à une destination et ceux ayant pour but de faire du sport sont sans aucun doute très différents, et peuvent difficilement être étudiés en parallèle. Les hypothèses formulées dans ce chapitre ne sont cohérentes que dans le cas de cyclistes cherchant à atteindre une destination. C'est pourquoi les 10% de traces faisant les détours les plus longs sont supprimés des 2 jeux de données. La Figure 3.5 montre les CDFs des différences de distances entre les traces expérimentées et les traces générées des 2 jeux de données après cette suppression. Le nombre de traces du jeu de données Véléal passe à 2300 et celui du jeu de données MonRésoVélo à 4365.



(A) Jeu de données Véléal.



(B) Jeu de données MonRésoVélo.

FIGURE 3.5 – CDF des différences de distances entre les traces expérimentées et les traces générées des jeux de données après la suppression des traces faisant de trop gros détours.

### 3.2.4 Résumé de la section

La Section 3.2 présente les jeux de données utilisés en détail. Un chemin le plus court (appelées traces générées dans la suite) est calculé entre l'origine et la destination de chaque trace GPS (appelées traces expérimentées dans la suite). Deux hypothèses sont alors formulées :

- les cyclistes n'empruntent pas le chemin le plus court au profit de segments de route qu'ils préfèrent
- les cyclistes ont une distance de détour maximum qu'ils acceptent d'effectuer pour utiliser ces segments

Ces hypothèses ne sont valables que dans le cadre de l'utilisation du vélo pour atteindre une destination. L'affichage de cartes de chaleur affichant les différents types de traces permet de confirmer la première hypothèse. Une CDF des différences de distance entre les traces générées et les traces expérimentées permet de confirmer la seconde.

## 3.3 Identification des segments de route préférés

Les cyclistes des 2 jeux de données étudiés ont tendance à faire des détours pour emprunter des segments de route qu'ils préfèrent, même si ceux-ci ont une limite à partir de laquelle ils considèrent que le détour est trop long. Afficher toutes les traces expérimentées sous la forme de carte de chaleurs permet d'identifier visuellement les segments de route utilisés par une majorité de cyclistes. Des segments de route préférés plus locaux, c'est-à-dire utilisés uniquement par des cyclistes passant par un même secteur, doivent cependant exister. Ces segments ne seraient que peu voire pas identifiables lors de l'affichage de l'entièreté des jeux de données. La décomposition des jeux de données en groupes de traces expérimentées partageant des segments de route permettrait de confirmer l'existence de ces segments de route préférés, et de les identifier par la même occasion. L'utilisation d'un algorithme de *clustering* paraît dès lors une approche intéressante pour ajouter de l'information aux jeux de données. Les algorithmes de *clustering* sont des algorithmes d'apprentissage non supervisé permettant de former des groupes de données similaires appelés *clusters*. Dans le cas des traces expérimentées, chaque *cluster* créé permettra l'identification d'un ou plusieurs segments de route préférés par les cyclistes. Les *clusters* les plus grands contiendront les segments de route préférés globaux, et les plus petits les segments de route préférés locaux. Dans cette section, 2 métriques de distance ainsi que 2 algorithmes de *clustering* sont testés pour construire des *clusters* pertinents.

### 3.3.1 Métriques de distance

Pour pouvoir regrouper des traces partageant des segments de route, il est nécessaire de quantifier la distance entre celles-ci. Il existe plusieurs méthodes pour y parvenir, la plus connue étant l'utilisation d'un algorithme de *mapmatching* (voir Section 2.3). La différence symétrique des ensembles d'arêtes de 2 traces permet de quantifier la distance entre celles-ci. Malgré la robustesse de ces algorithmes, 2 méthodes plus simples sont utilisées ici. Ce choix est motivé par la difficulté d'implémentation d'un algorithme de *mapmatching*, ainsi que par les besoins de ce travail qui se limitent à une quantification de segments en commun, sans avoir besoin de plus de détails. La première métrique utilise une méthode développée spécialement pour ces travaux. La deuxième utilise la distance géodésique, une distance couramment utilisée dans le cadre de points géographiques.

### Distance personnalisée

Cette métrique de distance a pour but de quantifier les segments de route communs entre 2 traces. Elle utilise pour se faire une réduction de l'espace. La carte est divisée en cellules. Pour limiter le coût en temps et en ressources, seules les cellules par lesquelles passe au moins une trace sont calculées. Chaque trace se voit attribuée un ensemble de cellules appelé ensemble direct. Cet ensemble de cellules contient toutes les cellules par lesquelles la trace passe directement. Les traces expérimentées de Véléval n'étant pas *mapmatchées*, il arrive que 2 traces sur un même segment de route passent par 2 cellules différentes. Pour corriger ce problème, un deuxième ensemble est attribué à chaque trace. Cet ensemble, appelé ensemble étendu et noté  $\tau$  par la suite, contient les cellules que la trace traverse directement ainsi que toutes les cellules existantes voisines de celles-ci. Pour rappel, une cellule est considérée comme existante uniquement si au moins une trace la traverse directement. Les ensembles étendus dépendent de l'ensemble de traces utilisé, et doivent donc être calculés après la génération des ensembles directs. La figure 3.6 montre un exemple graphique de l'intersection entre les ensembles étendus de 2 traces.

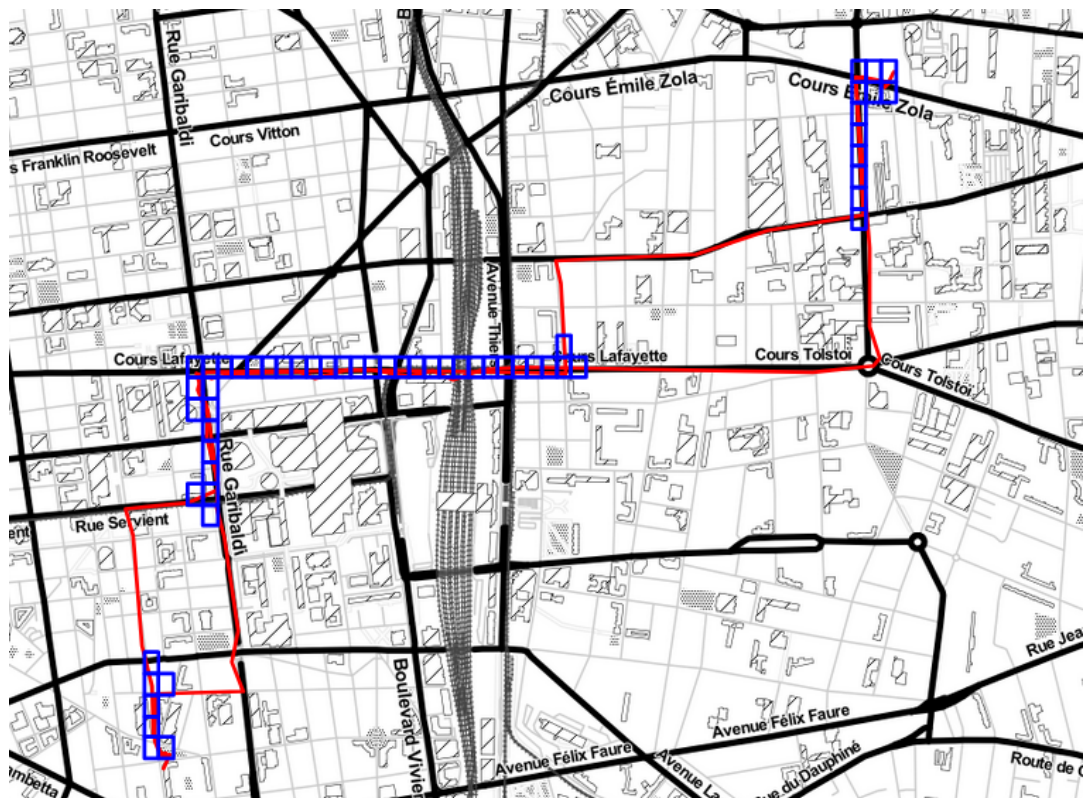


FIGURE 3.6 – Exemple graphique de l'intersection entre les ensembles étendus de 2 traces.

Soit  $t_1$  et  $t_2$  2 traces,  $\tau_1$  l'ensemble étendu de  $t_1$  et  $\tau_2$  l'ensemble étendu de  $t_2$ , la distance  $D_{t_1 t_2}$  entre  $t_1$  et  $t_2$  est la distance de Jaccard [99] entre  $\tau_1$  et  $\tau_2$ , définie comme suit :

$$D_{t_1 t_2} = 1 - \frac{|\tau_1 \cap \tau_2|}{|\tau_1 \cup \tau_2|}$$

## Distance géodésique

La distance géodésique entre 2 points géographiques est la plus courte distance à la surface d'un modèle ellipsoïdal de la terre entre ses 2 points. Cette distance est couramment utilisée dans le cadre de points géographiques. Dans ce chapitre, la distance géodésique est calculée en utilisant la méthode développée par Karney [100] avec le modèle ellipsoïdal WGS 84. La distance géodésique entre 2 traces est la moyenne des distances géodésiques entre les points de celles-ci. Les traces doivent donc être pré-traitées au préalable pour avoir le même nombre de points géographiques. Cette distance couramment utilisée dans le cadre de points géographiques a l'avantage de nécessiter moins de calculs que la distance personnalisée.

### 3.3.2 Algorithmes de *clustering*

Soit un ensemble de  $n$  observations  $(x_1, x_2, \dots, x_n)$  et une métrique permettant de mesurer la distance entre elles. L'objectif d'un algorithme de *clustering* est de former différents *clusters* d'observations proches les unes des autres. Il existe une multitude d'algorithmes de *clustering* possédant tous leurs avantages et leurs inconvénients. La pertinence de l'utilisation d'un algorithme de *clustering* ainsi que son paramétrage dépend du type de données (avec ou sans bruit, taille du jeu...) et des propriétés des *clusters* recherchés (nombre, taille, forme...) [101]. Deux des algorithmes les plus connus sont comparés dans ce chapitre : DBSCAN [102] et k-médoïdes [103] (basé sur k-means [104]). Le fonctionnement de chacun d'eux est d'abord expliqué.

#### DBSCAN

DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) est un algorithme de *clustering* basé sur la densité. Ce type d'algorithme est particulièrement adapté aux données contenant du bruit. DBSCAN permet de former des *clusters* de formes arbitraires et d'évincer les observations trop éloignées des autres (aussi appelé le bruit). DBSCAN possède 2 paramètres :  $minPts$  et  $\epsilon$ .  $minPts$  définit le nombre minimum de points dans une région dense pour qu'elle puisse appartenir à un *cluster*.  $\epsilon$  définit la distance maximale que 2 points peuvent avoir pour appartenir à la même région dense. DBSCAN parcourt les observations et construit le  $\epsilon$ -voisinage de celles-ci. Si ce voisinage contient un nombre de points supérieur ou égal à  $minPts$ , un nouveau *cluster* contenant l'observation et ses voisins est créé. DBSCAN tente ensuite d'agrandir le *cluster* en rajoutant les voisins des voisins ayant un  $\epsilon$ -voisinage suffisant et ainsi de suite, jusqu'à ce que plus aucun  $\epsilon$ -voisinage ne soit assez grand. L'algorithme 1 montre le fonctionnement détaillé de DBSCAN.

- **Avantage** : peu de temps de calcul (complexité d'environ  $O(n \log(n))$ ) avec des paramètres corrects)
- **Inconvénient** : assez difficile à paramétrer (2 paramètres pouvant être assez abstraits)

#### k-médoïdes

L'algorithme de *clustering* le plus connu et le plus utilisé est sûrement k-means. k-means prend en paramètre un nombre de *clusters*  $k$  et en entrée les observations

**Algorithme 1** : Algorithme DBSCAN

---

```

Data :  $\varepsilon$  une distance, minPts une taille de cluster minimum et  $X$  un
    ensemble de  $n$  observations
Result :  $S$  un ensemble de clusters
 $S \leftarrow \{\}$ ;
/* l'ensemble des clusters est initialisé vide */
 $B \leftarrow \{\}$ ;
/* l'ensemble du bruit est initialisé vide */
 $T \leftarrow \{\}$ ;
/* l'ensemble des observations traitées est initialisé vide */
for  $x \in X$  do
    /* pour toutes les observations */
    if  $x \notin T$  then
         $V_x \leftarrow \text{trouverVoisins}(x, \varepsilon)$ ;
        if  $|V_x| < \text{minPts}$  then
            /* si le nombre de voisin est insuffisant */
             $B \leftarrow B \cup \{x\}$ ;
             $T \leftarrow T \cup \{x\}$ ;
            /*  $x$  est considéré comme étant du bruit */
        else
             $c \leftarrow \{x\}$ ;
            /* un nouveau cluster  $c$  est initialisé avec  $x$  */
             $T \leftarrow T \cup \{x\}$ ;
            /* ...et à  $T$  l'ensemble des observations traitées */
            for  $v \in V_x$  do
                /* pour tous les voisins de  $x$  */
                if  $v \in B$  then
                    /* si  $v$  est considéré comme du bruit */
                     $B \leftarrow B \setminus \{v\}$ ;
                    /*  $v$  n'est plus considéré comme du bruit... */
                     $c \leftarrow c \cup \{v\}$ ;
                    /* ... et est intégré à  $c$  */
                else if  $v \notin T$  then
                    /* si  $v$  n'a pas encore été traité */
                     $c \leftarrow c \cup \{v\}$ ;
                    /* on ajoute  $v$  au cluster  $c$ ... */
                     $T \leftarrow T \cup \{v\}$ ;
                    /* ...et à  $T$  l'ensemble des observations traitées */
                 $V_v \leftarrow \text{trouverVoisins}(v, \varepsilon)$ ;
                if  $|V_v| \geq \text{minPts}$  then
                    /* si le nombre de voisin de  $v$  est suffisant... */
                     $V_x \leftarrow V_x \cup V_v$ ;
                    /* ... ils sont ajoutés aux candidats potentiels de  $c$  */
                end
            end
             $S \leftarrow S \cup c$ ;
            /* le cluster  $c$  est ajouté à l'ensemble des clusters */
        end
    end
end

```

end



sous la forme de vecteurs de même dimension. Son but est de partitionner les  $n$  observations en  $k$  clusters  $S = \{S_1, S_2, S_3, \dots, S_k\}$  en minimisant la variance inter-cluster. Formellement, l'objectif est de trouver :

$$\operatorname{argmin}_s \sum_{i=1}^k \sum_{x_j \in S_i} \|x_j - \mu_i\|^2$$

où  $\mu_i$  est le barycentre de  $S_i$  et  $\|\cdot\|^2$  la norme euclidienne. La notion de barycentre n'est pas définie pour la distance personnalisée. k-means est donc difficilement utilisable avec celle-ci. k-médoïdes est un algorithme similaire à k-means. Il prend lui aussi un nombre de clusters  $k$  en paramètre et minimise la distance entre les points d'un même cluster et un point désigné comme central. Cependant, ce point central n'est plus le barycentre, mais une des observations appelée médoïde. La distance personnalisée est donc utilisable avec cet algorithme. Plusieurs variantes de k-médoïdes existent. L'algorithme 2 décrit le fonctionnement de PAM (*Partitioning Around Medoids*), la variante utilisée dans ce chapitre. La fonction *intialiseMedoids* prend en paramètre un nombre  $k$  et l'ensemble des observations  $X$  et renvoie un ensemble de  $k$  observations médoïdes. La fonction *creerClusters* prend en paramètres l'ensemble des observations  $X$  et un ensemble de  $k$  médoïdes  $M$ . Chaque observation est mise dans le cluster du médoïde le plus proche. L'ensemble des clusters finaux ainsi que la somme des distances entre chaque observation et le médoïde de son cluster sont renvoyés.

- **Avantage** : facile à paramétrer (un seul paramètre, le nombre de clusters désiré)
- **Inconvénient** : temps de calcul assez long (complexité  $O(k(n-k)^2)$  quand non optimisé)

### 3.3.3 Paramétrage

Pour paramétrer correctement un algorithme de *clustering*, il est nécessaire de définir les caractéristiques des clusters recherchées. Le choix de ces caractéristiques dépend du cadre dans lequel ceux-ci vont être utilisés. Quatre caractéristiques sont choisies pour mesurer la qualité des clusters créés. Elles sont expliquées ainsi que les raisons de leur choix dans le cadre de l'identification de segments de route préférés. Ces caractéristiques, ainsi qu'une analyse visuelle des clusters, sont ensuite utilisées pour trouver les meilleurs paramètres des algorithmes testés. DBSCAN et k-médoïdes fonctionnant assez différemment, certaines caractéristiques ne sont pertinentes que pour un seul d'entre eux.

- **Nombre de grands clusters** : Un grand cluster est défini comme un cluster contenant au moins 2 % des observations du jeu de données. La présence de grands clusters permet de s'assurer que les segments de route préférés globaux sont bien identifiés. Cependant, ceux-ci demandent vérification pour s'assurer qu'ils ne contiennent pas de traces trop éloignées les unes des autres.
- **Coefficient de silhouette** : La silhouette est un coefficient couramment utilisé pour valider la pertinence de clusters. Le coefficient de silhouette d'une observation est la différence entre la distance moyenne avec les observations appartenant au même cluster (cohésion) et la distance moyenne avec les observations appartenant aux clusters voisins (séparation). La silhouette varie de -1 à 1. Une valeur élevée indique que l'observation est bien intégrée à

**Algorithme 2 : Algorithme PAM**


---

```

Data :  $k$  un nombre de clusters ,  $X$  un ensemble de  $n$  observations
Result : Un ensemble  $S$  de  $k$  clusters
 $M \leftarrow \text{intilialiseMedoides}(k, X);$ 
/*  $M$  est l'ensemble des médoïdes */
 $S, c_{best} \leftarrow \text{creerClusters}(M, X \setminus M);$ 
do
   $c \leftarrow c_{best};$ 
  for  $m \in M$  do
    /* pour tout les médoïdes  $m$  */
    for  $o \in X \setminus M$  do
      /* pour toutes les observations non médoïdes  $o$  */
       $M_{new} \leftarrow M \setminus \{m\};$ 
       $M_{new} \leftarrow M_{new} / \{o\};$ 
      /*  $M_{new}$  est l'ensemble des médoïdes  $M$  avec  $o$  à la place
         de  $m$  */
       $S_{new}, c_{new} \leftarrow \text{creerClusters}(M_{new}, X \setminus M_{new});$ 
      if  $c_{new} < c_{best}$  then
        /* si la nouvelle configuration est mieux que la
           meilleure vue jusqu'à maintenant, on la retient */
         $c_{best} \leftarrow c_{new};$ 
         $o_{best} \leftarrow o;$ 
         $m_{best} \leftarrow m;$ 
         $S_{best} \leftarrow S_{new};$ 
      end
    end
  end
  if  $c_{best} < c$  then
    /* si une configuration testée est meilleure que la
       configuration actuelle, la configuration actuelle
       devient la configuration testée */
     $M \leftarrow M \setminus \{m_{best}\};$ 
     $M \leftarrow M / \{o_{best}\};$ 
     $S \leftarrow S_{best};$ 
  end
while  $c_{best} < c;$ 
/* tant qu'une meilleure configuration a été trouvée */

```

---

son *cluster*, et suffisamment éloignée des *clusters* voisins. La moyenne de la silhouette de toutes les observations non considérées comme du bruit est calculée. Un bon score de silhouette signifie que les segments de route préférés sont bien discernables dans les *clusters*, et que chaque segment de route n'est identifié que dans un seul d'entre eux.

- **Bruit (uniquement pour DBSCAN)** : Comme dit plus haut, DBSCAN considère les observations trop éloignées des autres comme du bruit. Un DBSCAN mal paramétré peut soit considérer la majorité des données comme du bruit, soit ne pas détecter les observations trop éloignées des autres. Si une majorité des traces sont considérées comme du bruit, les segments de route identifiés ne seraient représentatifs que d'une petite partie des cyclistes ayant généré les jeux de données. D'autre part, si aucune trace n'est considérée comme du bruit, les *clusters* risquent de contenir des traces ne partageant pas ou peu de segments de route. La qualité des segments de route préférés identifiés serait alors moindre.
- **Nombre de *clusters* (uniquement pour DBSCAN)** : Un trop grand nombre de *clusters* peut signifier qu'ils ne contiennent en moyenne pas suffisamment de traces pour identifier les segments de route préférés globaux, ou que les mêmes segments de route préférés sont identifiées dans plusieurs *clusters*. Peu de *clusters* peut en revanche signifier que des segments de route locaux existent et ne sont pas identifiés. Afficher le nombre de *clusters* créé n'est pas pertinent pour k-médoïdes, car c'est  $k$  qui le définit.

## DBSCAN

Les Figures 3.7 et 3.8 montrent l'évolution de chaque variable pour les *clusters* produits par DBSCAN selon  $\varepsilon$ . Cet affichage est effectué dans un premier temps pour plusieurs *minPts* dans le but de le fixer.  $minPts = 3$  s'est avéré être le meilleur choix pour Véléal quelle que soit la métrique de distance utilisée (Figures 3.7a et 3.7b), ainsi que pour MonRésoVélo avec la distance personnalisée (Figure 3.8a). Pour MonRésoVélo et la distance géodésique,  $minPts = 2$  est utilisé (Figure 3.8b).  $\varepsilon$  varie de 0.1 à 0.9 pour la distance personnalisée et de 0.1 à 2 pour la distance géodésique. La distance personnalisée ne pouvant dépasser 1, un  $\varepsilon$  supérieur n'aurait pas de sens. Pour la distance géodésique, les résultats avec  $\varepsilon > 2$  sont peu intéressants avec très peu de *clusters* contenant quasiment toutes les traces. Le bruit diminue à mesure que  $\varepsilon$  augmente quel que soit le jeu de données ou la métrique utilisée. Une faible valeur de  $\varepsilon$  signifie que seuls les points très proches les uns des autres sont regroupés. À mesure que  $\varepsilon$  augmente, des points plus éloignés peuvent être placés dans le même *cluster*. DBSCAN trouve donc de moins en moins de points ne pouvant appartenir à aucun *cluster*.

Le comportement de DBSCAN avec le jeu de données Véléal diverge cependant selon la métrique de distance utilisée. Avec la distance personnalisée, le nombre de *clusters* ainsi que le score de silhouette commencent par augmenter jusqu'à atteindre une valeur maximum après laquelle ils diminuent. Lorsque le  $\varepsilon$ -voisinage des observations augmentent, plus de *clusters* se créent et ceux-ci s'étendent plus. À partir de  $\varepsilon = 0.4$ , le  $\varepsilon$ -voisinage des observations permet à certains *clusters* de trop s'étendre. Ces *clusters* absorbent des observations qu'il aurait mieux valu affilier à des *clusters* distincts, faisant ainsi chuter le score de silhouette et le nombre de *clusters*. À partir de  $\varepsilon = 0.5$ , le nombre de grands *clusters* chute aussi, car certains *clusters* s'étendent

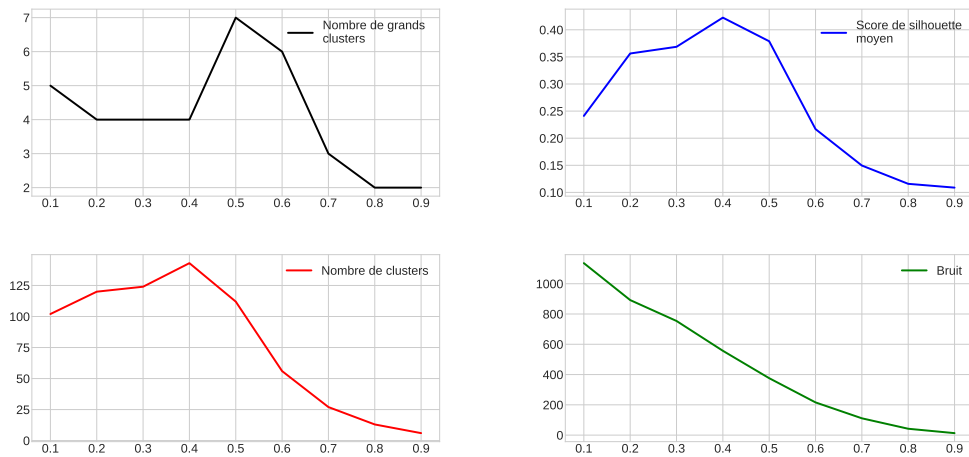
tellement qu'il est plus difficile de créer des grands *clusters* avec les observations restantes.

Avec la distance géodésique, il y a 2 grands *clusters* lorsque  $\varepsilon$  vaut 0.1 contre cinq avec la distance personnalisée. Le nombre maximum de grands *clusters* formés est de sept avec la distance personnalisée contre cinq avec la distance géodésique. Les traces semblent donc plus éloignées les unes des autres lorsque la distance géodésique est utilisée. Cela est confirmé par le nombre de *clusters*, constamment plus important avec la distance géodésique lorsque  $\varepsilon$  est dans l'intervalle  $[0.1, 0.9]$ . Le score de silhouette commence par augmenter jusqu'à  $\varepsilon = 0.3$  avant de diminuer jusqu'à son minimum à  $\varepsilon = 0.6$  puis de réaugmenter jusqu'à son maximum à  $\varepsilon = 2$ . Là encore, certains *clusters* s'étendent de plus en plus, améliorant les performances dans un premier temps puis les dégradant ensuite. Mais, avec la distance géodésique, les performances augmentent lorsque certains de ces grands *clusters* deviennent assez volumineux pour empêcher la formation d'autres grands *clusters*. La meilleure configuration trouvée par DBSCAN est avec  $\varepsilon = 2$ . Cette configuration crée trois *clusters*, dont un possédant 76 % des traces et un autre en possédant 23 %. Les traces semblent tellement uniformément distantes les unes des autres que DBSCAN ne peut diviser ces grands *clusters* sans diminuer le score de silhouette moyen. DBSCAN est donc peu adapté à la distance géodésique pour le jeu de données Véléal.

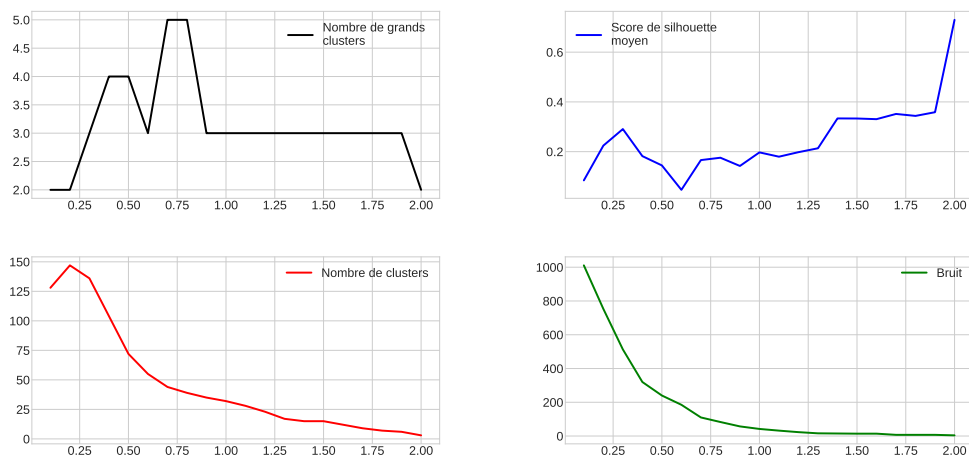
On observe les mêmes phénomènes lorsque DBSCAN est utilisé avec le jeu MonRésoVélo (Figure 3.8). Les différences avec Véléal ne se situent pas sur les variations des caractéristiques selon le paramétrage, mais sur les valeurs qu'elles atteignent. Par exemple, il y a systématiquement moins de grands *clusters* créés avec MonRésoVélo. MonRésoVélo contient des traces crowdsourcées et situées sur une zone plus large que celle de Véléal. Les différences entre les traces de MonRésoVélo sont telles que DBSCAN a plus de mal à étendre les *clusters*. Pour les mêmes raisons, le score de silhouette moyen est systématiquement plus bas avec MonRésoVélo. On peut enfin noter que la meilleure configuration trouvée avec la distance géodésique avec MonRésoVélo est constitué de 13 *clusters* dont un contenant 97 % des traces. Il semble que DBSCAN ne soit pas non plus adapté à la distance géodésique lorsqu'il est utilisé avec MonRésoVélo.

### k-médoïdes

La Figure 3.9 montre l'évolution du nombre de grands *clusters* et du score de silhouette moyen des *clusters* produits par k-médoïdes selon  $k$  pour le jeu de données Véléal. Le nombre de *clusters*  $k$  varie de 10 à 300 avec un pas de 10.  $k$  ne dépasse pas 300 car le temps de calcul devient trop long. La Figure 3.9a montre ces résultats pour la métrique de distance personnalisée et la Figure 3.9b pour la métrique de distance géodésique. On remarque que l'évolution des 2 caractéristiques est quasiment la même quelle que soit la métrique de distance utilisée. Le nombre de grands *clusters* commence par augmenter rapidement jusqu'à ce que  $k$  atteigne 30. La majorité des *clusters* créés jusque-là sont des grands *clusters*. Le nombre de *clusters* total semble insuffisant, car k-médoïdes est obligé de mettre une grande partie du jeu de données dans chacun d'eux. Cela est confirmé par l'évolution du score de silhouette moyen qui augmente grandement entre  $k = 10$  et  $k = 30$  *clusters*. Le nombre de grands *clusters* chute ensuite entre  $k = 30$  et  $k = 130$ , passant de 24 à 7 pour la distance personnalisée et de 21 à 5 pour la distance géodésique. Le score de silhouette augmente lui constamment sur cet intervalle.  $k$  a atteint une valeur suffisante pour permettre à k-médoïdes de diviser les grands *clusters* en plus petits. Les traces trop

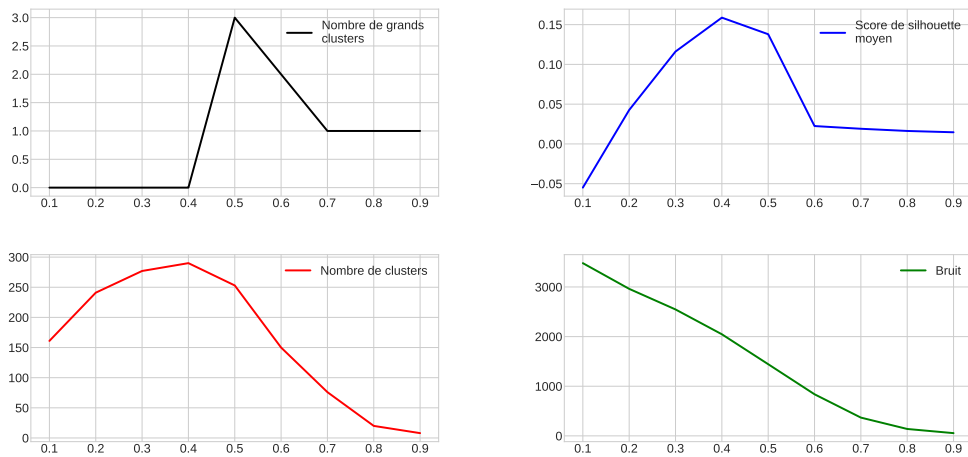


(A) Distance personnalisée ( $minPts = 3$ ).

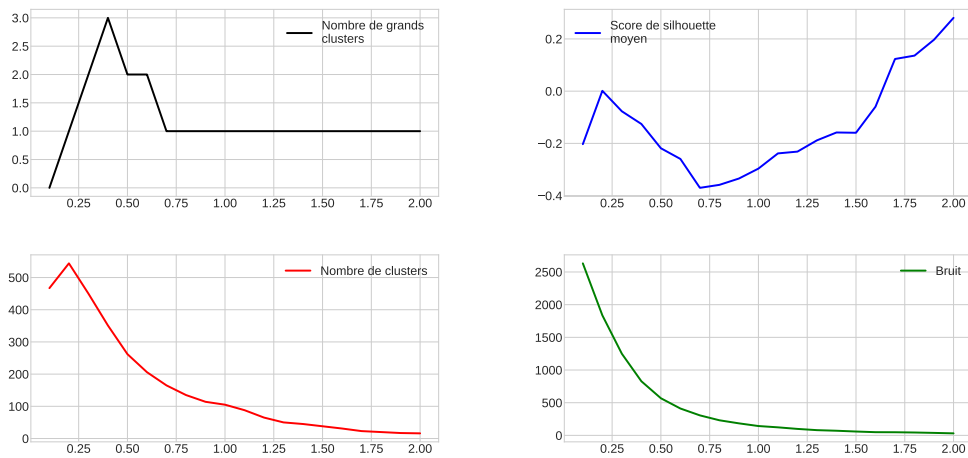


(B) Distance géodésique ( $minPts = 3$ ).

FIGURE 3.7 – Évolution du nombre de grands *clusters* (courbe noire), du score de silhouette moyen (courbe bleue), du nombre de *clusters* (courbe rouge) et du bruit (courbe verte) des *clusters* créés par DBS-CAN selon  $\epsilon$  pour le jeu de données Véléval.



(A) Distance personnalisée. ( $minPts = 3$ )



(B) Distance géodésique ( $minPts = 2$ ).

FIGURE 3.8 – Évolution du nombre de grands *clusters* (courbe noire), du score de silhouette moyen (courbe bleue), du nombre de *clusters* (courbe rouge) et du bruit (courbe verte) des *clusters* créés par DBS-CAN selon  $\epsilon$  pour le jeu de données MonRésoVélo.

éloignées ne sont désormais plus dans les mêmes grands *clusters*. Cela augmente naturellement le score de silhouette moyen. Cette tendance continue plus doucement jusqu'aux alentours de  $k = 200$ . Le nombre de grands *clusters* et le score de silhouette stagnent ensuite, respectivement à 5 et 0.48 pour la distance personnalisée et 3 et 0.54 pour la distance géodésique. Une légère augmentation du nombre de grands *clusters*, accompagnée d'une légère diminution du score de silhouette, sont visibles sur un intervalle aux alentours de  $k = 250$ . Ce phénomène est probablement dû à un minimum local trouvé par k-médoïdes.

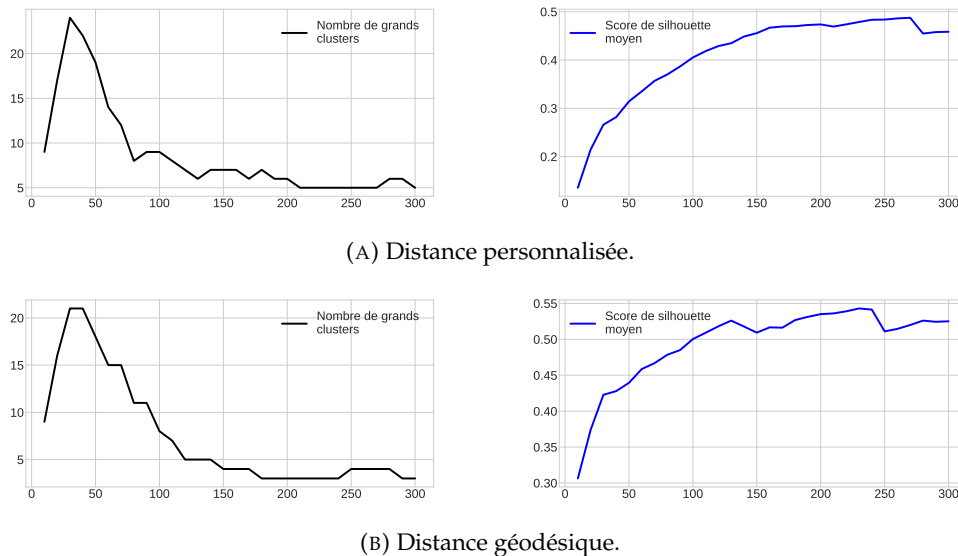


FIGURE 3.9 – Évolution du nombre de grands *clusters* (courbe noire) et du score de silhouette moyen (courbe bleue) des *clusters* créés par k-médoïdes selon  $k$  pour le jeu de données Véléal.

On observe une évolution similaire des 2 caractéristiques pour le jeu de données MonRésoVélo (Figure 3.10). On peut noter cependant quelques différences. Premièrement, le nombre de grands *clusters* augmente cette fois-ci jusqu'à  $k = 50$ . MonRésoVélo étant plus grand que Véléal, il est logique que k-médoïdes ait besoin d'un nombre de *clusters* plus important avant de pouvoir se permettre de commencer à diviser ceux-ci. La chute du nombre de grands *clusters* est aussi plus rapide, se finissant cette fois-ci à  $k = 100$ . La tendance à la baisse continue jusqu'à  $k = 200$ , où le nombre de grands *clusters* atteint 0. Les différences plus importantes entre les traces MonRésoVélo en sont là encore la cause. Celles-ci semblent être trop éloignées les unes des autres pour permettre la création d'un *cluster* contenant 2 % du jeu de données, quelle que soit la métrique de distance utilisée. Cela explique aussi le score de silhouette moyen qui augmente moins vite et n'atteint que 0.22 pour la distance personnalisée et 0.31 pour la distance géodésique.

### Algorithmes et paramètres retenus

Au vu des courbes détaillées précédemment, DBSCAN ne semble pas être un algorithme adapté lorsque utilisé avec la distance géodésique. Cependant, les performances de celui-ci sont proches de celles de k-médoïdes avec la distance personnalisée. En parallèle, les performances de k-médoïdes sont assez similaires quelle que soit la métrique de distance utilisée. Une analyse plus détaillée des *clusters* est

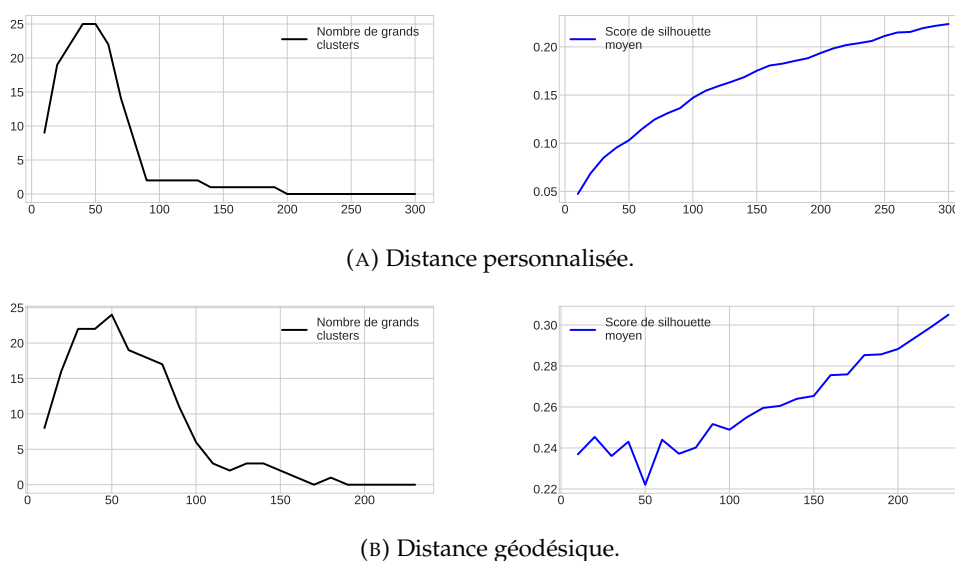


FIGURE 3.10 – Évolution du nombre de grands *clusters* (courbe noire) et du score de silhouette moyen (courbe bleue) des *clusters* créés par  $k$ -médoïdes selon  $k$  pour le jeu de données MonRéoVélo.

nécessaire pour trouver le paramétrage le plus adapté à la problématique de cette section pour chaque jeu. L'analyse effectuée précédemment permet tout de même de ne retenir que six paramétrages. Ceux-ci permettent un compromis entre le score de silhouette moyen et le nombre de grands *clusters*. Ils sont présentés dans le tableau 3.3.3.

Jeu de données	Algorithme	Distance	Paramétrage	<i>clusters</i>	Grands <i>clusters</i>	Silhouette
Véléal	DBSCAN	personnalisée	$\varepsilon = 0.4$	144	4	0.42
	k-médoïdes	personnalisée	$k = 150$	150	7	0.45
	k-médoïdes	géodésique	$k = 110$	110	7	0.5
MonRéoVélo	DBSCAN	personnalisée	$\varepsilon = 0.5$	254	3	0.13
	k-médoïdes	personnalisée	$k = 190$	190	1	0.18
	k-médoïdes	géodésique	$k = 180$	180	1	0.28

### 3.3.4 Analyse visuelle des plus grands *clusters*

Une analyse visuelle des *clusters* est effectuée pour chaque paramétrage. L'objectif des *clusters* est d'identifier des segments de route préférés. Afficher toutes les traces contenues dans un *cluster* permet de confirmer ou non si celles-ci passent effectivement par un ou plusieurs segments de route communs. Des cartes de chaleurs sont affichées dans un souci de visibilité. Celles-ci utilisent les mêmes cases que la métrique de distance personnalisée (voir Section 3.3.1). Toutes les cases traversées par au moins une trace d'un *cluster* sont générées. Un point est affiché pour chacune d'entre elles. La position du point est le centre de la case, et plus la case est traversée par des traces, plus le point est gros et rouge. Les Figures 3.3 et 3.4 utilisent le même principe. Pour rester concis, seules les cartes de chaleurs des plus grands *clusters* de chaque paramétrage sont affichées. Plus un *cluster* contient de traces, plus il est probable qu'au moins l'une d'entre elles ne partage de segments de route avec aucune

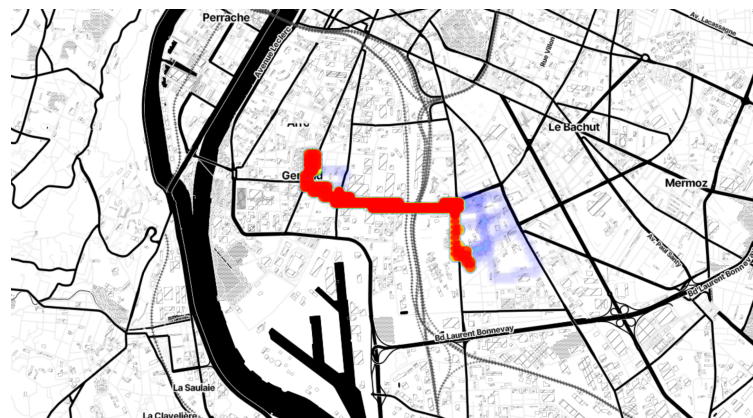


autre. Cela fait de la qualité du plus grand *cluster* créé un indicateur assez pertinent de la qualité générale d'un paramétrage.

Les cartes de chaleur des trois paramétrages retenus pour le jeu de données Vé-léal sont affichées en Figure 3.11. Le *cluster* contenant le plus de traces est celui créé par DBSCAN (Figure 3.11a) avec 70 traces. On discerne trois segments de route particulièrement rouges. Ces segments sont utilisés par quasiment toutes les traces du *cluster*. Une zone bleue assez importante est visible à l'est de ces segments. Cette zone peut être représentative de traces ne passant pas par les segments rouges, ou le début de traces passant par ceux-ci. Une petite zone bleue est aussi visible au Nord-Ouest. Le plus grand *cluster* créé par k-médoïdes avec la distance personnalisée contient les mêmes segments rouges (Figure 3.11b). Celui-ci ne contient cependant que 61 traces. Il est intéressant de constater que les mêmes segments de route préférés sont identifiables dans le plus grand *cluster* des 2 algorithmes. La petite zone bleue au Nord-Ouest est visible elle aussi. Le *cluster* créé par k-médoïdes ne contient néanmoins pas la zone bleue à l'est des segments rouges. Celle-ci est remplacée par un mince filet bleu, probablement causé par une ou 2 traces rejoignant ceux-ci. Le *cluster* créé par k-médoïdes avec la distance géodésique contient lui aussi 61 traces (Figure 3.11c). Deux segments de route sont visibles, différents de ceux des 2 premiers *clusters*. Ceux-ci sont reliés à 2 segments rouge-verts, utilisés par une majorité des traces, mais non par la quasi-totalité de celles-ci. Une grande zone bleue semblable à celle présente sur la Figure 3.11a s'étend dans toutes les directions autour de ces segments. La distance géodésique quantifie la proximité entre les points géographiques des traces. Logiquement, les traces proches géographiquement sont regroupées dans les mêmes *clusters*. Cependant, 2 traces proches géographiquement ne partagent pas systématiquement de segments de route. Des segments de route préférés sont donc logiquement moins identifiables sur la Figure 3.11c.

La Figure 3.12 montre les cartes de chaleurs des paramétrages retenus pour MonRésoVélo. Le plus grand *cluster* créé par DBSCAN contient 552 traces (Figure 3.12a), soit cinq fois plus que les *clusters* créés par k-médoïdes. Celui-ci contient de nombreux segments rouges, reliés à quelques segments rouge-verts. Une grande zone bleue s'étend dans toutes les directions tout autour de ces segments. Le *cluster* créé par k-médoïdes avec la distance personnalisée est assez surprenant (Figure 3.12b). Il contient certes un segment de route rouge bien défini en son centre, mais celui-ci n'est pas relié aux autres zones visibles. Beaucoup des traces affichées dans ce *cluster* créent des zones indépendantes les une des autres, bleues ou bleu-vertes dans la plupart des cas. Les traces de MonRésoVélo semblent tellement éloignées les unes des autres que l'algorithme n'arrive pas à créer un grand *cluster* contenant des traces partageant toutes ou presque un segment de route. k-médoïdes fait cependant mieux avec la distance géodésique (Figure 3.12b) et est capable de créer un *cluster* avec 93 traces dans une même zone géographique. Aucun segment de route n'est cependant visible dans ce *cluster*. Celui-ci ne contient qu'une zone rouge sur plusieurs segments indistincts. La distance géodésique a une fois de plus permis de rassembler des traces proches géographiquement, mais qui ne partagent pas forcément de segments de route.

Les scores de silhouette moyens que les 2 algorithmes arrivent à obtenir sont systématiquement plus importants avec la distance géodésique qu'avec la distance personnalisée. Cependant, une analyse visuelle montre que les *clusters* créés avec cette



(A) DBSCAN avec distance personnalisée (70 traces).

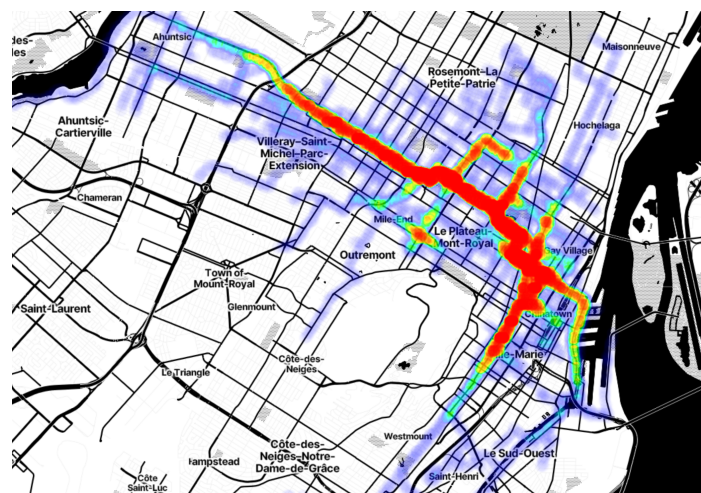


(B) k-médoides avec distance personnalisée (61 traces).

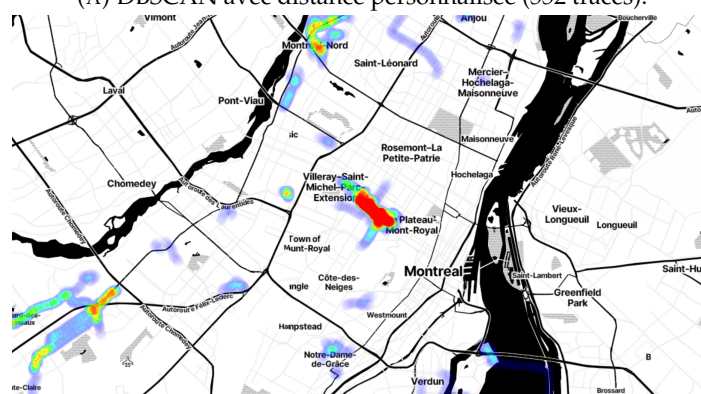


(C) k-médoides avec distance géodésique (61 traces).

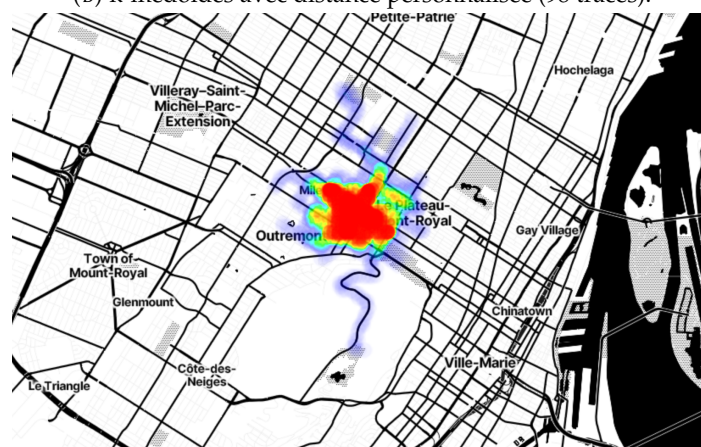
FIGURE 3.11 – Cartes de chaleurs des plus grands *clusters* pour les trois paramétrages retenus avec le jeu de données Véléal



(A) DBSCAN avec distance personnalisée (552 traces).



(B) k-médoides avec distance personnalisée (98 traces).



(C) k-médoides avec distance géodésique (93 traces).

FIGURE 3.12 – Cartes de chaleurs des plus grands *clusters* pour les trois paramétrages retenus avec le jeu de données MonRésoVélo.

métrique ne permettent pas l'identification de segments de route préférés. Pour ce qui est des différences de performances entre k-médoïdes et DBSCAN lorsque utilisés avec la distance personnalisée, elles dépendent du jeu de données utilisé. Avec le jeu de données MonRésoVélo, le plus grand *cluster* créé par k-médoïdes contient des traces ne partageant aucun segment de route. DBSCAN quant à lui génère un plus grand *cluster* contenant beaucoup de traces (12 % du jeu de données). Malgré sa taille importante, il permet l'identification de plusieurs segments de route préférés. Cependant, ces segments sont entourés d'une zone utilisée de manière homogène par très peu de traces. La combinaison entre DBSCAN et la distance personnalisée est tout de même acceptable, car plusieurs segments de route préférés sont identifiables. De plus, cette combinaison est meilleure que les 2 autres paramétrages testés. Pour Véléal, les performances de k-médoïdes sont bien plus acceptables. En effet, les segments de route visibles dans le plus grand *cluster* qu'il crée sont identiques et autant identifiables que ceux présents dans le plus grand *cluster* de DBSCAN. Cependant, le plus grand *cluster* formé par DBSCAN contient plus de traces passant par les segments de route préférés identifiables. C'est pour cette raison que DBSCAN est aussi choisi pour le jeu de données Véléal. DBSCAN identifie certaines traces comme étant du bruit, respectivement 1444 traces pour MonRésoVélo et 557 pour Véléal. Celles-ci sont supprimées des jeux de données pour la suite de ce travail. Véléal ne contient donc plus que 1743 traces et MonRésoVélo 2921 traces.

### 3.3.5 Résumé de la section

L'objectif de la Section 3.3 est d'identifier les segments de route préférés des cyclistes des 2 jeux de données. Une approche utilisant un algorithme de *clustering* est proposée. Diviser les jeux de données en groupes de traces partageant un ou plusieurs segments de route permet d'identifier les segments de route préférés, et ainsi d'ajouter de l'information à ceux-ci. Deux métriques de distance sont proposées pour quantifier les différences entre 2 traces. La première est une distance personnalisée utilisant une réduction de l'espace pour quantifier les segments de route communs à 2 traces. La deuxième, plus classique, est la distance géodésique. Celle-ci permet d'estimer le nombre de kilomètres entre 2 points géographiques. Deux algorithmes de *clustering* sont ensuite proposés. DBSCAN qui est un algorithme basé sur la densité. Celui-ci sépare les zones denses des jeux de données en *cluster*. k-médoïdes est lui un algorithme basé sur k-means. Les *clusters* formés par celui-ci possèdent chacun un médoïde. k-médoïdes minimise la somme des distances entre les médoïdes et les points appartenant à leurs *clusters*. Une analyse quantitative est d'abord effectuée pour trouver quelques paramétrages acceptables. Cette analyse est basée sur des caractéristiques de *clusters* intéressantes au vu de l'objectif fixé. Enfin, une analyse qualitative est faite pour trouver le meilleur paramétrage pour chaque jeu de données. Cette analyse qualitative s'effectue en affichant les *clusters* et en déterminant si les traces qu'ils contiennent partagent des segments de route. Les paramétrages retenus sont les suivants :

- Pour Véléal, DBSCAN est utilisé avec la distance personnalisée,  $minPts = 3$  et  $\epsilon = 0.4$ .
- Pour MonRésoVélo, DBSCAN est utilisé avec la distance personnalisée,  $minPts = 3$  et  $\epsilon = 0.5$ .

### 3.4 Générateur de traces basées sur l'expérience

Les modèles de choix d'itinéraire implicites finaux doivent être capables d'utiliser les segments de route préférés identifiés dans la section précédente. Il est montré que les cyclistes des 2 jeux de données font un compromis entre le chemin le plus court et des segments de route préférés en Section 3.2. C'est sur cette hypothèse que se base la méthode de génération de traces basées sur l'expérience détaillée dans cette section. Celle-ci utilise une paire d'O/D ainsi qu'un *cluster* de traces expérimentées pour créer un compromis entre le chemin le plus court et l'utilisation des segments de route préférés. L'algorithme de Dijkstra ainsi qu'une pondération des poids d'un graphe routier sont utilisés.

#### 3.4.1 Coefficient de cyclabilité

Les cellules créées pour la métrique de distance personnalisée (Section 3.3.1) sont réutilisées. Chaque trace possède 2 ensembles de cases, un ensemble direct et un ensemble étendu. De la même manière, chaque cellule possède 2 ensembles de traces. Soit une cellule  $c$ . Son premier ensemble  $\delta_c$  contient les traces passant directement par celle-ci, et son deuxième ensemble  $\nu_c$  contient les traces passant par ses voisines directes. Un coefficient de cyclabilité normalisé quantifiant l'utilisation de  $c$  par les cyclistes peut être calculé à partir de ces ensembles. Celui-ci est défini comme suit :

$$\mathcal{C}_c = \frac{|\delta_c \cup \nu_c|}{\max_m |\delta_m \cup \nu_m|}$$

Le coefficient de cyclabilité se situe donc dans l'intervalle  $]0, 1]$ , car les cellules par lesquelles ne passent aucune trace ne sont pas générées et que  $\mathcal{C}_m = 1$ . Il peut être calculé en considérant les cellules générées par la totalité d'un jeu de données, ou seulement celles d'un sous-groupe spécifique. Le coefficient de cyclabilité est utilisé pour modifier un graphe routier comme suit.

#### 3.4.2 Modification du graphe routier

Soit un *cluster*  $cl$ . L'Algorithme 3 modifie les poids des arêtes d'un graphe routier  $G$  à partir des traces contenues dans celui-ci. La fonction *creerCellules* prend en entrée une ou plusieurs traces et renvoie l'ensemble étendu des cellules traversées par ces traces. Il est possible de générer les cellules traversées par une arête de  $G$ . En effet, une arête représente une route délimitée par 2 intersections possédant une position géographique. Le poids d'une arête  $e$  est modifié en fonction du coefficient de cyclabilité moyen des cellules qu'elle traverse. Les coefficients de cyclabilité sont ici calculés en ne considérant que les cellules générées avec les traces de  $cl$ . Les arêtes de  $G$  représentant les routes les plus utilisées par les traces de  $cl$  auront les poids les plus faibles à la sortie de l'algorithme 3. L'algorithme de Dijkstra peut ensuite être utilisé sur le graphe modifié. Celui-ci aura plus de chances de sélectionner les routes les plus utilisées lors du calcul du chemin le plus court entre une origine et une destination. L'exploration de chaque arête de  $G$  est coûteux en temps, mais ne doit être faite qu'une seule fois par *cluster* si les modifications à faire sont enregistrées.

---

**Algorithme 3** : Modification d'un graphe routier à partir d'un *cluster* de traces.

---

```

Data :  $G$  un graphe routier et  $cl$  un cluster
Result :  $G$  avec des poids modifiés
 $\tau_{cl} \leftarrow \text{creerCellules}(cl)$ ;
/*  $\tau_{cl}$  contient l'ensemble des cellules par lesquelles les traces
de  $cl$  passent */
for  $e \in G$  do
  /* pour chaque arête  $e$  de  $G$  */
   $\tau_e \leftarrow \text{creerCellules}(e)$ ;
  /*  $\tau_e$  contient l'ensemble des cellules par lesquelles  $e$  passe
  */
   $\tau \leftarrow \tau_{cl} \cap \tau_e$ ;
   $m \leftarrow \text{moyenne}(\mathcal{C}_c, c \in \tau)$ ;
  /*  $\mathcal{C}_c$  est le coefficient de cyclabilité de la cellule  $c$  */
   $e.w \leftarrow e.w * (1 - m)$ ;
  /*  $e.w$  est le poids de  $e$  */
end

```

---

### 3.4.3 Qualité des traces basées sur l'expérience

Une trace basée sur l'expérience est calculée pour chaque trace expérimentée des 2 jeux de données. Un oracle est tout d'abord utilisé pour affilier les traces expérimentées à un *cluster*. L'oracle est capable de renvoyer le numéro de *cluster* dans lequel se trouve une trace expérimentée à partir de son origine et de sa destination. La Section 3.5 est consacrée à l'implémentation d'une solution remplaçant cet oracle. Pour mesurer la pertinence des traces basées sur l'expérience, la distance entre celles-ci et les traces expérimentées qu'elles tentent de reproduire est calculée. La métrique de distance personnalisée est utilisée, car elle quantifie les parties identiques et non pas seulement les parties proches géographiquement. Les traces expérimentées que le générateur doit reproduire sont toutefois présentes dans les *clusters* utilisés pour générer les traces basées sur l'expérience. Pour les *clusters* ne contenant pas beaucoup de traces, cela pourrait grandement influencer la modification du graphe routier, créant ainsi un biais important. Pour résoudre ce problème, 20 % des traces expérimentées de chaque jeu de données sont choisies au hasard et retirées des *clusters*. Ce groupe de traces forme les sous-jeux de test (460 traces pour Véléal et 873 pour MonRésoVélo). Les 80 % de traces restantes forment les sous-jeux d'entraînement (1840 traces pour Véléal et 3492 pour MonRésoVélo). À noter que l'oracle est toujours capable de retrouver le *cluster* dans lequel les traces du jeu de test se trouvaient initialement.

### 3.4.4 Traces de comparaison

Trois autres types de traces sont utilisés pour comparer la qualité des traces basées sur l'expérience. Trois traces de comparaisons sont calculées pour chaque trace expérimentée, à partir de l'origine et de la destination de celles-ci. Les distances entre la trace expérimentée et les trois traces de comparaison sont ensuite calculées en utilisant là aussi la métrique de distance personnalisée. Les trois types de traces de comparaison sont :

- **Traces générées** : Pour rappel, celles-ci sont les chemins le plus court entre l'origine et la destination des traces expérimentées. Elles sont utilisées comme traces de comparaison pour montrer la nécessité de développer d'autres approches.
- **Traces basées sur l'expérience globale** : Un graphe routier global est généré pour chaque jeu de données. L'Algorithme 3 est utilisé avec toutes les traces du jeu de données pour le créer. Les traces basées sur l'expérience globale sont ensuite tracées en utilisant l'algorithme de Dijkstra sur ce graphe. Ces traces permettent de démontrer l'importance d'identifier les segments de route préférés plus locaux pour calculer des traces proches du comportement réel des cyclistes. Ces traces sont appelées "traces globales" dans la suite de ce travail.
- **Traces Mapbox** : Mapbox<sup>1</sup> propose une API permettant entre autres de générer un itinéraire entre une origine et une destination. Ce service commercial permet de sélectionner le mode de transport utilisé, et le vélo fait partie des possibilités. Mapbox ne divulgue pas l'algorithme utilisé pour le calcul de trajet. Dans le cadre du cyclisme, celui-ci est néanmoins probablement le chemin le plus court sur un graphe pondéré en utilisant les infrastructures cyclables existantes. De nombreux services de calcul de trajet existe, mais seul Mapbox est utilisé pour rester concis. Comparer les traces basées sur l'expérience aux traces Mapbox permet d'évaluer la méthode de génération proposée dans cette section par rapport à une solution commerciale déployée.

### 3.4.5 Résultats

Les Figures 3.13a et 3.13b montrent les résultats avec les sous-jeux d'entraînement et de test de Véléal, et les Figures 3.14a et 3.14b avec les sous-jeux d'entraînement et de test de MonRésoVélo. Ces résultats sont affichés sous la forme de boîtes à moustaches montrant les ensembles de distances entre les quatre types de traces et les traces expérimentées. Les boîtes contiennent les distances dans l'intervalle  $[Q1, Q3]$  et les moustaches celles dans l'intervalle  $[Q1 - 1.5 * IQR, Q3 + 1.5 * IQR]$ , avec  $Qx$  le quartile  $x$  et  $IQR = (Q3 - Q1)$  l'écart interquartile. Les distances hors de ces intervalles sont affichées individuellement, car considérées comme des valeurs aberrantes. La valeur médiane des ensembles de distances est représentée par une ligne orange. On observe tout d'abord que les résultats des 2 sous-jeux de données ne diffèrent pas beaucoup. La présence des traces expérimentées à reproduire dans les *clusters* n'influe pas beaucoup sur les performances de la méthode de génération de traces. C'est une preuve que les *clusters* sont plutôt bien construits, car retirer 20 % des traces les formant à peu d'impact sur les segments de route qu'ils identifient. Seuls les résultats avec les sous-jeux de test sont détaillés ici, car les modèles finaux doivent être capables de générer des trajets qui ne sont pas dans les *clusters* qu'ils utilisent.

#### Veleval

Les traces générées sont les plus éloignées des traces expérimentées, suivies de près par les traces Mapbox. Cela tend à confirmer que les traces Mapbox sont des chemins le plus court calculés sur un graphe routier légèrement modifié. Les distances dans le premier quartile se situent entre 0 et 0.4 pour ces 2 types de traces. Le dernier quartile s'étend jusqu'à la distance maximale pour les 2. La médiane est

---

1. <https://www.mapbox.com/>

légèrement inférieure pour les traces Mapbox, avec environ 0.67 contre environ 0.69. La borne supérieure du troisième quartile est de 0.8 pour les traces générées contre 0.75 pour les traces Mapbox. L'algorithme de Mapbox arrive donc à faire légèrement mieux que l'algorithme de Dijkstra sur un graphe routier non modifié, particulièrement lorsque la trace expérimentée s'éloigne fortement du chemin le plus court. Les cyclistes s'éloignent du chemin le plus court dans le but d'accéder à une infrastructure cyclable. Si le graphe routier de Mapbox est bien modifié à l'aide de ces infrastructures, il est logique que les traces Mapbox performent mieux lorsque les traces générées sont très différentes des traces expérimentées.

Les traces globales font mieux que les 2 types précédents. 75 % de celles-ci ont une distance inférieure à la distance médiane des traces Mapbox. Cependant, le quatrième quartile des distances des traces globales s'étend aussi jusqu'à un, montrant que cette méthode génère de manière non négligeable des traces n'ayant aucun segment de route commun à la trace expérimentée originale. Les traces globales sont calculées sur un graphe routier modifié en prenant en compte l'entièreté du jeu de données. Les poids des segments de route affichés en rouge sur la Figure 3.3 sont donc fortement diminués, et les poids de ceux affichés en bleu très peu. Les traces globales calculées dans des secteurs peu utilisés à l'échelle du jeu de données entier sont donc très proches, voire identique aux traces générées. Au vu des distances entre les traces générées et les traces expérimentées, il n'est pas étonnant qu'un nombre assez important de traces globales soient très différentes des traces expérimentées. Ce sont les traces basées sur l'expérience qui se rapprochent le plus des traces expérimentées, avec une distance médiane d'environ 0.2. 75 % des traces basées sur l'expérience ont une distance inférieure à 0.25. C'est aussi la seule méthode de génération de traces qui ne produit aucune trace entièrement différente de la trace expérimentée originale. Les traces ayant une distance supérieure à 0.5 sont considérées comme aberrantes tant elles sont peu nombreuses et seule une est à une distance supérieure à 0.7. Utiliser les *clusters* pour cibler les modifications à faire sur le graphe routier permet de calculer des traces qui sont assez proches des traces expérimentées dans la majorité des cas, et de ne jamais totalement diverger de celles-ci.

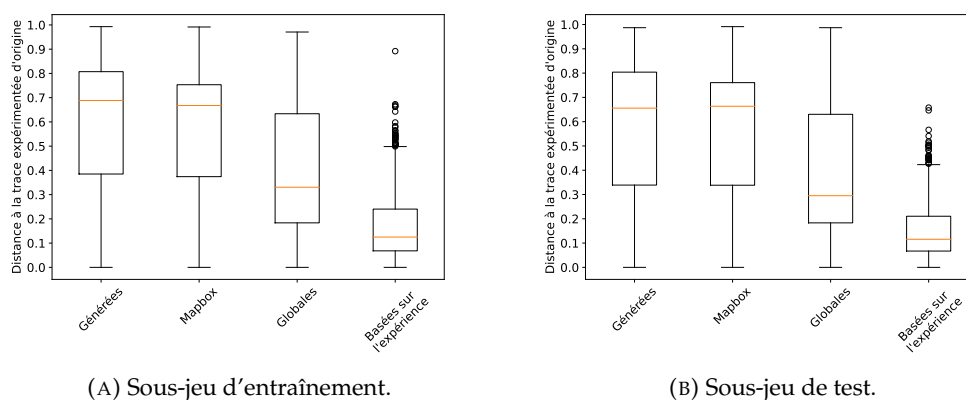


FIGURE 3.13 – Boîtes à moustaches des distances entre les traces expérimentées de Véléval et les quatre autres types de traces.



## MonRésoVélo

La comparaison des performances entre les différents types de traces sont les mêmes pour MonRésoVélo (Figure 3.14). Les traces générées sont les plus éloignées, suivies de près par les traces Mapbox. Les traces Mapbox se différencient moins cette fois-ci, avec une médiane certes un peu moins élevée mais 50 % des traces dans le même intervalle de distance que les traces générées. C'est surprenant, car Montréal possède plus d'infrastructures cyclables que Lyon et Saint-Etienne, ce qui aurait dû améliorer les performances de Mapbox. Il est possible que Montréal ait un nombre suffisant d'infrastructures pour que la modification des poids du graphe routier de Mapbox soit assez homogène. Le cas échéant, le comportement d'un algorithme de chemin le plus court serait le même que sur un graphe non modifié. En effet, ces algorithmes effectuent leurs choix en prenant en compte la différence de poids entre les arêtes, et non les valeurs absolues des poids de celles-ci.

Les traces globales sont de meilleures qualités, mais la différence est moins marquée que pour Véléal. La médiane des distances des traces globales est cette fois-ci aux alentours de 0.5. L'intervalle des quartiles médians commence à environ 0.15 et termine aux alentours de 0.75 pour les traces globales, contre environ 0.3 et 0.85 pour les traces Mapbox. Le problème de la génération de traces globales évoqué pour Véléal semble être plus présent pour MonRésoVélo. Les traces MonRésoVélo sont réparties de manière plus homogène que celles de Véléal, et les segments de route les plus utilisés par les traces expérimentées sont en partie dans le centre. Les segments de route les plus utilisés par les traces générées se situent, eux aussi, dans le centre (voir Figure 3.4). Le graphe modifié utilisé pour calculer les traces globales possède donc beaucoup d'arêtes peu ou pas modifiées, et celles modifiées sont situées dans le centre là où passent les traces générées. Ce n'est donc pas étonnant que les performances des traces globales soient plus proches de celles des traces générées pour le jeu MonRésoVélo. Les traces basées sur l'expérience sont les traces les plus proches des traces expérimentées, avec cette fois-ci 75 % ayant une distance inférieure à 0.3 et une médiane d'environ 0.15. Les valeurs aberrantes commencent cette fois-ci à 0.7. Cela signifie que contrairement à Véléal, de nombreuses traces basées sur l'expérience de MonRésoVélo ont une distance comprise entre 0.5 et 0.7. Aucune des valeurs aberrantes n'atteint une distance de un, même si certaines s'en rapprochent. Les traces basées sur l'expérience de MonRésoVélo sont moins pertinentes que celles de Véléal de manière générale. Cela s'explique par la qualité des *clusters*. En effet, le plus grand *cluster* créé avec MonRésoVélo possède beaucoup de zones utilisées par peu de traces autour des segments de route préférentiels en rouge (voir Figure 3.12). Les modifications du graphe routier avec les *clusters* sont moins précises que celles faites avec Véléal, diminuant la précision des traces basées sur l'expérience. Celles-ci se rapprochent tout de même bien plus des traces expérimentées originales que les types de traces utilisés en comparaison.

### 3.4.6 Résumé de section

La Section 3.4.3 est dédiée à la mise en place d'une méthode de génération de traces basées sur l'expérience. Les segments de route préférés identifiés grâce aux *clusters* de la Section 3.3 ainsi qu'une pondération de poids de graphe routier sont utilisées. Les poids des arêtes sont diminués proportionnellement à leur utilisation par les traces d'un *cluster*. L'algorithme de Dijkstra peut ensuite être utilisé sur le

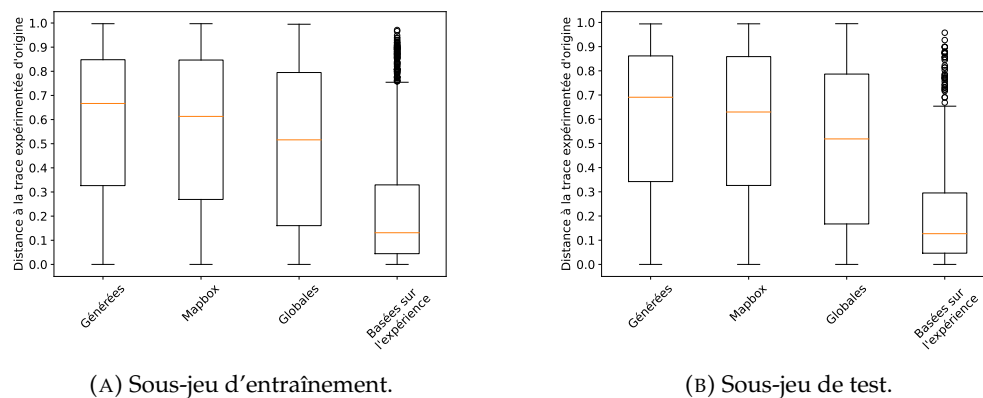


FIGURE 3.14 – Boîtes à moustaches des distances entre les traces expérimentées de MonRésoVélo et les quatre autres types de traces.

graphe modifié pour créer un compromis entre le chemin le plus court et l'utilisation de segments de route préférés, imitant ainsi le comportement des cyclistes. Les traces calculées grâce à cette méthode sont ensuite comparées à trois types de traces (générées, globales et Mapbox). Pour cela, les distances entre les traces expérimentées et les quatre autres types de traces sont comparées. La métrique de distance personnalisée est utilisée. Les traces basées sur l'expérience sont les plus proches des traces expérimentées, bien que les performances soient différentes selon le jeu de données utilisé. Ces différences de performances mettent en lumière l'importance d'utiliser des *clusters* avec des segments de route préférés bien définis.

### 3.5 Classifieur

La méthode de génération de traces basées sur l'expérience se montre assez performante pour calculer des traces proches de traces expérimentées. Cependant, celle-ci a besoin d'un *cluster* de traces expérimentées pour fonctionner. Dans la section précédente, un oracle était utilisé. Celui-ci était capable de renvoyer le numéro de *cluster* d'une trace à partir de sa paire O/D. Le problème de cet oracle est qu'il n'est utilisable qu'avec des O/D de traces expérimentées présentes dans le jeu de données de base. Or les modèles de choix d'itinéraire implicites doivent être capable de créer un itinéraire quelles que soient l'origine et la destination choisies dans le secteur qu'il couvre. Il est donc nécessaire de développer une approche capable de renvoyer un *cluster* à partir d'une paire O/D quelconque. Ce problème est un problème de classification. Ce type de problème est solvable avec un algorithme d'apprentissage supervisé, appelé classifieur par la suite. Ce classifieur (voir le bloc *Prediction* sur la Figure 3.1) doit être implicite pour être entraîné sur les traces sans avoir à définir de variables descriptives. Le classifieur choisi prend en entrée un chemin le plus court et renvoie un numéro de *cluster*. Les traces générées sont utilisées pour l'entraînement. En effet, chaque trace générée correspond à une trace expérimentée. Il est donc possible d'affilier un numéro de *cluster* à chaque trace générée, en prenant celui dans lequel se trouve la trace expérimentée correspondante. Le modèle de choix d'itinéraire final peut ensuite calculer un nouveau chemin le plus court à partir d'une paire O/D et envoyer celui-ci au classifieur entraîné pour obtenir un numéro de *cluster*. La méthode décrite dans la section précédente peut ensuite être appliquée pour générer une trace basée sur l'expérience. Dans cette section, le choix du classifieur est

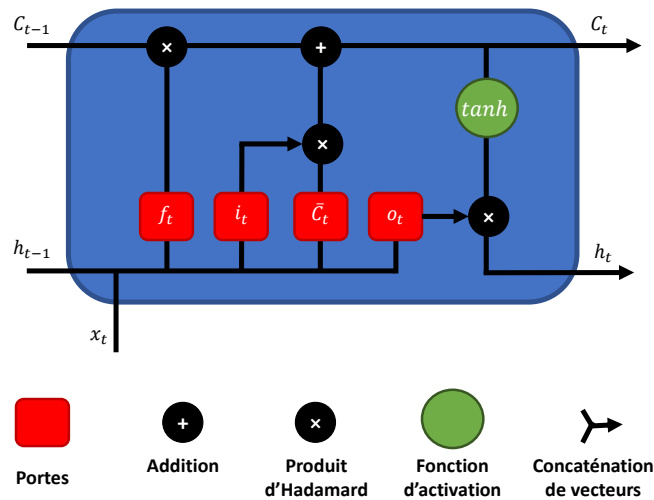


FIGURE 3.15 – Diagramme montrant la structure d'une cellule LSTM.

argumenté. Une description de celui-ci et de son entraînement est ensuite effectuée. Enfin, les résultats obtenus sont discutés.

### 3.5.1 Réseau de neurones LSTM

Une trace générée est une séquence de points géographiques successifs. L'ordre de ces points est un aspect important. Le classifieur à entraîner doit donc être capable de traiter des données séquentielles. Les réseaux de neurones récurrents (RNN pour *Recurrent Neural Network*) sont un type de classifieur ayant montré de bons résultats sur ce type de données. En plus des données d'entrée, un RNN possède un état caché ( $h_t$  pour *hidden state*) qui contient des informations sur les données précédemment traitées.  $h_t$  sert en quelque sorte de mémoire. Cet état caché lui permet de prendre en compte l'aspect temporel des données qu'il doit traiter. Après expérience, un simple RNN ne s'est cependant pas révélé performant. La raison est la longueur des traces générées. En effet, les réseaux de neurones artificiels sont soumis au problème de disparition du gradient. Dans le cas des RNNs, ce problème empêche l'apprentissage des dépendances à long terme [105, 106]. Les LSTM (pour *Long Short Term Memory*) [107] sont un type de réseau de neurones développés spécialement pour faire de l'apprentissage sur de longues séquences. Ceux-ci sont constitués d'un état cellulaire ( $C_t$  pour *cell state*) en plus de l'état caché. L'objectif de l'état cellulaire est de conserver les informations sur le long terme, permettant aux LSTM de performer sur des données séquentielles longues. La Figure 3.15 montre une cellule LSTM.

Les informations contenues dans l'état cellulaire  $C_t$  sont contrôlées et modifiées par trois portes (*gates*). Tout d'abord, la porte d'oubli  $f_t$  supprime les valeurs inutiles contenues dans l'état cellulaire. Ensuite, la porte d'entrée contrôle les nouvelles valeurs qui y seront ajoutées. La porte d'entrée est divisée en 2 parties. La première partie  $i_t$  a pour but de mettre à jour les valeurs existantes, et la seconde partie  $\tilde{C}_t$  génère des valeurs candidates qui peuvent être ajoutées à l'état cellulaire. Enfin, la porte de sortie  $o_t$  décide quelles valeurs de l'état cellulaire sont utilisées pour créer le nouvel état caché  $h_t$ . Les trois portes sont définies comme suit :

$$\begin{aligned}
f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\
i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\
\tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \\
o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)
\end{aligned}$$

$x_t$  est l'élément de la séquence d'entrée  $x$  à l'instant  $t$  et  $h_{t-1}$  est l'état caché renvoyé le LSTM à l'instant précédent ( $t - 1$ ).  $h_1$  est initialisé de manière aléatoire.  $W_f$ ,  $W_i$ ,  $W_C$ ,  $W_o$  et  $b_f$ ,  $b_i$ ,  $b_C$ ,  $b_o$  sont respectivement les matrices de poids et les vecteurs de biais des différentes portes. L'état de la cellule  $C_t$  et l'état caché  $h_t$  à l'instant présent  $t$  sont calculés à l'aide des résultats des différentes portes :

$$\begin{aligned}
C_t &= f_t * C_{t-1} + i_t * \tilde{C}_t \\
h_t &= o_t * \tanh(C_t)
\end{aligned}$$

avec l'opérateur '\*' le produit d'Hadamard et  $C_{t-1}$  l'état cellulaire calculé par le LSTM à l'étape précédente ( $t - 1$ ).  $C_1$  est initialisé aléatoirement.  $\sigma$  et  $\tanh$  sont des fonctions d'activations non linéaires définies par :

$$\begin{aligned}
\sigma &= \frac{1}{1 + e^{-x}} \\
\tanh &= \frac{e^x - e^{-x}}{e^x + e^{-x}}
\end{aligned}$$

Lorsque les  $N$  éléments de la séquence d'entrée  $x = \{x_1, \dots, x_{t-1}, x_t, \dots, x_N\}$  ont été traités par le LSTM, le dernier état caché  $h_N$  est utilisé pour la classification.

### 3.5.2 Prétraitement des traces

Une trace générée est une séquence de  $N$  points ordonnés  $t = \{p_1, \dots, p_N\}$  avec chaque point  $p_t = (lat_t, lon_t)$  contenant une latitude et une longitude. Les réseaux LSTM sont certes adaptés aux longues séquences, mais l'espace de valeurs des latitudes et des longitudes est trop continu pour permettre une convergence. Il est donc nécessaire d'effectuer une discrétisation. La discrétisation utilisée ici est inspirée des travaux de Crivellari et al. [81], qui ont utilisé un réseau LSTM pour prédire les prochaines étapes d'un chemin touristique à partir des étapes déjà effectuées. Ils ont pour cela transformé les traces GPS en séquences de zones touristiques avant de les envoyer à leur LSTM. L'utilisation des infrastructures cyclables en tant que zones cyclables peut sembler être un moyen de discrétiser les traces générées. Cependant, les chemins le plus court ne passent pas nécessairement par des routes en possédant. Pour garder toutes les informations de celles-ci, il faut créer des zones cyclables personnalisées. Les cellules générées pour la métrique de similarité personnalisée sont encore une fois utilisées. Celles-ci sont regroupées en *clusters* avec l'algorithme *k-means*. Celui-ci utilise leurs positions géographiques ainsi que leurs coefficients de cyclabilité. Cela permet aux zones cyclables d'être constituées de cellules proches géographiquement, avec un taux d'utilisation par les traces générées proche. Le résultat est 1565 zones cyclables pour Véléal et 3087 pour MonRésoVélo. Une séquence en entrée du LSTM est une suite d'entiers représentant les numéros des zones cyclables par lesquelles la trace générée passe.

### 3.5.3 Structure du classifieur et hyperparamétrage

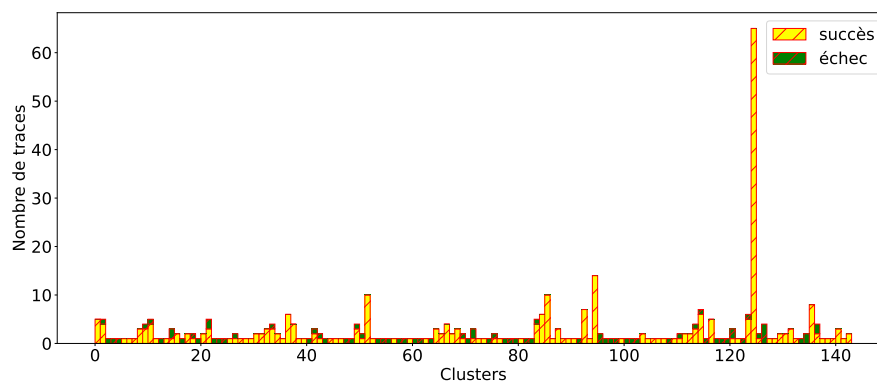
Un classifieur est entraîné par jeu de données. Chaque classifieur est composé de 2 couches de LSTM suivies d'une couche de 256 neurones *fully-connected*. La fonction d'activation ReLU [108] est utilisée après chaque couche, exceptée la couche de sortie, afin de donner des propriétés de non-linéarité au classifieur. L'optimiseur Adam [109] est utilisé avec un taux d'apprentissage de  $5e^{-4}$ . Les apprentissages durent 10 000 étapes pendant lesquelles les traces sont envoyées par groupe de 30. La sortie d'un classifieur est une liste de  $N$  log-probabilités obtenues en utilisant  $\log(\text{Softmax}(x))$ , avec  $N$  le nombre de *clusters* créés avec le jeu de données utilisé. Chaque valeur de cette liste représente la probabilité que la trace expérimentée correspondant à la trace générée en entrée appartienne à un des *clusters*. La fonction de perte utilisée est la *negative log likelihood*. Les sous-jeux d'entraînements et de tests formés dans la Section 3.4.3 sont réutilisés. Seules les traces des sous-jeux d'entraînement sont envoyées lors des entraînements. Les traces appartenant aux sous-jeux de test sont ensuite utilisées pour mesurer la capacité des classifieurs à généraliser les connaissances qu'ils ont acquis pendant l'entraînement.

### 3.5.4 Performances

Le classifieur entraîné avec Véléal atteint un taux de réussite de 96 % avec le sous-jeu d'apprentissage et de 78.6 % avec le sous-jeu de test. Celui entraîné avec MonRésoVélo atteint un taux de réussite de 96.6% avec le sous-jeu d'apprentissage et de 56.6% avec le sous-jeu de test. Les différences de performances entre les 2 types de sous-jeux montrent que les classifieurs sur-apprennent pendant l'entraînement. Deux stratégies sont classiquement utilisées pour diminuer le phénomène. Tout d'abord, la dimension ainsi que le nombre des différentes couches des classifieurs ont été diminués. Diminuer la dimension d'un classifieur lui permet d'oublier les caractéristiques les moins importantes des données par faute de neurones dans lesquels les stocker. Un classifieur trop petit n'aura cependant pas assez de neurones pour stocker toutes les caractéristiques importantes des données. Deux cellules LSTM et une couche de 256 neurones apparaît comme la meilleure configuration. La deuxième méthode est le *dropout* qui consiste à mettre certaines sous-parties des neurones d'un classifieur à 0 aléatoirement. Cela permet de réduire la co-adaptation des neurones, permettant à chacun d'entre eux de retenir une caractéristique importante générale des données [110]. Malheureusement, le *dropout* n'a fait que réduire les performances du classifieur et n'est donc pas utilisé.

La Figure 3.16 montre le détail des prédictions pour chaque *cluster* avec les sous-jeux de test. Pour Véléal, plus le *cluster* est grand, moins il y a de chances que le classifieur se trompe dans ses prédictions. Le classifieur semble bien avoir assimilé les caractéristiques communes des traces présentes dans les plus grands *clusters*. Cependant, lorsque les *clusters* sont plus petits, les prédictions sont plus aléatoires. En effet, certains petits *clusters* voient la totalité de leurs traces classifiées correctement, certains ne sont constitués que de traces mal classifiées et certains voient le classifieur prédire correctement environ la moitié de leurs traces. Le classifieur entraîné avec Véléal semble délaisser les caractéristiques des traces de certains petits *clusters* au profit des caractéristiques de celles dans les plus grands. Pour MonRésoVélo, la taille des *clusters* ne semble pas affecter les capacités de classification. En effet, une partie non négligeable (autour de 30 %) des traces des plus grands *clusters* est mal classifiée. Pour les plus petits *clusters*, le constat est similaire que pour Véléal. La

différence de performances entre le classifieur de Véléal et de MonRésoVélo semble être concentrée sur les plus grands *clusters*. Les traces des plus grands *clusters* de Véléal ont toutes une grande partie de segments en communs (voir Figure 3.7a). Les caractéristiques de celles-ci sont donc très proches. Mais le nombre conséquent de *clusters*, et l'hétérogénéité de leurs tailles empêche le classifieur de retenir ces caractéristiques en même temps que celles de traces dans les plus petits *clusters*. Le problème se pose aussi pour les *clusters* MonRésoVélo qui sont plus nombreux, et plus hétérogènes en termes de taille. À cela s'ajoute des traces dans les plus grands *clusters* possédant des parties communes moins importantes. Tout cela explique les différences de performances entre les 2 jeux de données. Le sur-apprentissage n'est donc pas dû à la structure des classifieurs, mais bien à l'hétérogénéité des données.



(A) Véléal.



(B) MonRésoVélo.

FIGURE 3.16 – Prédictions des classifieurs avec les sous-jeux de test.

### 3.5.5 Résumé de la section

Dans la Section 3.5, une méthode pour remplacer l'oracle de la section précédente est développée. Un classifieur par jeu de données est entraîné. Celui-ci a pour but de retrouver le numéro de *cluster* d'une trace expérimentée à partir du chemin le plus court entre son origine et sa destination. Un LSTM est utilisé, car ce type de réseau de neurones est spécialisé dans l'apprentissage de longues séquences. La structure d'une trace GPS étant dans un espace trop continu pour permettre la convergence d'un réseau de neurones, l'espace est discrétisé à l'aide des cellules générées pour

la métrique de distance personnalisée. Les sous-jeux d'entraînements et de test sont utilisés pour évaluer la capacité des classifieurs à généraliser ce qu'il apprend sur les données. Les performances sont différentes selon le jeu de données. Le taux de réussite est plutôt bon pour Véléval, avec plus de 75 %. Pour MonRésoVélo, le classifieur n'arrive qu'à un taux de réussite de 56 %. Une analyse détaillée des prédictions permet de conclure que ces différences de performances sont dues aux différences importantes entre les 2 jeux de données.

## 3.6 Modèles finaux

### 3.6.1 Analyse quantitative

La même méthode qu'en Section 3.4 est utilisée pour évaluer les performances des modèles finaux. Une trace basée sur l'expérience est calculée pour chaque trace expérimentée des jeux. Cependant, c'est le classifieur qui est utilisé pour le choix du *cluster* et non plus l'oracle. La distance personnalisée entre la trace expérimentée et la trace basée sur l'expérience est calculée. Cette distance est comparée à celle entre la trace expérimentée et les 2 autres types de traces basées sur l'expérience (globales et avec l'oracle). Les résultats pour Véléval sont affichés en Figure 3.17 et ceux pour MonRésoVélo en Figure 3.18. Les traces calculées avec les LSTM sont très proches de celles calculées avec l'oracle pour les 2 sous-jeux d'entraînements. Ce n'est pas étonnant, car les LSTM entraînés avec les 2 jeux de données sont proches d'une précision de 100 % avec les sous-jeux d'entraînements. Les différences entre les LSTM et les oracles résident dans les quelques pourcents de traces mal classifiées. Les traces mal classifiées pèsent logiquement plus sur les performances avec les sous-jeux de test.

Pour Véléval, la borne supérieure des 2 quartiles du milieu est 2 fois plus grande avec le LSTM qu'avec l'oracle (environ 0.4 contre 0.2). De plus, la moustache supérieure avec le LSTM atteint 0.8, contre 0.45 avec l'oracle. La médiane est cependant la même entre les 2 manières de trouver le *cluster*. Les 20 % de traces mal classifiées par le LSTM semblent être assez éloignées des traces expérimentées d'origine. Cependant, les traces créées à l'aide du LSTM sont meilleures que celles des traces globales, avec par exemple aucune trace n'atteignant une distance de 1, et seules quelques traces aberrantes ayant une distance supérieure à 0.8. Cela signifie que, pour le jeu de données Véléval, il vaut mieux se tromper de *clusters* plutôt que d'utiliser tout le jeu de données pour créer une trace basée sur l'expérience.

Ce constat n'est qu'à moitié partagé par le jeu de données MonRésoVélo. En effet, les traces calculées à l'aide du LSTM sont plus proches des traces expérimentées d'origines que les traces globales, avec une distance médiane bien meilleure (environ 0.25 contre 0.5) et un intervalle des 2 quartiles centraux un peu plus bas (de 0.15 à 0.8 pour les traces globales et de 0.08 à 0.75 pour le LSTM). Cependant, une partie non négligeable des traces calculées à l'aide du LSTM n'a aucun segment de route en commun avec la trace expérimentée d'origine. Cela signifie que dans certains cas, entraîner et utiliser un LSTM ne permet pas de créer une trace basée sur l'expérience de meilleure qualité que la trace globale. Pire, il peut arriver que la trace globale soit de meilleure qualité que celle calculée avec le LSTM. Cependant, il est important de noter que la distance médiane des traces créées avec le LSTM est bien plus proche de celle des traces créées avec l'oracle que de celle des traces globales. Cela signifie que le LSTM est tout de même utile dans beaucoup de cas. Les traces qu'il génère se

rapprochent de manière générale plus des traces expérimentées que les traces créées sans utiliser d'oracle.

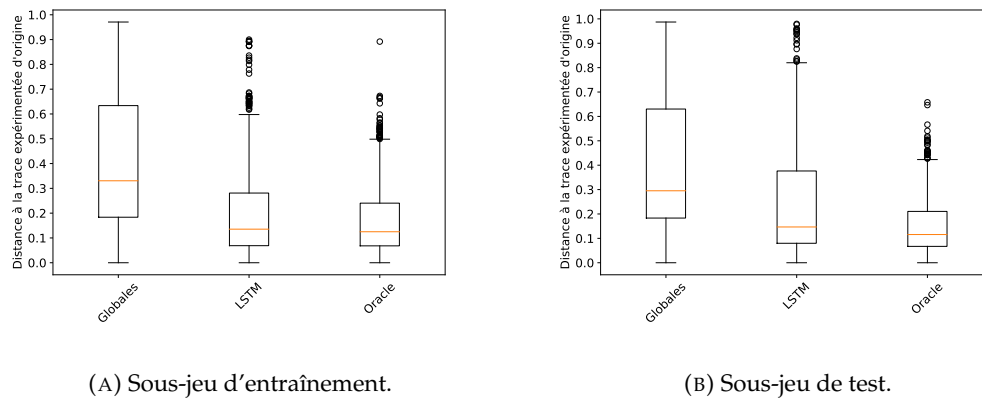


FIGURE 3.17 – Boîtes à moustaches des distances entre les traces expérimentées de Véléval et les trois types de traces basées sur l'expérience.

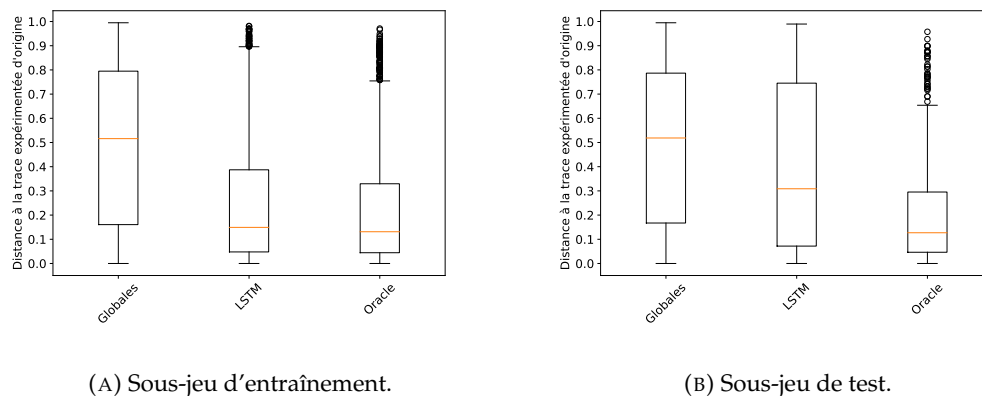


FIGURE 3.18 – Boîtes à moustaches des distances entre les traces expérimentées de MonRésoVélo et les trois types de traces basées sur l'expérience.

### 3.6.2 Analyse visuelle de traces basées sur l'expérience

Les Figures 3.19 et 3.20 montrent chacune 2 cartes sur lesquelles sont affichées une trace expérimentée, sa trace générée correspondante ainsi que la trace basée sur l'expérience créée par le modèle entraîné avec le bon jeu de données. Des traces ayant été calculées avec le bon *cluster* sont montrées en Figures 3.19a et 3.20a. Les Figures 3.19b et 3.20b affichent des traces calculées avec le mauvais *cluster*. Cela permet de montrer les performances des 2 modèles dans un cas favorable ainsi que dans un cas défavorable. Toutes les traces expérimentées d'origine font partie du sous-jeu de test et ne sont donc jamais dans le *cluster* utilisé pour calculer la trace basée sur l'expérience.

La Figure 3.19a est un très bon exemple des capacités de simulation de comportement de cycliste d'un modèle implicite. Sur cette carte, on observe trois traces dont



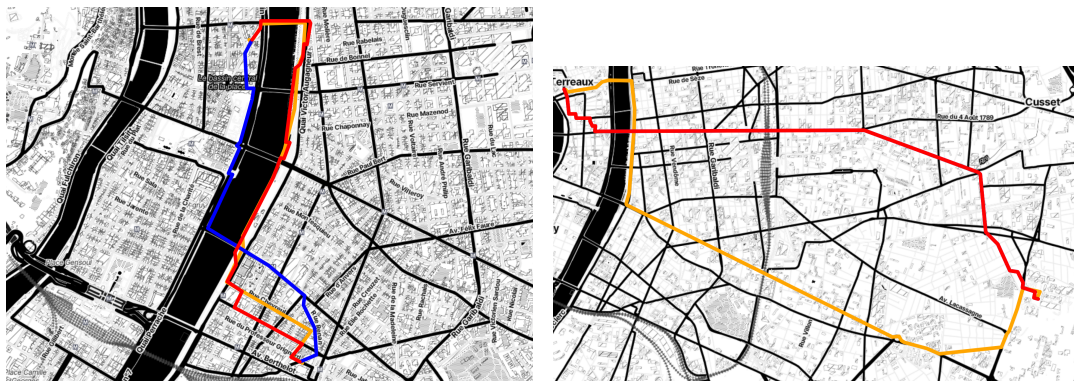
les points d'origine et de destination sont proches du Rhône. Le point de départ est proche de la rive droite du fleuve, et en amont du point de destination que se situe proche de la rive gauche de celui-ci. Comme dit précédemment, les berges gauches du Rhône sont très prisées par les cyclistes, grâce à des infrastructures cyclables séparées des voitures entre autres. Il est donc assez logique de voir la trace expérimentée atteindre assez rapidement ces berges puis les longer. Le cycliste démarre même en allant à l'opposé de sa destination pour prendre le pont le plus proche et parcourir une plus grande distance sur celles-ci. Celui-ci quitte les berges par une sortie proche de sa destination (les berges étant en contrebas, on ne peut pas y entrer ou les quitter n'importe où). La trace générée commence par longer les berges droites jusqu'au pont le plus proche de la destination, qu'elle traverse. Elle continue sur la route face au pont puis tourne à droite pour minimiser la distance entre cette route et la destination. La trace générée est sans surprise plus courte que la trace expérimentée. La trace créée par le modèle est très proche de la trace expérimentée. En effet, les poids des routes formant les berges gauche semblent avoir été tellement diminués que le chemin le plus court sur le graphe modifié part lui aussi à l'opposé de la destination pour les atteindre. Il entre sur les berges par la même entrée que la trace expérimentée, et en sort par une sortie plus proche de la destination. La fin du chemin jusqu'à celle-ci est ensuite différente. La trace créée par le modèle passe par une rue parallèle à celle utilisée par le cycliste. La diminution des poids semble être moins marquée à cet endroit-là, empêchant le modèle d'effectuer le bon détour. On peut aussi noter qu'à peu près à mi-chemin, la trace du modèle sort puis rentre de nouveau rapidement sur les berges. Cela est dû à la taille des cellules utilisées. La route collée aux berges à cet endroit a vraisemblablement vu son poids diminué car il traverse les mêmes cellules que les segments de berges. L'algorithme de Dijkstra prend donc la décision de l'utiliser malgré le détour que cela implique.

Sur la Figure 3.19b, la trace expérimentée démarre d'un point à l'Est pour atteindre un point au Nord-Ouest. Une fois encore, le cycliste est parti à l'opposé de sa destination en se dirigeant plein Sud. Il atteint rapidement une avenue menant directement au Rhône. Il longe ensuite les berges gauches avant de finalement traverser le fleuve et atteindre sa destination. La trace générée démarre en direction du Nord-Ouest. Elle met plus de temps à atteindre une avenue donnant sur le Rhône, et celle-ci est située plus au Nord que celle empruntée par le cycliste. Une fois arrivée au Rhône, la trace le traverse sans emprunter les berges, par un pont plus au Sud que celui utilisé par le cycliste. Finalement, la trace créée par le modèle rejoint sa destination en se dirigeant vers le Nord-Ouest. On ne distingue pas la trace générée, car celle-ci est entièrement recouverte par la trace créée par le modèle. En effet, le mauvais *cluster* ayant été choisi, des routes peu pertinentes ont vu leurs poids diminuer. Dans le cas présenté, ces routes ne semblent pas être dans le secteur entre l'origine et la destination de la trace expérimentée. La trace basée sur l'expérience est donc le chemin le plus court sur un graphe routier non modifié.

La Figure 3.20a affiche des traces allant d'une origine au Sud-Est à une destination au Nord-Ouest. La trace générée montre qu'aller toujours tout droit sur une route (qui devient un boulevard à terme) permet de quasiment atteindre la destination. Cependant, le cycliste a décidé de partir au Sud pour effectuer un itinéraire bien plus sinueux qui longe cette route, pour finalement la croiser en direction du Nord au dernier moment pour atteindre sa destination. Ce choix a probablement été fait dans le but de traverser le parc visible au Sud de la carte, et que la route empruntée par la trace générée ne fait que longer. Faire du vélo dans un parc permet d'être éloigné des voitures et dans un endroit vert, 2 caractéristiques d'itinéraires recherchées

par les cyclistes. La trace calculée par le modèle est très similaire à la trace expérimentée. Celui-ci simule bien le détour débuté dès le départ par le cycliste dans le but de traverser le parc. Que ce soit avant ou après cette traversée, la trace basée sur l'expérience est parfaitement identique à la trace expérimentée. Les seules différences entre les 2 traces se situent dans le parc. En effet, la trace calculée par le modèle semble plus directe que la trace expérimentée. Les différents chemins dans le parc sont très proches, comparés aux routes dans la ville. Il est probable que plusieurs chemins voisins passent par les mêmes cellules, et que leurs poids aient été diminués de la même façon. Le modèle crée donc un plus court chemin dans le parc, sans parvenir à simuler les petits détours du cycliste.

Les traces sur la Figure 3.20b vont d'un point à l'Ouest à un point à l'Est. La trace générée entre ces 2 points tente le plus possible d'aller en direction de l'Est. À cause de la disposition des routes disponibles, le résultat est une trace avec beaucoup de tournants. Le cycliste a lui décidé de rejoindre un chemin séparé des voitures et qui longe une voie de chemin de fer. Les rails et le chemin sont séparés par un grillage par sécurité. Lorsque le cycliste est proche de sa destination, il quitte ce chemin pour la rejoindre. La trace calculée par le modèle ne partage que peu de segments de route avec les 2 autres traces. De plus, elle semble plus longue que la trace expérimentée. Il semble que le *cluster* choisi ait entraîné une modification des poids des routes dans le secteur entre l'origine et la destination de la trace expérimentée. Cependant, le *cluster* choisi n'étant pas le plus pertinent, ces modifications créent des détours non pertinents. La trace calculée par le modèle est donc une trace plus longue que le chemin le plus court, et ne parvenant pas à reproduire le comportement d'un cycliste.



(A) Trace générée bien classifiée par le LSTM.

(B) Trace générée mal classifiée par le LSTM.

FIGURE 3.19 – Cartes montrant 2 traces expérimentées (en orange), les traces générées correspondantes (en bleu) et les traces calculées par le modèle de choix d'itinéraire entraîné avec le jeu Véléval (en rouge).

### 3.6.3 Résumé de la section

La Section 3.6 évalue les modèles finaux créés à partir de la méthode présentée dans ce chapitre. Deux analyses sont effectuées. La première est quantitative. La même méthode qu'en Section 3.2.3 est utilisée. Des traces sont calculées par les modèles entre les origines et les destinations de traces expérimentées. La distance entre ces traces basées sur l'expérience et les traces d'origine est calculée, et comparée à d'autres traces. Les modèles parviennent à créer des traces plus similaires aux traces

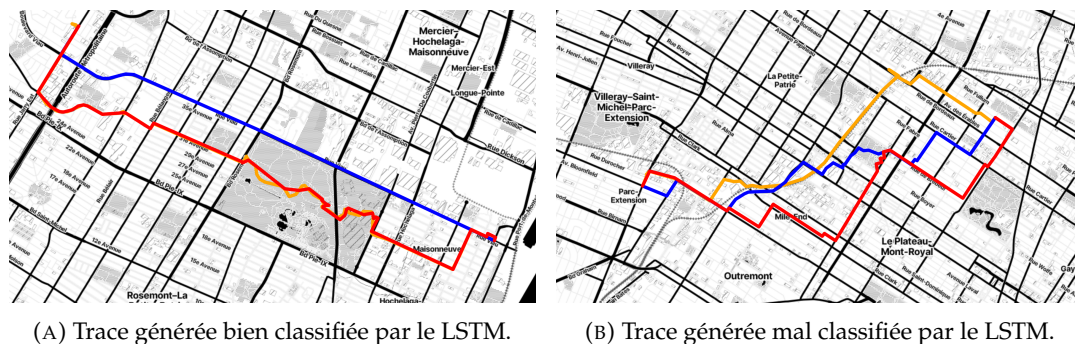


FIGURE 3.20 – Cartes montrant 2 traces expérimentées (en orange), les traces générées correspondantes (en bleu) et les traces calculées par le modèle de choix d'itinéraire entraîné avec le jeu MonRésVo (en rouge).

expérimentées que les traces globales. Cependant, les traces créées avec l'oracle sont de meilleures qualités, démontrant l'importance d'utiliser le bon *cluster* pour modifier le graphe routier. Enfin, une analyse visuelle des traces créées par les modèles est effectuée. Cette analyse confirme l'importance du choix du *cluster*. Les modèles créent des traces identiques ou de moins bonne qualité que les traces générées lorsqu'ils utilisent le mauvais *cluster*. Avec le bon *cluster* cependant, ils sont capables de simuler de manière très précise le comportement des cyclistes, et ce même lorsque ceux-ci décident d'aller à l'opposé de leurs destinations.

### 3.7 Conclusion et discussion

Il est important d'être capable de reproduire le comportement des cyclistes dans le but de mieux le comprendre. Cela participe à la création d'infrastructures plus pertinentes, incitant à un changement modal vers le cyclisme au sein des villes. Ce chapitre propose une méthode de création de modèles de choix d'itinéraire implicites pour les cyclistes. À partir d'une paire O/D, les modèles finaux sont capables de générer un trajet individuel de cyclistes en se basant sur des données comportementales. L'utilisation d'algorithmes d'apprentissage spécifiques permet d'utiliser les données sans définir de variables descriptives. Cela permet aussi de se passer d'ensemble de trajets possibles. Les modèles sont construits grâce à des traces GPS générées par des cyclistes. Un algorithme de *clustering* utilisant une métrique de distance personnalisée regroupe les traces passant par les mêmes routes. Cela permet d'identifier les segments de route préférés des cyclistes. Une méthode est ensuite déployée pour créer des traces basées sur l'expérience à partir de ces segments. Ces traces sont un compromis entre le chemin le plus court et l'utilisation de ces segments. Un classifieur utilisant des couches LSTM est ensuite entraîné à retrouver un numéro de *cluster* à partir d'un chemin le plus court. Cela permet aux modèles finaux de calculer une trace à partir de n'importe quelle paire O/D. Les modèles finaux fonctionnent en 4 étapes :

- L'algorithme de Dijkstra est utilisé sur un graphe routier non modifié pour créer un chemin le plus court entre une origine et une destination.
- Le chemin le plus court est envoyé au classifieur qui renvoie un numéro de *cluster*.
- Les traces GPS de ce *cluster* sont utilisées pour modifier les poids des arêtes du graphe routier.

- L'algorithme de Dijkstra est utilisé sur le graphe routier modifié pour générer un itinéraire cyclable.

Deux jeux de traces GPS sont utilisés pour créer 2 modèles distincts. Le premier, appelé Véléval, est formé en majorité de trajets domicile/travail à Lyon et à Saint-Etienne. Le deuxième, appelé MonRésoVélo, est *crowdsourcé*. Les traces qu'il contient se situent dans la ville de Montréal. Chaque trace calculée par un des modèles est comparée au chemin le plus court, à un itinéraire calculé par un service commercial comprenant le vélo comme mode de transport et à une trace calculée avec un modèle similaire, mais n'utilisant pas le *clustering*. Les traces créées par les modèles s'approchent plus des traces GPS d'origines que tous les autres types de traces utilisées en comparaison. Cependant, former des *clusters* identifiant correctement des segments de route préférés est primordial à la création de traces simulant correctement le comportement des cyclistes. De plus, choisir un *cluster* identifiant les mauvais segments de route préférés empêche la création de traces se rapprochant du comportement d'un cycliste.

Ce travail s'accompagne de limites. La plus évidente est montrée par la différence de performances entre les 2 modèles finaux. Le modèle utilisant Véléval obtient de très bons résultats à toutes les étapes. Celui utilisant MonRésoVélo est systématiquement moins performant. Cela s'explique par les biais des 2 jeux de données utilisés qui sont diamétralement opposés. En effet, Véléval est majoritairement constitué de trajets domicile/travail effectués par environ 40 cyclistes. Plusieurs traces sont forcément très similaires. Il est logique que l'algorithme de *clustering* trouve facilement des *patterns* dans les données, représentés par les segments de route préférés. Ces segments étant bien définis, les traces basées sur l'expérience créées par le modèle final s'approchent beaucoup des traces GPS d'origines. La similitude entre les traces ainsi que la qualité des *clusters* permet au classifieur d'avoir une bonne précision même sur des traces non utilisées lors de son entraînement. De son côté, MonRésoVélo est constitué de traces créées par un plus grand nombre de cyclistes. Ces cyclistes n'ont de plus pas tous utilisés le vélo avec les mêmes objectifs. Il y a donc forcément moins de *patterns* dans les données, expliquant la moins bonne qualité des *clusters*. La zone dans laquelle les traces sont situées est aussi plus grande. Le classifieur doit donc faire face à plus de zones cyclables possibles en plus des *clusters* de moins bonne qualité. Enfin, la méthode de création des modèles est basée sur des hypothèses formulées dans le contexte d'un cycliste utilisant le vélo pour atteindre une destination. Malgré la suppression des traces faisant de trop gros détours, il est probable qu'un certain nombre de trajets de MonRésoVélo n'aient pas été effectués dans cet objectif. Ces traces se situent donc hors des objectifs du modèle. Malgré ces biais, le modèle entraîné avec MonRésoVélo parvient à créer des traces de meilleure qualité que celles utilisées en comparaison. Cela démontre une certaine robustesse dans la méthode expliquée dans ce chapitre. Les algorithmes d'apprentissage parviennent à trouver certains *patterns* présents dans MonRésoVélo et à les reproduire, même si ceux-ci ne sont présents que dans un sous-ensemble du jeu. Un jeu de données plus représentatif que Véléval, mais formé par des cyclistes utilisant le vélo comme un moyen de transport, pourrait permettre une meilleure évaluation des modèles de choix d'itinéraire implicites. Un jeu provenant d'un BSS pourrait par exemple parfaitement convenir.

Une deuxième limite est l'absence d'explication des comportements simulés. Les choix effectués par les modèles sont faits à partir des représentations implicites qu'ils se font des données, et sont donc par nature difficilement explicables. Un moyen de comprendre ces choix est d'analyser les traces calculées. Cette analyse demande

néanmoins une connaissance préalable du comportement général des cyclistes, ainsi que du secteur dans lequel les traces se situent.

Enfin, les traces GPS possèdent des propriétés temporelles en plus des propriétés spatiales. Seules les propriétés spatiales ont été utilisées dans ce chapitre, dans le but de faciliter l'apprentissage. L'intégration des propriétés temporelles dans de futurs travaux pourraient néanmoins améliorer les performances des algorithmes d'apprentissage.

## Chapitre 4

# Feu de circulation intelligent pour cyclistes

### 4.1 Introduction

La création des modèles de choix d'itinéraire implicites détaillée dans le chapitre précédent a mis en lumière l'utilisation par les cyclistes de segments de route préférés. Une manière d'étendre ces travaux est de s'intéresser à la manière de transformer un segment de route quelconque en segment de route préféré par les cyclistes. La sécurité étant un point important dans le choix d'itinéraire des cyclistes, séparer les flux de vélos et de véhicules motorisés est une manière efficace d'inciter à l'utilisation de certaines routes. Cela se fait classiquement par la construction de pistes cyclables séparées physiquement des voies de voitures. Dans ce chapitre, un feu de circulation séparant les flux de vélos et les flux de véhicules motorisés est développé. Cette solution est moins coûteuse en place que la construction de pistes cyclables séparées. Le déploiement d'un tel feu permettrait de plus la sécurisation du passage des vélos aux intersections, celles-ci étant connues pour être des lieux dangereux pour les cyclistes [111]. L'apprentissage par renforcement profond est utilisé pour limiter l'augmentation du temps d'attente des véhicules produite par la sécurisation du feu.

#### 4.1.1 Les cyclistes et feux de signalisation

Les cyclistes sont connus pour ne pas toujours respecter les feux rouges. La proportion de cyclistes traversant une intersection lorsque le feu est rouge diffère selon les études. Elle varie de 40 % [112] à 60 % [113] des cyclistes pour les plus récentes observations. Schleinitz et al. [114] ont montré en 2013 que les cyclistes australiens, un pays où l'on roule à gauche, sont plus à même de griller un feu rouge lorsqu'ils tournent à gauche. Les auteurs ne sont pas parvenus à démontrer une augmentation du risque d'accident lorsqu'un cycliste traverse au rouge durant leurs observations. Hollingworth et al. [115] ont montré une petite augmentation du risque d'accident avec blessures chez les cyclistes ne respectant pas les feux rouges. Ils notent cependant que cette augmentation peut être due au comportement général plus risqué des cyclistes le faisant, plus que l'action même de griller un feu rouge. Les intersections contrôlées par feux restent cependant des endroits dangereux pour les cyclistes. Miranda et al. [111] ont étudié en 2011 les occurrences de cyclistes blessés à ce type d'intersection. Leurs résultats suggèrent que la sécurité des cyclistes aux intersections à feux est significativement affectée par le volume de cyclistes et le trafic routier. Plus important, les conflits entre les trajectoires des cyclistes et des véhicules motorisés augmentent significativement les risques de collisions, particulièrement dans le cas

de trajectoires tournant à droite. Cette étude a été menée à Montréal, où les véhicules roulent à droite. Que ce soit au Canada ou en Australie, les comportements dangereux effectués ou subis par les cyclistes augmentent dans le cas de trajectoires n'impliquant pas la traversée d'autres voies. Des solutions existent déjà pour réduire le problème. En France, par exemple, les panneaux M12 permettent aux cyclistes de traverser l'intersection même si le feu est rouge, à condition de laisser la priorité aux véhicules et aux piétons qui ont le feu vert.

La sécurité des cyclistes étant diminuée aux intersections à feux, des expériences tentent de faciliter leur passage. En 2019, Andres et al. [116] ont créé un e-vélo conçu pour permettre aux cyclistes de prendre une vague verte. Une vague verte est une succession de feux synchronisés pour que les véhicules les traversant successivement et à la bonne vitesse arrivent à chacun d'eux pendant une phase verte. Le e-vélo adapte son assistance électrique pour permettre au cycliste de garder la bonne vitesse pendant une vague verte. De manière similaire, Frohlich et al. [117] ont développé en 2016 une application mobile suggérant une plage de vitesse à l'utilisateur pour atteindre le prochain feu de circulation pendant une phase verte. D'autres études ont pour but de modifier les intersections pour les cyclistes. Cela a pour avantage de profiter à tous les cyclistes, et non pas uniquement à ceux possédant le bon équipement. De Angelis et al. [118] ont demandé à différents cyclistes de noter différentes interfaces à des feux de signalisation indiquant si ceux-ci sont dans les temps pour une vague verte. Leurs résultats montrent des différences significatives selon les nationalités. Anagnostopoulos et al. [119] ont eux proposé des feux de signalisations donnant la priorité aux cyclistes en les détectant grâce à leurs smartphones. Ils n'ont cependant pas évalué l'impact d'un tel système sur le trafic des véhicules motorisés.

#### 4.1.2 DRL pour le contrôle de feux de signalisation en simulation

Différentes études s'intéressent à l'optimisation d'un ou plusieurs feux de signalisation [120, 121]. L'apprentissage par renforcement profond (DRL pour *Deep Reinforcement Learning*) a été plusieurs fois utilisé dans ce but. Le DRL est basé sur l'apprentissage par renforcement (RL pour *Reinforcement Learning*), un domaine de l'apprentissage automatique dans lequel un agent apprend par l'expérience. L'agent évolue dans un environnement durant des épisodes finis ou infinis, et a la possibilité d'effectuer des actions modifiant celui-ci. À chaque étape  $t$ , l'agent reçoit l'état de son environnement  $s_t \in S$  et choisit une action  $a_t \in A$ , avec  $S$  l'ensemble des états de l'environnement possible et  $A$  l'ensemble des actions que l'agent peut effectuer. Une fois l'action exécutée, l'agent reçoit le nouvel état de l'environnement  $s_{t+1} \in S$  ainsi qu'une récompense  $r_t$ .  $r_t$  est une valeur numérique quantifiant la qualité de l'action effectuée. Le but de l'agent est de développer une politique  $\pi$  associant une action à un état  $\pi(s) = a$  permettant de maximiser la somme des récompenses  $R = \sum_{t=0}^T \gamma^t r_t$  avec  $\gamma \in [0, 1)$ .  $\gamma$  est appelé le *discount rate* et pondère l'importance des futures récompenses. Les algorithmes de DRL sont des algorithmes de RL qui utilisent l'apprentissage profond pour choisir les meilleures actions (plus d'explications sont données dans la Section 4.2). Certaines études utilisent le DRL pour modifier le comportement de feux de circulation. Certaines utilisent un cycle prédéfini et utilisent le DRL pour choisir quand effectuer un changement de phase [122, 123, 124]. Dans ces études, l'agent a le choix de rester à la phase actuelle ou de passer à la suivante, avec un temps prédéfini entre 2 décisions. En 2019, Genders et al. [125] ont utilisé un feu avec un cycle prédéfini et des temps de phases vertes

modifiables. À la fin de chaque cycle, l'agent a la possibilité d'augmenter ou de réduire de 5 secondes la durée d'une phase. Le but de l'agent est de trouver la durée de chaque phase permettant de minimiser le temps d'attente des véhicules à l'intersection. Mais le DRL est assez puissant pour apprendre un cycle dynamique, et des études l'ont utilisé dans ce but. Dans celles-ci, l'agent DRL sélectionne la voie à mettre au vert à intervalles réguliers. Celui-ci ne choisit donc non plus seulement le moment de changement des phases, mais aussi l'ordre de celles-ci. Les différences entre ces études résident dans leurs méthodes de comparaison, et dans le paramétrage de leurs simulations. Certains auteurs comparent leur approche DRL à une approche d'apprentissage profond [126] ou à d'autres approches DRL [127]. Plusieurs études se comparent à des méthodes statiques et dynamiques de contrôle de feu de circulation existantes. Wang et al. [128] l'ont par exemple fait en 2019 dans des simulations avec une demande de trafic évoluant de manière arbitraire. Genders et al. [125] ont eux effectués les mêmes comparaisons, mais en simulant des heures de pointes pour tester la robustesse du DRL dans des situations plus réalistes. Cette dernière étude montre que le DRL peut être plus efficace en simulation que certaines méthodes de contrôle de feu déjà déployées.

### 4.1.3 Positionnement

Des expérimentations sont prévues en France pour permettre aux cyclistes attendant à un feu rouge de partir plus tôt que les voitures. Cela permet de séparer les 2 flux le temps que les cyclistes atteignent une vitesse suffisante. L'idée derrière le travail expliqué dans ce chapitre est d'étendre ce concept et d'ajouter des phases vertes distinctes pour les cyclistes. Cela permettrait de totalement séparer les flux de vélos des flux de voitures, comme le ferait une infrastructure physique dédiée. L'avantage de séparer les flux de cette manière est le coût bien moins important en termes de place. Cependant, c'est le temps d'attente des véhicules qui serait grandement impacté. En effet, plus de phases vertes signifie des cycles plus longs, et donc un temps d'attente supérieur entre 2 phases vertes pour toutes les voies. Ce chapitre propose une méthode de sélection de phase basée sur le DRL, permettant la création de phases vertes pour cyclistes avec un impact limité sur le temps d'attente des véhicules à l'intersection. L'agent contrôle l'ordre ainsi que le moment de changement des phases. Des données de comptages de vélos ainsi que de comptages de voitures sont utilisées pour comparer cette approche à d'autres déjà déployées dans un contexte assez réaliste. Ce travail est motivé par l'espoir qu'un tel type d'infrastructure avec un impact suffisamment modéré sur le temps d'attente favoriserait un changement modal vers le vélo, augmentant ainsi la sécurité des cyclistes [129]. Dans ce chapitre, 2 agents DRL sont entraînés, chacun utilisant un algorithme DRL différent. Les 2 algorithmes DRL testés sont d'abord expliqués. Les composantes nécessaires pour entraîner les agents sont ensuite définies. Les méthodes de simulation de l'environnement et d'entraînement des agents sont ensuite décrites. Le meilleur des 2 agents est retenu, et ses performances en termes de temps d'attente sont comparées à celles d'autres méthodes de contrôle de feu de circulation. Enfin, le comportement du feu lorsqu'il est contrôlé par l'agent DRL retenu est expliqué.

## 4.2 Apprentissage par renforcement profond

Une solution à base de DRL est proposée pour limiter l'augmentation du temps d'attente des véhicules causée par l'ajout des phases vertes pour cyclistes. Il existe



différents algorithmes de DRL, pouvant être divisés en 3 catégories. Les algorithmes dits *value-based* ont pour but d'estimer une fonction donnant une valeur à chaque paire état-action, notées  $Q(s, a)$ . Une phase d'exploration est effectuée durant l'entraînement pour estimer au mieux ces valeurs. La politique  $\pi$  est explicite dans ce type d'algorithme. Lorsque l'agent reçoit un état  $s$ , la politique de celui-ci choisit l'action ayant la plus grande valeur  $\pi(s) = \operatorname{argmax}_a Q(s, a)$ . Les algorithmes dits *policy-based* estiment eux directement la politique  $\pi$ . Celle-ci renvoie une distribution de probabilités, et un échantillonnage est effectué pour choisir une action  $a$ . L'objectif est d'augmenter les probabilités des bonnes actions et de diminuer les probabilités des mauvaises actions, jusqu'à arriver à la politique optimale. Enfin, les algorithmes dits *actor-critic* sont un mix des 2 autres catégories. Une partie *actor* estime la politique  $\pi$  et une partie *critic* donne une valeur aux choix que celle-ci effectue. Ces dernières années, les algorithmes *value-based* et *actor-critic* montrent de meilleures performances que les algorithmes *policy-based*. Deux algorithmes récents, un *value-based* et un *actor-critic*, sont expliqués dans cette section puis comparés dans la suite de ce chapitre.

### 4.2.1 Double Dueling Deep Q-Network (3DQN)

3DQN est un algorithme *value-based*. Il est basé sur le *Deep Q-Network*, l'algorithme de DRL ayant permis la démocratisation du domaine.

#### Deep Q-Network

En 2015, Mnih et al. [130] ont développé un algorithme appelé *Deep Q-Network* (DQN) capable d'apprendre des politiques de niveau humain. DQN est basé sur un algorithme d'apprentissage par renforcement appelé *Q-learning* [131]. Le *Q-learning* utilise une fonction  $Q : S \times A \rightarrow \mathbb{R}$  pour calculer la valeur d'une paire état-action, appelée Q-valeur. À chaque fois que l'agent choisit une action  $a_t$ , la Q-valeur  $Q(s_t, a_t)$  est mise à jour en utilisant l'équation de Bellman comme suit :

$$Q^{new}(s_t, a_t) \leftarrow (1 - \alpha) \cdot Q(s_t, a_t) + \alpha \cdot (r_t + \gamma \max_a Q(s_{t+1}, a))$$

avec  $\alpha$  le taux d'apprentissage. La politique finale choisit l'action ayant la meilleure Q-valeur. Le DQN utilise un réseau de neurones appelé le Q-réseau pour approximer  $Q$ . Celui-ci est noté  $Q_\theta$ ,  $\theta$  représentant les paramètres (c.-à-d. les poids) du réseau de neurones. Le Q-réseau est entraîné à minimiser la fonction de perte  $L$  définie par :

$$L(\theta) = (Y_t^{DQN} - Q_\theta(s_t, a_t))^2$$

$$Y_t^{DQN} = r_t + \gamma \max_{a'} Q_{\theta'}(s_{t+1}, a')$$

$Y_t^{DQN}$  est appelée la valeur cible et  $\theta'$  représente les paramètres du réseau cible, un second réseau de neurones ayant la même architecture que le Q-réseau. Le réseau cible est uniquement utilisé pour calculer  $Y_t$ , car cela est nécessaire à la convergence du Q-réseau. Le réseau cible est mis à jour en direction du Q-réseau régulièrement au cours de l'entraînement.

### Double Deep Q-Network

En 2016, van Hasselt et al. [132] ont démontré que l'algorithme DQN souffre d'un biais de surestimation des Q-valeurs et ont proposé une solution inspirée de l'algorithme double *Q-learning* [133]. Cette solution, appelée *Double Deep Q-Network* (DDQN) est très similaire à DQN, à l'exception près de la valeur cible qui est cette fois-ci calculée comme suit :

$$Y_t^{DDQN} = r_t + \gamma Q_{\theta'}(s_{t+1}, \underset{a'}{\operatorname{argmax}} Q_{\theta}(s_{t+1}, a'))$$

L'algorithme DQN se sert du réseau cible pour choisir et évaluer l'action  $a'$  utilisée pour calculer la valeur cible. Dans l'algorithme DDQN en revanche,  $a'$  est choisie par le Q-réseau puis évaluée par le réseau cible. Cela réduit la surestimation des Q-valeurs, améliorant ainsi les politiques produites.

### Dueling Deep Q-Network

La fonction d'avantage est définie par  $A(s, a) = Q(s, a) - V(s)$  où  $V(s)$  est appelée la fonction de valeur et représente la récompense moyenne à long terme pouvant être attendu lorsque l'environnement se trouve dans l'état  $s$ .  $A(s, a)$  représente donc l'avantage de choisir  $a$  par rapport aux autres actions qu'il est possible de choisir lorsque l'environnement est dans l'état  $s$ . À partir de cette équation,  $Q$  peut être décomposée comme étant la somme de  $V(s)$  et  $A(s, a)$ . L'idée derrière le *Dueling Deep Q-Network* développé par Wang et al. en 2016 [134] est de décomposer le Q-réseau en 2 flux :  $V_{\theta}(s)$  qui approxime  $V(s)$  et  $A_{\theta}(s, a)$  qui approxime  $A(s, a)$ . Le Q-réseau et le réseau cible sont dès lors définis par :

$$Q_{\theta}(s_t, a_t) = V_{\theta}(s_t) + (A_{\theta}(s_t, a_t) - \frac{1}{|A|} \sum_{a'} A_{\theta}(s_t, a'))$$

$$Q_{\theta'}(s_t, a_t) = V_{\theta'}(s_t) + (A_{\theta'}(s_t, a_t) - \frac{1}{|A|} \sum_{a'} A_{\theta'}(s_t, a'))$$

avec  $\theta$  les paramètres du Q-réseau et  $\theta'$  les paramètres du réseau cible. La moyenne des approximations des avantages est soustraite de l'approximation de l'avantage  $A(s_t, a_t)$  pour augmenter la stabilité et les performances de l'apprentissage.

### Algorithme final

Le Double DQN et le *Dueling DQN* peuvent être combinés pour obtenir le *Double Dueling Deep Q-Network* (3DQN). Le 3DQN est utilisé dans ce chapitre, car celui-ci offre un apprentissage plus stable et plus performant que DQN de manière générale. Quelques détails sur 3DQN sont explicités ici. Ces détails sont valables pour 3DQN mais aussi pour DQN et tous ses dérivés. Dans ces algorithmes, l'agent n'apprend pas après chaque action qu'il effectue. À la place, une mémoire tampon retient les transitions  $(s_t, a_t, s_{t+1}, r_t, d)$ . Cette transition signifie qu'à l'étape  $t$ , l'état de l'environnement était  $s_t$ , que l'agent a choisi l'action  $a_t$ , qu'il a obtenu une récompense  $r_t$  et que cela a fait passer l'environnement dans l'état  $s_{t+1}$ .  $d$  permet de savoir si  $s_{t+1}$  est un état final ou pas. L'épisode en cours se termine lorsque l'environnement atteint un état final. La mémoire tampon a une taille finie. Si une nouvelle transition doit être stockée quand celle-ci est pleine, la plus ancienne transition est remplacée par la nouvelle. Durant chaque phase d'apprentissage, l'agent crée un paquet de transitions choisies aléatoirement et calcule la perte  $L(\theta)$  de chacune d'elles. La moyenne

des pertes est ensuite utilisée pour modifier le Q-réseau par descente de gradient. Enfin, une politique  $\epsilon$ -greedy est utilisée pour l'exploration.  $\epsilon$  est un hyperparamètre dans l'intervalle  $[0, 1]$ . Quand l'agent doit choisir une action, les Q-valeurs sont calculées à l'aide du Q-réseau. Un nombre aléatoire  $r \in [0, 1]$  est alors généré. Si  $r < \epsilon$ , l'action est choisie aléatoirement. Sinon, c'est celle dont la Q-valeur est la plus élevée qui est choisie.  $\epsilon$  a une valeur haute au début de l'entraînement qui diminue durant celui-ci. Cela permet d'avoir un agent qui explore au départ et exploite ses connaissances à la fin de l'entraînement. La Section 4.5.2 donne plus de détails sur l'implémentation de 3DQN faite pour ce chapitre.

## 4.2.2 Proximal Policy Optimization

En 2017, Schulman et al. [135] ont développé un algorithme nommé *Proximal Policy Optimization* (PPO). PPO étant un algorithme de type *actor-critic*, la politique  $\pi$  est directement approximée par un réseau de neurones noté  $\pi_\theta$ . Un problème connu des algorithmes de type *actor-critic* est que de légères modifications de la politique peuvent entraîner d'importants changements de comportement. Ces changements de comportement peuvent annihiler dans certains cas les performances de l'agent. PPO a été développée pour pallier ce problème de manière moins coûteuse en calcul que la solution utilisée jusqu'alors. De manière générale, PPO apprend plus rapidement de meilleures politiques que les algorithmes développés avant lui. C'est pour cette raison qu'il est utilisé dans ce chapitre. Lorsque PPO est utilisé, la fonction objective de  $\pi_\theta$  est définie par :

$$L_{ac}(\theta) = \min\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} A^{\pi_{\theta_{old}}(s_t, a_t)}, g(\epsilon, A^{\pi_{\theta_{old}}(s_t, a_t)})\right)$$

$$g(\epsilon, A) = \begin{cases} (1 + \epsilon)A & \text{si } A \geq 0 \\ (1 - \epsilon)A & \text{sinon} \end{cases}$$

avec  $\epsilon$  un hyperparamètre de petite valeur,  $\pi_{\theta_{old}}$  une ancienne politique et  $A^{\pi_{\theta_{old}}}$  une méthode quelconque d'estimation de l'avantage.  $\pi_{\theta_{old}}(a_t|s_t)$  est la probabilité renvoyée par  $\pi_\theta$  de choisir l'action  $a_t$  lorsque l'environnement est dans l'état  $s_t$ .  $\pi_{\theta_{old}}(a_t|s_t)$  est la probabilité renvoyée par l'ancienne politique. Contrairement à DQN et à ses dérivés, PPO fonctionne avec des trajectoires  $\tau$ . Une trajectoire est une suite d'actions ayant été faites par l'agent. Lors de l'entraînement,  $n_{tr}$  trajectoires sont créées en utilisant  $\pi_\theta$  et stockées.  $k$  phases d'apprentissages sont ensuite effectuées sur ces trajectoires. Lors de la première phase d'apprentissage,  $\pi_\theta$  et  $\pi_{\theta_{old}}$  sont identiques.  $L_{ac}$  est calculée pour chaque action des trajectoires, et la moyenne des valeurs permet de modifier  $\pi_\theta$  grâce à une montée de gradient.  $\frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)}$  mesure la distance entre la politique actuelle et l'ancienne politique.  $A^{\pi_{\theta_{old}}(s_t, a_t)}$  permet d'évaluer la pertinence de  $a_t$ . Les modifications de  $\pi_\theta$  sont donc proportionnelles à la distance entre la politique actuelle et la politique ayant généré les trajectoires, ainsi qu'à la qualité des actions choisies. Le terme  $g(\epsilon, A^{\pi_{\theta_{old}}(s_t, a_t)})$  permet de s'assurer que les modifications de la politique ne sont pas trop brusques. Comme la fonction d'avantage, l'estimation de l'avantage  $A^{\pi_{\theta_{old}}}$  a besoin d'une fonction de valeur pour être calculée. Celle-ci est approximée par un deuxième réseau de neurones noté  $V_\phi$ .  $V_\phi$  a comme objectif de minimiser la fonction :

$$L_{cr}(\phi) = (V_\phi(s_t) - R_t)^2$$

avec  $R_t = \sum_t^T \gamma^t r_t$  la somme pondérée des futures récompenses de la trajectoire. Cette somme est ici calculable, car l'entièreté des trajectoires sont connues. Lors des phases d'entraînements,  $L_{cr}$  est calculée pour chaque étape des trajectoires, et la moyenne est utilisée pour modifier  $V_\phi$  par descente de gradient. L'estimation de l'avantage utilisée dans ce chapitre est la *Generalized Advantage Estimation*, développée par Schulman et al. en 2015 [136].

### 4.3 Approche DRL

Quel que soit l'algorithme utilisé, un agent DRL a besoin de certaines composantes pour approximer une politique optimale. Tout d'abord, un environnement ainsi que les états de celui-ci qui seront envoyés à l'agent doivent être définis. L'agent doit ensuite pouvoir effectuer des actions pour modifier cet environnement, et ces actions doivent être récompensées. Dans cette section, le type d'environnement dans lequel l'agent évolue est d'abord détaillé. Ensuite, les états décrivant cet environnement sont expliqués. Enfin, les actions que l'agent pourra effectuer, ainsi que la fonction de récompense permettant d'évaluer celles-ci sont définies.

#### 4.3.1 Environnement

Comme expliqué en Section 4.1.3, le feu de circulation proposé possède des phases vertes pour cyclistes. L'environnement dans lequel l'agent va évoluer est donc une intersection, constituée de plusieurs axes se croisant, et contrôlée par un feu de circulation. Par souci de simplicité, chaque axe est bidirectionnel et possède une voie cyclable par sens. Si toutes les voitures d'un axe sont au vert en même temps (c.-à-d. qu'il n'y a pas de phases vertes spécifiques pour les voitures qui tournent), le feu d'une intersection de  $n$  axes aura  $2n$  phases vertes différentes. Par exemple, l'ensemble des phases vertes d'un feu contrôlant une intersection constituée de 2 axes sera  $G = \{g_{voitures}^{ax_1}, g_{velos}^{ax_1}, g_{voitures}^{ax_2}, g_{velos}^{ax_2}\}$  avec  $g_t^{ax_x}$  signifiant que les véhicules de type  $t$  sur l'axe  $ax_x$  ont le feu vert. Un exemple graphique d'une intersection constituée de 2 axes est montré en Figure 4.1,  $ax_1$  étant l'axe Nord-Sud (NS) et  $ax_2$  l'axe Est-Ouest (EO).

#### 4.3.2 États

Les états envoyés à l'agent doivent condenser les informations pertinentes de l'environnement. Deux types d'informations relatives aux véhicules sont retenus. Premièrement, l'agent doit savoir combien de véhicules sont présents à l'intersection et sur quelles voies ils se situent. La position des véhicules est donc importante. Ensuite, l'agent doit pouvoir faire la différence entre les véhicules qui attendent, ceux qui sont en train de traverser l'intersection et ceux qui sont entre le début de leur voie et l'éventuelle file d'attente sur celle-ci. En effet, plus la file d'attente d'une voie est grande, plus il est nécessaire de lui octroyer une phase verte. La vitesse des véhicules est donc elle aussi importante.

Inspirés par les travaux de Liang et al. [137], les états sont 2 matrices de mêmes dimensions. Elles sont nommées respectivement matrice de position et matrice de vitesse. L'environnement est divisé en cellules de 5 mètres de long. Liang et al. ont utilisé toutes les cellules de l'intersection. Il a été considéré que seules les cellules appartenant à une des voies arrivant à l'intersection sont utiles à la prise de décisions. La dimension des matrices est donc  $N \times P$  avec  $N$  le nombre de voies arrivant à l'intersection et  $P$  le nombre de cellules que chaque voie contient. La matrice de

position contient le nombre de véhicules qui sont dans chaque case et la matrice de vitesse la moyenne des vitesses de ceux-ci. Ces matrices peuvent être reconstruites grâce à des caméras pointées sur les voies arrivant à l'intersection couplées à des méthodes récentes d'estimation de position et de vitesse de véhicules [138]. Un exemple graphique de la construction d'une matrice de position est montré en Figure 4.1.

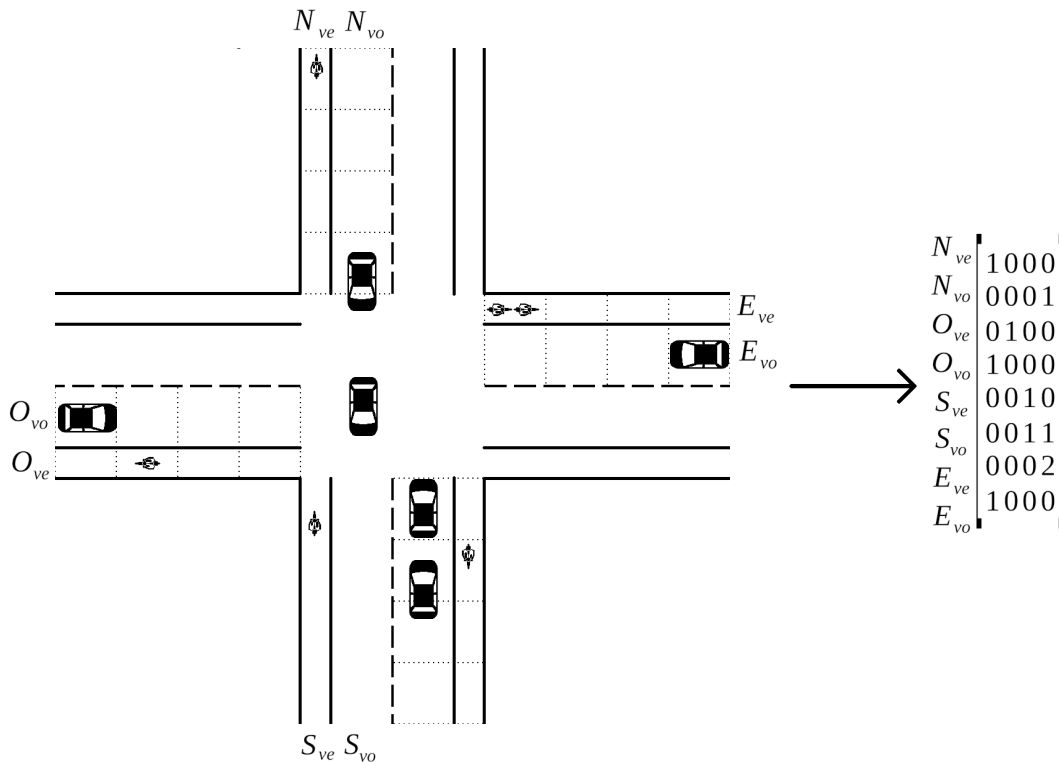


FIGURE 4.1 – Diagramme de la construction d'une matrice de position à partir du schéma d'une intersection.

### 4.3.3 Actions

Les actions pouvant être effectuées par l'agent doivent modifier le comportement du feu de circulation contrôlant l'intersection. Comme d'autres auteurs [125, 127, 128], l'ensemble des phases vertes  $G$  est utilisé comme ensemble d'actions possibles (voir Section 4.3.1). Une phase verte a un temps minimum de 10s. Lorsqu'une phase verte est active depuis 10s, l'agent choisit la future phase verte. Si la phase choisie est différente de la phase actuelle, une phase orange de 4s impliquant toutes les voies au vert jusqu'à la décision démarre. Après cette phase orange, ou directement lors de la décision si la phase choisie est la même que la phase actuelle, la phase sélectionnée par l'agent est activée pour une durée de 10s. Mettre en place des phases vertes qui durent 10s minimum permet de ne pas effectuer de changements de phases trop brutaux pouvant surprendre les véhicules. Cela améliore aussi la stabilité de l'apprentissage.

### 4.3.4 Récompenses

Genders et al. [125] font partie des auteurs ayant utilisé  $G$  comme ensemble d'actions possibles et ont donc développé une fonction de récompenses fonctionnant avec celui-ci. Les récompenses qu'ils ont utilisé sont adaptées et expliquées ci-dessous.

## Fonction de récompenses

La fonction de récompenses est définie comme :

$$r_t = -(w_{velos} + w_{voitures})^2$$

avec  $w_{velos}$  et  $w_{voitures}$  respectivement le nombre de vélos en attente et le nombre de voitures en attente. Un véhicule est considéré en attente lorsque sa vitesse est inférieure à 0.5 km/h. Les récompenses sont donc nulles ou négatives, et l'agent doit minimiser le nombre de véhicules en attente pour maximiser la somme de celles-ci. La somme de  $w_{velos}$  et  $w_{voitures}$  est mise au carré pour faciliter l'entraînement. En effet, la différence entre 2 décisions menant à un nombre important de véhicules attendant sera plus grande, car celui-ci est mis au carré. Cela permet à l'agent de mieux discerner quelle décision est la meilleure, notamment au début de l'entraînement.

## Facteur d'étalement

Les valeurs de la fonction de récompenses pouvant être largement négatives, elles peuvent compromettre la convergence des apprentissages. Genders et al. [125] ont résolu le problème en divisant les récompenses par  $r_{max}$  la plus grande récompense calculée. Dans le cadre de ce travail, cette solution permet une convergence de l'agent mais pas la création de politiques performantes.  $r_{mean}$ , la moyenne de toutes les récompenses calculées, est utilisée à la place. Pendant les entraînements, toutes les récompenses calculées ainsi que le nombre d'actions réalisées par l'agent entraîné sont stockées.  $r_{mean}$  est actualisée à la fin de chaque épisode (qui sont détaillés en Section 4.5). Utiliser  $r_{mean}$  comme facteur d'étalement incite l'agent à effectuer des actions qui sont en moyenne de meilleures qualités que celles qu'il a effectué jusqu'alors pour maximiser la somme des récompenses. Au fur et à mesure qu'un agent s'améliore, la somme des récompenses augmente, le poussant à faire des choix encore plus pertinents pour améliorer la somme de ses récompenses. Cela lui permet de finir l'entraînement avec une politique performante.

## 4.4 Simulation de l'environnement

Une fois les composantes de l'approche DRL conçues, l'environnement dans lequel l'agent va évaluer doit être simulé. Dans cette section, la méthode de simulation de l'environnement (cyclistes, voitures et feux de signalisation) est présentée. La demande de trafic basée sur des comptages de véhicules est ensuite détaillée.

### 4.4.1 Environnement simulé

SUMO (*Simulation of Urban MObility*) est un micro-simulateur de trafic permettant de reproduire la dynamique de l'écoulement routier. SUMO est communément utilisé dans les études s'intéressant aux feux de signalisation en simulation. La Figure 4.2 montre une capture d'écran de l'environnement. On y voit une intersection composée de 2 axes se croisant (NS pour Nord-Sud et EO pour Est-Ouest) et contrôlée par un feu de circulation. Chaque route arrivant à l'intersection mesure 150m de long et est constituée d'une voie pour voitures et d'une voie pour vélos. Le feu de circulation proposé a donc 4 phases vertes  $G = \{\mathcal{G}_{voitures}^{NS}, \mathcal{G}_{velos}^{NS}, \mathcal{G}_{voitures}^{EO}, \mathcal{G}_{velos}^{EO}\}$ .  $\mathcal{G}_{velos}^{NS}$  est activé sur la Figure 4.2. Les véhicules apparaissent à l'extrémité d'une des routes arrivant et ont l'extrémité d'une autre route comme destination. Lorsqu'un véhicule est au vert et veut tourner à gauche, il doit donner la priorité aux véhicules arrivant

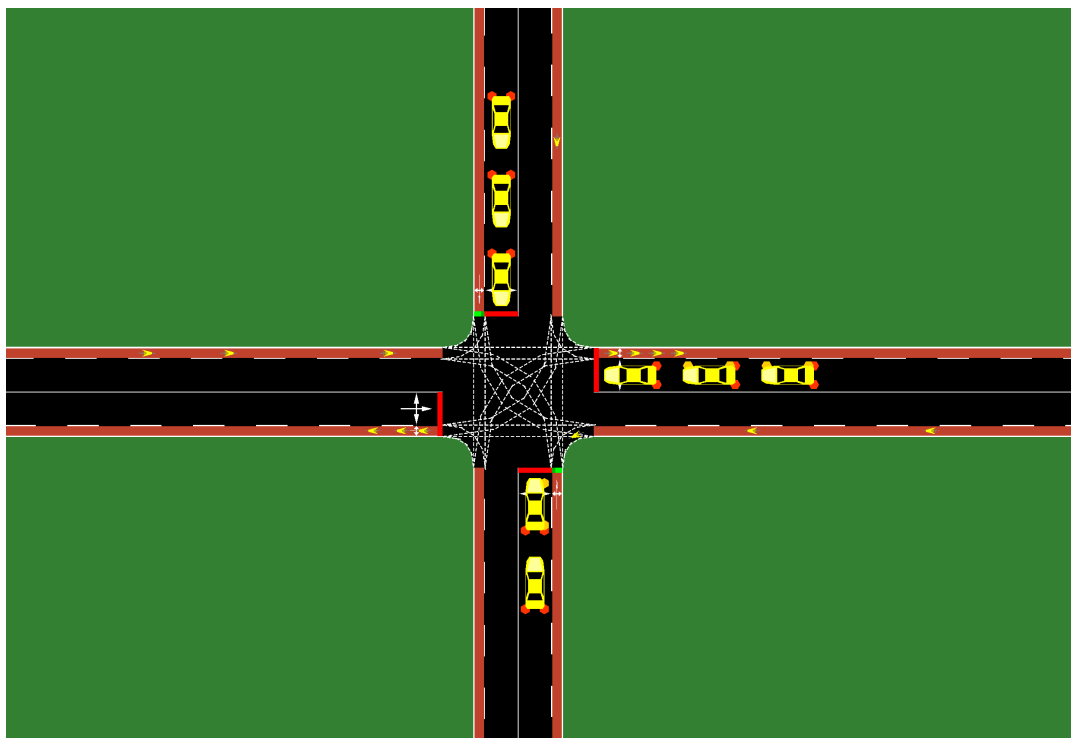


FIGURE 4.2 – Capture d'écran de l'environnement simulé par SUMO.

en face. Cela ajoute du temps d'attente qui n'est pas dû au comportement du feu. De plus, lorsque c'est possible, les véhicules se positionnent au milieu de l'intersection pour laisser les véhicules derrière eux ne désirant pas tourner à gauche les dépasser. SUMO ne permet pas ce comportement, ce qui a pour conséquence de faire attendre tous les véhicules de la voie lorsqu'un véhicule veut tourner à gauche et laisse la priorité. Cela ajoute d'autant plus de temps d'attente à l'échelle de l'intersection qui n'est pas due à la phase verte active. Les véhicules n'ont pas le droit de tourner à gauche pour cette raison. Ceux-ci ont la même probabilité d'avoir comme destination l'une des 2 voies auxquelles ils peuvent accéder sans tourner à gauche.

#### 4.4.2 Données de comptage et demande de trafic

La demande de trafic est conçue en fonction d'observations réelles. La ville de Paris met à disposition les données de ces compteurs automatiques de véhicules sur son site *d'open data*<sup>1</sup>. Différentes agrégations temporelles sont disponibles, allant de l'année jusqu'à l'heure. L'agrégation horaire est utilisée, car c'est la plus précise. Les données de 2 compteurs voitures unidirectionnels ainsi que d'un compteur bidirectionnel de vélos sont collectées. Ces compteurs sont situés sur le boulevard Montparnasse et sont proches les uns des autres (environ 500m). Le boulevard Montparnasse est à double sens, avec 2 voies pour voitures et une voie pour vélo dans chaque direction. Le nombre de voitures comptées est donc divisé par 2, car l'environnement SUMO ne possède qu'une voie pour voiture dans chaque sens. Les données datent du 20 juin 2023, un mardi avec beau temps.

La Figure 4.3 montre la somme des véhicules comptés par heure dans les 2 directions. Des différences entre les distributions de voitures et de vélos sont observables.

1. <https://opendata.paris.fr/pages/home/>

Il n’y a quasiment aucun vélo compté la nuit, un nombre relativement stable de vélos comptés entre 10 :00 et 16 :00 et 2 gros pics à 08 :00 et 18 :00, les heures de trajets domicile-travail. Pour les voitures, le nombre diminue pendant toute la nuit avant d’augmenter aux alentours de 06 :00. Le nombre de voitures comptées atteint ensuite un plateau qui dure jusqu’à 17 :00. Le nombre augmente ensuite légèrement avec un pic à 19 :00 avant une diminution qui se poursuit jusqu’à 00 :00. Ces différences dans les distributions démontrent l’importance d’utiliser des données réelles à l’échelle de la journée. L’approche 3DQN doit être capable d’adapter ses décisions aux changements des 2 types de trafic pour être efficace. À noter que ces 2 distributions sont assez proches des distributions de véhicules comptés à l’échelle de la ville montrées en Annexe A. La seule différence notable est une diminution du nombre de voitures comptées à l’échelle de la ville à 17 :00 qui n’est pas visible sur les données des compteurs de Montparnasse.

La demande de trafic est modélisée par des processus de Poisson. Chaque voie  $v$  arrivant à l’intersection a son propre processus  $p^v$  qui reproduit la génération. L’intensité  $\lambda_{p^v}(t)$  de ce processus à l’instant  $t$  est fixée pour chaque heure pour coller aux données de comptage agrégées :

$$\lambda_{p^v}(t) = \frac{c_{h(t)}^v}{3600}$$

avec  $c_{h(t)}^v$  le nombre de véhicules (voitures ou vélos) comptés pendant l’heure  $h(t)$  sur la voie  $v$ .

Pour résumer, l’environnement simule 2 boulevards de Montparnasse se croisant pendant une journée entière. Ces boulevards n’ont qu’une voie pour voitures. Le trafic est synthétisé en extrapolant les données agrégées à l’échelle de l’heure de compteurs de véhicules existants.

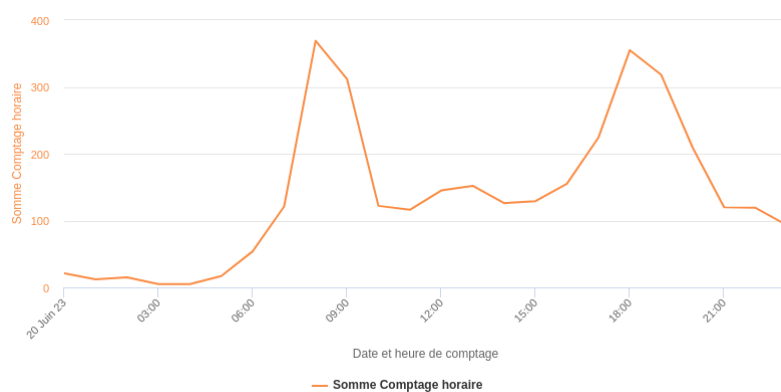
## 4.5 Entraînements

Un agent de chaque type (3DQN et PPO) est entraîné. L’entraînement s’effectue dans les mêmes conditions pour les 2 agents. Cette section présente l’architecture des réseaux de neurones entraînés, ainsi que les hyperparamètres utilisés lors de l’entraînement. Les performances de chaque agent pendant l’entraînement sont ensuite détaillées et expliquées.

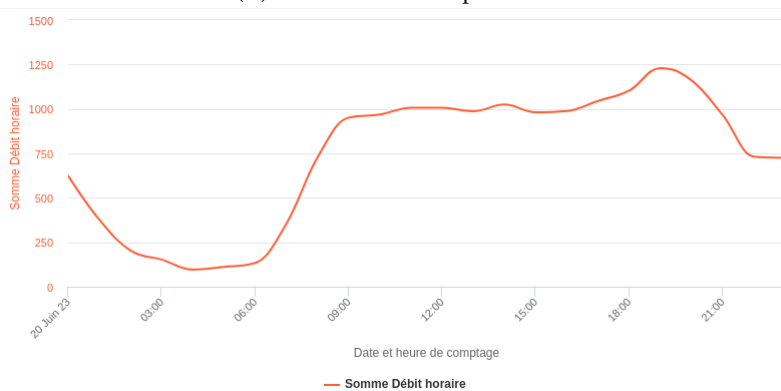
### 4.5.1 Architecture des réseaux de neurones

Un état est constitué de 2 matrices, pouvant être vues comme 2 images condensant les informations utiles de l’environnement. Les réseaux de neurones des agents doivent donc être capables d’identifier les *patterns* présents dans une image pour traiter correctement les états qu’ils reçoivent. Les réseaux de neurones convolutifs [139] ont montré d’impressionnantes performances dans différents domaines, notamment dans la classification d’images, la reconnaissance d’objets ou encore de visages [140]. Les couches convolutives extraient certaines caractéristiques des images. Une couche convolutive est constituée de masques, qui sont des matrices de petites dimensions. Chaque masque a pour but de trouver un type de *pattern* spécifique dans l’image. Une convolution est effectuée entre les différents masques et l’image. Il y a 3 structures de réseaux de neurones différentes. La première structure est celle du Q-réseau  $Q_\theta$  de l’agent 3DQN. Cet agent a aussi besoin d’un réseau cible  $Q_{\theta'}$ ,





(A) Nombre de vélos par heure.



(B) Nombre de voitures par heure.

FIGURE 4.3 – Somme des véhicules comptés par heure dans les 2 directions sur le boulevard Montparnasse le 20 juin 2023.

mais celui-ci a la même structure que le Q-réseau. Les 2 autres structures sont la politique  $\pi_\theta$  et la fonction de valeur  $V_\phi$  de l'agent PPO. Celui-ci a aussi besoin d'une ancienne politique  $\pi_{\theta_{old}}$ , mais celle-ci a la même structure que  $\pi_\theta$ . Tous les réseaux de neurones sont constitués de 2 couches convolutives initiales, ayant 16 filtres de dimension  $2 \times 2$ . Celles-ci sont systématiquement suivies d'une couche de 128 neurones *fully-connected*. Les couches de sorties diffèrent selon les réseaux :

- $Q_\theta$  et  $Q_{\theta'}$  possèdent 2 couches de sorties, respectivement pour la fonction d'avantage et la fonction de valeur
- $\pi_\theta$  et  $\pi_{\theta_{old}}$  possèdent une couche de sortie suivie de la fonction *Softmax* permettant d'obtenir des valeurs de sortie dans l'intervalle  $[0, 1]$  et dont la somme est égale à 1
- $V_\phi$  possèdent une seule couche de sortie

La fonction d'activation ReLU est utilisée après chaque couche, excepté les couches de sortie. La Figure 4.4 résume graphiquement l'architecture du Q-réseau  $Q_\theta$ .

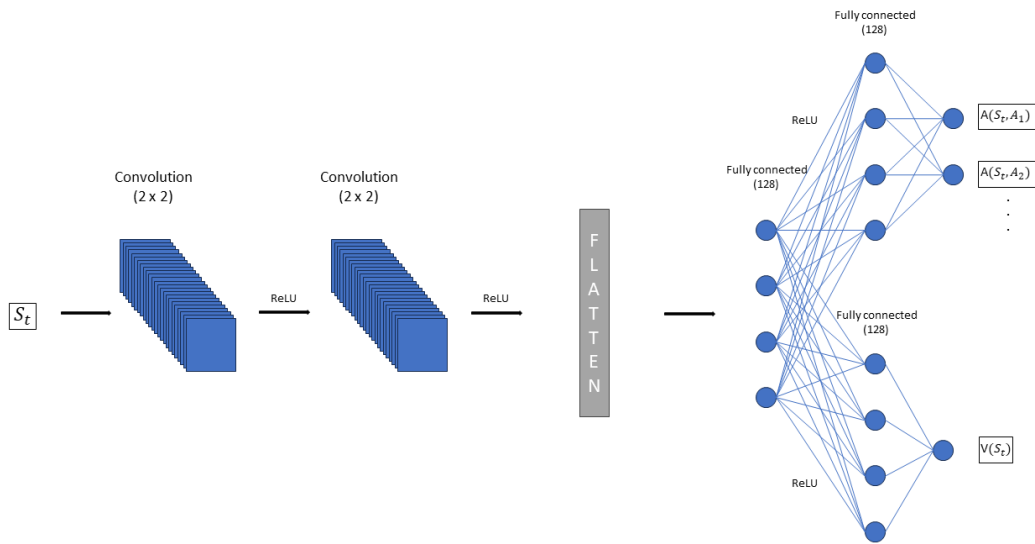


FIGURE 4.4 – Diagramme de la structure du Q-réseau.

## 4.5.2 Hyperparamétrage

Les valeurs et explications des hyperparamètres utilisés dans ce chapitre sont montrées dans le Tableau 4.1. Les agents sont entraînés par épisodes. Pendant chaque épisode, les véhicules peuvent apparaître pendant 6 heures simulées (c.-à-d.  $3600 \times 6 = 21600$  étapes). Un épisode s'arrête lorsque tous les véhicules sont apparus et que plus aucun ne subsiste dans l'environnement. Un véhicule disparaît de l'environnement lorsque celui-ci a atteint sa destination, située au bout d'une route quittant l'intersection. L'entraînement s'arrête lorsque que l'agent a choisi  $f$  actions.

L'agent 3DQN commence par stocker de l'information sans apprendre durant  $pt$  actions. Une phase d'apprentissage se lance ensuite après chaque épisode. La valeur d' $\epsilon$  diminue linéairement chaque fois que l'agent 3DQN choisit une action, et atteint sa valeur finale à la  $f^{eme}$  action. Les paramètres du réseau cible sont mis à jour en étant remplacés par ceux du Q-réseau toutes les  $v$  actions.

L'agent PPO s'entraîne sur des trajectoires. Une trajectoire correspond à un épisode entier.  $k$  phases d'apprentissage sont donc effectuées tous les  $n_{tr}$  épisodes dès le début de l'entraînement.

Algorithme	Paramètre	Valeur	Description
3DQN (voir Section 4.2.1)	$f$	1500000	Nombre d'actions durant l'entraînement.
	Taille de la mémoire tampon	25000	Taille maximum de la mémoire tampon.
	Taille des paquets	128	Taille des paquets de transitions utilisés pour l'apprentissage.
	$\epsilon$ initial	1	Valeur d' $\epsilon$ au début de l'entraînement.
	$\epsilon$ final	0.01	Valeur d' $\epsilon$ à la fin de l'entraînement.
	$pt$	10000	Nombre d'actions effectuées par l'agent sans avant la première phase d'apprentissage.
	$\gamma$	0.99	<i>Discount rate</i> pondérant l'importance des futures récompense.
	$v$	7500	Nombre d'actions à effectuer entre 2 mises à jour du réseau cible.
	$\alpha$	0.001	Taux d'apprentissage de $Q_\theta$ .
PPO (voir Section 4.2.2)	$f$	1500000	Nombre d'actions durant l'entraînement.
	$k$	4	Nombre de phases d'apprentissage effectuées à la suite.
	$n_{tr}$	5	Nombre de trajectoires à stocker entre 2 apprentissages.
	$\gamma$	0.99	<i>Discount rate</i> pondérant l'importance des futures récompense.
	$\alpha$	0.0001	Taux d'apprentissages de $\pi_\theta$ et $V_\phi$ .

TABLE 4.1 – Tableau des hyperparamètres.

### 4.5.3 Performances

Les données de chaque épisode effectué durant les entraînements sont sauvegardées. Trois variables sont affichées pour mesurer l'amélioration des politiques au cours de ceux-ci : le temps d'attente moyen des vélos, le temps d'attente des voitures et le temps moyen des véhicules.

#### Agent 3DQN

La Figure 4.5 montre l'évolution des différentes variables pendant l'entraînement de l'agent 3DQN. Il faut un peu moins de 1000 épisodes pour atteindre la  $f^{eme}$  action. Le temps d'attente moyen des vélos est assez bas dès le début de l'entraînement (Figure 4.5a). Celui-ci commence par légèrement augmenter jusqu'à l'épisode 150 puis diminue progressivement jusqu'à l'épisode final. On observe des périodes de stagnation et des périodes de légère baisse, signe que l'agent a du mal à l'améliorer. Le temps d'attente moyen des voitures commence lui bien plus haut (Figure 4.5b), entre 50 et 150 étapes contre 45/50 étapes pour les vélos. On observe encore une fois une

augmentation jusqu'à l'épisode 150. Le temps moyen maximum des voitures est 10 fois plus grand que celui des vélos, avec 800 étapes contre 80. Une nette diminution est ensuite visible. Cette diminution continue jusqu'à l'épisode 500 puis le temps d'attente moyen des voitures est assez stable jusqu'à la fin de l'entraînement, même si celui-ci continue à légèrement réduire. L'évolution du temps d'attente moyen des véhicules (Figure 4.5c) ressemble beaucoup à celle du temps d'attente moyen des voitures. Des différences sont tout de même visibles après l'épisode 500. Le temps d'attente moyen des véhicules est plus variable que celui des voitures, et semble descendre plus bas.

Les différences entre vélos et voitures s'expliquent par la manière dont SUMO les simule ainsi que par leurs trafics respectifs. Premièrement, SUMO autorise les cyclistes à être plus proches les uns des autres lorsqu'ils sont à l'arrêt (0.5m pour les vélos contre 2.5m pour les voitures). De plus, la longueur des vélos est moins importante que celle des voitures, avec 1.6m contre 5m. Les files de vélos sont donc moins longues que celles des voitures. Même si l'accélération des voitures est plus importante ( $2.6m/s^2$  contre  $1.2m/s^2$ ), les vélos sont plus proches du feu lorsque celui-ci passe au vert. Cette différence de point de départ est de plus en plus grande à mesure que le nombre de véhicules dans la file augmente.

Deuxièmement, le trafic de voitures est plus important que le trafic vélos, même après la division par 2 du nombre de voitures (expliquée en Section 4.4.2). Au début de l'entraînement,  $\epsilon$  vaut 1. Toutes les phases ont donc la même probabilité d'être choisies par l'agent. Les 10 secondes minimum des phases vertes semblent être suffisantes pour vider les voies vélos, mais pas les voies voitures pour les raisons évoquées précédemment. Certaines voitures doivent donc attendre plusieurs phases vertes pour franchir l'intersection, augmentant significativement le temps d'attente moyen de celles-ci. Après 150 épisodes, les Q-valeurs des phases vertes pour voitures commencent à être suffisamment supérieures à celles des phases vertes pour voies vélos pour que l'agent sélectionne les premières plus souvent. Cela a pour conséquence de faire chuter le temps d'attente moyen des voitures. Le temps d'attente moyen des vélos diminue lui aussi. L'agent semble aussi commencer à sélectionner l'axe ayant le plus de vélos en attente lorsqu'il décide de donner le vert à ce type de véhicules, expliquant la diminution de leur temps d'attente moyen. À mesure que l'entraînement continue, les Q-valeurs s'affinent et la valeur de  $\epsilon$  diminue, permettant à l'agent de diminuer le temps d'attente moyen des 2 types de véhicules.

### Agent PPO

La Figure 4.6 montre l'évolution des différentes variables pendant l'entraînement de l'agent PPO. L'évolution de l'apprentissage entre l'agent 3DQN et l'agent PPO est similaire. Cela n'est pas surprenant car les 2 algorithmes ont le même objectif, et doivent l'atteindre dans les mêmes conditions en utilisant les mêmes représentations de leur environnement. Des différences sont tout de même notables. Le temps d'attente moyen des vélos commence aussi par augmenter, mais de manière plus significative (Figure 4.6a). Celui-ci atteint même les 100 étapes lors de plusieurs épisodes, là où l'agent 3DQN ne dépasse jamais les 80 étapes. L'augmentation du temps d'attente moyen des vélos arrête d'augmenter aux alentours de l'épisode 200, puis diminue doucement jusqu'à retrouver des valeurs proches de celles du début de l'entraînement. Pour les voitures en revanche, aucune augmentation du temps d'attente moyen n'est observable (Figure 4.6b). Celui-ci commence entre 50 et 250 et ne fait que diminuer au cours de l'entraînement. Cette diminution est progressive,

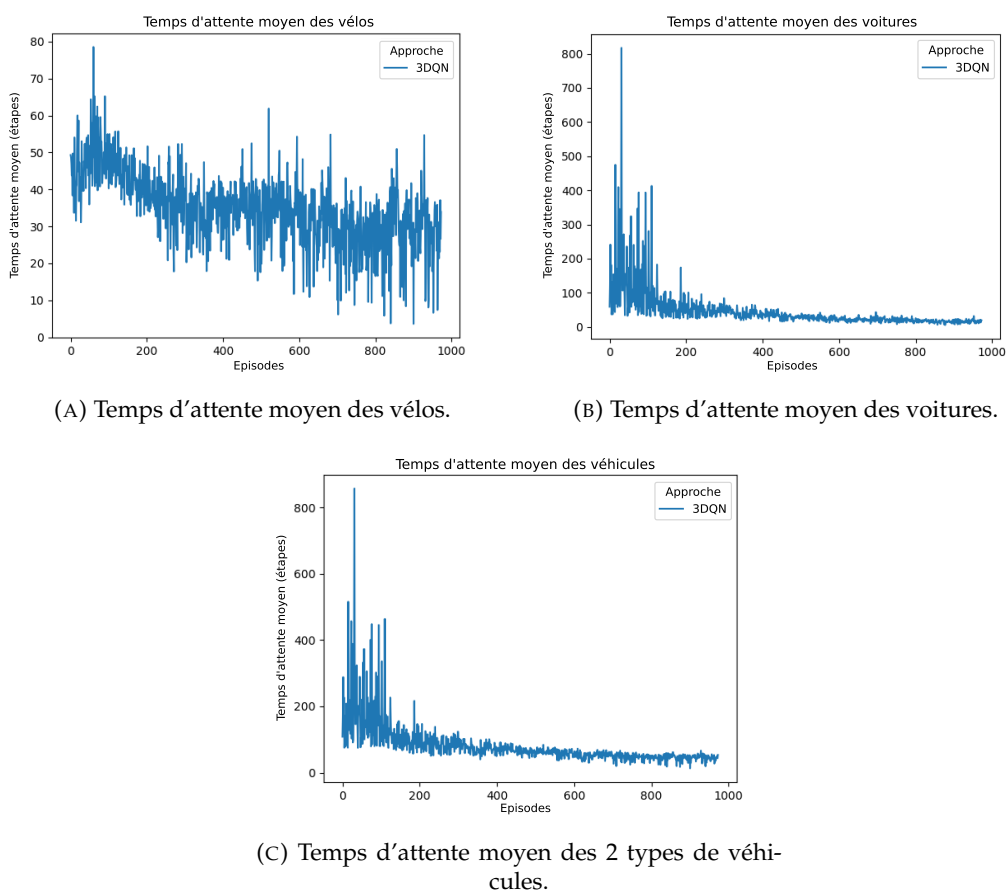


FIGURE 4.5 – Évolution du temps d’attente moyen des véhicules pendant l’entraînement de l’agent 3DQN.

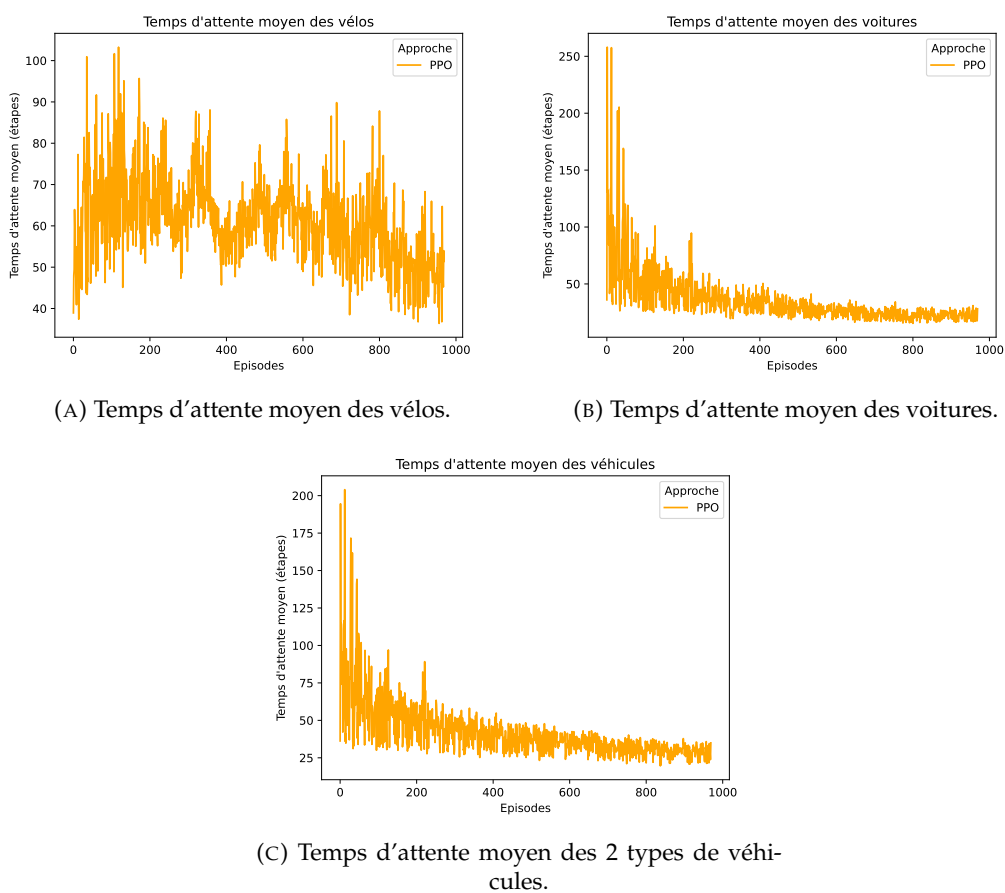


FIGURE 4.6 – Évolution du temps d'attente moyen des véhicules pendant l'entraînement de l'agent PPO.

là où l'agent 3DQN voit le temps d'attente moyen des voitures diminuer brusquement aux alentours de l'épisode 150. Le temps d'attente moyen des véhicules suit là encore l'évolution de celui des voitures (Figure 4.6c). Il est donc là encore plus progressif que lors de l'entraînement de l'agent 3DQN, et atteint des valeurs moins élevées.

Les différences d'apprentissage entre les 2 agents sont dues aux fonctionnements de 3DQN et de PPO qui diffèrent. 3DQN commence par une phase d'exploration. Lors de celle-ci, les phases sont choisies aléatoirement, et seules les Q-valeurs des actions choisies sont modifiées. À mesure que l'entraînement avance, les phases sont moins souvent choisies aléatoirement. Cependant, trop peu de Q-valeurs ont été modifiées, et les phases que l'agent choisit non aléatoirement ne sont pas encore pertinentes. Celui-ci peut par exemple trop privilégier une voie, faisant en moyenne plus attendre les véhicules que lorsque le choix est complètement aléatoire. Ce n'est que lorsque assez de Q-valeurs ont été ajustées et qu' $\epsilon$  est assez bas que les performances commencent à s'améliorer. Lors de l'entraînement affiché, cela se produit à l'épisode 150 environ. Le principe de PPO est en revanche de s'éloigner de sa politique actuelle dans la bonne direction par petits pas. La pire politique est donc celle de départ, lorsque la probabilité de choisir chaque action est la même. Cela est visible sur la Figure 4.6c, avec le temps d'attente moyen des véhicules qui ne fait que diminuer en moyenne. Pour arriver à ce résultat, l'agent PPO privilégie les voies voitures, car celles-ci sont plus susceptibles d'augmenter le temps d'attente des véhicules. Cela a pour conséquence de dégrader le temps d'attente moyen des vélos. Ce n'est que lorsque le temps d'attente moyen des voitures est assez bas que l'agent PPO commence à s'améliorer en diminuant le temps d'attente moyen des vélos.

## 4.6 Résultats

Après plusieurs entraînements, il s'avère que les politiques de l'agent 3DQN sont systématiquement meilleures que celles de l'agent PPO, et ce quel que soit l'état du trafic. Seuls les résultats de l'approche 3DQN sont donc décrits. L'approche 3DQN désigne l'utilisation de l'agent 3DQN entraîné pour contrôler le feu de circulation. Pour évaluer l'approche 3DQN, les temps d'attente qu'elle génère sont comparés aux temps d'attente générés par 3 autres méthodes de contrôle de feu. Celles-ci sont expliquées ci-après. Le trafic n'est jamais saturé à la sortie de l'intersection avec les demandes de trafic utilisées. La performance d'une solution est donc le temps passé par les véhicules avant de franchir l'intersection au feu vert. Après l'entraînement de l'agent, des journées entières ( $3600 \times 24 = 86400$  étapes) sont simulées et le temps d'attente des véhicules est calculé pour chaque heure. Les temps d'attente générés par l'approche 3DQN sont comparés à ceux générés par 3 approches comparées expliquées dans cette section. Les résultats sont ensuite en 2 parties. La première partie est à l'échelle horaire pour une simulation d'une journée. La deuxième est à l'échelle de la journée, avec des variations de la demande de trafic vélos.

### 4.6.1 Approches de comparaison

L'approche 3DQN est comparée à 3 méthodes de contrôle de feu décrites ci-dessous :

- *statique non sécurisée* : La première approche de comparaison, appelée *non sécurisée* dans la suite, sert de base de référence pour quantifier l'augmentation du temps d'attente de la sécurisation du feu de circulation pour les cyclistes. Le feu de circulation a un cycle statique classique, avec une phase verte de

40s et une phase orange de 4s par axe pour une durée totale de 88s. Les vélos et les voitures traversent l'intersection en même temps.

- *statique sécurisée* : Cette approche est une approche naïve. Elle consiste simplement à ajouter des phases vertes pour les cyclistes, et à les empêcher de traverser en même temps que les voitures. Le feu a donc un cycle statique avec 4 phases vertes de 40s dans l'environnement simulé. Cette approche permet de montrer l'explosion du temps d'attente si la sécurisation du feu pour les cyclistes est implémentée naïvement.
- *actuated* : L'approche *statique sécurisée* sert à démontrer l'importance d'une approche dynamique de changement de phase lors de l'ajout de phases vertes pour cyclistes. Comparer l'approche 3DQN qui est hautement dynamique uniquement à une approche statique ne serait pas acceptable. C'est pourquoi l'approche *actuated* est utilisée, avec un feu sécurisé pour les cyclistes. *actuated* est une méthode de contrôle de feu dynamique communément mise en œuvre en Allemagne. [141]. Un feu de circulation en mode *actuated* possède un détecteur de véhicules sur chacune des voies arrivant à l'intersection, à environ 50m de celle-ci. Le feu a un paramètre *duree*, et chaque phase verte a une durée minimum *minDur* et une durée maximum *maxDur*. Lorsqu'une phase verte débute, le feu attend *minDur* secondes. Si un détecteur sur une voie au vert est activé lorsque les *minDur* secondes sont écoulées, un compte à rebours de *duree* secondes est lancé. Une fois que le compte à rebours atteint 0, le feu passe à la phase suivante de son cycle. Si un détecteur sur une voie au vert détecte un véhicule avant que le compte à rebours n'atteigne 0, celui-ci est réinitialisé à *duree* secondes. Si une phase verte est active depuis *maxDur* secondes, le feu de circulation passe à la phase suivante quel que soit l'état du compte à rebours. Dans l'implémentation d'*actuated* utilisée dans ce chapitre, les 4 phases vertes ont une durée minimum *minDur* de 10s, une durée maximum *maxDur* de 40s et le paramètre *duree* est de 5s. *actuated* est communément utilisé pour tester les performances d'approches DRL pour le contrôle de feux de signalisation [128, 125, 123].

#### 4.6.2 Résultats horaires

Pour les résultats horaires, une simulation initiale est lancée avec l'approche 3DQN et la demande de trafic basée sur les données de comptage. Les moments et endroits de génération de tous les véhicules sont sauvegardés. Trois autres simulations sont ensuite lancées avec une approche de comparaison différente. Cela permet d'effectuer la comparaison de performances dans des conditions strictement identiques.

La Figure 4.7 montre les résultats horaires pour une journée. L'axe des abscisses représente l'heure de la journée simulée, allant de 0 à 23. L'axe des ordonnées montre le nombre de véhicules pour la Figure 4.7a et le temps d'attente moyen des véhicules pour la Figure 4.7b. Les courbes de distributions des véhicules ont la même forme que celles montrées en Figure 4.3 sans être parfaitement identiques. Cela est dû à l'aléatoire créé par les processus de Poisson.

L'approche *non sécurisée* est sans surprise l'approche générant le moins de temps d'attente. Contrairement aux autres courbes, l'approche *non sécurisée* est plutôt plate et ne change que très peu selon le trafic. Celui-ci n'est en effet pas assez fort pour saturer les voies arrivant à l'intersection avec cette approche. Tous les véhicules attendant pendant une phase rouge peuvent traverser l'intersection lors de la première phase verte accordée à leurs voies. Ajouter des phases vertes dédiées aux cyclistes



augmente la congestion sur chaque voie, car les voies de voitures et de vélos sur le même axe se vident alors successivement.

La pire approche est logiquement la *statique sécurisée*. Doubler le temps du cycle du feu de circulation fait exploser le temps d'attente des véhicules, car chaque voie doit attendre plus longtemps entre 2 phases vertes. Réduire le temps des phases vertes à 20s augmente encore plus le temps d'attente, avec des voies beaucoup plus saturées. Celles-ci n'ont en effet plus le temps de se vider entièrement pendant une phase verte, obligeant certains véhicules à attendre plusieurs cycles complets avant de traverser l'intersection.

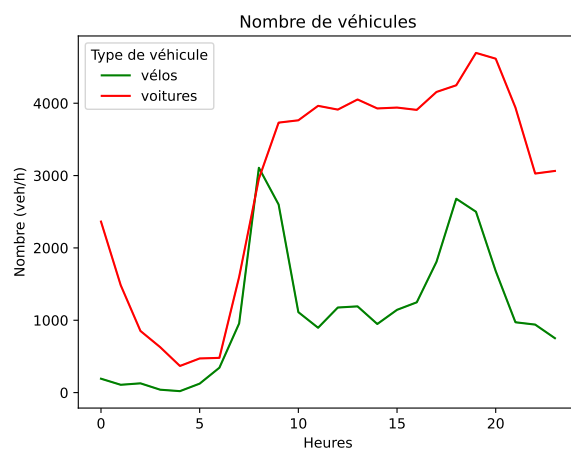
*actuated* fait bien mieux que l'approche *statique sécurisée* quelle que soit l'heure de la journée. Elle gère les 2 doubles pics de voitures et vélos à 08 :00 et 19 :00 beaucoup plus facilement, avec seulement une légère augmentation de temps d'attente. Les durées des phases vertes étant adaptées selon le trafic, les voies sont peu saturées. De plus, le temps perdu à laisser une voie vide au vert est grandement réduit.

L'approche 3DQN fait encore mieux que l'approche *actuated*, avec un temps d'attente moyen inférieur à quasiment toutes les heures. Le gain de temps d'attente est plus grand durant les heures à faible trafic. C'est logique, car *actuated* est moins efficace lorsque les voies sont vides ou presque. En effet, quelles que soient les conditions de trafic, le feu de circulation en mode *actuated* effectue le même cycle, et doit attendre 10s minimum pour chaque phase verte. 3DQN est capable de détecter les véhicules et de choisir la phase verte la plus adaptée en conséquence. Mais à mesure que les voies se remplissent, l'ordre est de moins en moins important au profit du moment de changement de phases. L'approche 3DQN étant capable de gérer les 2, elle reste meilleure qu'*actuated* durant les heures de fort trafic, mais l'écart entre les 2 approches diminue. Les performances de l'approche 3DQN sont les plus proches de celles de l'approche *non sécurisée*. Il est cependant important de noter que ses performances sont meilleures lorsque le trafic est faible.

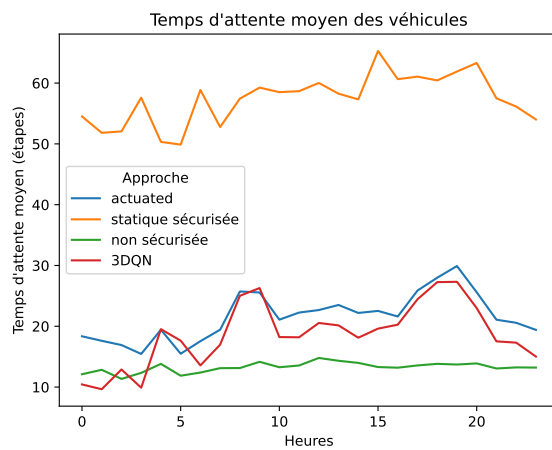
À l'échelle de la journée, ajouter des phases vertes spécifiques aux cyclistes multiplie le temps d'attente moyen des véhicules par 4.35 en utilisant l'approche naïve *statique sécurisée*, par 1.69 en utilisant l'approche *actuated* et par 1.55 en utilisant l'approche 3DQN. Travailler avec un trafic plus important permettrait certainement d'atteindre le taux de saturation de l'approche *non sécurisée*, diminuant ainsi ses performances. Cependant, cette saturation serait d'autant plus présente lors de l'ajout des phases vertes pour cyclistes. Cela dégraderait les performances des autres approches, et pourrait même compromettre l'apprentissage de l'agent 3DQN.

### 4.6.3 Robustesse aux changements du trafic vélos

L'approche 3DQN limite l'augmentation du temps d'attente moyen des véhicules lors d'une journée avec un trafic similaire à celui vu par l'agent pendant son entraînement. Cependant, le choix d'utiliser son vélo ou non est fortement corrélé avec la météo. La demande de trafic étant basée sur des données récupérées pendant une journée ensoleillée, le nombre de cyclistes serait sûrement moins important un jour de pluie. À l'inverse, le but de sécuriser un feu de circulation est de permettre aux cyclistes d'être plus en sécurité. Le déploiement d'une telle infrastructure pourrait attirer des cyclistes, augmentant ainsi le trafic vélo. La robustesse de l'approche 3DQN aux changements de trafic vélo apparaît dès lors comme un point important à vérifier. Un coefficient multiplicateur allant de 0.5 à 1.5 avec un pas de 0.1 est défini. Le nombre de vélos comptés chaque heure est multiplié par ce coefficient pour faire varier linéairement le trafic vélo de 50% à 150% du trafic observé par le compteur de Paris. Cinq simulations initiales (c.-à-d. avec le feu contrôlé avec l'approche 3DQN)



(A) Nombre de véhicules par heure.



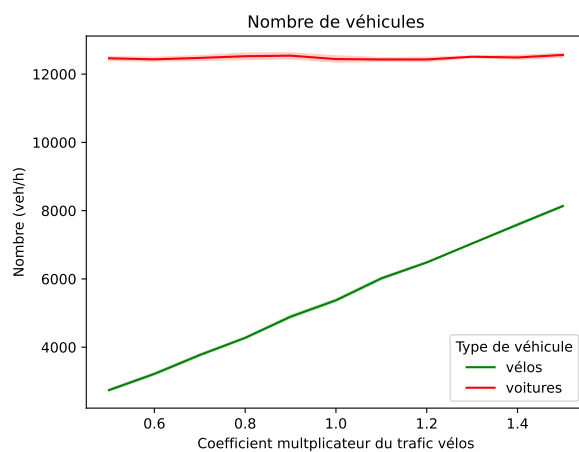
(B) Temps d'attente moyen par heure.

FIGURE 4.7 – Résultats d'une simulation d'un jour.

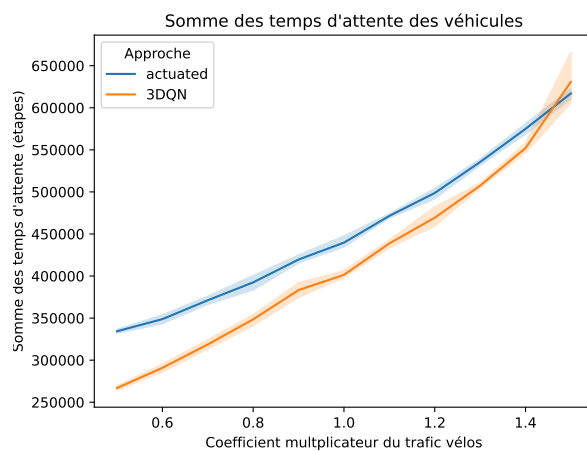
sont lancées pour chaque demande de trafic vélos. Une simulation de comparaison est ensuite exécutée par simulation initiale avec *actuated*. *actuated* est choisie pour comparaison car c'est l'approche sécurisée montrant les meilleures performances après 3DQN. Les heures de la nuit n'étant pas les plus intéressantes à cause d'une intensité trafic trop peu élevée, les résultats montrés ne concernent que les heures entre 6h et 20h.

La Figure 4.8 montre le nombre de véhicules (4.8a) et la somme des temps d'attente de tous les véhicules (4.8b) pour chaque coefficient multiplicateur. Logiquement, le nombre de voitures par simulation est stable et le nombre de vélos augmente linéairement. La somme des temps d'attente générés par 3DQN commence plus bas que celle des temps d'attente générés par *actuated*. Les 2 augmentent progressivement, et se rejoignent lorsque le coefficient atteint 1.5. Lorsque le coefficient vaut 1.5, 3DQN fait davantage attendre les véhicules qu'*actuated*. Cela est consistant avec les résultats horaires. Moins il y a de véhicules à l'intersection, plus la probabilité que l'approche *actuated* laisse une voie vide au vert est importante à cause de son cycle statique et de son temps de phase verte minimum. Cela favorise 3DQN qui est plus dynamique. Ces résultats montrent tout de même que 3DQN est capable de s'adapter à une baisse de trafic vélo. Celle-ci ne semble pas avoir d'impact significatif sur les prises de décisions de l'agent. Cette capacité d'adaptation est sûrement due aux variétés de demande de trafic auquel l'agent doit faire face pendant son entraînement. La demande de trafic varie en effet beaucoup durant les journées simulées, avec des heures de la nuit au trafic très faible. Les performances de 3DQN sont aussi assez stables lorsque le trafic vélo augmente. La somme des temps d'attente augmente logiquement, car plus de vélos à l'intersection signifie un plus grand besoin d'activation des phases vertes spécifiques aux voies vélos. Cela impacte le temps d'attente de tous les véhicules à l'intersection, qui doivent davantage attendre entre 2 phases vertes. Cependant, la différence de performances entre 3DQN et *actuated* diminue à mesure que le coefficient augmente. La première raison à ce constat est que les performances d'*actuated* sont meilleures lorsqu'il y a plus de véhicules à l'intersection (explications en Section 4.6.2). Mais les décisions de l'agent 3DQN sont aussi probablement moins pertinentes à mesure que le trafic vélo augmente. En effet, plus le trafic vélo s'éloigne de celui utilisé pendant l'entraînement, plus la probabilité que l'agent reçoive un état qu'il n'a pas l'habitude de gérer est importante. Cette situation peut mener l'agent à prendre une décision peu pertinente, augmentant la saturation des voies et provoquant une situation qu'il est encore moins capable de gérer. Cela explique aussi l'écart de performances entre les simulations qui est plus élevé lorsque le coefficient atteint 1.5. L'approche 3DQN apparaît donc robuste aux changements du trafic vélos, tant que le trafic varie entre 50% et 140% de celui utilisé pendant l'entraînement.

Pour aller plus loin, la Figure 4.9 montre le temps d'attente moyen des vélos (4.9a) et des voitures (4.9b). Pour *actuated*, le temps d'attente moyen des 2 types de véhicules évolue de manière similaire, avec une légère augmentation linéaire à mesure que le coefficient augmente. Ce constat n'est pas le même pour 3DQN. Pour les voitures, l'augmentation est plutôt linéaire, mais plus prononcée. Le temps d'attente moyen des voitures généré par 3DQN dépasse d'ailleurs celui généré par *actuated* lorsque le coefficient atteint 1.1. Pour les vélos, le temps d'attente moyen généré par 3DQN est plus stable et diminue même jusqu'à ce que le coefficient multiplicateur atteigne 1.1. Après cela, il augmente, mais ne dépasse jamais celui d'*actuated*. C'est une surprise, car il était attendu que les temps d'attente moyens des 2 types de véhicules évoluent de la même manière que la somme des temps d'attente de la Figure



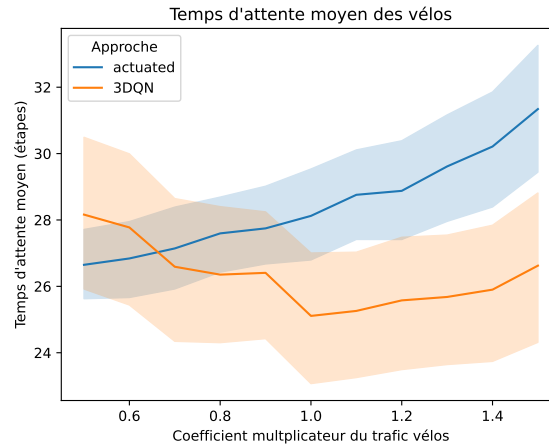
(A) Nombre de véhicules.



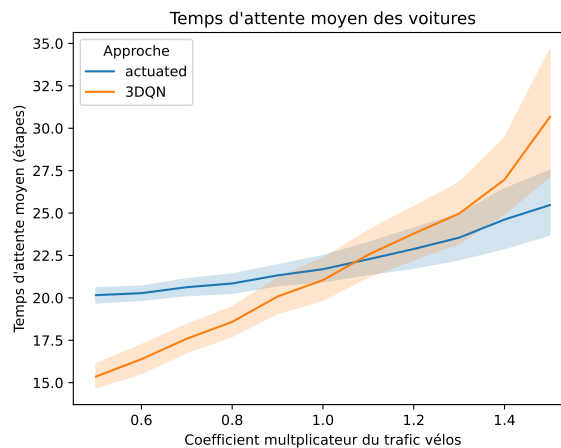
(B) Somme du temps d'attente des véhicules.

FIGURE 4.8 – Nombre de véhicules et sommes des temps d'attentes entre selon le coefficient multiplicateur du trafic vélo (de 6h à 20h).

4.8. À la place, le temps d’attente moyen des voitures augmente de manière plus forte, permettant aux vélos de moins attendre. Ce résultat peut s’expliquer par les différences entre les vélos et les voitures dans SUMO (détaillées en Section 4.5), ou par le comportement de l’agent 3DQN qui privilégie les voies vélos lors de ses choix. Une analyse du comportement du feu avec les 2 approches est nécessaire pour départager ces hypothèses.



(A) Temps d’attente moyen des vélos.



(B) Temps d’attente moyen des voitures.

FIGURE 4.9 – Temps d’attente moyen des véhicules selon le coefficient multiplicateur du trafic vélo (de 6h à 20h).

## 4.7 Analyse du comportement du feu

L’approche 3DQN permet bien de limiter l’augmentation du temps d’attente causée par la sécurisation du feu pour les cyclistes. Cependant, lorsque le trafic vélo augmente, c’est le temps d’attente des voitures qui est le plus impacté par les choix de l’agent. Il est nécessaire d’étudier le comportement du feu pour comprendre les causes de ce phénomène. Pour ce faire, une analyse à l’échelle de la journée entière ainsi qu’à l’échelle d’une heure de pointe sont effectuées. *actuated* est encore une

fois utilisée en comparaison, car c'est l'approche la plus similaire et ayant les performances les plus proches de 3DQN. Les simulations effectuées en Section 4.6.2 sont réutilisées.

#### 4.7.1 Journée entière

La première partie de cette analyse est à l'échelle de la journée entière. La Figure 4.10 montre le nombre d'étapes passées en phase verte ainsi qu'en phase orange par type de voies sur chacun des axes. L'approche *actuated* ayant un cycle statique, le nombre de phases orange entre 2 types de voies ne peut différer que de 1. Lors de la simulation de test, *actuated* venait de finir un cycle lorsque celle-ci s'est arrêté, car le temps passé en phase orange est identique pour toutes les voies (Figure 4.10a). Les voies vélos sont restées aux alentours de 13700 étapes en phase verte et les voies voitures aux alentours de 19500 étapes. La similitude entre les temps passés en phase verte des voies du même type n'est pas surprenante, car les demandes de trafic des 2 axes sont basées sur le même comptage de véhicules. Les différences entre les voies vélos et les voies voitures sont proportionnelles aux différences de trafic entre les 2 types de véhicules. Plus il y a de véhicules, plus le détecteur en amont du feu sera déclenché et augmentera la durée de la phase verte. Les vélos étant moins nombreux que les voitures, il est logique que l'approche *actuated* leur attribue moins de temps en phase verte.

Pour l'approche 3DQN (Figure 4.10b), le nombre d'étapes passées en phase orange est autour de 6000 pour toutes les voies sauf celles pour vélos l'axe Est-Ouest. Cela signifie que l'agent 3DQN choisit peu ces voies lors de ces décisions. Le faible nombre d'étapes passées en phase verte confirme que les voies vélos de cet axe sont délaissées par l'agent, car celui-ci ne prolonge pas les phases vertes qu'il leur octroie. Celles-ci ont d'ailleurs le plus faible nombre d'étapes passées en phase verte, quelle que soit l'approche. Pour compenser, l'agent permet aux vélos de l'axe Nord-Sud de très souvent franchir l'intersection. Ce sont d'ailleurs les voies vélos sur cet axe qui bénéficient du plus d'étapes passées en phase verte, quelle que soit l'approche. Ces voies passent aussi moins de temps à l'orange que les voies pour voitures avec 3DQN, montrant que l'agent prolonge les phases vertes qu'il leur octroie. Pour les voies voitures, le temps passé en phase verte est plus faible avec 3DQN qu'avec *actuated*. Cela est étonnant au vu des différences de trafic entre voitures et vélos. Le comportement de l'agent peut cependant s'expliquer. La récompense est liée au nombre de véhicules attendant à l'intersection. L'agent apprend donc à minimiser le nombre de véhicules qui attend. Celui-ci doit cependant choisir une phase chaque fois que cela lui est demandé. Il arrive, pendant les périodes de faible trafic, que sa décision n'ait pas d'effet sur le nombre de véhicules en attente. Cela se produit lorsque l'intersection est vide ou que seuls quelques véhicules assez éloignés sont en approche. Le cas échéant, l'agent a besoin d'une phase "par défaut". Lors de l'entraînement affiché, cette phase est celle des voies vélos de l'axe Nord-Sud. Cela signifie que les vélos sur cet axe sont privilégiés lorsque le trafic commence à augmenter. Comme vu en Section 4.5, ce sont les voitures qui sont les plus susceptibles d'attendre au vu de leurs caractéristiques en simulation et des demandes de trafic utilisées. L'agent doit donc compenser en partageant une majorité du temps restant entre les voies voitures des 2 axes, délaissant les voies vélos de l'axe Est-Ouest.

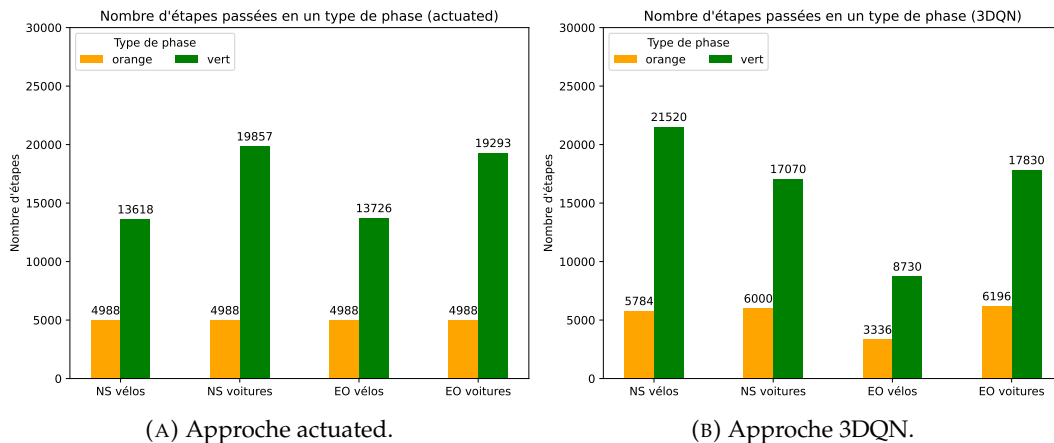
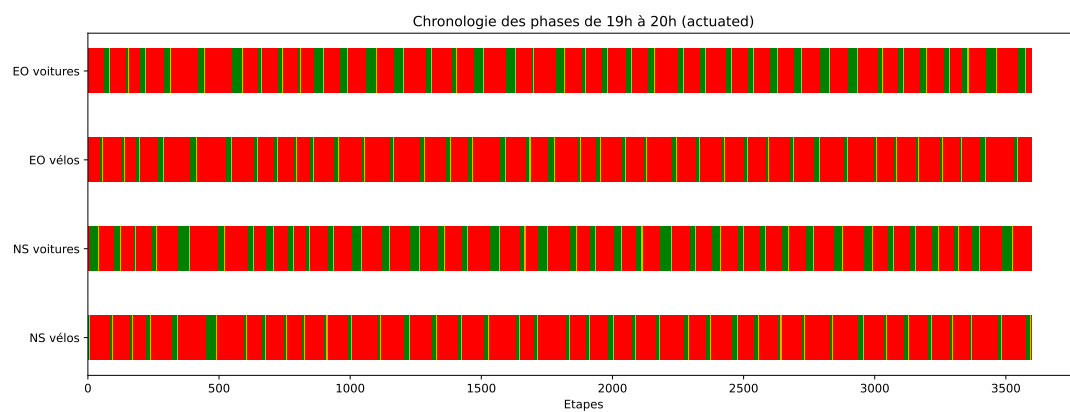


FIGURE 4.10 – Nombre d'étapes passées en phase verte et orange par voie durant une simulation d'un jour.

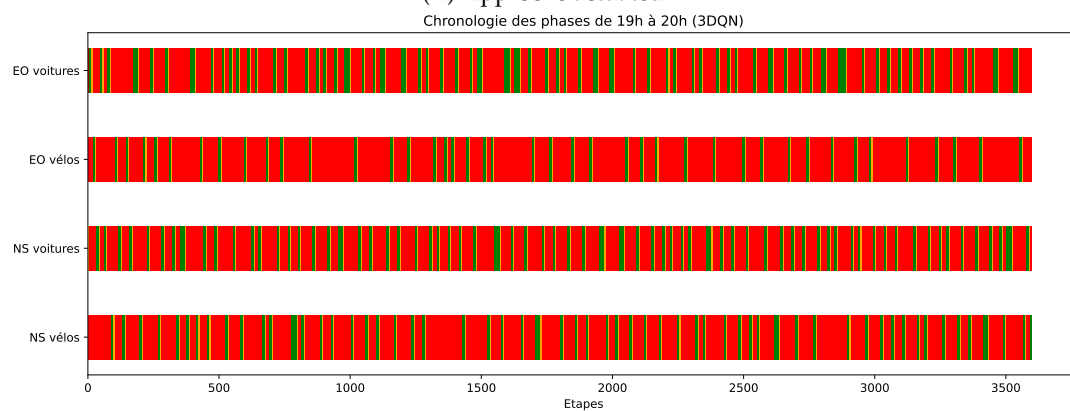
#### 4.7.2 Heure de pointe

Pour étudier le comportement du feu plus en détail, la Figure 4.11 montre la chronologie des phases pour chaque type de voies des 2 axes durant une heure de pointe. L'heure choisie est 19h, celle-ci ayant le trafic le plus important avec un total de 1799 véhicules. Les phases orange sont affichées, mais sont peu visibles en raison de leur courte durée. Sans surprise, les phases vertes s'enchaînent à intervalles réguliers avec l'approche *actuated* (Figure 4.11a). Les différences entre les types de voies tiennent à l'épaisseur des bandes vertes, proportionnelle au temps de la phase. Tout comme sur la Figure 4.10a, on observe que le comportement du feu est quasi identique entre voies pour même type de véhicules. Les phases vertes pour voitures sont plutôt longues tout au long de l'heure, avec quelques exceptions probablement dues à l'aléatoire des processus de Poisson. Les phases vertes des voies vélos sont beaucoup plus fines, conséquence d'un trafic plus faible.

L'approche 3DQN adopte un comportement opposé. Les phases vertes sont irrégulières et dans la majorité des cas très courtes. Les voies pour voitures de l'axe Est-Ouest sont celles avec le plus de bandes épaisses, mais celles-ci restent minoritaires. Quelques rares phases vertes plus longues sont aussi visibles pour les voies voitures et vélos de l'axe Nord-Sud. L'agent accorde environ le même nombre de phases vertes aux voies pour voitures des 2 axes. Les voies pour voitures Est-Ouest passent donc un peu plus d'étapes en phase verte, car les phases qui leur sont accordées sont plus souvent plus longues. Cela est cohérent avec l'analyse de la journée entière, avec 17830 étapes passées en phase verte pour les voies voitures Est-Ouest contre 17070 pour les voies voitures Nord-Sud (Figure 4.10b). Les voies voitures ont cependant plus de temps en phases vertes que les voies vélos sur l'axe Nord-Sud. Cela n'est pas en adéquation avec les résultats à l'échelle de la journée, mais confirme en partie la théorie de la phase "par défaut". En effet, si les voies voitures ont plus de phases vertes que les voies vélos Nord-Sud pendant les heures de pointe, mais que l'inverse se produit à l'échelle de la journée, cela signifie que le déséquilibre est en faveur des voies vélos Nord-Sud pendant les heures creuses. Cependant, cela n'explique pas les différences entre les voies vélos des 2 axes pendant l'heure de pointe. Les voies pour vélos de l'axe Est-Ouest ont beaucoup moins de phases vertes que celles de l'axe Nord-Sud malgré un trafic très similaire. Cela découle probablement des couches convolutives de Q-réseau. Celles-ci ont en effet pour but de repérer les *patterns* importants dans les états qui sont envoyés à l'agent. La gestion des voitures



(A) Approche actuated.



(B) Approche 3DQN.

FIGURE 4.11 – Chronologie des phases par voie de 19h à 20h.



étant prioritaire à celle des vélos pour minimiser le temps d'attente, les couches convolutives doivent donner plus d'importance aux informations représentant les voies voitures. Le Q-réseau interprète donc un environnement sans véhicule et un environnement sans voiture et avec vélos de manière similaire, l'incitant à prendre la même décision dans les 2 cas. Celui-ci privilégiant les voies vélos Nord-Sud lors des heures creuses, il le fait donc aussi lorsque peu de voitures sont présentes à l'intersection en heure de pointe.

## 4.8 Conclusion et discussion

Dans ce chapitre, un feu de circulation permettant aux cyclistes de traverser une intersection de manière sécurisée lors de phases vertes dédiées est proposé. Les résultats montrent qu'adapter le comportement du feu au trafic en temps réel grâce au DRL permet de réduire l'augmentation du temps d'attente créée par ces nouvelles phases vertes. Un agent 3DQN est capable de contrôler ce type de feu de manière efficace avec différentes intensités de trafic. Cela lui permet d'absorber les fluctuations de trafic durant une journée typique. Les performances de l'agent restent stables lors de modifications modérées du trafic vélos. Cependant, l'analyse du comportement du feu lorsque celui-ci est contrôlé par l'agent 3DQN montre un biais de préférence pour les vélos sur l'axe Nord-Sud. Les voitures sont plus nombreuses, et mettent plus de temps à atteindre l'intersection lorsque la file d'attente s'étend. L'agent décide tout de même d'accorder plus de phases vertes aux vélos sur l'axe Nord-Sud qu'aux voitures de manière générale. Bien que contrintuitif, ce comportement n'empêche pas l'agent de moins faire attendre les véhicules à l'intersection qu'une approche dynamique déjà déployée. Ce biais poserait tout de même un problème dans l'éventualité du déploiement d'une telle infrastructure. Inciter l'agent à ne pas effectuer la même action successivement trop de fois pourrait réduire ce biais. Cela demanderait une modification de la fonction de récompenses, ou une nouvelle conception des actions pouvant être effectuées par l'agent.

Ce travail s'accompagne aussi d'autres limites. Bien que des données réelles de comptage soient utilisées pour simuler le trafic, celui-ci est modélisé par des processus de Poisson. Ces processus constituent une simplification peu réaliste de la demande de trafic. Dans le même souci de réalisme, les véhicules dans les simulations utilisent le comportement par défaut de SUMO. Ce comportement est un comportement idéal, respectant à la lettre toutes les règles du trafic. Des simulations comprenant des trafics et des comportements individuels plus réalistes doivent être conduites avant un éventuel déploiement. Ces premiers résultats sont tout de même concluants. Une perspective intéressante de ce travail serait d'étudier la distribution des véhicules en sortie d'intersection, puis d'entraîner un autre agent DRL à partir de celles-ci. Cela permettrait de chaîner plusieurs feux chacun contrôlés par un agent DRL dans le but de créer des vagues vertes optimisées en temps réel.

Au vu des résultats mitigés de l'agent PPO, il est aussi possible qu'un autre algorithme de DRL puisse faire mieux que l'agent 3DQN entraîné dans ce chapitre. Le *Rainbow DQN* [142], un 3DQN avec des améliorations augmentant de manière générale les performances des politiques créées, serait par exemple intéressant à tester.

## Chapitre 5

# Conclusion

L'augmentation du nombre de cyclistes en ville a permis la multiplication des données décrivant leurs comportements. Dans cette thèse, ces données sont utilisées dans le but de créer des outils innovants d'aide aux politiques urbaines d'incitation au cyclisme.

### 5.1 Résumé

Dans un premier temps, les principales sources de données comportementales de cyclistes sont présentées. Elles sont au nombre de 4 : les réponses à des questionnaires de préférences, les systèmes de vélos en libre-service, les capteurs GPS et les compteurs de vélos. Ces sources de données sont étudiées par le biais de 4 caractéristiques : l'accessibilité, la compatibilité avec les méthodes d'apprentissage, le degré d'informations sur les comportements individuels et les problématiques auxquelles elles permettent de répondre dans la littérature scientifique. Au terme de cette étude, 2 jeux de traces GPS sont retenus pour la génération d'itinéraires cyclables.

Une méthode de création de modèles de choix d'itinéraire implicites est ensuite développée autour de ces jeux de données. Après une analyse tant qualitative que quantitative des jeux, un algorithme d'apprentissage non supervisé est choisi pour regrouper les traces autour de segments préférés des cyclistes ayant généré les jeux. Une métrique de distance ainsi qu'une méthode de paramétrage sont développées pour cela. Les *clusters* ainsi créés permettent de démontrer l'existence de ces segments de route préférés ainsi que de les identifier. Une pondération de graphe routier utilisant un *cluster* est ensuite effectué pour permettre la génération de chemins privilégiant un groupe de segments de route préférés. Un réseau de neurones profond permettant de traiter des données séquentielles est ensuite proposé pour affilier un chemin le plus court entre une origine et une destination à un cluster. Toutes ces composantes assemblées constituent un modèle de choix d'itinéraire implicite permettant la génération d'itinéraires cyclables. Ces itinéraires sont de meilleures qualités que ceux proposés par des services commerciaux.

Le développement de modèles de choix d'itinéraire implicites a mis en lumière l'utilisation hétérogène des segments de routes par les cyclistes en ville. Les préférences des cyclistes ont largement été étudiées et la sécurité sur leurs trajets est une des caractéristiques principales recherchées. Séparer les flux de vélos des flux de véhicules motorisés est une manière efficace de sécuriser les cyclistes. Cela se fait classiquement par la construction de pistes cyclables séparées des voies voitures, coûteuses en place. Une séparation virtuelle des flux est proposée. Celle-ci tire profit des possibilités offertes par les feux de signalisation. Un feu ayant des phases vertes

spécifiques pour les cyclistes est proposé. Celui-ci, si implémenté naïvement, fait exploser le temps d'attente à l'intersection. Une solution à base d'apprentissage par renforcement est proposée pour limiter le coût en temps d'attente de l'implémentation d'un tel feu. Cette solution est testée dans des simulations utilisant des données de compteurs de véhicules pour générer le trafic. L'apprentissage par renforcement surpasse les performances d'une méthode de contrôle de feu de signalisation dynamique déjà déployée.

## 5.2 Discussion et perspectives

Le choix des sources de données ainsi que celui de les utiliser individuellement sont justifiés dans le Chapitre 2. Ces choix structurant cette thèse, il est intéressant d'y revenir un instant. Les avancées de l'intelligence artificielle dans le traitement de texte, notamment grâce aux grands modèles de langages, pourraient permettre un traitement plus rapide des réponses aux questionnaires de préférences. Cela rend envisageable le déploiement de questionnaires à plus grande échelle, limitant le biais de représentativité des personnes interrogées. Cependant, les algorithmes d'apprentissage sont dans ce cas de figure entraînés au préalable, et non sur les données comportementales récupérées à l'aide des questionnaires. L'innovation voulue des outils développés au cours de cette thèse réside dans les représentations implicites que se font les algorithmes d'apprentissage de données comportementales de cyclistes. Pour les modèles implicites, ces représentations servent à faire correspondre un chemin le plus court à certains segments préférés. Pour le feu de circulation intelligent, elles permettent de prendre des décisions minimisant le temps d'attente des véhicules. Le seul moyen trouvé dans la littérature d'entraîner un algorithme d'apprentissage sur des réponses de questionnaires de préférences est d'utiliser une régression, avec des variables choisies explicitement décrivant les réponses. La sélection explicite des variables remplace alors les représentations implicites. Les questionnaires de préférences n'ont pour cette raison pas pu être utilisés au cours de celle-ci.

Les systèmes de vélos en libre-service fournissent eux des données facilement utilisables pour entraîner des outils d'apprentissage. Celles-ci ont l'avantage d'être systématiques et donc plus représentatives que les jeux de traces GPS contributifs, tout comme les données des compteurs vélos. L'utilisation des données de compteurs vélos s'est seulement avérée plus pertinente lorsqu'il a été décidé de s'intéresser à la simulation d'un feu de signalisation. Entraîner des outils d'apprentissage à l'aide de données de systèmes de vélos en libre-service pourrait permettre d'autres innovations que celles décrites dans cette thèse. On peut par exemple imaginer un système d'inférence de quantité de cyclistes dans différentes zones de la ville, utilisant le nombre de vélos BSS en circulation et les stations dans lesquelles ils ont été empruntés. Être capable de quantifier en temps réel l'utilisation d'une zone par les cyclistes pourrait permettre aux villes de s'adapter, en offrant plus de feux verts aux axes les plus utilisés par ceux-ci notamment.

Enfin, il existe d'autres sources de données qui n'ont pas été prises en compte dans cette thèse, car moins utilisées dans la littérature. Les caméras de surveillance en sont un exemple. Un traitement agrégé des flux de celles-ci permettrait la récupération de données comportementales respectant la vie privée des cyclistes. Ces données offriraient de nouvelles opportunités de travailler sur la compréhension du comportement des cyclistes.

Les modèles de choix d'itinéraire implicites créés dans le Chapitre 3 ont l'avantage de pouvoir générer des itinéraires cyclables. Ils ont cependant l'inconvénient de ne pas pouvoir expliquer les choix effectués lors de cette génération. L'intelligence artificielle explicable (XAI pour *eXplainable Artificial Intelligence*) est un domaine de recherche visant à rendre les décisions prises par les outils d'apprentissage plus compréhensibles et transparents. La *layer wise relevance propagation* pourrait être utilisé pour comprendre quelles parties des chemins le plus court utilisés en entrée sont importantes dans la prise de décision du réseau de neurones. Cela permettrait de comprendre quelles zones de la ville sont évitées par les cyclistes, et quelles zones ceux-ci choisissent à la place. Une autre manière d'explicitier les choix effectués par les modèles de choix d'itinéraire serait d'étudier les segments de route préférés identifiés, en comparant leurs caractéristiques dominantes à des modèles de choix d'itinéraire discrets existants.

D'autre part, les modèles de choix d'itinéraire implicites créent des itinéraires plus proches du comportement réel des cyclistes que ceux générés par une solution commercialisée. Un modèle créé à partir d'un large jeu de traces GPS pourrait aboutir à la création d'une solution commerciale de génération d'itinéraires cyclables, si ces performances sont proches de celles des modèles développés durant cette thèse.

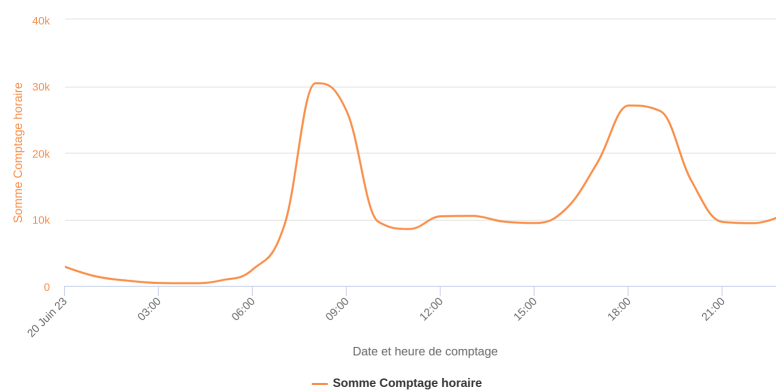
On observe déjà l'utilisation de feux de signalisation pour sécuriser le passage des cyclistes dans certaines villes de France. Cela se matérialise par la mise en place de panneaux permettant aux cyclistes de traverser certaines voies de certaines intersections lorsque leur feu est au rouge. Des expérimentations permettant aux cyclistes de démarrer plus tôt que les voitures sont aussi prévues. Séparer les flux de vélos des flux de voitures paraît donc être une idée d'actualité. L'agent DRL entraîné a démontré sa robustesse face à une modification du trafic de vélos. La robustesse de l'approche DRL face à des comportements humains ne respectant pas complètement la signalisation ou le Code de la route doit cependant être testé avant un éventuel déploiement. D'autre part, l'état de l'intersection reçu par l'agent était toujours exact lors des expériences menées, car celui-ci est donné par le simulateur. Une réflexion sur la manière de compter les véhicules et de mesurer leurs vitesses doit tout d'abord être menée. Ensuite, quelle que soit la méthode retenue, celle-ci risque de commettre des erreurs de mesure. La robustesse de l'approche DRL face à ce type d'erreurs doit donc aussi être testé. Une manière de faire pourrait être de rajouter manuellement des erreurs dans les états envoyés à l'agent, en utilisant une loi normale centrée pour rajouter du bruit lors du calcul de ceux-ci. Les états bruités pourraient être envoyés lors des tests dans un premier temps, puis utilisés pendant l'entraînement de l'agent ensuite.

Si l'approche DRL s'avère robuste au terme de ces expériences, il serait intéressant de chainer plusieurs feux de ce type dans le but de créer des pistes cyclables virtuelles. Ces pistes permettraient aux cyclistes de traverser plusieurs intersections de suite sans être confrontés aux flux de véhicules motorisés. De telles pistes cyclables installées dans une stratégie d'urbanisme tactique permettrait de mesurer l'attrait des cyclistes pour une nouvelle infrastructure sur une voie précise. Cela aiderait les décideurs politiques à mesurer la pertinence de l'installation d'une piste cyclable physique sur celle-ci. Enfin, le déploiement d'un tel feu pourrait s'accompagner d'une étude du comportement des cyclistes à l'intersection pour déceler

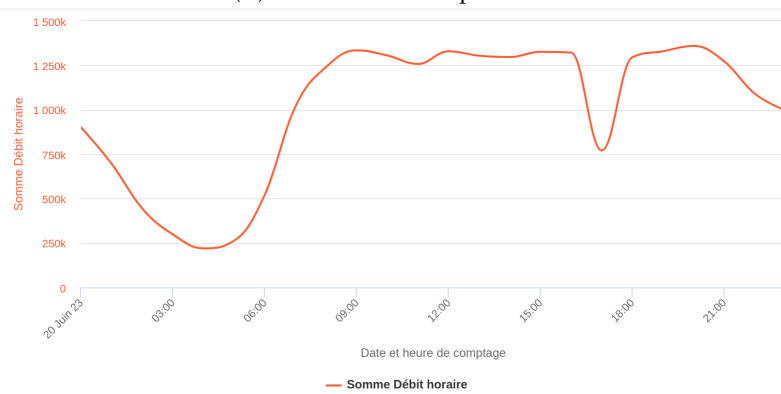
d'éventuelles modifications dans celui-ci, en termes de non-respect du feu rouge notamment.

## Annexe A

# Distribution des véhicules comptés à Paris



(A) Nombre de vélos par heure.



(B) Nombre de voitures par heure.

FIGURE A.1 – Somme des véhicules comptés dans la ville de Paris le 20 juin 2023.

# Bibliographie

- [1] Timothy R. WOJAN et Karen S. HAMRICK. « Can Walking or Biking to Work Really Make a Difference? Compact Development, Observed Commuter Choice and Body Mass Index ». en. In : *PLOS ONE* 10.7 (juill. 2015), e0130903. ISSN : 1932-6203. DOI : [10.1371/journal.pone.0130903](https://doi.org/10.1371/journal.pone.0130903). URL : <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0130903> (visité le 04/03/2021).
- [2] Louise-Ann LEYLAND et al. « The effect of cycling on cognitive function and well-being in older adults ». en. In : *PLOS ONE* 14.2 (fév. 2019). Sous la dir. de Maria Francesca PIACENTINI, e0211779. ISSN : 1932-6203. DOI : [10.1371/journal.pone.0211779](https://doi.org/10.1371/journal.pone.0211779). URL : <https://dx.plos.org/10.1371/journal.pone.0211779> (visité le 02/03/2021).
- [3] Yongping ZHANG et Zhifu MI. « Environmental benefits of bike sharing : A big data-based analysis ». In : *Applied Energy* 220 (juin 2018), p. 296-301. ISSN : 03062619. DOI : [10.1016/j.apenergy.2018.03.101](https://doi.org/10.1016/j.apenergy.2018.03.101). URL : <https://linkinghub.elsevier.com/retrieve/pii/S0306261918304392> (visité le 05/03/2021).
- [4] Ralph BUEHLER et al. « Reducing car dependence in the heart of Europe : lessons from Germany, Austria, and Switzerland ». In : *Transport Reviews* 37.1 (2 jan. 2017), p. 4-28. ISSN : 0144-1647, 1464-5327. DOI : [10.1080/01441647.2016.1177799](https://doi.org/10.1080/01441647.2016.1177799). URL : <https://www.tandfonline.com/doi/full/10.1080/01441647.2016.1177799> (visité le 09/11/2023).
- [5] Jordi HONEY-ROSÉS et al. « The impact of COVID-19 on public space : an early review of the emerging questions – design, perceptions and inequities ». In : *Cities & Health* 5 (sup1 21 juill. 2021), S263-S279. ISSN : 2374-8834, 2374-8842. DOI : [10.1080/23748834.2020.1780074](https://doi.org/10.1080/23748834.2020.1780074). URL : <https://www.tandfonline.com/doi/full/10.1080/23748834.2020.1780074> (visité le 09/11/2023).
- [6] Robert CERVERO, Benjamin CALDWELL et Jesus CUELLAR. « Bike-and-Ride : Build It and They Will Come ». In : *Journal of Public Transportation* 16.4 (déc. 2013), p. 83-105. ISSN : 1077-291X, 2375-0901. DOI : [10.5038/2375-0901.16.4.5](https://doi.org/10.5038/2375-0901.16.4.5). URL : <http://scholarcommons.usf.edu/jpt/vol16/iss4/5/> (visité le 26/07/2023).
- [7] Pablo VERA-VILLARROEL et al. « Perception of Safety and Liking Associated to the Colour Intervention of Bike Lanes : Contribution from the Behavioural Sciences to Urban Design and Wellbeing ». en. In : *PLOS ONE* 11.8 (août 2016). Sous la dir. d'Yanguang CHEN, e0160399. ISSN : 1932-6203. DOI : [10.1371/journal.pone.0160399](https://doi.org/10.1371/journal.pone.0160399). URL : <https://dx.plos.org/10.1371/journal.pone.0160399> (visité le 04/03/2021).

- [8] Heike MARQUART, Uwe SCHLINK et Maximilian UEBERHAM. « The planned and the perceived city : A comparison of cyclists' and decision-makers' views on cycling quality ». In : *Journal of Transport Geography* 82 (jan. 2020), p. 102602. ISSN : 09666923. DOI : [10.1016/j.jtrangeo.2019.102602](https://doi.org/10.1016/j.jtrangeo.2019.102602). URL : <https://linkinghub.elsevier.com/retrieve/pii/S0966692319303825> (visité le 09/11/2023).
- [9] Wei LU, Darren M. SCOTT et Ron DALUMPINES. « Understanding bike share cyclist route choice using GPS data : Comparing dominant routes and shortest paths ». en. In : *Journal of Transport Geography* 71 (juill. 2018), p. 172-181. ISSN : 09666923. DOI : [10.1016/j.jtrangeo.2018.07.012](https://doi.org/10.1016/j.jtrangeo.2018.07.012). URL : <https://linkinghub.elsevier.com/retrieve/pii/S0966692318300875> (visité le 04/03/2021).
- [10] Nebiyou Y. TILAHUN, David M. LEVINSON et Kevin J. KRIZEK. « Trails, lanes, or traffic : Valuing bicycle facilities with an adaptive stated preference survey ». In : *Transportation Research Part A : Policy and Practice* 41.4 (mai 2007), p. 287-301. ISSN : 09658564. DOI : [10.1016/j.tra.2006.09.007](https://doi.org/10.1016/j.tra.2006.09.007). URL : <https://linkinghub.elsevier.com/retrieve/pii/S096585640600108X> (visité le 09/02/2023).
- [11] Solveig NORDENGEN et al. « National Trends in Cycling in Light of the Norwegian Bike Traffic Index ». In : *International Journal of Environmental Research and Public Health* 18.12 (8 juin 2021), p. 6198. ISSN : 1660-4601. DOI : [10.3390/ijerph18126198](https://doi.org/10.3390/ijerph18126198). URL : <https://www.mdpi.com/1660-4601/18/12/6198> (visité le 08/02/2023).
- [12] Hua ZHANG, Susan A. SHAHEEN et Xingpeng CHEN. « Bicycle Evolution in China : From the 1900s to the Present ». In : *International Journal of Sustainable Transportation* 8.5 (3 sept. 2014), p. 317-335. ISSN : 1556-8318, 1556-8334. DOI : [10.1080/15568318.2012.699999](https://doi.org/10.1080/15568318.2012.699999). URL : <http://www.tandfonline.com/doi/abs/10.1080/15568318.2012.699999> (visité le 09/11/2023).
- [13] Konstantinos LIAKOS et al. « Machine Learning in Agriculture : A Review ». In : *Sensors* 18.8 (14 août 2018), p. 2674. ISSN : 1424-8220. DOI : [10.3390/s18082674](https://doi.org/10.3390/s18082674). URL : <http://www.mdpi.com/1424-8220/18/8/2674> (visité le 10/11/2023).
- [14] Christian SALIM et Nathalie MITTON. « K-predictions based data reduction approach in WSN for smart agriculture ». In : *Computing* 103.3 (mars 2021), p. 509-532. ISSN : 0010-485X, 1436-5057. DOI : [10.1007/s00607-020-00864-z](https://doi.org/10.1007/s00607-020-00864-z). URL : <http://link.springer.com/10.1007/s00607-020-00864-z> (visité le 27/11/2023).
- [15] Rahul C. DEO. « Machine Learning in Medicine ». In : *Circulation* 132.20 (17 nov. 2015), p. 1920-1930. ISSN : 0009-7322, 1524-4539. DOI : [10.1161/CIRCULATIONAHA.115.001593](https://doi.org/10.1161/CIRCULATIONAHA.115.001593). URL : <https://www.ahajournals.org/doi/10.1161/CIRCULATIONAHA.115.001593> (visité le 10/11/2023).
- [16] Aziliz LE GLAZ et al. « Machine Learning and Natural Language Processing in Mental Health : Systematic Review ». In : *Journal of Medical Internet Research* 23.5 (4 mai 2021), e15708. ISSN : 1438-8871. DOI : [10.2196/15708](https://doi.org/10.2196/15708). URL : <https://www.jmir.org/2021/5/e15708> (visité le 27/11/2023).



- [17] Abhishek SONI et al. « Design of a Machine Learning-Based Self-driving Car ». In : *Machine Learning for Robotics Applications*. Sous la dir. de Monica BIANCHINI et al. T. 960. Singapore : Springer Singapore, 2021, p. 139-151. ISBN : 9789811605970 9789811605987. DOI : [10.1007/978-981-16-0598-7\\_11](https://doi.org/10.1007/978-981-16-0598-7_11). URL : [https://link.springer.com/10.1007/978-981-16-0598-7\\_11](https://link.springer.com/10.1007/978-981-16-0598-7_11) (visité le 10/11/2023).
- [18] Pierre-Antoine LAHAROTTE, Romain BILLOT et Nour Eddin EL FAOUZI. « Detection of non-recurrent road traffic events based on clustering indicators ». In : *ESANN'17 : European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*. ESANN'17 : European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, Bruges, BELGIQUE, 26-/04/2017 - 28/04/2017. Bruges, Belgium, avr. 2017, p. -. URL : <https://hal.science/hal-01802649>.
- [19] David SILVER et al. « Reward is enough ». In : *Artificial Intelligence* 299 (2021), p. 103535.
- [20] Xiaojin Jerry ZHU. « Semi-supervised learning literature survey ». In : (2005).
- [21] Richard S SUTTON et Andrew G BARTO. *Reinforcement learning : An introduction*. MIT press, 2018.
- [22] Alex KRIZHEVSKY, Ilya SUTSKEVER et Geoffrey E HINTON. « ImageNet Classification with Deep Convolutional Neural Networks ». In : *Advances in Neural Information Processing Systems*. Sous la dir. de F. PEREIRA et al. T. 25. Curran Associates, Inc., 2012. URL : [https://proceedings.neurips.cc/paper\\_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf).
- [23] Ali Bou NASSIF et al. « Speech Recognition Using Deep Neural Networks : A Systematic Review ». In : *IEEE Access* 7 (2019), p. 19143-19165. ISSN : 2169-3536. DOI : [10.1109/ACCESS.2019.2896880](https://doi.org/10.1109/ACCESS.2019.2896880). URL : <https://ieeexplore.ieee.org/document/8632885/> (visité le 10/11/2023).
- [24] Tomas MIKOLOV et al. « Recurrent neural network based language model. ». In : *Interspeech*. T. 2. 3. Makuhari. 2010, p. 1045-1048.
- [25] Lucas MAGNANA, Herve RIVANO et Nicolas CHIABAUT. « Implicit GPS-based bicycle route choice model using clustering methods and a LSTM network ». In : *PLOS ONE* 17.3 (17 mars 2022). Sous la dir. de Jinjun TANG, e0264196. ISSN : 1932-6203. DOI : [10.1371/journal.pone.0264196](https://doi.org/10.1371/journal.pone.0264196). URL : <https://dx.plos.org/10.1371/journal.pone.0264196> (visité le 06/06/2022).
- [26] Lucas MAGNANA, Hervé RIVANO et Nicolas CHIABAUT. « A DRL solution to help reduce the cost in waiting time of securing a traffic light for cyclists. » working paper or preprint. Nov. 2023. URL : <https://inria.hal.science/hal-04300866>.
- [27] Matthieu ADAM et al. « Susciter la parole des cyclistes : traces GPS et vidéos au service de l'entretien ». In : (2020). DOI : [10.26151/ESPACESTEMPS.NET-Z2WH-1074](https://doi.org/10.26151/ESPACESTEMPS.NET-Z2WH-1074). URL : <https://www.espacestemps.net/articles/susciter-la-parole-des-cyclistes-traces-gps-et-videos-au-service-de-lentretien/> (visité le 24/03/2023).
- [28] Gustavo ROMANILLOS et al. « Big Data and Cycling ». In : *Transport Reviews* 36.1 (2 jan. 2016), p. 114-133. ISSN : 0144-1647, 1464-5327. DOI : [10.1080/01441647.2015.1084067](https://doi.org/10.1080/01441647.2015.1084067). URL : <http://www.tandfonline.com/doi/full/10.1080/01441647.2015.1084067> (visité le 09/01/2023).

- [29] Bandhan Bandhu MAJUMDAR et Sudeshna MITRA. « Analysis of bicycle route-related improvement strategies for two Indian cities using a stated preference survey ». In : *Transport Policy* 63 (avr. 2018), p. 176-188. ISSN : 0967070X. DOI : [10.1016/j.tranpol.2017.12.016](https://doi.org/10.1016/j.tranpol.2017.12.016). URL : <https://linkinghub.elsevier.com/retrieve/pii/S0967070X16306977> (visité le 14/02/2023).
- [30] Xuefeng LI et al. « Social Factors Influencing the Choice of Bicycle : Difference Analysis among Private Bike, Public Bike Sharing and Free-Floating Bike Sharing in Kunming, China ». In : *KSCE Journal of Civil Engineering* 23.5 (mai 2019), p. 2339-2348. ISSN : 1226-7988, 1976-3808. DOI : [10.1007/s12205-019-2078-7](https://doi.org/10.1007/s12205-019-2078-7). URL : <http://link.springer.com/10.1007/s12205-019-2078-7> (visité le 09/02/2023).
- [31] Ricardo HURTUBIA, Rodrigo MORA et Felipe MORENO. « The role of bike sharing stations in the perception of public spaces : A stated preferences analysis ». In : *Landscape and Urban Planning* 214 (oct. 2021), p. 104174. ISSN : 01692046. DOI : [10.1016/j.landurbplan.2021.104174](https://doi.org/10.1016/j.landurbplan.2021.104174). URL : <https://linkinghub.elsevier.com/retrieve/pii/S0169204621001377> (visité le 09/02/2023).
- [32] Monique A. STINSON et Chandra R. BHAT. « Commuter Bicyclist Route Choice : Analysis Using a Stated Preference Survey ». In : *Transportation Research Record : Journal of the Transportation Research Board* 1828.1 (jan. 2003), p. 107-115. ISSN : 0361-1981, 2169-4052. DOI : [10.3141/1828-13](https://doi.org/10.3141/1828-13). URL : <http://journals.sagepub.com/doi/10.3141/1828-13> (visité le 09/02/2023).
- [33] Brian CAULFIELD, Elaine BRICK et Orla Thérèse MCCARTHY. « Determining bicycle infrastructure preferences – A case study of Dublin ». In : *Transportation Research Part D : Transport and Environment* 17.5 (juill. 2012), p. 413-417. ISSN : 13619209. DOI : [10.1016/j.trd.2012.04.001](https://doi.org/10.1016/j.trd.2012.04.001). URL : <https://linkinghub.elsevier.com/retrieve/pii/S1361920912000363> (visité le 09/02/2023).
- [34] J. D. HUNT et J. E. ABRAHAM. « Influences on bicycle use ». In : *Transportation* 34.4 (25 juin 2007), p. 453-470. ISSN : 0049-4488, 1572-9435. DOI : [10.1007/s11116-006-9109-1](https://doi.org/10.1007/s11116-006-9109-1). URL : <http://link.springer.com/10.1007/s11116-006-9109-1> (visité le 04/03/2021).
- [35] Ipek N. SENNER, Naveen ELURU et Chandra R. BHAT. « An analysis of bicycle route choice preferences in Texas, US ». In : *Transportation* 36.5 (sept. 2009), p. 511-539. ISSN : 0049-4488, 1572-9435. DOI : [10.1007/s11116-009-9201-4](https://doi.org/10.1007/s11116-009-9201-4). URL : <http://link.springer.com/10.1007/s11116-009-9201-4> (visité le 04/03/2021).
- [36] Bibhas Kumar DEY et al. « Accommodating exogenous variable and decision rule heterogeneity in discrete choice models : Application to bicyclist route choice ». In : *PLOS ONE* 13.11 (30 nov. 2018). Sous la dir. de Baogui XIN, e0208309. ISSN : 1932-6203. DOI : [10.1371/journal.pone.0208309](https://doi.org/10.1371/journal.pone.0208309). URL : <https://dx.plos.org/10.1371/journal.pone.0208309> (visité le 25/02/2021).
- [37] Christopher M. MONSERE, Nathan W. MCNEIL et Rebecca L. SANDERS. « User-Rated Comfort and Preference of Separated Bike Lane Intersection Designs ». In : *Transportation Research Record : Journal of the Transportation Research Board* 2674.9 (sept. 2020), p. 216-229. ISSN : 0361-1981, 2169-4052. DOI : [10.1177/0361198120927694](https://doi.org/10.1177/0361198120927694). URL : <http://journals.sagepub.com/doi/10.1177/0361198120927694> (visité le 09/02/2023).

- [38] SINGHVI et al. *Predicting Bike Usage for New York City's Bike Sharing System*. 2015.
- [39] Lei LIN, Zhengbing HE et Srinivas PEETA. « Predicting station-level hourly demand in a large-scale bike-sharing network : A graph convolutional neural network approach ». In : *Transportation Research Part C : Emerging Technologies* 97 (déc. 2018), p. 258-276. ISSN : 0968090X. DOI : [10.1016/j.trc.2018.10.011](https://doi.org/10.1016/j.trc.2018.10.011). URL : <https://linkinghub.elsevier.com/retrieve/pii/S0968090X18300974> (visité le 09/03/2021).
- [40] Yan PAN et al. « Predicting bike sharing demand using recurrent neural networks ». In : *Procedia Computer Science* 147 (2019), p. 562-566. ISSN : 18770509. DOI : [10.1016/j.procs.2019.01.217](https://doi.org/10.1016/j.procs.2019.01.217). URL : <https://linkinghub.elsevier.com/retrieve/pii/S1877050919302364> (visité le 07/09/2021).
- [41] Tae San KIM, Won Kyung LEE et So Young SOHN. « Graph convolutional network approach applied to predict hourly bike-sharing demands considering spatial, temporal, and global effects ». In : *PLOS ONE* 14.9 (16 sept. 2019). Sous la dir. d'Yossiri ADULYASAK, e0220782. ISSN : 1932-6203. DOI : [10.1371/journal.pone.0220782](https://doi.org/10.1371/journal.pone.0220782). URL : <https://dx.plos.org/10.1371/journal.pone.0220782> (visité le 25/02/2021).
- [42] Yexin LI et Yu ZHENG. « Citywide Bike Usage Prediction in a Bike-Sharing System ». In : *IEEE Transactions on Knowledge and Data Engineering* 32.6 (1<sup>er</sup> juin 2020), p. 1079-1091. ISSN : 1041-4347, 1558-2191, 2326-3865. DOI : [10.1109/TKDE.2019.2898831](https://doi.org/10.1109/TKDE.2019.2898831). URL : <https://ieeexplore.ieee.org/document/8640047/> (visité le 03/02/2023).
- [43] Zidong YANG et al. « Mobility Modeling and Data-Driven Closed-Loop Prediction in Bike-Sharing Systems ». In : *IEEE Transactions on Intelligent Transportation Systems* 20.12 (déc. 2019), p. 4488-4499. ISSN : 1524-9050, 1558-0016. DOI : [10.1109/TITS.2018.2886456](https://doi.org/10.1109/TITS.2018.2886456). URL : <https://ieeexplore.ieee.org/document/8606241/> (visité le 07/09/2021).
- [44] Yexin LI et al. « Traffic prediction in a bike-sharing system ». In : *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*. SIGSPATIAL'15 : 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems. Seattle Washington : ACM, 3 nov. 2015, p. 1-10. ISBN : 9781450339674. DOI : [10.1145/2820783.2820837](https://doi.org/10.1145/2820783.2820837). URL : <https://dl.acm.org/doi/10.1145/2820783.2820837> (visité le 05/03/2021).
- [45] Yi AI et al. « A deep learning approach on short-term spatiotemporal distribution forecasting of dockless bike-sharing system ». In : *Neural Computing and Applications* 31.5 (1<sup>er</sup> mai 2019), p. 1665-1677. ISSN : 1433-3058. DOI : [10.1007/s00521-018-3470-9](https://doi.org/10.1007/s00521-018-3470-9). URL : <https://doi.org/10.1007/s00521-018-3470-9> (visité le 02/09/2021).
- [46] Chengcheng XU, Junyi JI et Pan LIU. « The station-free sharing bike demand forecasting with a deep learning approach and large-scale datasets ». In : *Transportation Research Part C : Emerging Technologies* 95 (oct. 2018), p. 47-60. ISSN : 0968090X. DOI : [10.1016/j.trc.2018.07.013](https://doi.org/10.1016/j.trc.2018.07.013). URL : <https://linkinghub.elsevier.com/retrieve/pii/S0968090X18306764> (visité le 09/03/2021).

- [47] Miriam RICCI. « Bike sharing : A review of evidence on impacts and processes of implementation and operation ». In : *Research in Transportation Business & Management* 15 (juin 2015), p. 28-38. ISSN : 22105395. DOI : [10.1016/j.rtbm.2015.03.003](https://doi.org/10.1016/j.rtbm.2015.03.003). URL : <https://linkinghub.elsevier.com/retrieve/pii/S2210539515000140> (visité le 30/01/2023).
- [48] Lu-Yi QIU et Ling-Yun HE. « Bike Sharing and the Economy, the Environment, and Health-Related Externalities ». In : *Sustainability* 10.4 (11 avr. 2018), p. 1145. ISSN : 2071-1050. DOI : [10.3390/su10041145](https://doi.org/10.3390/su10041145). URL : <http://www.mdpi.com/2071-1050/10/4/1145> (visité le 30/01/2023).
- [49] Xiaolu ZHOU. « Understanding Spatiotemporal Patterns of Biking Behavior by Analyzing Massive Bike Sharing Data in Chicago ». In : *PLOS ONE* 10.10 (7 oct. 2015). Sous la dir. d'Yanguang CHEN, e0137922. ISSN : 1932-6203. DOI : [10.1371/journal.pone.0137922](https://doi.org/10.1371/journal.pone.0137922). URL : <https://dx.plos.org/10.1371/journal.pone.0137922> (visité le 25/02/2021).
- [50] Longbiao CHEN et al. « Understanding bike trip patterns leveraging bike sharing system open data ». In : *Frontiers of Computer Science* 11.1 (fév. 2017), p. 38-48. ISSN : 2095-2228, 2095-2236. DOI : [10.1007/s11704-016-6006-4](https://doi.org/10.1007/s11704-016-6006-4). URL : <http://link.springer.com/10.1007/s11704-016-6006-4> (visité le 07/09/2021).
- [51] Ezgi EREN et Volkan Emre UZ. « A review on bike-sharing : The factors affecting bike-sharing demand ». In : *Sustainable Cities and Society* 54 (mars 2020), p. 101882. ISSN : 22106707. DOI : [10.1016/j.scs.2019.101882](https://doi.org/10.1016/j.scs.2019.101882). URL : <https://linkinghub.elsevier.com/retrieve/pii/S2210670719312387> (visité le 30/01/2023).
- [52] Jiawei ZHANG et al. « Bicycle-Sharing System Analysis and Trip Prediction ». In : *2016 17th IEEE International Conference on Mobile Data Management (MDM)*. 2016 17th IEEE International Conference on Mobile Data Management (MDM). Porto : IEEE, juin 2016, p. 174-179. ISBN : 9781509008834. DOI : [10.1109/MDM.2016.35](https://doi.org/10.1109/MDM.2016.35). URL : <http://ieeexplore.ieee.org/document/7517792/> (visité le 05/03/2021).
- [53] Brian CAULFIELD et al. « Examining usage patterns of a bike-sharing scheme in a medium sized city ». In : *Transportation Research Part A : Policy and Practice* 100 (juin 2017), p. 152-161. ISSN : 09658564. DOI : [10.1016/j.tra.2017.04.023](https://doi.org/10.1016/j.tra.2017.04.023). URL : <https://linkinghub.elsevier.com/retrieve/pii/S0965856416304141> (visité le 03/09/2021).
- [54] Leonardo CAGGIANI, Rosalia CAMPOREALE et Michele OTTOMANELLI. « A real time multi-objective cyclists route choice model for a bike-sharing mobile application ». In : *2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*. 2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS). Naples, Italy : IEEE, juin 2017, p. 645-650. ISBN : 9781509064847. DOI : [10.1109/MTITS.2017.8005593](https://doi.org/10.1109/MTITS.2017.8005593). URL : <http://ieeexplore.ieee.org/document/8005593/> (visité le 03/03/2021).
- [55] Hongmian GONG et al. « A GPS/GIS method for travel mode detection in New York City ». In : *Computers, Environment and Urban Systems* 36.2 (mars 2012), p. 131-139. ISSN : 01989715. DOI : [10.1016/j.compenvurbsys.2011.05.003](https://doi.org/10.1016/j.compenvurbsys.2011.05.003). URL : <https://linkinghub.elsevier.com/retrieve/pii/S0198971511000536> (visité le 15/03/2023).

- [56] Nadine SCHÜSSLER et Kay W. AXHAUSEN. « Identifying trips and activities and their characteristics from GPS raw data without further information ». In : (2008), 28 p. DOI : [10.3929/ETHZ-A-005589980](https://doi.org/10.3929/ETHZ-A-005589980). URL : <http://hdl.handle.net/20.500.11850/9471> (visité le 15/03/2023).
- [57] Li SHEN et Peter R. STOPHER. « Review of GPS Travel Survey and GPS Data-Processing Methods ». In : *Transport Reviews* 34.3 (4 mai 2014), p. 316-334. ISSN : 0144-1647, 1464-5327. DOI : [10.1080/01441647.2014.903530](https://doi.org/10.1080/01441647.2014.903530). URL : <http://www.tandfonline.com/doi/abs/10.1080/01441647.2014.903530> (visité le 15/03/2023).
- [58] Wendy BOHTE et Kees MAAT. « Deriving and validating trip purposes and travel modes for multi-day GPS-based travel surveys : A large-scale application in the Netherlands ». In : *Transportation Research Part C : Emerging Technologies* 17.3 (juin 2009), p. 285-297. ISSN : 0968090X. DOI : [10.1016/j.trc.2008.11.004](https://doi.org/10.1016/j.trc.2008.11.004). URL : <https://linkinghub.elsevier.com/retrieve/pii/S0968090X08000909> (visité le 15/03/2023).
- [59] Adrian C. PRELIPCEAN, Gyözö GIDÓFALVI et Yusak O. SUSILO. « Transportation mode detection – an in-depth review of applicability and reliability ». In : *Transport Reviews* 37.4 (4 juill. 2017), p. 442-464. ISSN : 0144-1647, 1464-5327. DOI : [10.1080/01441647.2016.1246489](https://doi.org/10.1080/01441647.2016.1246489). URL : <https://www.tandfonline.com/doi/full/10.1080/01441647.2016.1246489> (visité le 15/03/2023).
- [60] Sven LISSNER et Stefan HUBER. « Facing the needs for clean bicycle data – a bicycle-specific approach of GPS data processing ». In : *European Transport Research Review* 13.1 (déc. 2021), p. 8. ISSN : 1867-0717, 1866-8887. DOI : [10.1186/s12544-020-00462-2](https://doi.org/10.1186/s12544-020-00462-2). URL : <https://etr.r.springeropen.com/articles/10.1186/s12544-020-00462-2> (visité le 21/02/2023).
- [61] Kaiyue ZANG et al. « Assessing and Mapping of Road Surface Roughness based on GPS and Accelerometer Sensors on Bicycle-Mounted Smartphones ». In : *Sensors* 18.3 (19 mars 2018), p. 914. ISSN : 1424-8220. DOI : [10.3390/s18030914](https://doi.org/10.3390/s18030914). URL : <http://www.mdpi.com/1424-8220/18/3/914> (visité le 21/02/2023).
- [62] Federico RUPI, Cristian POLIZIANI et Joerg SCHWEIZER. « Data-driven Bicycle Network Analysis Based on Traditional Counting Methods and GPS Traces from Smartphone ». In : *ISPRS International Journal of Geo-Information* 8.8 (25 juill. 2019), p. 322. ISSN : 2220-9964. DOI : [10.3390/ijgi8080322](https://doi.org/10.3390/ijgi8080322). URL : <https://www.mdpi.com/2220-9964/8/8/322> (visité le 21/02/2023).
- [63] Muhammad GHANAYIM et Shlomo BEKHOR. « Modelling bicycle route choice using data from a GPS-assisted household survey ». In : *European Journal of Transport and Infrastructure Research* (1<sup>er</sup> avr. 2018), Vol 18 No 2 (2018). DOI : [10.18757/EJTIR.2018.18.2.3228](https://doi.org/10.18757/EJTIR.2018.18.2.3228). URL : <https://journals.open.tudelft.nl/ejtir/article/view/3228> (visité le 21/02/2023).
- [64] Danique TON et al. « How Do People Cycle in Amsterdam, Netherlands? : Estimating Cyclists' Route Choice Determinants with GPS Data from an Urban Area ». In : *Transportation Research Record* 2662.1 (1<sup>er</sup> jan. 2017), p. 75-82. ISSN : 0361-1981. DOI : [10.3141/2662-09](https://doi.org/10.3141/2662-09). URL : <https://doi.org/10.3141/2662-09> (visité le 30/09/2021).

- [65] Dillon T. FITCH et Susan L. HANDY. « Road environments and bicyclist route choice : The cases of Davis and San Francisco, CA ». In : *Journal of Transport Geography* 85 (mai 2020), p. 102705. ISSN : 09666923. DOI : [10.1016/j.jtrangeo.2020.102705](https://doi.org/10.1016/j.jtrangeo.2020.102705). URL : <https://linkinghub.elsevier.com/retrieve/pii/S0966692319303394> (visité le 08/09/2021).
- [66] Darren M. SCOTT, Wei LU et Matthew J. BROWN. « Route choice of bike share users : Leveraging GPS data to derive choice sets ». In : *Journal of Transport Geography* 90 (jan. 2021), p. 102903. ISSN : 09666923. DOI : [10.1016/j.jtrangeo.2020.102903](https://doi.org/10.1016/j.jtrangeo.2020.102903). URL : <https://linkinghub.elsevier.com/retrieve/pii/S0966692320309807> (visité le 08/09/2021).
- [67] Maëlle ZIMMERMANN, Tien MAI et Emma FREJINGER. « Bike route choice modeling using GPS data without choice sets of paths ». In : *Transportation Research Part C : Emerging Technologies* 75 (fév. 2017), p. 183-196. ISSN : 0968090X. DOI : [10.1016/j.trc.2016.12.009](https://doi.org/10.1016/j.trc.2016.12.009). URL : <https://linkinghub.elsevier.com/retrieve/pii/S0968090X16302637> (visité le 04/03/2021).
- [68] Bahar DADASHOVA et al. « Estimation of Average Annual Daily Bicycle Counts using Crowdsourced Strava Data ». In : *Transportation Research Record : Journal of the Transportation Research Board* 2674.11 (nov. 2020), p. 390-402. ISSN : 0361-1981, 2169-4052. DOI : [10.1177/0361198120946016](https://doi.org/10.1177/0361198120946016). URL : <http://journals.sagepub.com/doi/10.1177/0361198120946016> (visité le 08/02/2023).
- [69] Thomas NOSAL et Luis F. MIRANDA-MORENO. « The effect of weather on the use of North American bicycle facilities : A multi-city analysis using automatic counts ». In : *Transportation Research Part A : Policy and Practice* 66 (août 2014), p. 213-225. ISSN : 09658564. DOI : [10.1016/j.tra.2014.04.012](https://doi.org/10.1016/j.tra.2014.04.012). URL : <https://linkinghub.elsevier.com/retrieve/pii/S0965856414000998> (visité le 08/02/2023).
- [70] Daniel J FAGNANT et Kara KOCKELMAN. « A direct-demand model for bicycle counts : the impacts of level of service and other factors ». In : *Environment and Planning B : Planning and Design* 43.1 (jan. 2016), p. 93-107. ISSN : 0265-8135, 1472-3417. DOI : [10.1177/0265813515602568](https://doi.org/10.1177/0265813515602568). URL : <http://journals.sagepub.com/doi/10.1177/0265813515602568> (visité le 08/02/2023).
- [71] Peter SCHMIEDESKAMP et Weiran ZHAO. « Estimating Daily Bicycle Counts in Seattle, Washington, from Seasonal and Weather Factors ». In : *Transportation Research Record : Journal of the Transportation Research Board* 2593.1 (jan. 2016), p. 94-102. ISSN : 0361-1981, 2169-4052. DOI : [10.3141/2593-12](https://doi.org/10.3141/2593-12). URL : <http://journals.sagepub.com/doi/10.3141/2593-12> (visité le 08/02/2023).
- [72] Jillian STRAUSS, Luis F. MIRANDA-MORENO et Patrick MORENCY. « Mapping cyclist activity and injury risk in a network combining smartphone GPS data and bicycle counts ». In : *Accident Analysis & Prevention* 83 (oct. 2015), p. 132-142. ISSN : 00014575. DOI : [10.1016/j.aap.2015.07.014](https://doi.org/10.1016/j.aap.2015.07.014). URL : <https://linkinghub.elsevier.com/retrieve/pii/S0001457515300233> (visité le 24/03/2023).
- [73] Kristiann C. HEESCH et Michael LANGDON. « The usefulness of GPS bicycle tracking data for evaluating the impact of infrastructure change on cycling behaviour : GPS bicycle tracking data in evaluating cycling behaviour ». In : *Health Promotion Journal of Australia* 27.3 (déc. 2016), p. 222-229. ISSN : 10361073. DOI : [10.1071/HE16032](https://doi.org/10.1071/HE16032). URL : <http://doi.wiley.com/10.1071/HE16032> (visité le 20/03/2023).

- [74] Jillian STRAUSS et al. « Cyclist deceleration rate as surrogate safety measure in Montreal using smartphone GPS data ». In : *Accident Analysis & Prevention* 99 (fév. 2017), p. 287-296. ISSN : 00014575. DOI : 10.1016/j.aap.2016.11.019. URL : <https://linkinghub.elsevier.com/retrieve/pii/S0001457516304328> (visité le 24/03/2023).
- [75] Chen CHEN et al. « Using bicycle app data to develop Safety Performance Functions (SPFs) for bicyclists at intersections : A generic framework ». In : *Transportation Research Part A : Policy and Practice* 132 (fév. 2020), p. 1034-1052. ISSN : 09658564. DOI : 10.1016/j.tra.2019.12.034. URL : <https://linkinghub.elsevier.com/retrieve/pii/S0965856418308255> (visité le 20/03/2023).
- [76] Bahar DADASHOVA et Greg P. GRIFFIN. « Random parameter models for estimating statewide daily bicycle counts using crowdsourced data ». In : *Transportation Research Part D : Transport and Environment* 84 (juill. 2020), p. 102368. ISSN : 13619209. DOI : 10.1016/j.trd.2020.102368. URL : <https://linkinghub.elsevier.com/retrieve/pii/S1361920920305551> (visité le 20/03/2023).
- [77] Nathalie MITTON et Herve RIVANO. « On the use of city bikes to make the city even smarter ». In : *2014 International Conference on Smart Computing Workshops*. 2014 International Conference on Smart Computing Workshops (SMART-COMP Workshops). Hong Kong : IEEE, nov. 2014, p. 3-8. ISBN : 9781479964475. DOI : 10.1109/SMARTCOMP-W.2014.7046675. URL : <http://ieeexplore.ieee.org/document/7046675/> (visité le 27/11/2023).
- [78] Suresh K. LODHA et al. « Visualization of spatio temporal GPS uncertainty within a GIS environment ». In : *AeroSense 2002*. Sous la dir. de Nickolas L. FAUST, James L. KURTZ et Robert TREBITS. Orlando, FL, 30 juill. 2002, p. 216-227. DOI : 10.1117/12.488302. URL : <http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=888733> (visité le 08/06/2023).
- [79] S. LISSNER et al. « GPS-data in bicycle planning : “Which cyclist leaves what kind of traces?” Results of a representative user study in Germany ». In : *Transportation Research Interdisciplinary Perspectives* 7 (sept. 2020), p. 100192. ISSN : 25901982. DOI : 10.1016/j.trip.2020.100192. URL : <https://linkinghub.elsevier.com/retrieve/pii/S2590198220301032> (visité le 21/02/2023).
- [80] Danique TON et al. « Evaluating a data-driven approach for choice set identification using GPS bicycle route choice data from Amsterdam ». In : *Travel Behaviour and Society* 13 (oct. 2018), p. 105-117. ISSN : 2214367X. DOI : 10.1016/j.tbs.2018.07.001. URL : <https://linkinghub.elsevier.com/retrieve/pii/S2214367X18300292> (visité le 21/02/2023).
- [81] Alessandro CRIVELLARI et Euro BEINAT. « LSTM-Based Deep Learning Model for Predicting Individual Mobility Traces of Short-Term Foreign Tourists ». In : *Sustainability* 12.1 (1<sup>er</sup> jan. 2020), p. 349. ISSN : 2071-1050. DOI : 10.3390/su12010349. URL : <https://www.mdpi.com/2071-1050/12/1/349> (visité le 16/04/2020).
- [82] Kyuhyun LEE et Ipek Nese SENNER. « Strava Metro data for bicycle monitoring : a literature review ». In : *Transport Reviews* 41.1 (2 jan. 2021), p. 27-47. ISSN : 0144-1647, 1464-5327. DOI : 10.1080/01441647.2020.1798558. URL : <https://www.tandfonline.com/doi/full/10.1080/01441647.2020.1798558> (visité le 08/02/2023).

- [83] Kristiann C. HEESCH et al. « Evaluation of the Veloway 1 : A natural experiment of new bicycle infrastructure in Brisbane, Australia ». In : *Journal of Transport & Health* 3.3 (sept. 2016), p. 366-376. ISSN : 22141405. DOI : [10.1016/j.jth.2016.06.006](https://doi.org/10.1016/j.jth.2016.06.006). URL : <https://linkinghub.elsevier.com/retrieve/pii/S2214140516301748> (visité le 20/03/2023).
- [84] Yeran SUN et Amin MOBASHERI. « Utilizing Crowdsourced Data for Studies of Cycling and Air Pollution Exposure : A Case Study Using Strava Data ». In : *International Journal of Environmental Research and Public Health* 14.3 (8 mars 2017), p. 274. ISSN : 1660-4601. DOI : [10.3390/ijerph14030274](https://doi.org/10.3390/ijerph14030274). URL : <http://www.mdpi.com/1660-4601/14/3/274> (visité le 20/03/2023).
- [85] Kyuhyun LEE et Ipek SENER. « Understanding Potential Exposure of Bicyclists on Roadways to Traffic-Related Air Pollution : Findings from El Paso, Texas, Using Strava Metro Data ». In : *International Journal of Environmental Research and Public Health* 16.3 (29 jan. 2019), p. 371. ISSN : 1660-4601. DOI : [10.3390/ijerph16030371](https://doi.org/10.3390/ijerph16030371). URL : <http://www.mdpi.com/1660-4601/16/3/371> (visité le 20/03/2023).
- [86] Walter MUSAKWA et Kadibetso M. SELALA. « Mapping cycling patterns and trends using Strava Metro data in the city of Johannesburg, South Africa ». In : *Data in Brief* 9 (déc. 2016), p. 898-905. ISSN : 23523409. DOI : [10.1016/j.dib.2016.11.002](https://doi.org/10.1016/j.dib.2016.11.002). URL : <https://linkinghub.elsevier.com/retrieve/pii/S235234091630662X> (visité le 20/03/2023).
- [87] Zijing LIN et Wei “David” FAN. « Bicycle Ridership Using Crowdsourced Data : Ordered Probit Model Approach ». In : *Journal of Transportation Engineering, Part A : Systems* 146.8 (août 2020), p. 04020076. ISSN : 2473-2907, 2473-2893. DOI : [10.1061/JTEPBS.0000399](https://doi.org/10.1061/JTEPBS.0000399). URL : <https://ascelibrary.org/doi/10.1061/JTEPBS.0000399> (visité le 20/03/2023).
- [88] Ben JESTICO, Trisalyn NELSON et Meghan WINTERS. « Mapping ridership using crowdsourced cycling data ». In : *Journal of Transport Geography* 52 (avr. 2016), p. 90-97. ISSN : 09666923. DOI : [10.1016/j.jtrangeo.2016.03.006](https://doi.org/10.1016/j.jtrangeo.2016.03.006). URL : <https://linkinghub.elsevier.com/retrieve/pii/S0966692316301168> (visité le 20/03/2023).
- [89] Stefan HUBER et Sven LISSNER. « Disaggregation of aggregate GPS-based cycling data – How to enrich commercial cycling data sets for detailed cycling behaviour analysis ». In : *Transportation Research Interdisciplinary Perspectives* 2 (sept. 2019), p. 100041. ISSN : 25901982. DOI : [10.1016/j.trip.2019.100041](https://doi.org/10.1016/j.trip.2019.100041). URL : <https://linkinghub.elsevier.com/retrieve/pii/S2590198219300417> (visité le 22/03/2023).
- [90] Matthieu ADAM et al. « Conducting Interviews with Maps and Videos to Capture Cyclists’ Skills and Expertise ». In : *Becoming Urban Cyclists : From Socialization to Skills*. Sous la dir. de Matthieu ADAM et Nathalie ORTAR. University of Chester Press, jan. 2022, p. 18-43. URL : <https://hal.science/hal-03552634>.
- [91] Jillian STRAUSS et Luis F. MIRANDA-MORENO. « Speed, travel time and delay for intersections and road segments in the Montreal network using cyclist Smartphone GPS data ». In : *Transportation Research Part D : Transport and Environment* 57 (déc. 2017), p. 155-171. ISSN : 13619209. DOI : [10.1016/j.trd.2017.09.001](https://doi.org/10.1016/j.trd.2017.09.001). URL : <https://linkinghub.elsevier.com/retrieve/pii/S1361920916300736> (visité le 24/03/2023).



- [92] Sirisha KOTHURI et al. « Bicycle and Pedestrian Counts at Signalized Intersections Using Existing Infrastructure : Opportunities and Challenges ». In : *Transportation Research Record : Journal of the Transportation Research Board* 2644.1 (jan. 2017), p. 11-18. ISSN : 0361-1981, 2169-4052. DOI : [10.3141/2644-02](https://doi.org/10.3141/2644-02). URL : <http://journals.sagepub.com/doi/10.3141/2644-02> (visité le 08/02/2023).
- [93] Emily Hentz LEISTER et al. « Understanding bike share reach, use, access and function : An exploratory study ». In : *Sustainable Cities and Society* 43 (nov. 2018), p. 191-196. ISSN : 22106707. DOI : [10.1016/j.scs.2018.08.031](https://doi.org/10.1016/j.scs.2018.08.031). URL : <https://linkinghub.elsevier.com/retrieve/pii/S2210670718307893> (visité le 03/04/2023).
- [94] Yves VAN DE VYVERE. « Les modèles de choix discret en géographie : une introduction ». In : *L'Espace géographique* 24.1 (1995), p. 1-10. ISSN : 0046-2497. DOI : [10.3406/spgeo.1995.3341](https://doi.org/10.3406/spgeo.1995.3341). URL : [https://www.persee.fr/doc/spgeo\\_0046-2497\\_1995\\_num\\_24\\_1\\_3341](https://www.persee.fr/doc/spgeo_0046-2497_1995_num_24_1_3341) (visité le 07/11/2023).
- [95] Joseph BROACH, Jennifer DILL et John GLIEBE. « Where do cyclists ride? A route choice model developed with revealed preference GPS data ». In : *Transportation Research Part A : Policy and Practice* 46.10 (déc. 2012), p. 1730-1740. ISSN : 09658564. DOI : [10.1016/j.tra.2012.07.005](https://doi.org/10.1016/j.tra.2012.07.005). URL : <https://linkinghub.elsevier.com/retrieve/pii/S0965856412001164> (visité le 13/10/2023).
- [96] J G WARDROP et J I WHITEHEAD. « Correspondence. some theoretical aspects of road traffic research. » In : *Proceedings of the Institution of Civil Engineers* 1.5 (oct. 1952), p. 767-768. DOI : [10.1680/ipeds.1952.11362](https://doi.org/10.1680/ipeds.1952.11362). URL : <https://www.icevirtuallibrary.com/doi/10.1680/ipeds.1952.11362> (visité le 12/05/2021).
- [97] Antonio LIMA et al. « Understanding individual routing behaviour ». In : *Journal of The Royal Society Interface* 13.116 (mars 2016), p. 20160021. DOI : [10.1098/rsif.2016.0021](https://doi.org/10.1098/rsif.2016.0021). URL : <https://royalsocietypublishing.org/doi/10.1098/rsif.2016.0021> (visité le 12/05/2021).
- [98] David H DOUGLAS et Thomas K PEUCKER. « ALGORITHMS FOR THE REDUCTION OF THE NUMBER OF POINTS REQUIRED TO REPRESENT A DIGITIZED LINE OR ITS CARICATURE ». In : *Cartographica : The International Journal for Geographic Information and Geovisualization* 10.2 (1<sup>er</sup> déc. 1973), p. 112-122. ISSN : 0317-7173, 1911-9925. DOI : [10.3138/FM57-6770-U75U-7727](https://doi.org/10.3138/FM57-6770-U75U-7727). URL : <https://utpjournals.press/doi/10.3138/FM57-6770-U75U-7727> (visité le 11/10/2023).
- [99] Paul JACCARD. « Distribution de la Flore Alpine dans le Bassin des Dranses et dans quelques régions voisines. » In : *Bulletin de la Societe Vaudoise des Sciences Naturelles* 37 (jan. 1901), p. 241-72. DOI : [10.5169/seals-266440](https://doi.org/10.5169/seals-266440).
- [100] Charles F. F. KARNEY. « Algorithms for geodesics ». In : *Journal of Geodesy* 87.1 (jan. 2013), p. 43-55. ISSN : 0949-7714, 1432-1394. DOI : [10.1007/s00190-012-0578-z](https://doi.org/10.1007/s00190-012-0578-z). URL : <http://link.springer.com/10.1007/s00190-012-0578-z> (visité le 15/09/2023).
- [101] Osama ABU ABBAS. « Comparisons Between Data Clustering Algorithms ». In : *Int. Arab J. Inf. Technol.* 5 (juill. 2008), p. 320-325.
- [102] Martin ESTER et al. « A density-based algorithm for discovering clusters in large spatial databases with noise ». In : *kdd*. T. 96. 34. 1996, p. 226-231.

- [103] Leonard KAUFMAN et Peter J. ROUSSEEUW. *Finding Groups in Data : An Introduction to Cluster Analysis*. en. 1<sup>re</sup> éd. Wiley Series in Probability and Statistics. Wiley, mars 1990. ISBN : 9780471878766 9780470316801. DOI : [10.1002/9780470316801](https://onlinelibrary.wiley.com/doi/book/10.1002/9780470316801). URL : <https://onlinelibrary.wiley.com/doi/book/10.1002/9780470316801> (visité le 28/02/2024).
- [104] Edward W FORGY. « Cluster analysis of multivariate data : efficiency versus interpretability of classifications ». In : *biometrics* 21 (1965), p. 768-769.
- [105] Y. BENGIO, P. SIMARD et P. FRASCONI. « Learning long-term dependencies with gradient descent is difficult ». In : *IEEE Transactions on Neural Networks* 5.2 (mars 1994), p. 157-166. ISSN : 1045-9227, 1941-0093. DOI : [10.1109/72.279181](https://ieeexplore.ieee.org/document/279181/). URL : <https://ieeexplore.ieee.org/document/279181/> (visité le 08/04/2021).
- [106] Sepp HOCHREITER. « The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions ». en. In : *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 06.02 (avr. 1998), p. 107-116. ISSN : 0218-4885, 1793-6411. DOI : [10.1142/S0218488598000094](https://www.worldscientific.com/doi/abs/10.1142/S0218488598000094). URL : <https://www.worldscientific.com/doi/abs/10.1142/S0218488598000094> (visité le 06/04/2021).
- [107] Sepp HOCHREITER et Jürgen SCHMIDHUBER. « Long Short-Term Memory ». en. In : *Neural Computation* 9.8 (nov. 1997), p. 1735-1780. ISSN : 0899-7667, 1530-888X. DOI : [10.1162/neco.1997.9.8.1735](https://direct.mit.edu/neco/article/9/8/1735-1780/6109). URL : <https://direct.mit.edu/neco/article/9/8/1735-1780/6109> (visité le 08/04/2021).
- [108] Kunihiko FUKUSHIMA. « Visual feature extraction by a multilayered network of analog threshold elements ». In : *IEEE Transactions on Systems Science and Cybernetics* 5.4 (1969), p. 322-333.
- [109] Diederik P. KINGMA et Jimmy BA. « Adam : A Method for Stochastic Optimization ». In : (2014). DOI : [10.48550/ARXIV.1412.6980](https://arxiv.org/abs/1412.6980). URL : <https://arxiv.org/abs/1412.6980> (visité le 28/09/2023).
- [110] Geoffrey E. HINTON et al. « Improving neural networks by preventing co-adaptation of feature detectors ». In : (2012). DOI : [10.48550/ARXIV.1207.0580](https://arxiv.org/abs/1207.0580). URL : <https://arxiv.org/abs/1207.0580> (visité le 02/10/2023).
- [111] Luis F. MIRANDA-MORENO, Jillian STRAUSS et Patrick MORENCY. « Disaggregate Exposure Measures and Injury Frequency Models of Cyclist Safety at Signalized Intersections ». In : *Transportation Research Record : Journal of the Transportation Research Board* 2236.1 (jan. 2011), p. 74-82. ISSN : 0361-1981, 2169-4052. DOI : [10.3141/2236-09](http://journals.sagepub.com/doi/10.3141/2236-09). URL : <http://journals.sagepub.com/doi/10.3141/2236-09> (visité le 24/07/2023).
- [112] Katja SCHLEINITZ et al. « (E-)Cyclists running the red light – The influence of bicycle type and infrastructure characteristics on red light violations ». In : *Accident Analysis & Prevention* 122 (jan. 2019), p. 99-107. ISSN : 00014575. DOI : [10.1016/j.aap.2018.10.002](https://linkinghub.elsevier.com/retrieve/pii/S0001457518307590). URL : <https://linkinghub.elsevier.com/retrieve/pii/S0001457518307590> (visité le 24/07/2023).
- [113] Matthew RICHARDSON et Brian CAULFIELD. « Investigating traffic light violations by cyclists in Dublin City Centre ». In : *Accident Analysis & Prevention* 84 (nov. 2015), p. 65-73. ISSN : 00014575. DOI : [10.1016/j.aap.2015.08.011](https://linkinghub.elsevier.com/retrieve/pii/S0001457515300440). URL : <https://linkinghub.elsevier.com/retrieve/pii/S0001457515300440> (visité le 24/07/2023).

- [114] Marilyn JOHNSON et al. « Why do cyclists infringe at red lights? An investigation of Australian cyclists' reasons for red light infringement ». In : *Accident Analysis & Prevention* 50 (jan. 2013), p. 840-847. ISSN : 00014575. DOI : [10.1016/j.aap.2012.07.008](https://doi.org/10.1016/j.aap.2012.07.008). URL : <https://linkinghub.elsevier.com/retrieve/pii/S000145751200262X> (visité le 24/07/2023).
- [115] Milo A. HOLLINGWORTH, Alice J.L. HARPER et Mark HAMER. « Risk factors for cycling accident related injury : The UK Cycling for Health Survey ». In : *Journal of Transport & Health* 2.2 (juin 2015), p. 189-194. ISSN : 22141405. DOI : [10.1016/j.jth.2015.01.001](https://doi.org/10.1016/j.jth.2015.01.001). URL : <https://linkinghub.elsevier.com/retrieve/pii/S221414051500002X> (visité le 25/07/2023).
- [116] Josh ANDRES et al. « "Co-riding With My eBike to Get Green Lights" ». In : *Proceedings of the 2019 on Designing Interactive Systems Conference*. DIS '19 : Designing Interactive Systems Conference 2019. San Diego CA USA : ACM, 18 juin 2019, p. 1251-1263. ISBN : 9781450358507. DOI : [10.1145/3322276.3322307](https://doi.org/10.1145/3322276.3322307). URL : <https://dl.acm.org/doi/10.1145/3322276.3322307> (visité le 24/07/2023).
- [117] Sven FRÖHLICH et al. « BikeNow : a pervasive application for crowdsourcing bicycle traffic data ». In : *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing : Adjunct*. UbiComp '16 : The 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing. Heidelberg Germany : ACM, 12 sept. 2016, p. 1408-1417. ISBN : 9781450344623. DOI : [10.1145/2968219.2968419](https://doi.org/10.1145/2968219.2968419). URL : <https://dl.acm.org/doi/10.1145/2968219.2968419> (visité le 28/10/2022).
- [118] Marco DE ANGELIS et al. « Green wave for cyclists : Users' perception and preferences ». In : *Applied Ergonomics* 76 (avr. 2019), p. 113-121. ISSN : 00036870. DOI : [10.1016/j.apergo.2018.12.008](https://doi.org/10.1016/j.apergo.2018.12.008). URL : <https://linkinghub.elsevier.com/retrieve/pii/S0003687018307385> (visité le 24/07/2023).
- [119] Theodoros ANAGNOSTOPOULOS et al. « Cyclist-aware traffic lights through distributed smartphone sensing ». In : *Pervasive and Mobile Computing* 31 (sept. 2016), p. 22-36. ISSN : 15741192. DOI : [10.1016/j.pmcj.2016.01.012](https://doi.org/10.1016/j.pmcj.2016.01.012). URL : <https://linkinghub.elsevier.com/retrieve/pii/S1574119216000249> (visité le 28/10/2022).
- [120] Pietro GRANDINETTI, Carlos Canudas-de WIT et Federica GARIN. « Distributed Optimal Traffic Lights Design for Large-Scale Urban Networks ». In : *IEEE Transactions on Control Systems Technology* 27.3 (mai 2019), p. 950-963. ISSN : 1063-6536, 1558-0865, 2374-0159. DOI : [10.1109/TCST.2018.2807792](https://doi.org/10.1109/TCST.2018.2807792). URL : <https://ieeexplore.ieee.org/document/8307762/> (visité le 27/11/2023).
- [121] Willy Carlos TCHUITCHEU, Christophe BOBDA et Md Jubaer Hossain PANTHO. « Internet of smart-cameras for traffic lights optimization in smart cities ». In : *Internet of Things* 11 (sept. 2020), p. 100207. ISSN : 25426605. DOI : [10.1016/j.iot.2020.100207](https://doi.org/10.1016/j.iot.2020.100207). URL : <https://linkinghub.elsevier.com/retrieve/pii/S2542660520300433> (visité le 27/11/2023).
- [122] Yisheng Lv LI LI et Fei-Yue WANG. « Traffic signal timing via deep reinforcement learning ». In : *IEEE/CAA Journal of Automatica Sinica* 3.3 (10 juill. 2016), p. 247-254. ISSN : 2329-9266, 2329-9274. DOI : [10.1109/JAS.2016.7508798](https://doi.org/10.1109/JAS.2016.7508798). URL : <http://ieeexplore.ieee.org/document/7508798/> (visité le 02/02/2023).

- [123] Kai Liang TAN et al. « Deep Reinforcement Learning for Adaptive Traffic Signal Control ». In : *arXiv.org perpetual, non-exclusive license* (2019). DOI : [10.48550/ARXIV.1911.06294](https://doi.org/10.48550/ARXIV.1911.06294). (Visité le 27/07/2023).
- [124] Hua WEI et al. « IntelliLight : A Reinforcement Learning Approach for Intelligent Traffic Light Control ». In : *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (19 juill. 2018), p. 2496-2505. DOI : [10.1145/3219819.3220096](https://doi.org/10.1145/3219819.3220096). (Visité le 06/01/2023).
- [125] Wade GENDERS et Saiedeh RAZAVI. « Asynchronous  $n$ -step Q-learning adaptive traffic signal control ». In : *Journal of Intelligent Transportation Systems* 23.4 (4 juill. 2019), p. 319-331. ISSN : 1547-2450, 1547-2442. DOI : [10.1080/15472450.2018.1491003](https://doi.org/10.1080/15472450.2018.1491003). URL : <https://www.tandfonline.com/doi/full/10.1080/15472450.2018.1491003> (visité le 02/02/2023).
- [126] Wade GENDERS et Saiedeh RAZAVI. « Using a Deep Reinforcement Learning Agent for Traffic Signal Control ». In : *arXiv.org perpetual, non-exclusive license* (2016). DOI : [10.48550/ARXIV.1611.01142](https://doi.org/10.48550/ARXIV.1611.01142). (Visité le 01/02/2023).
- [127] Seyed Sajad MOUSAVI, Michael SCHUKAT et Enda HOWLEY. « Traffic light control using deep policy-gradient and value-function-based reinforcement learning ». In : *IET Intelligent Transport Systems* 11.7 (sept. 2017), p. 417-423. ISSN : 1751-9578, 1751-9578. DOI : [10.1049/iet-its.2017.0153](https://doi.org/10.1049/iet-its.2017.0153). URL : <https://onlinelibrary.wiley.com/doi/10.1049/iet-its.2017.0153> (visité le 06/01/2023).
- [128] Song WANG et al. « Deep Reinforcement Learning-Based Traffic Signal Control Using High-Resolution Event-Based Data ». In : *Entropy* 21.8 (29 juill. 2019), p. 744. ISSN : 1099-4300. DOI : [10.3390/e21080744](https://doi.org/10.3390/e21080744). URL : <https://www.mdpi.com/1099-4300/21/8/744> (visité le 02/02/2023).
- [129] Rune ELVIK. « The non-linearity of risk and the promotion of environmentally sustainable transport ». In : *Accident Analysis & Prevention* 41.4 (juill. 2009), p. 849-855. ISSN : 00014575. DOI : [10.1016/j.aap.2009.04.009](https://doi.org/10.1016/j.aap.2009.04.009). URL : <https://linkinghub.elsevier.com/retrieve/pii/S0001457509000876> (visité le 25/07/2023).
- [130] Volodymyr MNIH et al. « Human-level control through deep reinforcement learning ». In : *Nature* 518.7540 (26 fév. 2015), p. 529-533. ISSN : 0028-0836, 1476-4687. DOI : [10.1038/nature14236](https://doi.org/10.1038/nature14236). URL : <https://www.nature.com/articles/nature14236> (visité le 07/07/2023).
- [131] Christopher J. C. H. WATKINS et Peter DAYAN. « Q-learning ». In : *Machine Learning* 8.3 (mai 1992), p. 279-292. ISSN : 0885-6125, 1573-0565. DOI : [10.1007/BF00992698](https://doi.org/10.1007/BF00992698). URL : <http://link.springer.com/10.1007/BF00992698> (visité le 07/07/2023).
- [132] Hado van HASSELT, Arthur GUEZ et David SILVER. « Deep Reinforcement Learning with Double Q-learning ». In : *Proceedings of the AAAI conference on artificial intelligence* (2015). (Visité le 07/07/2023).
- [133] Hado HASSELT. « Double Q-learning ». In : sous la dir. de J. LAFFERTY et al. T. 23. Curran Associates, Inc., 2010.
- [134] Ziyu WANG et al. « Dueling Network Architectures for Deep Reinforcement Learning ». In : sous la dir. de Maria Florina BALCAN et Kilian Q. WEINBERGER. T. 48. Proceedings of Machine Learning Research. New York, New York, USA : PMLR, juin 2016, p. 1995-2003. URL : <https://proceedings.mlr.press/v48/wangf16.html>.

- [135] John SCHULMAN et al. « Proximal Policy Optimization Algorithms ». In : (2017). DOI : [10.48550/ARXIV.1707.06347](https://doi.org/10.48550/ARXIV.1707.06347). URL : <https://arxiv.org/abs/1707.06347> (visité le 27/10/2023).
- [136] John SCHULMAN et al. « High-Dimensional Continuous Control Using Generalized Advantage Estimation ». In : (2015). DOI : [10.48550/ARXIV.1506.02438](https://doi.org/10.48550/ARXIV.1506.02438). URL : <https://arxiv.org/abs/1506.02438> (visité le 27/10/2023).
- [137] Xiaoyuan LIANG et al. « A Deep Reinforcement Learning Network for Traffic Light Cycle Control ». In : *IEEE Transactions on Vehicular Technology* 68.2 (fév. 2019), p. 1243-1253. ISSN : 0018-9545, 1939-9359. DOI : [10.1109/TVT.2018.2890726](https://doi.org/10.1109/TVT.2018.2890726). URL : <https://ieeexplore.ieee.org/document/8600382/> (visité le 06/01/2023).
- [138] Alexander A S GUNAWAN, Deasy Aprilia TANJUNG et Fergyanto E. GUNAWAN. « Detection of Vehicle Position and Speed using Camera Calibration and Image Projection Methods ». In : *Procedia Computer Science* 157 (2019), p. 255-265. ISSN : 18770509. DOI : [10.1016/j.procs.2019.08.165](https://doi.org/10.1016/j.procs.2019.08.165). URL : <https://linkinghub.elsevier.com/retrieve/pii/S187705091931083X> (visité le 20/07/2023).
- [139] Y. LECUN et al. « Backpropagation Applied to Handwritten Zip Code Recognition ». In : *Neural Computation* 1.4 (déc. 1989), p. 541-551. ISSN : 0899-7667, 1530-888X. DOI : [10.1162/neco.1989.1.4.541](https://doi.org/10.1162/neco.1989.1.4.541). URL : <https://direct.mit.edu/neco/article/1/4/541-551/5515> (visité le 11/10/2023).
- [140] Zewen LI et al. « A Survey of Convolutional Neural Networks : Analysis, Applications, and Prospects ». In : *IEEE Transactions on Neural Networks and Learning Systems* 33.12 (déc. 2022), p. 6999-7019. ISSN : 2162-237X, 2162-2388. DOI : [10.1109/TNNLS.2021.3084827](https://doi.org/10.1109/TNNLS.2021.3084827). URL : <https://ieeexplore.ieee.org/document/9451544/> (visité le 11/10/2023).
- [141] Werner BRILON et Wolf LAUBERT. « Priority for public transit in Germany ». In : *Journal of Advanced Transportation* 28.3 (sept. 1994), p. 313-340. ISSN : 01976729, 20423195. DOI : [10.1002/atr.5670280309](https://doi.org/10.1002/atr.5670280309). URL : <https://onlinelibrary.wiley.com/doi/10.1002/atr.5670280309> (visité le 27/07/2023).
- [142] Matteo HESSEL et al. *Rainbow : Combining Improvements in Deep Reinforcement Learning*. 6 oct. 2017. arXiv : [1710.02298](https://arxiv.org/abs/1710.02298)[cs]. URL : <http://arxiv.org/abs/1710.02298> (visité le 31/10/2023).



## FOLIO ADMINISTRATIF

### THESE DE L'INSA LYON, MEMBRE DE L'UNIVERSITE DE LYON.

NOM : MAGNANA  
(avec précision du nom de jeune fille, le cas échéant)

DATE de SOUTENANCE : 14/02/2024

Prénoms : Lucas René André

TITRE : Algorithmes d'apprentissage et données comportementales pour la création d'outils innovants d'incitation au cyclisme

NATURE : Doctorat

Numéro d'ordre : 2024ISAL0019

Ecole doctorale : InfoMaths

Spécialité : Informatique

RESUME : L'augmentation de l'utilisation du vélo comme mode de transport est essentielle pour diminuer la congestion ainsi que la pollution dans les villes. Le développement de nouvelles technologies ainsi qu'une prise de conscience des dangers du dérèglement climatique ont permis entre autres une augmentation du nombre de cyclistes dans les villes au cours des 20 dernières années. Cela s'est naturellement accompagné de nouveaux moyens de récupérer des données comportementales de cyclistes, diversifiant et multipliant les jeux de données existants. Ces données peuvent être utilisées pour affiner les politiques urbaines visant à augmenter la part modale du vélo dans les transports urbains. L'objectif de cette thèse est d'utiliser certaines données comportementales de cyclistes ainsi que des algorithmes récents d'intelligence artificielle pour créer des outils innovants d'aide aux politiques urbaines d'incitation au cyclisme. Dans un premier temps, un état de l'art des différentes sources de données de comportements de cyclistes est dressé. Cet état de l'art permet de se rendre compte de la diversité ainsi que du nombre de jeux de données, mais aussi des difficultés d'utilisation accompagnant chaque source. Cet état de l'art est aussi l'occasion de justifier les sources de données utilisées ensuite. Dans un second temps, une méthode de création de modèles de choix d'itinéraire implicites est développée à partir de traces GPS. Les modèles créés avec cette méthode permettent de générer un itinéraire cyclable à partir d'une origine et d'une destination. Le développement d'un modèle commence par l'identification des segments de route préférés par les cyclistes ayant généré les traces GPS. Un réseau de neurones artificiels choisi ensuite un groupe de segments de route préférés pertinents à partir d'une origine et d'une destination. Une pondération de graphe routier est ensuite effectuée à l'aide des segments de route sélectionnés pour générer un itinéraire cyclable. Les modèles créés permettent de créer des itinéraires plus proches de comportements réels de cyclistes lorsque comparés à d'autres méthodes de génération d'itinéraires. Enfin, un feu de signalisation intelligent permettant de sécuriser le passage des cyclistes à l'intersection qu'il régule est développé. L'utilisation de l'intelligence artificielle permet de séparer les flux de vélos et de véhicules motorisés en limitant l'impact sur le trafic à l'intersection. Les simulations dans lesquelles ce feu est testé ont un volume de trafic de voitures et de vélos réaliste grâce à l'utilisation de données de compteurs. La méthode d'apprentissage utilisée pour la régulation surpasse les performances d'autres méthodes de gestion de feu utilisées en comparaison.

MOTS-CLÉS : Algorithmes d'apprentissage, mobilités urbaines, cyclistes, itinéraires, feu de circulation

Laboratoire (s) de recherche : CITI

Directeur de thèse: Hervé RIVANO

Président de jury :

Composition du jury :

MITTON Nathalie	Directrice de Recherche	INRIA Lille-Nord Europe	Rapportrice
BILLOT Romain	Habilité à Diriger des Recherches	IMT Atlantique	Rapporteur
GARIN Federica	Chargée de Recherche	INRIA Grenoble	Examinatrice
GARCIA Christophe	Professeur des Universités	INSA Lyon	Examineur
DUCOURTHIAL Bertrand	Professeur des Universités	Heudiasyc	Examineur
ROCHE Stéphane	Professeur des Universités	Université Laval	Examineur
RIVANO Hervé	Professeur des Universités	INSA Lyon	Directeur de thèse
CHIABAUT Nicolas	Directeur des Nouvelles Mobilités	Département de la Haute-Savoie	Co-directeur de thèse