



HAL
open science

Prévision de charge dans les transports en commun en situation atypique par traitement d'image

Thomas Bapaume

► **To cite this version:**

Thomas Bapaume. Prévision de charge dans les transports en commun en situation atypique par traitement d'image. Modélisation et simulation. Université Gustave Eiffel, 2023. Français. NNT : 2023UEFL2063 . tel-04550138

HAL Id: tel-04550138

<https://theses.hal.science/tel-04550138>

Submitted on 17 Apr 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.



Prévision de charge dans les transports en commun en situation atypique par traitement d'image

Thèse de doctorat de l'Université Gustave Eiffel

École doctorale mathématiques et sciences et technologies de l'information et de la communication
Discipline : Informatique

Thèse présentée et soutenue à l'Université Gustave Eiffel,
le ..., par : **Thomas Bapaume**

Composition du Jury

| | |
|--|-----------------------|
| Alexandre ALAHI Professeur, Ecole Polytechnique Fédérale de Lausanne | Examineur |
| Laurent HEUTTE Professeur, Université de Rouen Normandie | Rapporteur |
| Vincent GUIGUE Professeur, AgroParisTech - Université Paris-Saclay | Rapporteur |
| Latifa OUKHELLOU Directrice de recherche, Université Gustave Eiffel | Directrice de thèse |
| Etienne COME Chargé de recherche, Université Gustave Eiffel | Co-Directeur de thèse |
| Jérémy ROOS Régie Autonome des Transports Parisiens | Invité |

Remerciement

Abstract

Public transport is essential in shaping urban development and fostering sustainable practices in major cities. As Paris gears up for significant events like the 2024 Olympic Games, the Rugby World Cup, and the expansion of its metro network through the Greater Paris project, the public transport system faces new and unprecedented challenges. While the primary goal remains to provide accurate load forecasts under normal conditions, accounting for atypical situations observed in recent years, such as COVID outbreaks, strikes, or incidents, is equally crucial. This proactive approach ensures a high standard of service that promotes the continued use of public transportation.

In this context, this thesis seeks to build a short-term load prediction model focusing on atypical situations on the network, whether or not they are foreseen or recurring. The model is based on a faithful representation of train movements in an image way. The advantage of this type of data representation is that it encodes all the trains running on a transport line, avoiding any temporal aggregation, thus enabling line-wide prediction and a detailed understanding of load dynamics, particularly in atypical cases.

The task of load prediction is transformed into an image processing problem akin to image inpainting, where we aim to reconstruct a portion of the image. This process is carried out simultaneously for all trains and stations along a metro line. Drawing inspiration from image processing techniques, we introduce a methodology for short-term load forecasting in public transport, which accounts for image-specific aspects like missing data and the spatio-temporal relationships between trains and stations. One of the critical challenges in this thesis is to validate the proposed methodology using a three-year dataset encompassing various atypical situations (such as sporting events, Covid-related disruptions, strikes, incidents, etc.), leveraging diverse architectural approaches ranging from diffusion models to deep learning, particularly U-net models and transformers (specifically Vision Transformers designed for images).

Finally, this research is mainly dedicated to atypical scenarios, encompassing both disruptions within the rail system (such as delays and high loads) and external events impacting the service (including sporting events, lockdowns, and strikes). These unique rail traffic contexts pose a significant challenge for predictive tools since they are not accounted for in standard theoretical models. Our approach involves two key strategies. Firstly, we employ statistical indicators and latent space to label the images, creating specialized sub-datasets. Subsequently, we conduct a differentiated analysis of prediction performance for each image class, explicitly focusing on the most complex cases to evaluate the models' generalization capabilities. Furthermore, we explore two methods for enhancing prediction accuracy. One approach involves oversampling atypical images, while the other generates new ones. These techniques evaluate load prediction models across the resulting datasets, ultimately contributing to a more comprehensive understanding of predicting load under diverse and challenging conditions.

Résumé

Les transports en commun jouent un rôle essentiel dans le développement du tissu urbain en offrant des services de transport contribuant à la durabilité des grandes villes. Avec la tenue d'événements majeurs (Jeux Olympiques 2024, Coupe du Monde de Rugby) et la création de nouvelles lignes de métro dans la capitale parisienne (projet du Grand Paris), les transports en commun seront soumis à des contraintes nouvelles et atypiques. La tâche de prédiction de charge à bord doit bien sûr fournir une prédiction la plus fiable possible en situation nominale mais elle doit aussi tenir compte de ces futures évolutions en considérant les situations atypiques observées au cours des dernières années (Covid, Grève, incident) afin de garantir un niveau de service élevé favorisant l'usage des transports en commun.

Dans ce contexte, ces travaux de thèse cherchent à construire un modèle de prédiction court-terme de charge en focalisant sur les situations atypiques du réseau, qu'elles soient prévues ou non, récurrentes ou non. La modélisation est construite sur une représentation fidèle des mouvements de trains sous la forme d'une image. Une telle représentation des données a l'avantage d'encoder l'intégralité des trains circulant sur une ligne de transport en évitant toute agrégation temporelle, autorisant ainsi une prédiction à l'échelle de la ligne et une compréhension fine des dynamiques de charges en particulier lors des cas atypiques.

La prédiction de charge est ainsi transposée sous la forme d'une problématique de traitement d'image visant à reconstruire une partie de l'image (tâche d'inpainting). Elle est effectuée simultanément pour tous les trains et stations d'une ligne de métro. En s'inspirant des modèles de traitement d'image, nous proposons une méthodologie de prédiction court-terme des charges dans les transports en commun, tenant compte des spécificités propres à l'image comme les données manquantes, l'aspect spatio-temporelle entre trains et stations. Un des challenge de la thèse est de valider la méthodologie proposée sur un jeu de trois ans de données avec une variété de situations atypiques (événements sportifs, Covid, grèves, incidents, ...) tout en utilisant diverses architectures allant de la diffusion à l'apprentissage profond en particulier les modèles U-net et transformers (Vision Transformer pour les images).

Enfin, ces travaux mettent l'accent sur les situations atypiques, regroupant à la fois les perturbations d'une ligne (retards, fortes charges), et des événements impactant la ligne (événements sportifs, confinements, grèves). Ces contextes spéciaux du trafic ferroviaire constituent un réel enjeu pour les outils de prédiction car ils ne sont pas considérés dans l'offre théorique. Dans un premier temps, nous exploitons conjointement des indicateurs statistiques et l'espace latent pour labelliser les images et constituer des sous-bases spécifiques. Nous procédons ensuite à une analyse différenciée des performances de prédiction sur chacune des classes d'images. Il s'agit d'évaluer les modèles pour les cas les plus complexes afin de juger de leur capacité de généralisation. Dans un second temps, nous explorons deux pistes pour améliorer ces performances, l'une se base sur le sur-échantillonnage des images atypiques tandis que l'autre s'attache à générer de nouvelles images atypiques. Nous évaluons les modèles de prédiction de charges sur les bases de données ainsi constituées.

Table des matières

| | |
|---|-----------|
| List of Figures | 11 |
| List of Tables | 12 |
| List of Acronyms | 13 |
| Publications | 14 |
| Introduction | 15 |
| 1 Prévision de la charge des réseaux de transport en commun : méthodologies, applications et données | 21 |
| 1.1 Introduction | 21 |
| 1.2 Contexte applicatif et données mobilisées | 22 |
| 1.2.1 Ligne 9 | 23 |
| 1.2.2 Sources de données de la ligne 9 | 23 |
| 1.2.3 Affluence de la ligne 9 | 25 |
| 1.2.4 Offre et demande | 27 |
| 1.3 Profil par station | 32 |
| 1.3.1 Station Havre-Caumartin | 32 |
| 1.3.2 Station Jasmin | 34 |
| 1.3.3 Situations atypiques | 37 |
| 1.4 Prédiction court terme d'un réseau de transport : les contraintes et objectifs | 39 |
| 1.4.1 Granularité spatiale | 39 |
| 1.4.2 Échantillonnages temporels | 41 |
| 1.4.3 Prédiction de situations atypiques | 42 |
| 1.5 Conclusion | 42 |
| 2 Formalisation du problème de prévision multiple comme une tâche de traitement d'image | 44 |
| 2.1 Introduction | 44 |
| 2.2 Le format image | 45 |
| 2.3 Représentation de l'état d'une ligne de métro dans un format image | 45 |
| 2.3.1 Extension possible à des réseaux | 50 |
| 2.4 Encodage des différentes variables | 51 |
| 2.5 Données manquantes | 53 |
| 2.5.1 Données manquantes aléatoires | 53 |
| 2.5.2 Données manquantes non aléatoires | 54 |
| 2.5.3 Le cas de la ligne 9 du métro Parisien | 54 |

| | | |
|----------|---|-----------|
| 2.6 | Masque de prédiction | 55 |
| 2.6.1 | Définition des masques | 56 |
| 2.7 | Utilisation des images en prédiction | 57 |
| 2.7.1 | Prévision multiple à court terme par inpainting | 57 |
| 2.7.2 | Images d'entrée et de sortie | 58 |
| 2.7.3 | Horizon de prédiction | 59 |
| 2.8 | Conclusion | 59 |
| 3 | Inpainting et architectures profondes pour la prédiction de charge | 60 |
| 3.1 | Tâche d'Inpainting | 60 |
| 3.1.1 | Applications de l'approche Inpainting | 60 |
| 3.1.2 | Adaptation à notre contexte de prédiction | 61 |
| 3.1.3 | Les tâches de type image vers image | 62 |
| 3.2 | Résolution par diffusion | 62 |
| 3.2.1 | Modèle naïf | 63 |
| 3.3 | Approches à base d'apprentissage profond | 64 |
| 3.3.1 | Réseau de neurones | 64 |
| 3.3.2 | Les modèles convolutionnels | 66 |
| 3.3.3 | U-net | 68 |
| 3.3.4 | Modèle génératif adversarial | 68 |
| 3.3.5 | Les modèles de type transformers | 69 |
| 3.3.6 | Fonction de coût | 73 |
| 3.4 | Limitation de comparaison | 74 |
| 3.5 | Conclusion | 75 |
| 4 | Une approche de type image pour la prédiction des charges d'une ligne de métro | 76 |
| 4.1 | Introduction | 76 |
| 4.2 | Expérimentation sur la ligne 9 du métro Parisien | 76 |
| 4.2.1 | Architectures des modèles de prédiction | 77 |
| 4.2.2 | Modèles de réseaux de neurones profond | 77 |
| 4.2.3 | Modèles de réseaux de neurones Convolutifs | 77 |
| 4.2.4 | Modèles d'attention et Transformer | 78 |
| 4.2.5 | Modèle U-net | 78 |
| 4.2.6 | Modèle U-transformer | 78 |
| 4.2.7 | Modèle Channel ViT | 79 |
| 4.2.8 | Benchmark de prédiction | 79 |
| 4.2.9 | Taux d'échantillonnage | 79 |
| 4.2.10 | Bases d'apprentissage et de test | 81 |
| 4.2.11 | Contraintes lors de l'entraînement et plan d'expérience | 82 |
| 4.3 | Résultats de prédiction | 82 |
| 4.3.1 | Résultats globaux | 82 |
| 4.3.2 | Résultats par pixels | 84 |
| 4.3.3 | Analyse spatio-temporelle des résultats | 86 |
| 4.3.4 | Analyse des scores d'attention | 88 |
| 4.4 | Conclusion | 89 |

| | | |
|----------|---|------------|
| 5 | Labellisation d'images et amélioration des prédictions en situations atypiques | 91 |
| 5.1 | Introduction | 91 |
| 5.1.1 | Protocole pour la gestion des cas atypiques | 92 |
| 5.2 | Données calendaires et évènements atypiques | 93 |
| 5.3 | Détection statistique des cas atypiques | 93 |
| 5.3.1 | Statistiques de charges | 94 |
| 5.3.2 | Statistiques sur les temps de parcours | 96 |
| 5.3.3 | Pixels manquants | 97 |
| 5.4 | Exploration et annotation semi-automatique de l'espace latent | 99 |
| 5.4.1 | Outliers au sein des groupes nominaux | 103 |
| 5.5 | Impact des différentes situations sur les performances | 103 |
| 5.5.1 | Résultats | 104 |
| 5.6 | Augmentation de données et sur-échantillonnage | 106 |
| 5.6.1 | Sur-échantillonnage des cas atypiques | 107 |
| 5.6.2 | Génération d'images pseudo-aléatoires | 111 |
| 5.7 | Conclusion | 113 |
| | Conclusion and Perspectives | 115 |
| | Résumé détaillé de la thèse en français | 117 |
| | Bibliography | 119 |

Table des figures

| | | |
|------|--|----|
| 2 | Panneau d’affichage à quai de l’information voyageur à Saint Lazare [44] | 16 |
| 3 | Situation de forte charge sur la ligne 9 [29] | 17 |
| 4 | Carte des sites olympiques de Paris 2024 [21] | 18 |
| 1.1 | Nombre d’entrant sur la ligne 9 par jour d’exploitation depuis le 1er Janvier 2019. | 22 |
| 1.2 | Plan de la ligne 9 du métro. | 23 |
| 1.3 | Service réalisé le 15 février 2021 pour les trains au départ de Pont de Sèvres. | 24 |
| 1.4 | Graphe de Marey pour le 15 février 2021 pour les trains partant de Pont de Sèvres. La taille et la couleur dénotent le nombre de passagers par trains. | 25 |
| 1.5 | Parcours complet d’un train pour la journée du 15 février 2020. Chaque couleur représente un parcours entre deux terminus. | 26 |
| 1.6 | Boxplot du nombre d’entrants (validation télé-billettique) par jour et par station. | 26 |
| 1.7 | Boxplot du nombre de passagers par trains sur la ligne, i.e. serpent de charge. | 27 |
| 1.8 | Comparaison des variations du nombre de passagers à bord des trains selon le sens de circulation. La variation de charge représente le gain ou la perte de passagers au départ d’une station. Les couleurs représentent le sens de circulation. | 28 |
| 1.9 | Répartition des temps d’attente pour la ligne 9. De gauche à droite, les profils de temps d’attente pour les jours ouvrés et les jours de week-end. Les couleurs représentent les intervalles de temps de départ de trains. | 29 |
| 1.10 | Profil du nombre de passagers à bord des trains selon le sens de circulation au cours de la journée. Les couleurs représentent le sens de circulation, vert pour les départs de pont de Sèvres et Rouge pour les départs de Mairie de Montreuil (ce code couleur est commun à toutes les figures introduisant le sens de circulation). | 30 |
| 1.11 | Comparaison entre 2019 et 2019 de l’offre réel en nombre de missions par heures de pointes et la charge par rame durant les mêmes périodes. | 31 |
| 1.12 | Comparaison entre 2019 et 2022 de la charge des trains pour les vendredi pour chacun des sens de circulation. En vert les trains au départ de Pont de Sèvres, en Rouge les trains au départ de Mairie de Montreuil. | 32 |
| 1.13 | Médianes, 1 ^{er} et 3 ^e quartiles du nombre de passagers à bord des trains au départ de la station Havre-Caumartin par année entre 2019 et 2022. | 33 |
| 1.14 | Médianes, 1 ^{er} et 3 ^e quartiles du nombre de validation télé-billettique à la station Havre-Caumartin par année entre 2019 et 2022. | 33 |

| | | |
|------|---|----|
| 1.15 | Médianes, 1 ^{er} et 3 ^e quartiles de l'évolution du nombre de passagers (delta de charge) à bord d'un train à Havre-Caumartin. La couleur définit le sens de circulation. Une valeur négative indique que le train a perdu des usagers et une valeur positive indique un gain de passagers. | 34 |
| 1.16 | Médianes, 1 ^{er} et 3 ^e quartiles du nombre de passagers à bord des trains au départ de la station Jasmin par année entre 2019 et 2022. | 35 |
| 1.17 | Médianes, 1 ^{er} et 3 ^e quartiles du nombre de validation télé-billettique à la station Jasmin par année entre 2019 et 2022. | 36 |
| 1.18 | Répartition des temps d'attente pour la ligne 9. De gauche à droite, les profils de temps d'attente pour les jour de weekend et jour ouvrée. Les couleurs représentent les intervalles de temps de depart de trains. | 36 |
| 1.19 | Graphe de Marey pour une situation atypique sur la ligne 9 le 17 février 2020. Entouré en pointillé l'allongement des temps d'attente et en trait plein des demi-tours effectué par des trains. | 38 |
| 2.1 | Parcours de l'ensemble des trains de la ligne 9 de la station Pont de Sèvres à Mairie de Montreuil. La couleur est associée au nombre de passagers à bord des rames. | 45 |
| 2.2 | Courses des trains entre Pont de Sèvres et Mairie de Montreuil entre 15 :30 et 19 :00 le 15 février 2020. Une ligne représente le parcours d'un train et les points un arrêt. | 46 |
| 2.3 | Schéma présentant la transposition d'un ensemble de quatre trains en image, les couleurs encodent les trains. | 47 |
| 2.4 | Image typique de charge | 48 |
| 2.5 | Relation d'ordre entre les différents pixels de l'image. | 48 |
| 2.6 | Canaux d'une image de trafic de la ligne 9. (A) canal de charges passagers, (B) canal des intervals de départ, (C) canal des validations télé-billettiques, (D) canal des temps de parcours | 49 |
| 2.7 | Schéma d'une image représentant un réseau de transport complet. | 51 |
| 2.8 | Masque indiquant en noir les stations fermées. | 53 |
| 2.9 | Image de charge avec différents cas de données manquantes en gris | 54 |
| 2.10 | Image temps réel $I^{in}(t)$ de la ligne 9 du métro. Les pixels gris indiquent les départs de trains futurs au moment t de sa génération. En noir les données manquantes de l'image. | 56 |
| 2.11 | A gauche : Une image avec l'horizon passé en gradient de gris avec la cible de prévision en couleur. À droite, le masque associé qui code la position de chaque pixel à prédire. Les pixels gris à droite de l'image correspondent aux pixels non prédits. | 57 |
| 2.12 | Schéma de principe du fonctionnement de la tâche de prédiction multiple. | 58 |
| 3.1 | Exemple d'inpainting dans une image. La région en blanc dans l'image représente la cible à combler par la tâche d'inpainting | 61 |
| 3.2 | Schéma de la position des pixels dans le cadre d'une diffusion | 63 |
| 3.3 | | 64 |
| 3.4 | Schéma de principe d'une architecture type de CNN | 66 |
| 3.5 | Schéma de principe du fonctionnement d'une couche de maximum pooling de taille 2. | 67 |
| 3.6 | Schéma de principe d'une architecture de type U-net.[6] | 68 |
| 3.7 | Transformer Encoder [72] | 70 |

| | | |
|------|--|-----|
| 3.8 | Schéma de principe de l'architecture de type U-Transformer.[5] | 72 |
| 3.9 | Schéma de principe de l'architecture de type Channel Vision Transformer .[5] | 73 |
| 4.1 | Évolution des erreurs du modèle de prévision de type U-net en fonction de différents taux d'échantillonnage | 80 |
| 4.2 | Prédiction du modèle U-Transformer pour le 25 juin 2020 pour la station de Havre Caumartin en heure creuse. Les lignes verticales représentent les départs de trains à la station. | 84 |
| 4.3 | Prédiction du modèle U-Transformer pour le 25 juin 2020 pour la station de Havre Caumartin en heure de pointe. Les lignes verticales représentent les départs de trains à la station. | 85 |
| 4.4 | Visualisation par pixel de la tâche de prédiction | 86 |
| 4.5 | Erreur en WMAPE par heure de la journée | 87 |
| 4.6 | Erreur en WMAPE par station de la ligne (sens de circulation de gauche à droite) | 87 |
| 4.7 | Visualisation des scores d'attention pour un seul pixel dans 4 têtes du modèle de transformation du canal de vision utilisé pour la prédiction de la charge des passagers : (a) $I(t)$ étudié avec le pixel cible; (b) scores d'attention du canal de charge des passagers; (c) scores d'attention du canal de temps de trajet; (d) scores d'attention du canal de temps d'attente; [une couleur rouge plus intense indique un score d'attention plus élevé] | 88 |
| 5.1 | Image de Grève | 93 |
| 5.2 | Distribution de la charge maximale notée cm | 94 |
| 5.3 | Distribution de la variation absolue moyenne de la charge entre deux départs de train vc | 95 |
| 5.4 | Image de forte Charge | 96 |
| 5.5 | Image du groupe atypique <i>Variation</i> | 96 |
| 5.6 | Distribution des temps de parcours maximaux par image | 97 |
| 5.7 | Image des temps de parcours lors de forts délais | 98 |
| 5.8 | Image avec une forte proportion de données manquantes | 98 |
| 5.9 | | 99 |
| 5.10 | Représentation t-SNE de l'espace latent du modèle U-transformer, les couleurs encodent l'heure de la journée | 100 |
| 5.11 | Représentation t-SNE de l'espace latent de la charge de passagers correspondant au modèle U-Transformer. Des couleurs différentes distinguent les 20 classes obtenues à partir du regroupement par K-means, des labels manuels permettent de situer les différents groupes extraits. | 101 |
| 5.12 | Image issues du groupe <i>Evènements</i> | 102 |
| 5.13 | Image issues du groupe <i>Correspondances</i> | 102 |
| 5.14 | Image issues du groupe <i>Perturbations</i> | 103 |
| 5.15 | Exemple d'une image de charge se variante pseudo-randomisée | 112 |

Liste des tableaux

| | | |
|-----|---|-----|
| 1.1 | Résumé et exemple des 3 types de situations atypiques. | 37 |
| 4.1 | The settings of all the benchmarked models. | 80 |
| 4.2 | Résultats de l’inpainting et de la prévision des prochains départs en pourcentage d’erreur absolue moyenne pondérée (WMAPE) et en erreur quadratique moyenne (RMSE). | 83 |
| 5.1 | Nombre et proportion des images de groupes atypiques à la fois pour la base d’apprentissage et la base de test. | 104 |
| 5.2 | Résultats de la prévision de la charge de passagers sur la base du pourcentage d’erreur absolue moyenne pondérée (WMAPE) pour l’ensemble des modèles et les situations atypiques. | 105 |
| 5.3 | Résultats des modèles par groupes d’images atypiques en équilibrant le nombre d’images par classes | 109 |
| 5.4 | Résultats des modèles par groupes d’images atypiques selon le sur-échantillonnage des images de <i>Outliers</i> , <i>Perturbations</i> , <i>Délais</i> | 110 |
| 5.5 | Résultats des modèles par groupes d’images atypiques selon le sur-échantillonnage de toutes les images atypiques | 110 |
| 5.6 | Comparaison des performances pour les cas atypiques entre les modèles U-net, U-Transformer et Unet 50% (Unet avec une base d’apprentissage comprenant 50% d’image généré) | 113 |

List of Acronyms

| | |
|---------------------|---|
| RATP | Régie Autonome des Transports Parisiens |
| TP | Transport Public |
| I | Image de trafic ferroviaire |
| t | temps |
| $I(t)$ | Image de trafic à l'instant t |
| $I^{in}(t)$ | Image d'entrée |
| $\hat{I}_{pred}(t)$ | Image de prédiction |
| $\hat{I}_{pred}(t)$ | Image de prédiction |

Publications

Revue internationale à comité de lecture

1. Thomas Bapaume, Etienne Côme, Jérémy Roos, Mostafa Ameli and Latifa Oukhellou, "Image inpainting and deep learning to forecast short-term train loads", IEEE Access, 2021, vol. 9, pp. 98506-98522.
2. Thomas Bapaume, Etienne Côme, Jérémy Roos, Mostafa Ameli and Latifa Oukhellou, "Forecasting passenger flows and headway at train level for a public transport line : Focus on atypical situations", Transport Research Part C, 2023, vol. 13.

Conférences internationales

1. Thomas Bapaume, Etienne Côme, Jérémy Roos, Mostafa Ameli and Latifa Oukhellou, "Passenger flow forecasting framework based on vision transformer and inpainting : Application to a public transport system" TRISTAN XI, Mauritius, 2022.
1. Thomas Bapaume, Etienne Côme, Jérémy Roos, Mostafa Ameli and Latifa Oukhellou, "Vision transformer and inpainting approach for short-term passenger loads forecasts during disruptive events" Transport Research Board (TRB) 102nd annual conference, Washington DC., United States of America, 2023.

Introduction

Les transports en commun (TC) font partie des éléments clés pour que l'urbanisation et le développement des villes se fasse de manière pertinente. Ils offrent un service de transport de masses (mass transit), qui permet le développement d'espaces d'attractivité économique, social et démographique en fournissant aux usagers des services efficaces et peu chères [50]. Ils occupent bien entendu un rôle primordial dans le développement durable des grandes villes qui exige de passer de modes de transport individuels plus coûteux et polluants à des modes de transport plus respectueux de l'environnement. Pour le cas de la France, les transports en commun et service de transport représentent 2,5% des dépenses des ménages en 2019 contre 11% pour les véhicules individuels ([24]). Une dépense des ménage qui est en croissance avec une augmentation annuelle de 4% depuis 2003.

Toutefois, les transports en commun ont connu, suite à la crise du Covid 19 à partir de 2020, une évolutions des mobilités urbaines et de leur usage par la démocratisation du télétravail. La première conséquence est la baisse du volume total de voyageur qui s'établie en 2022 à un niveau entre 80 % à 85 % du niveau de 2019 (selon l'institut Paris Région [65]). Vient s'ajouter également une modification des comportement des usagers avec l'apparition des "jours de pointe" en raison du télé-travail. Il s'agit des mardi et jeudi pour la capitale française avec 18% de télétravail représentant les jours avec la fréquentation maximale. A l'opposé, les autres jours ouvrés de la semaine voient une baisse de la fréquentation des usagers avec comme exemple les vendredi où l'on recense en 2022 38% de télétravail. Ceci représente un écart de 18 % du nombre d'usagers utilisant les TC entre les mardi et vendredi (on notait un écart de 7% avant 2019 [71]).

Cette évolution des comportements de mobilité pose la question pour des système de transport intelligent (ITS) de la gestion de contextes atypiques dans la mise en place des outils comme la prédiction qui est le sujet de ces travaux. La séparation des heures de pointes et des heures creuses au sein d'une journée n'est plus suffisante pour caractériser les profils journaliers de demande d'une ligne de transport en commun ni pour évaluer les performances opérationnelles.

En plus de ces évolutions induisant des changements profonds et à moyen/long terme de la structure de la demande, le problème de la prédiction à court terme, qui va occuper la majeure partie de cette thèse, doit prendre en compte des événements complexes tels que des congestions ou des perturbations, afin d'anticiper l'évolution des flux de passagers et de permettre l'ajustement des horaires pendant ces périodes perturbées.

Prédire les charges, objectifs et enjeux

La prédiction à court terme de charge ou de temps de parcours fait partie des outils standards des ITS depuis plusieurs dizaines d'années. L'intérêt derrière la prédiction est multiple. Premièrement, elle peut fournir une information riche tant aux opérateurs des

lignes qu'aux passagers. En ce qui concerne le cadre parisien, seul les temps d'attente théorique en station pour quelques départs comme dans l'image 2 sont pour le moment communiqués au usagers et disponibles pour l'aide à la régulation. Deuxièmement, elle peut être utilisée pour l'optimisation des flux de passagers ou de l'offre de transport en temps réel. Enfin, une prédiction à court terme peut aider à la résolution de contextes atypiques.



FIGURE 2 – Panneau d’affichage à quai de l’information voyageur à Saint Lazare [44]

La prédiction de charge connaît un intérêt accru ces dernières années grâce à la mise en valeur et la mise à disposition de nouvelles données de transport en temps quasi réel, ainsi que l’augmentation des capacités de traitement des données.

La prédiction à court terme du trafic des transports en commun peut être formalisée à différentes échelles : microscopiques (station, véhicule) ou macroscopique (réseau entier). De plus, plusieurs challenges [75] restent encore à relever ; on peut définir trois catégories suivant qu’ils concernent : les données, la méthodologie, ou bien encore la compréhension et l’explication des résultats. Pour les données de transport, il s’agit en particulier de mettre en valeur les caractéristiques spatiales et temporelles tout en ayant une agrégation pertinente pour les cas d’étude. Pour la méthodologie, on pose la question de la comparaison des modèles d’intelligence artificiel qui pour la plupart sont spécifiques à des zones urbaines distinctes ayant chacune leurs contraintes et caractéristiques. Enfin le dernier challenge consiste à améliorer la transparence des approches basées sur les données. Il s’agit d’expliquer les résultats des modélisations pour la prise de décision et de mettre en lumière les forces et faiblesses des modèles au regard de situations atypiques opérationnelles réelles.

Prédire les cas sensibles et rares

Dans le cadre des travaux de cette thèse, nous allons particulièrement nous intéresser aux cas atypiques, ceux-ci sont en effet un réel enjeu pour la mise en production des modèles de prédiction dans le contexte des ITS, le cas nominal étant déjà connu et traité par l’offre théorique par les opérateurs franciliens. Les cas atypiques peuvent impacter de

diverses manières les transports en commun mais ils se dénotent par une congestion à bord des trains et en station mais également par des temps de parcours allongés. Ces situations atypiques ont pour principales conséquences une baisse du bien être à bord des trains [42] et la redirection des trajets sur d'autres lignes de transport en commun. Cependant, les cas atypiques peuvent avoir des effets inverses comme le COVID 19 qui a provoqué une chute de la fréquentation des transports en commun durant les confinements (La charge des trains était presque nulle durant la période de mars-avril 2020).



FIGURE 3 – Situation de forte charge sur la ligne 9 [29]

D'un point de vue opérationnel, ce sont les situations atypiques ayant un impact sur la qualité de service et le bien être des passagers qui sont à cibler pour la prédiction de charge des transports en commun. Les causes de changement de contextes peuvent être liées à des événements internes au système de transport (incident, panne, fermeture de stations) ou à des événements externes (grèves, événements culturels). Enfin, l'intérêt pour les situations atypiques est double car il évalue la robustesse des méthodologies de prédiction en tenant compte des cas opérationnels présents dans les données mais également les cas futurs liés à des événements ou évolution des mobilités ("jours de pointes", ou prolongement de ligne, diminution du télétravail). Le but est d'utiliser les cas atypiques afin de juger de la généralisation d'un modèle de prédiction.

Faire face à des contextes futurs et inconnus

L'île de France connaît depuis ces dernières années un ensemble de travaux et d'aménagements de son réseau de transports avec l'aboutissement des premières lignes du "Grands Paris". Il s'agit d'un agrandissement du réseau existant (RER E, ligne 14 [62]) et la construction de nouvelles lignes (ligne 15 à 18) pour desservir les villes de proches banlieues. Ce plan d'urbanisation [20] a pour but d'intégrer des régions sans réseau ferré et de fournir un réseau de transport de banlieue à banlieue sans nécessairement passer par Paris comme les lignes de RER actuelles. L'agrandissement du système de métros parisiens implique une évolution des flux de passagers qui sera effective à chaque mise en route d'une nouvelle ligne. La mise en place de stratégie de prédiction de charges ne peut

compte de contexte imprévisible. La seule distinction entre heures de pointes et heures creuses n'est pas suffisante pour juger des performances réelles d'un modèle de prédiction de charge. Il est nécessaire d'élargir le spectre des contextes complexes de charge à bord.

Motivations et Contributions

Mettre en avant les trains dans les prédictions

La première tâche menée au sein de cette thèse est de tenir compte des trains en tant que base de l'information ; il s'agit ici d'éviter l'agrégation de données à un intervalle temporel fixe. Une telle agrégation ne permet pas de tenir compte des déplacements réels des trains. Connaître les mouvements de chacun des trains permet de visualiser avec précisions les contextes de charge atypiques, d'en connaître l'origine spatiale et temporelle tout en observant son impact sur l'ensemble des trains ainsi que les choix opérationnelles mis en place.

Pour ce faire, ces travaux ont opté pour l'usage d'un format original, le format image qui permet de conserver les dynamiques réelles des trains le long d'une ligne de métro. Les images offrent également un format de données malléable qui s'adapte à différentes lignes de transport en commun et intègre différentes sources de données. De plus, ce choix guidé par notre volonté de travailler à l'échelle des trains ouvre la voie à l'utilisation des outils de computer vision ayant émergé récemment.

Utiliser les forces de la computer Vision pour la prédiction

La prédiction de charge des transport en commun est une tâche courante dans la littérature. Elle fournit un outil de décision opérationnel pour un système de transport complexe. Cette problématique est majoritairement réalisée via des approches d'apprentissage profond en utilisant des procédés de récurrence pour l'aspect temporelle ou par l'usage de graphes (simulant la réalité physique du réseau) pour l'aspect spatiale. Cependant, ces méthodes reposent sur des données agrégées par intervalle de temps ne permettant pas de tenir compte des déplacements réels des trains.

La transposition en image d'une ligne de métro impose l'usage de méthodes issues d'un tout autre domaine qu'est le traitement d'image (computer vision en anglais). L'intégration d'image tant en entrée qu'en sortie des modèles implique l'usage de méthodologie de computer vision qui s'éloigne de l'état de l'art pour la prédiction de charge des transports en commun (modèle de graphe, réseaux de neurones récurrents). Il s'agit alors de transcrire des procédés de traitement d'image à la tâche spécifique de la prédiction de charge des transport en commun.

Focaliser la prédiction sur les cas atypiques

Comme nous allons le voir, les mobilités se sont transformées au cours de ces dernières années avec l'évolution des comportement des usagers en particulier avec le développement du télé-travail. Comme nous l'avons déjà expliqué La prédiction de charge ne peut plus seulement être résumé aux phénomène d'heure de pointe et d'heure creuses. Il faut pouvoir tenir de compte de nouvelles saisonnalités comme l'apparition des "jours de pointes". On peut ajouter la tenu d'évènement éphémère comme les futurs Jeux Olympiques ou des contextes atypiques comme les incidents ou grèves impactant le bien être à bord des rames ainsi que la qualité du service de transport. Ainsi, réaliser une prédiction des charges ne

peut pas être évalué uniquement sur les seules distinctions de jour ouvrés et/ou heure de pointes car d'une part ces situations sont déjà comprises dans l'offre de transport théoriques et d'autre part ne témoigne pas de la robustesse des modèles face à des contextes de charge inconnues. L'évaluation d'un modèle doit intégrer comme enjeu la gestion de situations atypiques comme une des cibles de prévision et ne pas seulement tenir compte de la performance général du modèle car ces lors de situations atypiques que la prévision de charge peut apporter une plus value tant pour les opérateurs (gestion des flux, anticiper l'impact d'une situation atypiques pour une ligne) que pour les usagers (bien-être, éviter les zones impactées lors de la planification des trajets). La labellisation des caractéristiques des trains devient alors obligatoire pour juger des performances réelles d'un modèle de prédiction.

Faire progresser les modèles pour des cas rare ou non observée

La labellisation des contextes des parcours des trains peut également servir à d'autres tâches comme la détection de cas atypiques ou afin d'améliorer les modèles pour les situations ainsi labellisées. Les situations atypiques deviennent alors le vrai enjeu des modèle de prévisions. L'objectif est alors de prédire au mieux les situations atypiques sans pour autant dégrader les performances pour les situations nominales.

Structuration du manuscrit

Le premier chapitre introduit le cas d'étude de la ligne 9 du métro parisiens dans l'optique de la prédiction de charge. Le deuxième chapitre présente la génération d'image d'un réseau de transport public intégrant les contraintes opérationnelles et techniques de l'opérateur de transport. Le troisième est dédié à la modélisation de la prévision de charge par des techniques de computer vision et plus particulièrement via de l'inpainting. Le quatrième chapitre présente les résultats pour la ligne 9 du métro parisien avec l'approche par inpainting. Le dernier chapitre propose une analyse sur les situations atypique et détaille des pistes en vue d'améliorer les performances de prédiction dans les cas les plus complexes de transport. Enfin nous concluons et présentons les perspectives de ces travaux.

Chapitre 1

Prévision de la charge des réseaux de transport en commun : méthodologies, applications et données

1.1 Introduction

Dans les systèmes de transport en commun, les méthodes de prédiction dépendent principalement de deux paramètres : les orientations et objectifs de l'opérateur de transport et les données disponibles pour effectuer les prédictions. Les choix opérés sont limités par les données, en particulier leur structure ou leur disponibilité. Par exemple, une prédiction à court terme ne peut être réalisée avec des données quotidiennes ou mensuelles, elle nécessite une remontée quasi temps réel des données. Ces contraintes se reflètent sur la diversité d'approches et de modèles mis en place dans la littérature. On retrouve ainsi comme la prédiction des charges au niveau d'une station [63], mais également la prédiction peut avoir comme focus les véhicules d'une ligne de bus ou de métro [18], enfin des approches macroscopique cherchant à prédire la totalité ou un sous ensemble d'un réseau de transport [14] [6] par des représentations intelligentes des données (graphes, images).

L'objectif dans ces travaux de thèse est de fournir une méthode applicable aux systèmes d'informations des opérateurs de Transport et plus particulièrement à celui de la RATP. L'opérateur RATP met à disposition plusieurs sources de données représentant divers aspects d'un réseau de transport en commun. On dénote deux grandes familles de données, celles relatives aux trains et à leur déplacement et celles relatives au passagers telles que les données de charge ou de télé-billettique. La première décrit le parcours réalisé des trains ainsi que les temps d'attente à chaque station. La seconde fournit des indicateurs de charges passagers dans les trains ainsi que des comptage de validations télé-billettiques unitaires.

Le focus est porté ici sur les données de charges passagers à bord. Cette donnée définit à l'échelle d'une voiture ou d'une rame, représente la cible de prédiction. La charge passager présente deux enjeux notables, le premier concerne la qualité de service et vise à fournir aux usagers des transport en commun une prédiction fiable de la charge des trains leur permettant ainsi d'optimiser leurs trajets pour voyager dans des trains moins saturés. Le second enjeu concerne l'opérateur de transport qui peut anticiper les fortes affluences à bord des rames, lui permettant ainsi d'adapter l'offre lors de situations atypiques.

Pour la prédiction du nombre de passager par rame, il faut tenir compte des spécificités du réseau parisien au cours de ces 4 dernières années. La métropole de Paris a connu plusieurs périodes atypiques avec une succession d'évènements ayant eu un fort impact sur la demande et l'offre des transports en commun.

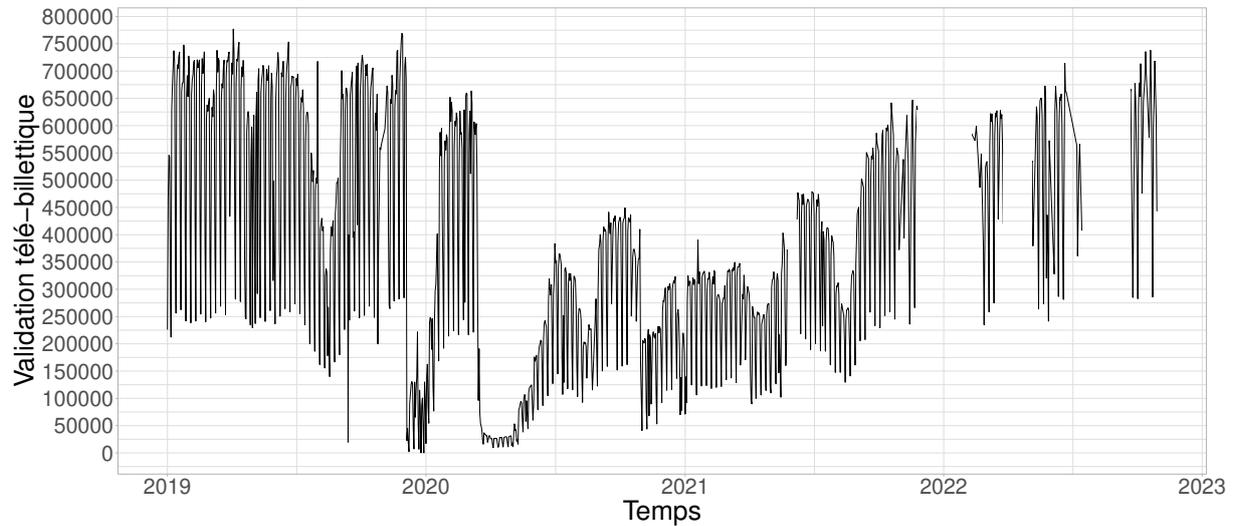


FIGURE 1.1 – Nombre d’entrant sur la ligne 9 par jour d’exploitation depuis le 1er Janvier 2019.

Le réseau de transport a d’abord connu fin 2019 une des grèves les plus importantes depuis les années 90 qui a entraîné une chute du nombre de validations télé-billettiques. Cette grève a été suivie comme illustré dans la figure 1.1¹ par plusieurs périodes de faible fréquentation dont la plus notable est celle de mars 2020. Ces périodes dénotent la crise du Covid et les confinements successifs ayant eu lieu en région Parisienne. De plus, à partir du premier confinement, la demande augmente progressivement pour un retour vers une situation normale comme en 2019 (avant la crise sanitaire).

La figure 1.1 représente l’évolution du nombre de validations sur la ligne 9 entre 2019 et 2023. On note la forte variabilité des charges passagers qui induit la nécessité de prédire les charges dans divers contextes et situations. En plus d’évènements exogènes connus, on retrouve les situations atypiques du réseau comme les perturbations ou incidents. Ces situations atypiques sont des évènements rares pouvant avoir des conséquences locales (quelques stations) voire plus globales sur l’ensemble de la ligne. Enfin, l’ouverture du grand Paris et la tenue des Jeux Olympique en 2024 impose de généraliser des modèles de prédiction à des contextes nouveaux et inconnus.

Pour introduire les travaux réalisés durant cette thèse nous allons, dans la Section 1.2 présenter les différentes sources de données fournies par la RATP et leurs caractéristiques. Ensuite, la Section 1.4 présentera l’état de l’art de travaux ayant trait à la prédiction à court terme dans le domaine du transport public.

1.2 Contexte applicatif et données mobilisées

Nous présentons dans cette section les différentes sources de données ainsi que leurs caractéristiques spatiales et temporelles. Nous focalisons notre analyse uniquement sur

1. Certaines données en 2022 ne sont pas disponibles

la ligne 9 du métro Parisien, laquelle constitue avec les lignes 2 et 5 les lignes de métro ayant la plus grande maturité des données à la fois au niveau de l'offre et de la demande (disponibilité de données de pesées en particulier permettant une mesure fiable de la charge).

1.2.1 Ligne 9

Dans le cadre de ces travaux, seule la ligne 9 du métro Parisien sera considérée. Cette ligne de métro est exploitée par la Régie Autonome des Transports Parisiens (RATP) et relie l'ouest et l'est de Paris. Comme le montre la figure 1.2, cette ligne comprend 37 stations, ce qui en fait la plus longue ligne du métro parisien avec 136 millions de passagers pour l'année 2019.



FIGURE 1.2 – Plan de la ligne 9 du métro.

Le parcours de la ligne traverse les quartiers d'Auteuil et de Boulogne à l'ouest pour rejoindre Montreuil à l'est. Elle dessert des régions touristiques comme le Trocadéro et des gares multimodales (train de banlieue, bus, métro) comme Nation ou Havre-Caumartin (Reliée à Saint-Lazare) qui représentent les deux principales stations de la ligne en terme de flux voyageur. De plus, on retrouve des stations accueillant des événements sportifs comme la station Porte de Saint-Cloud pour l'accès au Parc des Princes (stade de football du Paris-Saint-Germain) et la station "Jardins d'Auteuil" pour le tournoi de tennis Roland Garros.

La ligne est à l'image du réseau dense de Paris : elle présente la particularité d'être interconnectée avec la presque intégralité du réseau de métro. Seule la ligne 12 n'est pas reliée directement à la ligne 9, elle l'est via le hub de Havre-Caumartin et Saint-Lazare. De plus, Havre-Caumartin avec Saint Augustin sont les points d'entrée de la ligne avec la Gare de Saint-Lazare qui est une des 7 gares principales² de Paris opéré par SNCF pour des trains régionaux et de banlieue. Elle est la troisième station en volume de passager derrière Gare du Nord et Gare de Lyon. On note également la présence de la station République qui est, après Chatelet les Halles, la station de métro avec le plus de correspondances (5 au total). On retrouve également des connexions avec d'autres transports collectifs. Pour les lignes de transport plus lourd, on dénombre des correspondances avec les 3 RER (Réseau Express Régional) A, C et E qui connectent principalement l'Ouest et l'Est Parisien. On note également deux lignes de Tramway (le T2 à pont de sèvres et T3b à Porte de Montreuil) et le RoissyBus à Havre-Caumartin.

1.2.2 Sources de données de la ligne 9

Sur la ligne 9, la RATP dispose de diverses données relatives à l'offre et à la demande effective historisées depuis 2019. Ces travaux utilisent comme données principales les données d'affluence, c'est à dire le nombre de passagers à bord des rames au départ de chaque train. Ces données présentent diverses caractéristiques des flux de trains au sein

2. Gare de Lyon, Gare du Nord, Gare de l'Est, Montparnasse, Austerlitz, Bercy

de la ligne et représentent la fusion de diverses sources de données fournies par la RATP. Nous comptons trois principales sources de données :

- La charge des trains (SYLEVE) : cette base indique le nombre de passagers à bord des trains ainsi que le parcours de chaque train d'une ligne
- Les validations télé-billettiques en station (Optimus) : il s'agit du nombre d'usagers validant leur titre de transport au sein de chaque station
- Les temps d'attente à quai (IV) : l'information voyageur pour les prochains trains à chacune des stations du réseau.

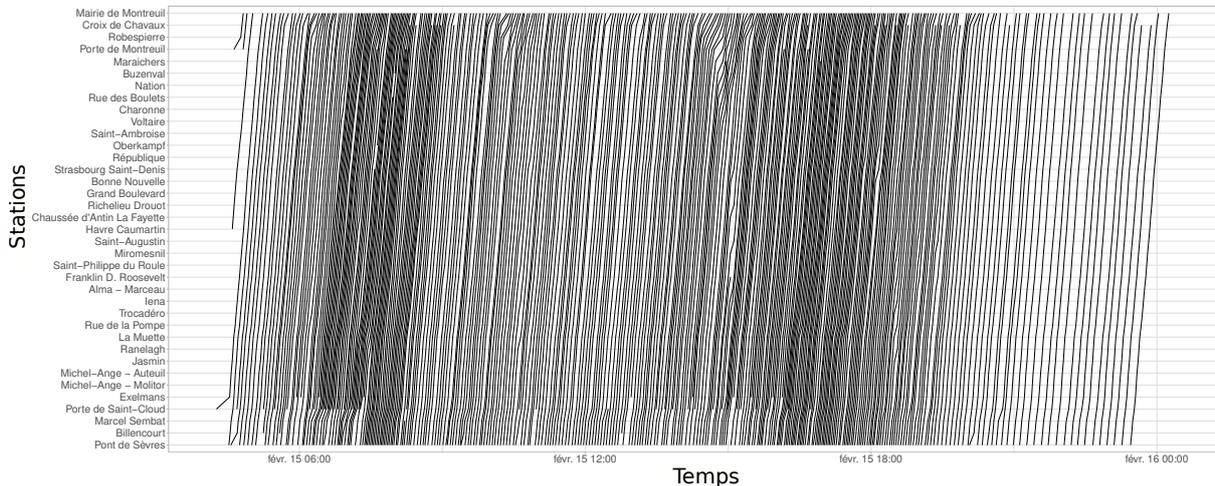


FIGURE 1.3 – Service réalisé le 15 février 2021 pour les trains au départ de Pont de Sèvres.

A titre d'exemple, la figure 1.3 représente l'extraction des parcours de trains du 15 février 2021 pour les trains parcourant la ligne de la station Pont de sèvres à Mairie de Montreuil. On y observe les variations de l'offre avec les deux heures de pointes en début de matinée et entre 16 heures et 19 heures.

La première base nommée SYLEVE représente le socle sur lequel les autres sources de données viennent se greffer. Une vision "train" des données permet d'une part d'avoir une représentation fine des dynamiques d'une ligne et des stations et d'autre part de pouvoir agréger les données selon nos propres critères. Ainsi pour chaque départ de trains nous nous retrouvons avec les variables d'intérêt suivantes :

- Date et heure du départ d'un train à une station
- Nombre de passagers à bord du train
- Nombre de validation entre deux départs de trains
- ID du train
- Sens de circulation
- Heures estimées des prochains départs à la station

Ainsi, nous pouvons agréger la Figure 1.3 avec les différentes sources de données pour mettre en valeur les différentes caractéristiques des parcours des trains. La figure 1.4 montre un sous-ensemble de départ de trains du 15 février 2021 en associant à chaque départ de trains l'affluence (ou charge).

Cependant, au sein de nos données, bien que certaines informations ne soient pas directement fournies, elles peuvent être estimées à partir des données. Il s'agit en particulier des courses des trains. La course d'un train représente un parcours de la ligne entre deux terminus. Seul le parcours complet des trains est présent dans la base de données. Le premier traitement a pour but de reconstruire les missions des trains comme montré dans

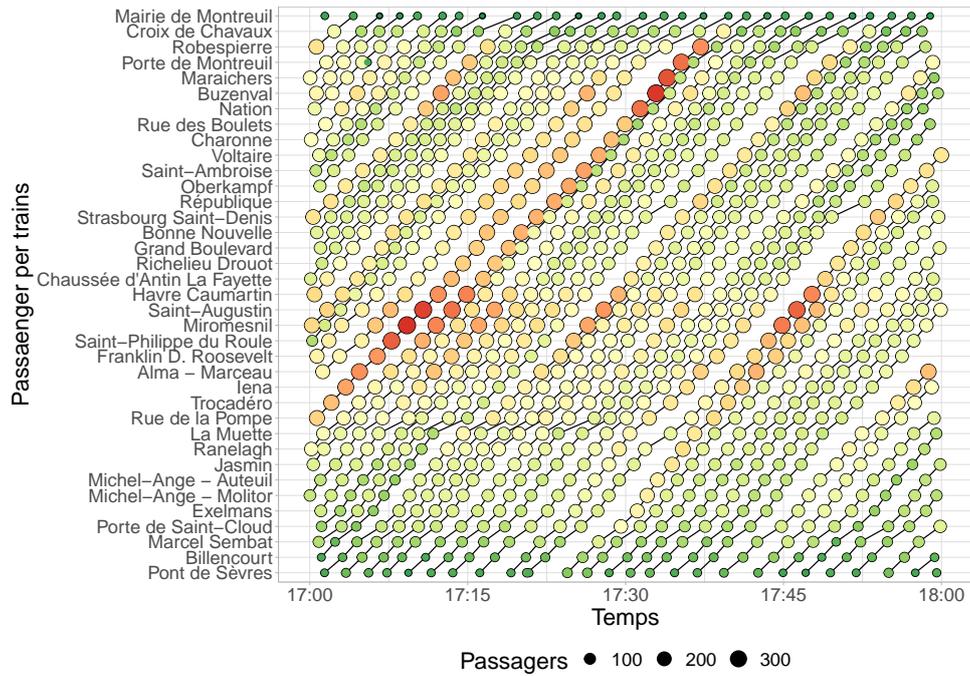


FIGURE 1.4 – Graphe de Marey pour le 15 février 2021 pour les trains partant de Pont de Sèvres. La taille et la couleur dénotent le nombre de passagers par trains.

la Figure 1.5 qui présente le parcours d’un train le 15 février 2020. Les couleurs y représentent les différentes missions réalisées par un train. De plus, un nettoyage des données est effectué simultanément lors de la génération des missions afin de détecter et supprimer les faux départs de train ou les missions ne présentant qu’un seul arrêt (en particulier tôt le matin avant l’ouverture de la ligne). D’autres variables peuvent être estimées à partir des données comme les temps de parcours réels des trains ou la variation de charge entre deux départs d’un même train.

1.2.3 Affluence de la ligne 9

Pour entrer plus en détail dans les données de charge, nous allons nous intéresser à l’affluence des trains pour la ligne 9 du métro Parisien à une échelle macroscopique. Pour débiter, nous observons les points d’entrée des passagers sur la ligne en s’intéressant aux validations télé-billettique avec la figure 1.6. Les validations télé-billettique représentent les titres de transport magnétiques (Pass Navigo, Pass Imaginaire). A chaque fois qu’un usager entre dans le réseau de transport ou en sort, l’information est stockée selon l’heure de la validation, l’espace contrôlé (station), la borne utilisée et le sens de l’usager (voix publique vers station pour une entrée). Pour les lignes de métro, seulement les données d’entrée du réseau sont connues, les usagers ne validant pas en sortie du réseau. De plus, les correspondances entre deux lignes de métro ne sont pas observables dans ces données car les usagers ne valident généralement pas entre les différentes lignes de métro.

La figure 1.6 met en lumière 3 stations : Havre-Caumartin, République et Nation. Ces stations possèdent de nombreuses connexions avec d’autres lignes de métro ou trains de banlieue (RER, TER) en particulier Nation et Havre-Caumartin(cf. le plan de la ligne 1.2). Hormis ces 3 stations, on note que la majorité des validations ont lieu dans la partie centrale de la ligne entre Trocadéro et Nation et également pour les stations proches des

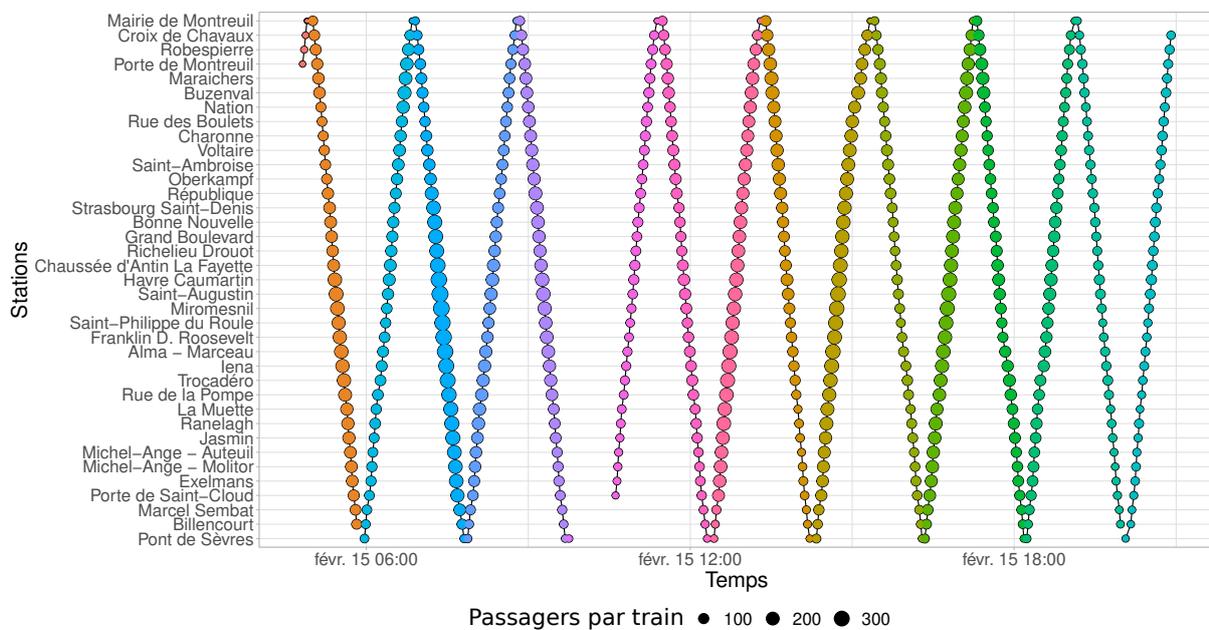


FIGURE 1.5 – Parcours complet d'un train pour la journée du 15 février 2020. Chaque couleur représente un parcours entre deux terminus.

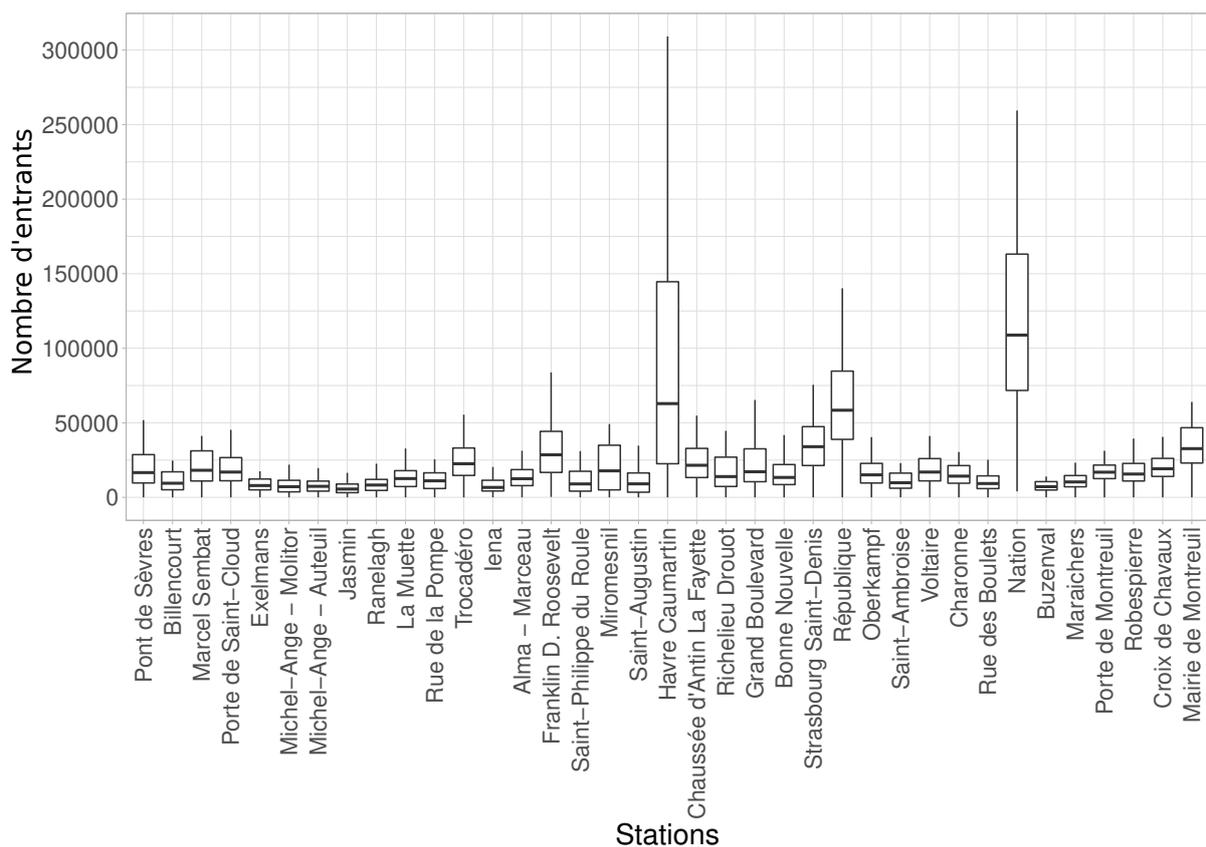


FIGURE 1.6 – Boxplot du nombre d'entrants (validation télé-billettique) par jour et par station.

terminus (Pont de Sèvres à Porte de Saint Cloud et Mairie de Montreuil).

Pour étayer cette première vision de l'affluence observée au travers des validations,

nous nous intéressons à la charge moyenne des trains pour chacune des stations avec la figure 1.7 qui présente la répartition du nombre moyen de passagers par jour et station à bord des trains. Par rapport à la figure précédente, on note que les validations ne se traduisent pas forcément en charge (comme avec le cas de la stations Nation). Ce phénomène est dû au fait qu’il est impossible à partir des validations de connaître la direction des usagers. Les usagers ne valident leur titre de transport qu’une seul fois en entrée dans le réseau³. De plus, il est impossible de connaître les sens de circulation et ligne empruntée par l’usager.

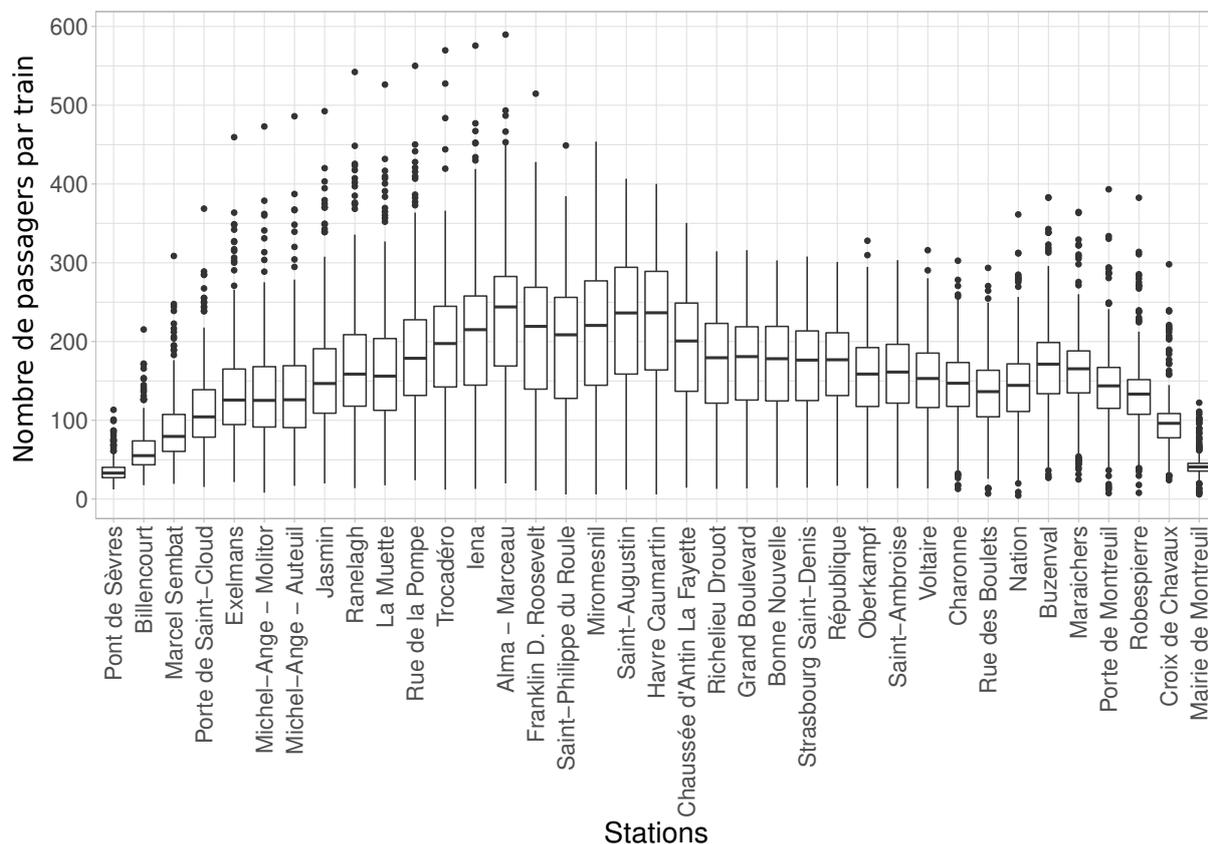


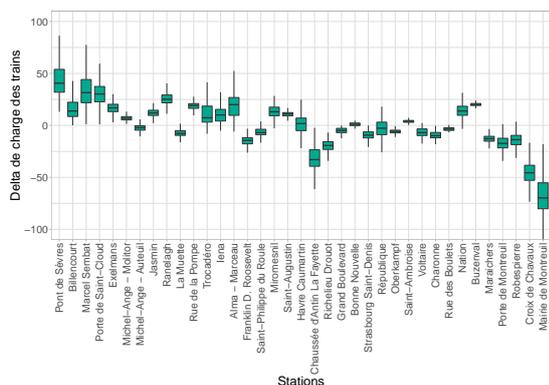
FIGURE 1.7 – Boxplot du nombre de passagers par trains sur la ligne, i.e. serpent de charge.

Concernant l’affluence à bord des trains, la charge moyenne des trains est plus importante pour les stations entre Havre-Caumartin et Iéna avec une médiane de passagers de plus de 200 par train. On note également la présence de nombreux outliers pour les stations entre pont de Sèvres et Iéna dont la moyenne journalière est proche de la capacité maximal des trains.

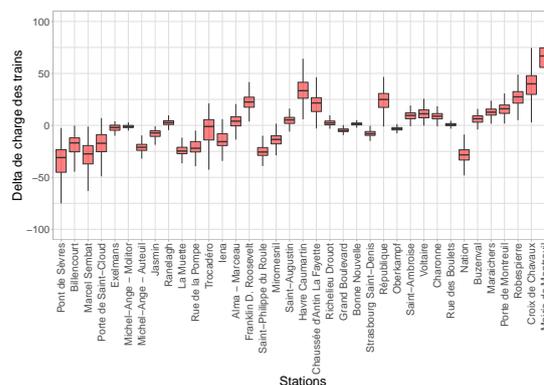
1.2.4 Offre et demande

La seconde caractéristique importante des données est l’offre de transport donnée par l’heure de départ des trains. Cette information est utilisée pour décrire les temps de parcours inter-stations et les intervalles de départ des trains qui correspondent aux temps d’attente réels en station entre deux trains.

3. Uniquement le cas pour le métro, Pour les trains express RER, le titre est validé en entrée et sortie du réseau.



(a) Variation de la charge des trains par station pour les trains partant de Pont de Sèvres.



(b) Variation de charge des trains par station pour les trains partant de Mairie de Montreuil.

FIGURE 1.8 – Comparaison des variations du nombre de passagers à bord des trains selon le sens de circulation. La variation de charge représente le gain ou la perte de passagers au départ d’une station. Les couleurs représentent le sens de circulation.

La figure 1.9 illustre la répartition des temps d’attente selon l’heure de la journée depuis 2019. Comme on peut le constater, les profils des temps d’attente diffèrent principalement entre les jours ouvrés et les week-ends. L’offre est adaptée à la demande respective de chaque type de jour. Pour visualiser les profils journaliers, les temps d’attente sont divisés en intervalles de temps. Pour des temps d’attente inférieurs à 10 minutes, on construit 5 intervalles de 2 minutes chacun. Pour les temps d’attente supérieurs à 10 minutes, on construit un dernier intervalle.

La figure 1.9 met en lumière les heures de pointe de la ligne. Pour les jours ouvrés, on observe deux périodes de pointe une le matin entre 7 heures et 10 heures et une seconde le soir entre 16 heures et 20 heures. On retrouve aussi une heure de pointe entre 16 heures et 21 heures pour les week-ends. Durant ces périodes, la fréquence des trains augmente réduisant ainsi le temps d’attente. 75% des trajets possèdent un temps d’attente inférieur à 5 minutes les jours ouvrés, alors que cette proportion est de seulement 40% les jours de week-ends. Pour les jours ouvrés, les heures de pointes voient plus de 75% des trains sous les 3 minutes d’attente. Ce graphique montre l’adaptation de l’offre lors des pics d’affluence de la ligne. De plus il est important de préciser que ce profil est similaire pour les 4 années de données. Les périodes de confinement ont peu impacté l’offre qui, suite aux directives de l’autorité organisatrice de transport (IDFM), a été maintenue à une offre nominale alors que la demande avait énormément chuté (entre 70% et 90%).

En observant cette fois la charge des trains en fonction de l’heure de la journée (cf. figure 1.10), nous retrouvons les heures de pointe déjà observées dans la figure 1.10. Les pics d’affluence sont corrélés à l’offre de la ligne, et même si l’offre est plus importante en heure de pointe, les trains sont tout de même plus chargés. On note toutefois une différence entre la répartition des charges suivant les sens de circulation. Pour les trains au départ de Mairie de Montreuil, le pic d’affluence est en matinée avec une médiane supérieure à 200 passagers alors que pour le sens inverse, c’est le pic du soir qui comporte le maximum de charge (une médiane de 200 passagers par train). On observe également un pic d’affluence en matinée pour ce sens de circulation mais présentant une charge médiane de 150 passagers nettement inférieure à celle observée pour les trains partant de Mairie de Montreuil. Cette inversion de l’affluence entre la matinée et soirée indique la présence de

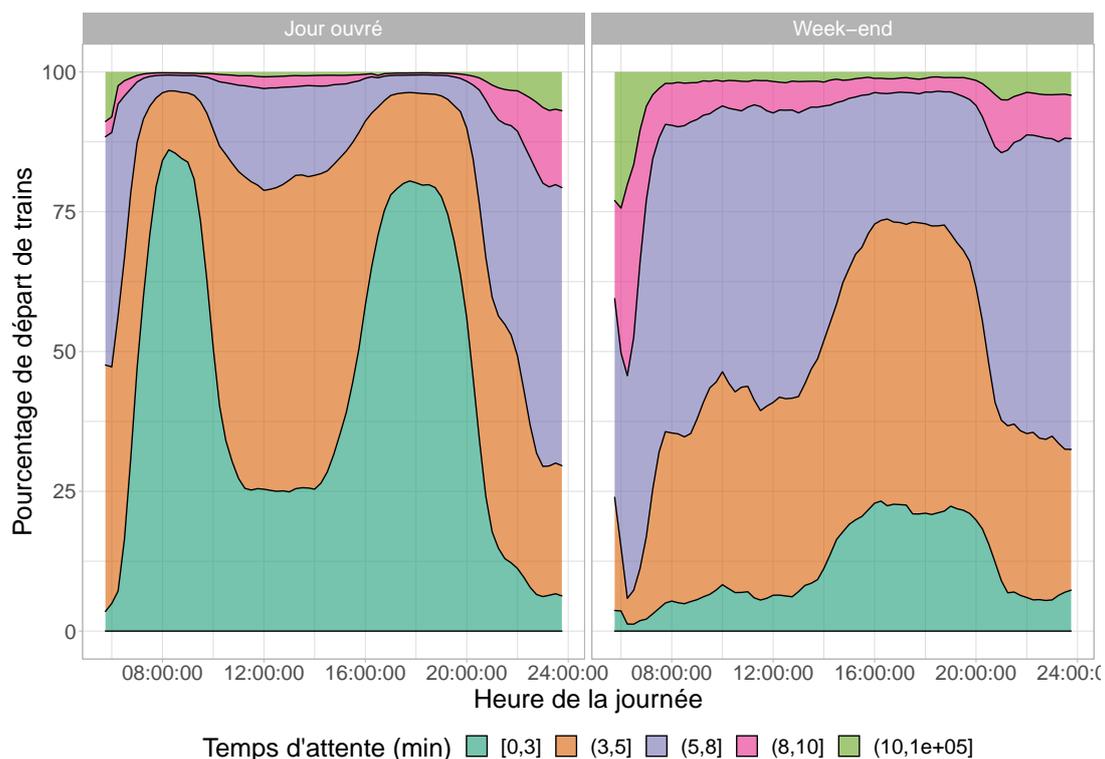
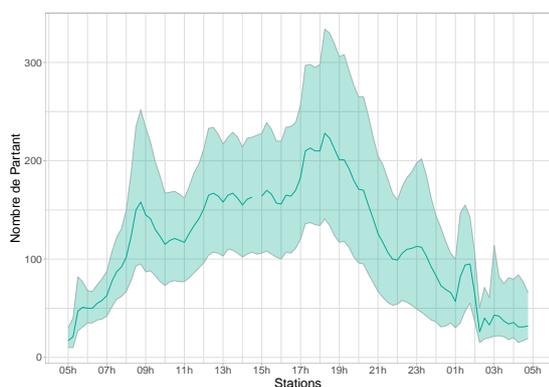


FIGURE 1.9 – Répartition des temps d’attente pour la ligne 9. De gauche à droite, les profils de temps d’attente pour les jours ouvrés et les jours de week-end. Les couleurs représentent les intervalles de temps de départ de trains.

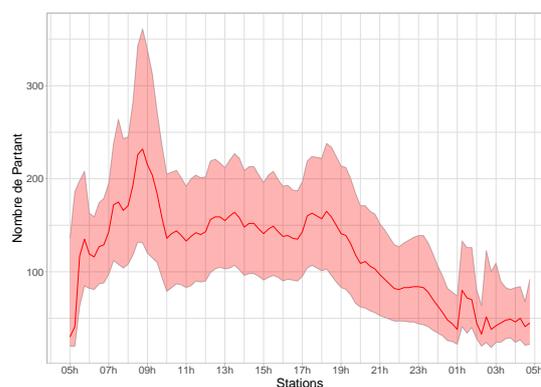
trajet domicile travail principalement dans un sens de circulation (de l’Est vers l’Ouest le matin et de l’Ouest vers l’Est le soir). En recoupant avec les figures 1.8 et 1.7, ces trajets domicile travail ont lieu dans la portion de ligne située entre Havre-Caumartin et Pont de Sèvres qui présente des trains plus chargés que le reste de la ligne et des stations qui voient majoritairement les passagers descendre à ces stations pour les trains partant de mairie de Montreuil et monter dans le sens inverse. Enfin, on note un pic de validation entre 23 heures et minuit relatif à des activités nocturnes de loisirs dans le centre de Paris.

Afin de compléter cette analyse exploratoire, spatiale et temporelle, de l’offre et de la demande, nous allons observer les dynamiques et corrélations entre nos différentes variables et comprendre ainsi les enjeux de la prédiction de charges par rame. Pour débuter, nous allons observer l’évolution de la dynamique de charge par jour de la semaine pour mettre en lumière l’évolution de la demande entre 2019 et 2022. Nous nous concentrons uniquement sur ces deux années pour ne pas tenir compte des deux années atypiques que furent 2020 et 2021.

La Figure 1.11 compare le nombre de mission par année (1.11a et 1.11c) d’une part et la charge par train (1.11b et 1.11d) d’autre part. Elle met en évidence une évolution des comportements des passagers entre 2019 et 2022 avec une baisse globale du nombre de passagers par train pour les jours ouvrés tandis que ceux du week-end restent égaux entre 2019 et 2022. On notera que l’offre reste équivalente entre ces deux années comme le montre 1.11a et 1.11c où le nombre de mission est identique. De plus, le comportement des usagers lors des jours ouvrés a changé en 2022 avec des pics d’affluence présents désormais les mardi et jeudi contrairement à 2019 où l’affluence était homogène entre les jours ouvrés. Ce phénomène est du à la généralisation du télétravail lors des 3 dernières



(a) Médianes, 1^{er} et 3^e quartiles du nombre de passagers par train pour les départs de Pont de Sèvres.

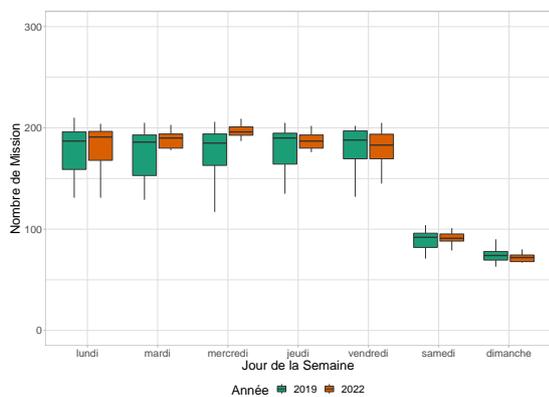


(b) Médianes, 1^{er} et 3^e quartiles du nombre de passagers par train pour les départs de mairie de Montreuil.

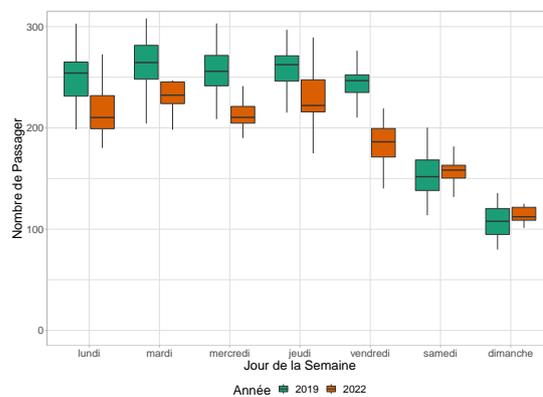
FIGURE 1.10 – Profil du nombre de passagers à bord des trains selon le sens de circulation au cours de la journée. Les couleurs représentent le sens de circulation, vert pour les départs de pont de Sèvres et Rouge pour les départs de Mairie de Montreuil (ce code couleur est commun à toutes les figures introduisant le sens de circulation).

années qui se traduit par une baisse notable du nombre de passagers le vendredi.

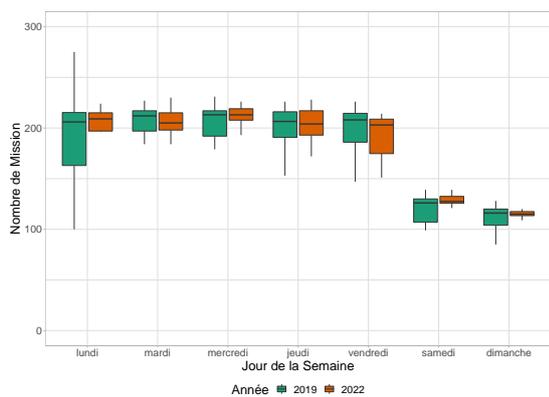
Pour mieux montrer l'évolution de la demande, nous proposons de faire un focus sur les vendredi entre 2019 et 2022 (2020 et 2021 sont exclus de part l'impact de la crise du COVID). La figure 1.12 présente pour les deux sens de circulation de la ligne le profil de charge entre les deux années sélectionnées. Pour les figures 1.12a et 1.12b, nous notons que les dynamiques (pic d'affluence, heure creuse) sont similaires entre les deux années mais le volume de passagers a diminué suite à la crise du COVID. Pour la figure 1.12a, on note une baisse de 50 passagers pour la médiane entre 17 heures et 19 heures et d'environ 100 passagers pour le pic du matin passant de 250 passagers à plus de 150 en 2022. Pour l'autre sens de circulation, la baisse du nombre de passagers est la plus notable pour le pic d'affluence du matin avec une médiane proche de 350 passagers en 2019 qui a chuté à 250 passagers en 2022.



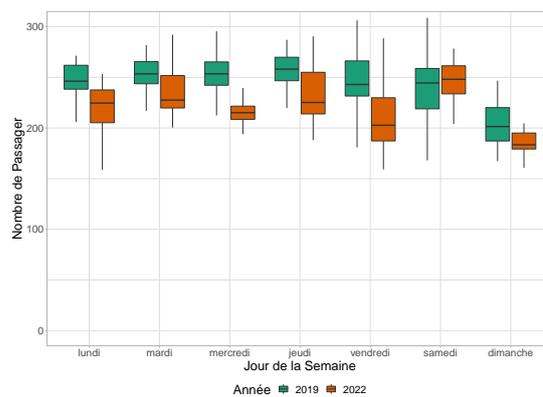
(a) Nombre de mission par jour entre 2019 et 2022 pour l'heure de pointe du matin (16 :00-20 :00).



(b) Charge des trains par jour entre 2019 et 2022 pour l'heure de pointe du matin (16 :00-20 :00).

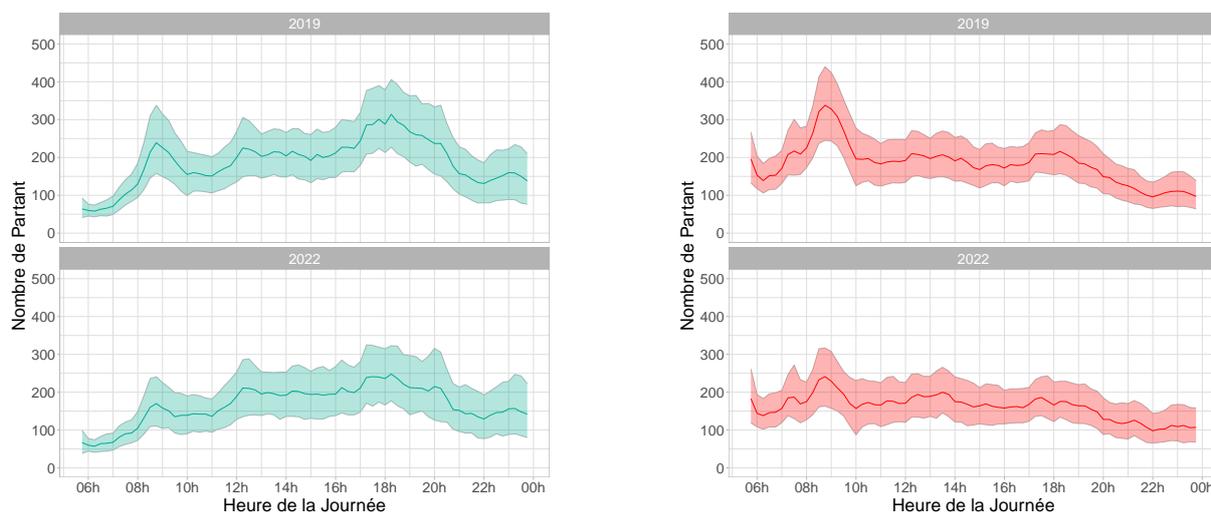


(c) Nombre de mission par jour entre 2019 et 2022 pour l'heure de pointe du soir (7 :30-9 :30).



(d) Charge des trains par jour entre 2019 et 2022 pour l'heure de pointe du soir (7 :30-9 :30).

FIGURE 1.11 – Comparaison entre 2019 et 2019 de l'offre réel en nombre de missions par heures de pointes et la charge par rame durant les mêmes périodes.



(a) Médianes, 1^{er} et 3^e quartiles du nombre de passagers par train pour les départs de Pont de Sèvres

(b) Médianes, 1^{er} et 3^e quartiles du nombre de passagers par train pour les départs de mairie de Montreuil

FIGURE 1.12 – Comparaison entre 2019 et 2022 de la charge des trains pour les vendredi pour chacun des sens de circulation. En vert les trains au départ de Pont de Sèvres, en Rouge les trains au départ de Mairie de Montreuil.

1.3 Profil par station

Pour l’heure, nous avons présenté les caractéristiques de la ligne 9 et les dynamiques d’offre et de demande. Nous pouvons maintenant mener une analyse à une plus petite échelle, celle des stations afin, de mettre en lumière les différentes spécificités des stations par rapport au fonctionnement globale de la ligne. Pour ce faire, nous allons nous intéresser à 2 stations, Havre-Caumartin et Jasmin qui représentent respectivement un hub de la ligne et du métro parisien et une station simple sans correspondance en périphérie de la ligne.

1.3.1 Station Havre-Caumartin

La première station mise en lumière est Havre-Caumartin le hub principal de la ligne. Cette station est reliée (avec Saint-Augustin) au pôle multimodale de la Gare Saint Lazarre et au RER A avec la station Auber. Elle représente un des point d’entrée principale de la ligne 9.

Pour débuter, nous nous intéressons au profil de charge par jour de Havre-Caumartin avec la figure 1.13 qui présente le profil par an du nombre de passager par train. On retrouve les pics d’affluence de la ligne en matinée et en soirée. On peut toutefois observer que le nombre de passagers par train est plus élevé que les statistiques générales de la ligne comme on pouvait le noter dans la figure 1.10.

Pour compléter les profils de charge, nous montrons aussi le nombre de validation télébilletique entre chaque départ de trains à Havre-Caumartin avec la Figure 1.14. On note une corrélation entre les pics d’affluence de la ligne pour la charge et celle des validations particulièrement pour les années 2019 et 2022. On note également la baisse de la demande en 2020 et 2021 avec le COVID avec une chute des validations sous les 100 validations entre départs de trains pour la médiane.

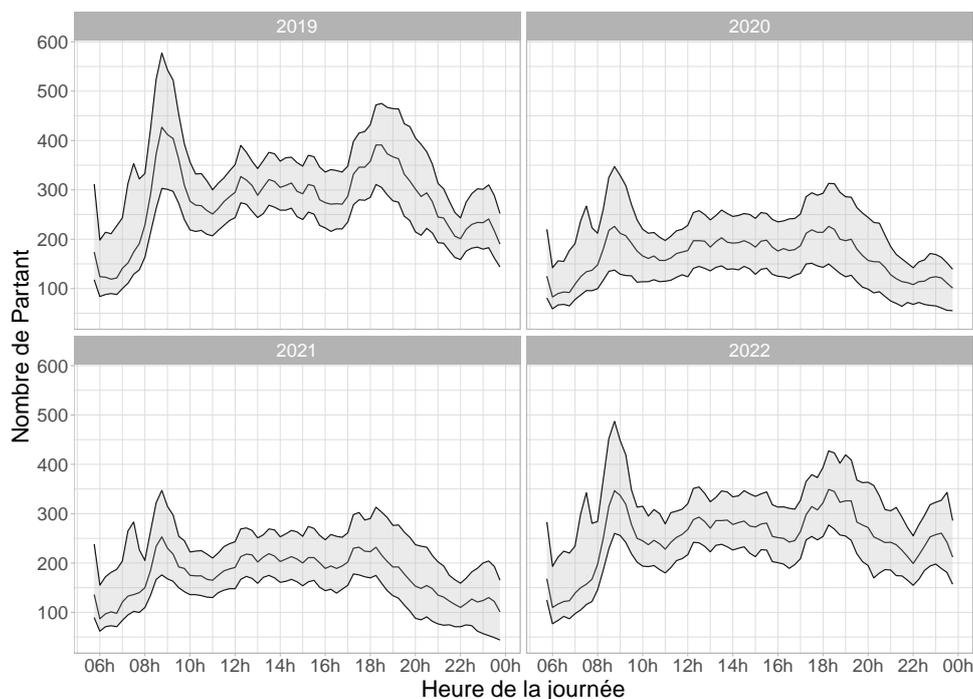


FIGURE 1.13 – Médianes, 1^{er} et 3^e quartiles du nombre de passagers à bord des trains au départ de la station Havre-Caumartin par année entre 2019 et 2022.

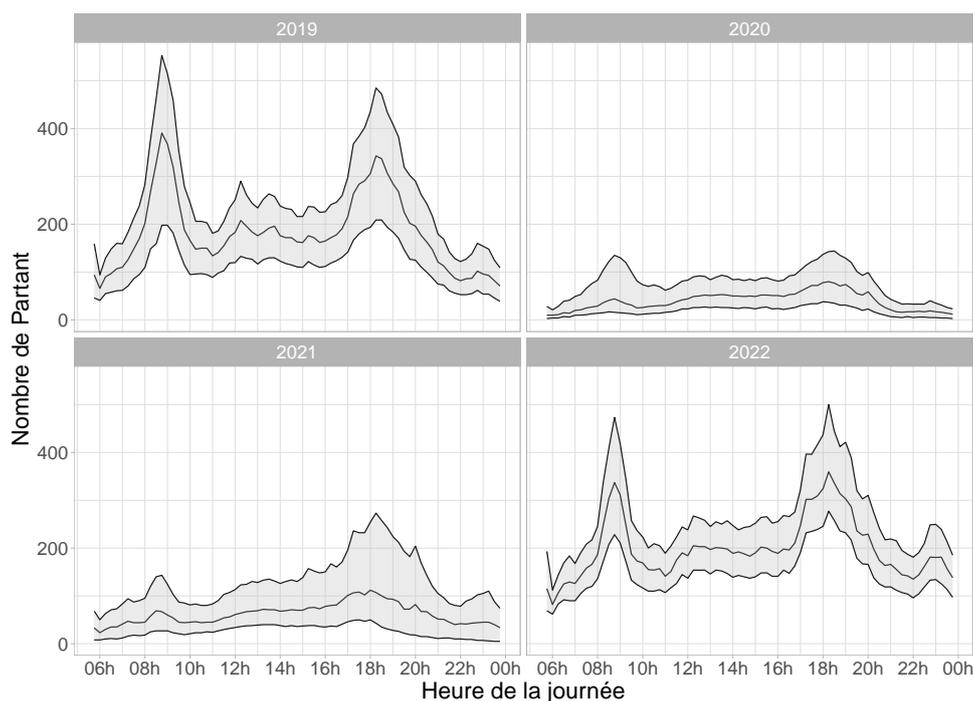


FIGURE 1.14 – Médianes, 1^{er} et 3^e quartiles du nombre de validation télé-billettique à la station Havre-Caumartin par année entre 2019 et 2022.

Il est intéressant d'analyser la dynamique des passagers à Havre-Caumartin et le nombre de montants ou de descendants. Il ne s'agit pas du nombre réel de descendants ou montants mais du gain ou perte de passagers suite au départ de Havre-Caumartin, nous parlerons de delta de charge. Par exemple, un delta de 0 peut correspondre à différentes

situations soit aucun passager n'est descendu du train, soit il y a eu autant de montants que descendants. La figure 1.15 présente le delta de charge d'un train entre le départ à la station et à la station précédente (en soit l'évolution du nombre de passager à bord d'un trains après chaque stations). Cette figure met en lumière les dynamiques des trajets domicile travail avec en matinée une augmentation de 75 passagers par train pour les trains en direction de Pont de Sèvres et un delta médian à la station de 25 passagers pour le pic d'affluence du soir. De plus, toujours en direction de Pont de Sèvres, on observe que sur l'ensemble de la journée, les trains présentent en majorité des deltas de charge positifs. La situation est sensiblement inverse pour la direction opposée avec plus de descentes que de montées et donc des deltas négatifs et un pic plus important en soirée.

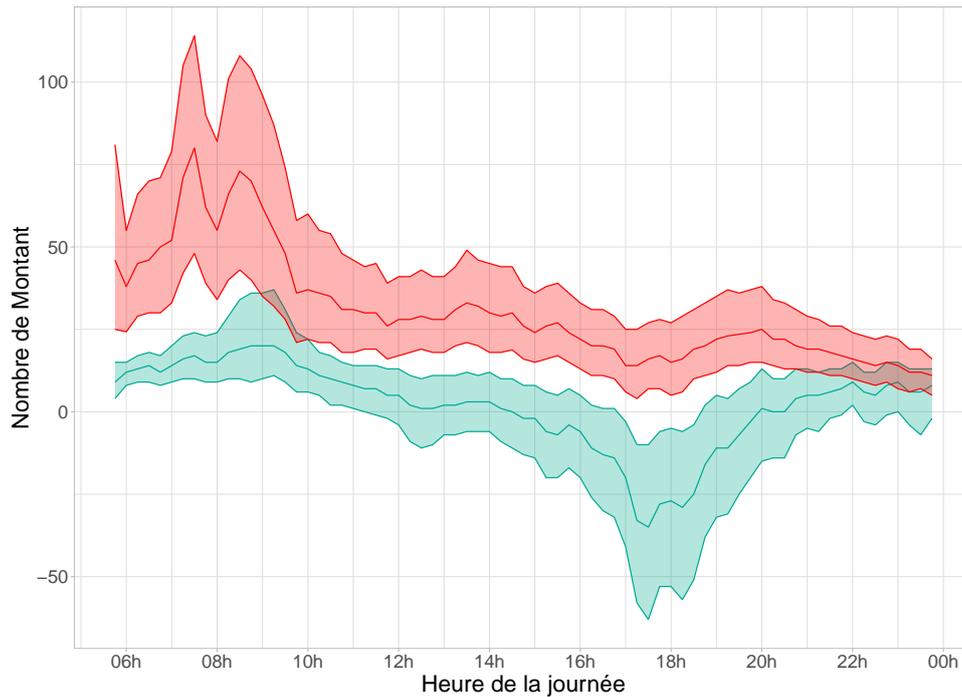


FIGURE 1.15 – Médianes, 1^{er} et 3^e quartiles de l'évolution du nombre de passagers (delta de charge) à bord d'un train à Havre-Caumartin. La couleur définit le sens de circulation. Une valeur négative indique que le train a perdu des usagers et une valeur positive indique un gain de passagers.

Ainsi, on note que pour la station Havre-Caumartin, les dynamiques observées pour la charge et les validations sont similaires à la dynamique globale de la ligne 9, ce qui peut être induit par la position de Havre-Caumartin comme hub principal de la ligne, influençant grandement son fonctionnement.

1.3.2 Station Jasmin

Pour la seconde station, nous avons choisi Jasmin, une station en périphérie de Paris et sans correspondance. Nous présentons la station de manière similaire à Havre-Caumartin en débutant par la charge avec la figure 1.16 qui représente la charge par train au départ de Jasmin. On retrouve les pics d'affluence de la ligne avec une charge médiane inférieure à celle observée à Havre-Caumartin (une médiane de 400 passagers pour 2019 et 2022). Cette baisse peut s'expliquer par la position de la ligne en dehors de la partie centrale de la ligne.

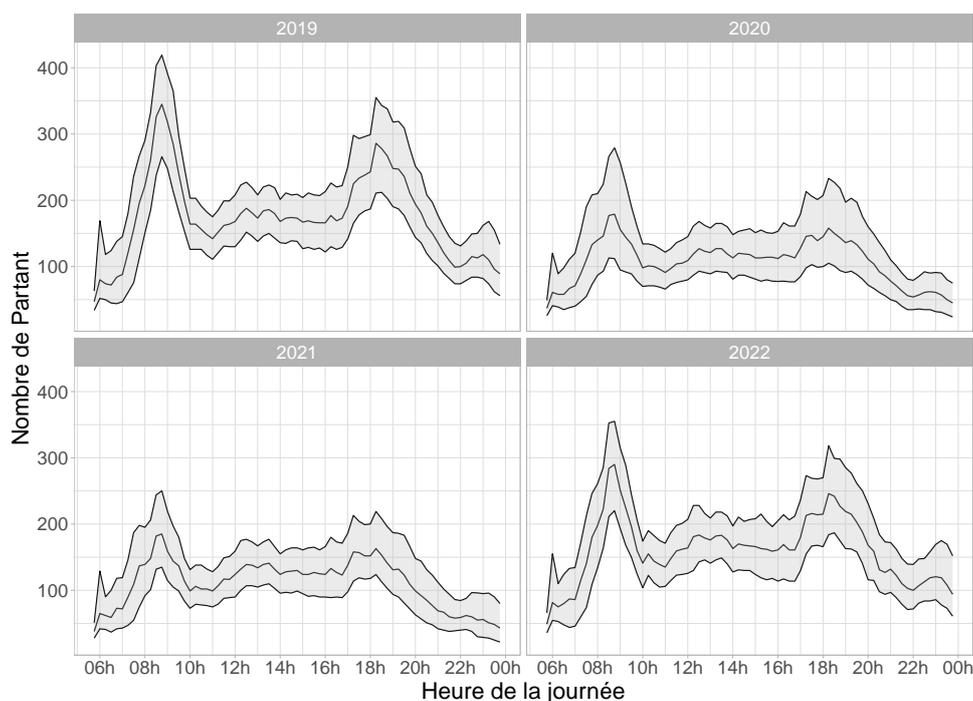


FIGURE 1.16 – Médianes, 1^{er} et 3^e quartiles du nombre de passagers à bord des trains au départ de la station Jasmin par année entre 2019 et 2022.

La principale différence entre Havre-Caumartin et Jasmin se situe sur les validations qui ne sont plus corrélées à la charge des trains pour des stations de petite envergure comme Jasmin. La figure 1.17 montre l'évolution des validation télé-billettique depuis 2019 pour chaque départ de train. On note une médiane de validation grandement inférieure à Havre-Caumartin (supérieure à 200 en 2019 et d'environ 50 en 2020 et 2021) ne dépassant pas les 20 passagers entre 2019 et 2022. Cependant contrairement à Havre-Caumartin, les validations restent constantes même avec la crise du COVID. Enfin on note une augmentation des validations en 2022 pour la station avec une médiane qui atteint environ 30 passagers par train entre 8 heures et 9 heures.

Enfin, de manière similaire à Havre-Caumartin, la figure 1.18 présente le nombre de montants pour chaque départ de trains à la station. Nous observons ici deux comportements distincts pour chaque sens de circulation. Pour les trains partant de Pont de Sèvres, on note une majorité de montant à la station qui plus est, est corrélée aux validations de la figure 1.17. Ainsi, de par sa position en début de ligne, les personnes entrant à Jasmin se dirigent principalement en direction du centre de la capitale. Au contraire, les trains issus de Mairie de Montreuil ne présentent principalement que des descendants à la station. On ne retrouve donc pas les dynamiques des trajets domicile travail comme à Havre-Caumartin. Cette information nous renseigne également sur la direction choisie en majorité par les entrants, comme nous renseigne la corrélation entre montant et validation. Cependant, cette information ne permet pas d'anticiper la charge des trains à partir des données de la station. La charge des trains au départ de Jasmin n'est en effet pas corrélée aux validations comme c'est le cas à Havre-Caumartin, celle-ci étant largement influencée par les montants descendants des autres stations.

Ainsi, à travers l'exemple de ces deux stations, Jasmin et Havre-Caumartin, nous avons pu voir que la position et les caractéristiques d'une station (hub, sans correspondance) influent sur les dynamiques locales de charge et de validation de la station. Il est, dans

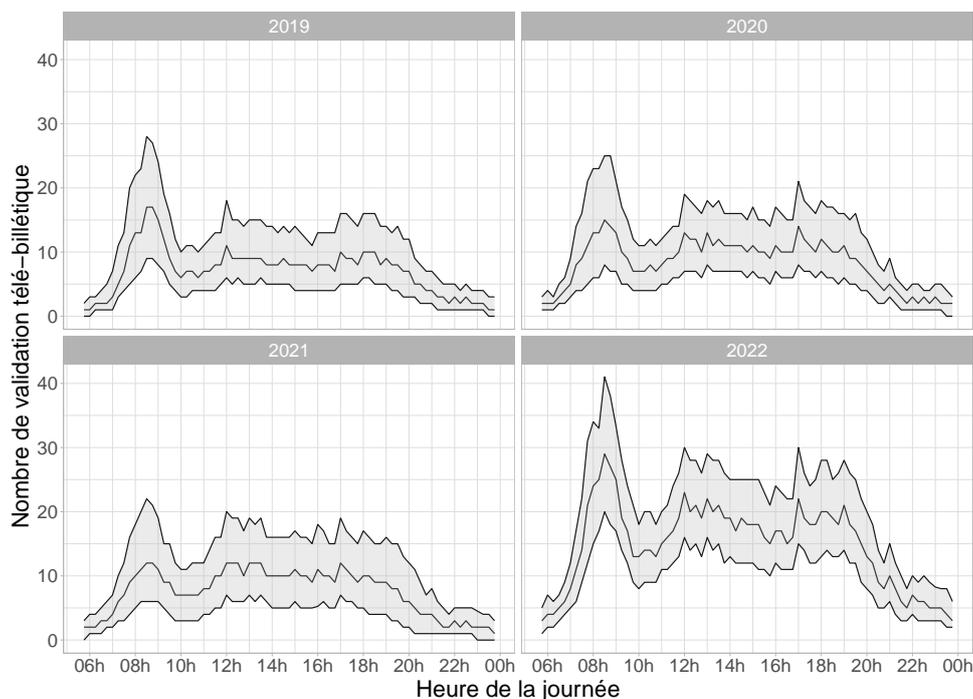


FIGURE 1.17 – Médianes, 1^{er} et 3^e quartiles du nombre de validation télé-billettique à la station Jasmin par année entre 2019 et 2022.

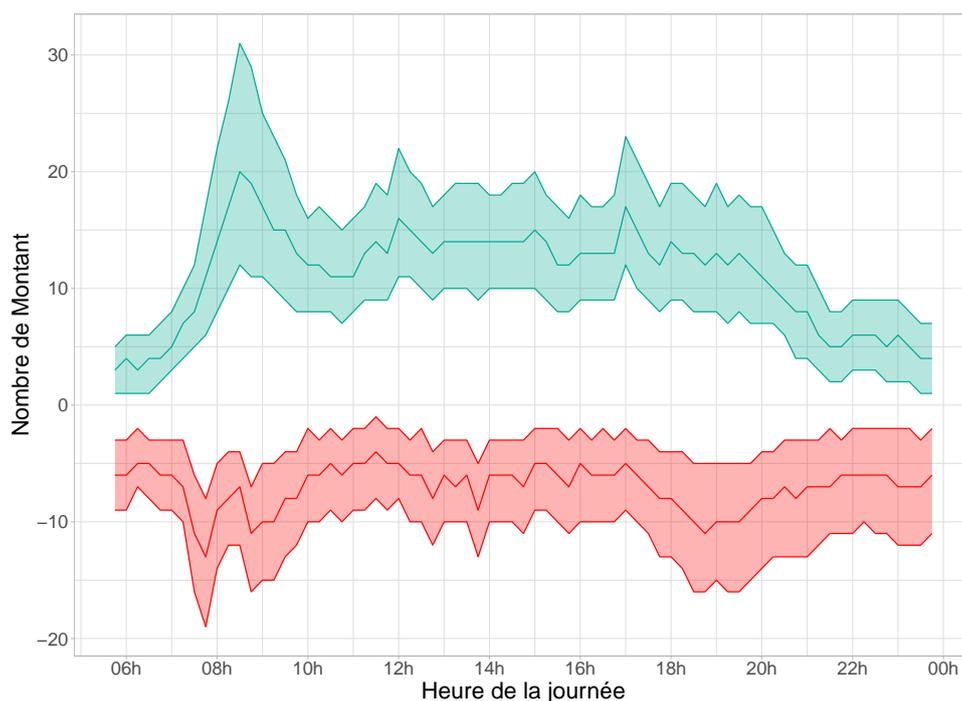


FIGURE 1.18 – Répartition des temps d’attente pour la ligne 9. De gauche à droite, les profils de temps d’attente pour les jour de weekend et jour ouvrée. Les couleurs représentent les intervalles de temps de départ de trains.

la cas de Jasmin, facile de reporter les validations à la charge selon l’heure de la journée et sens de circulation du en partie à la station ne desservant que la ligne 9. Au contraire pour Havre-Caumartin, il est beaucoup plus difficile d’anticiper la direction choisie par

les usagers entre l'ensemble des transport en commun disponible (RER, Métro, Trains de Banlieue).

Le premier commentaire concerne le volume par train au cours des 4 années que composent la base de données. On perçoit l'impact de la crise du covid et des confinements successifs lors des années 2020 et 2021. La charge moyenne par train pour les trois stations chute entre 2019 et 2020, 75% des charges deviennent inférieures au 3ème quartile de 2019 pour l'heure de pointe. Le volume passager opère un retour à la normale sur les 3 dernières années mais reste encore inférieur à 2019. La charge moyenne en 2019 pour la pointe du soir à Havre-Caumartin était de 450 passagers par train contre 400 passagers en 2022. Cette observation est commune à l'ensemble des stations.

1.3.3 Situations atypiques

En disposant de plus de trois années de données, on note la présence de situations où le trafic des métros est perturbé, nous utiliserons la dénomination situations atypiques pour ces cas de figure. Un contexte atypique peut avoir comme conséquence une augmentation du nombre de passagers par rame ou un allongement des temps d'attente dans les espaces contrôlés. Il est également possible de considérer des choix opérationnels comme atypiques car ils ne respectent pas l'offre de transport initialement prévue. Les conséquences d'une situation atypique peuvent impacter seulement la circulation des trains ou la charge ou les deux. Par exemple la période du Covid en 2020 que nous considérons comme atypique a induit une baisse de la charge sans pour autant modifier les temps d'attente. Ces situations sont par définition rares et de diverses natures et peuvent être le résultat d'évènements sur la ligne ou en dehors de la ligne. Nous avons identifié dans ces travaux 3 types de situations selon si la situation est prévue ou récurrente comme résumer dans le tableau 1.1 suivant :

| cas | Récurrent | Prévu | Exemple |
|-----|-----------|-------|-------------------------|
| 1 | Oui | Oui | Match de Football |
| 2 | Non | Oui | Travaux, Grève |
| 3 | Non | Non | Incident d'exploitation |

TABLE 1.1 – Résumé et exemple des 3 types de situations atypiques.

Pour la prédiction de charge, ces cas de figures représentent un réel enjeu, car ils sont rares et difficiles à prédire. Il faut cependant noter une différences notable entre les trois cas de situations atypiques mentionnés dans le tableau 1.1. En effet, dans le cas 1 et 2, l'offre de transport est souvent adaptée pour répondre à la modification de la demande. Cependant pour les situations prévues est non récurrentes, l'impact sur la ligne varie en fonction de l'origine de la situation atypique, une fermeture de station n'a nullement le même impact qu'une période de grève. Le dernier type de situations atypiques (non prévu et non récurrent) est le plus difficile à prédire car les patterns varient entre chacune des situations atypiques qu'il n'est pas possible d'anticiper.

Pour illustrer plus clairement une situation atypique, la figure 1.19 présente un grappe de Marey pour la journée du 17 février 2020 entre 17h et 18h30. Cette situation est non prévue et non récurrente et a lieu lors du pic d'affluence en fin de journée. On note que pour ce contexte anormal, l'ensemble de la ligne est impacté et que la situation atypique s'étale sur presque une heure de 17 :15 à 18 :15. On note également dans cette figure deux comportements, l'allongement des temps de parcours des trains et le demi tour des trains.

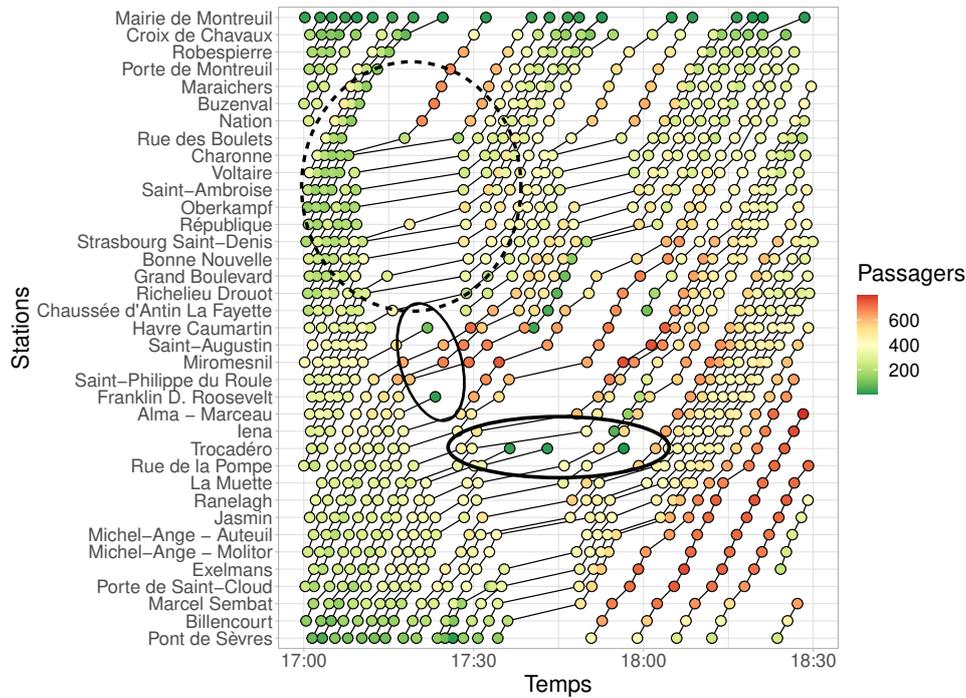


FIGURE 1.19 – Graphe de Marey pour une situation atypique sur la ligne 9 le 17 février 2020. Entouré en pointillé l’allongement des temps d’attente et en trait plein des demi-tours effectué par des trains.

L’allongement des temps de parcours est facilement visible sur la figure 1.19. Pour les demi-tours de trains, ils peuvent être observés grâce à l’apparition ou disparition de trains à des stations qui ne sont pas les terminus habituels (Mairie de Montreuil, Pont de Sèvres⁴). Toutefois ces manoeuvres ne sont réalisables qu’à certains emplacements de la ligne comme Trocadéro ou Havre-Caumartin par exemple. Ainsi, ces manoeuvres sont observables dans la figure 1.19 vers 17h20 à Trocadéro ou Havre-Caumartin et à 18h à la station Chaussée d’Antin. Il est toutefois impossible de comprendre l’origine exacte de la situation atypique à partir de ces données mais seulement d’en observer les conséquences sur la ligne et les trains impactés. Enfin la visualisation offerte par les graphe de Marey permet de mettre en évidence aisément le début et l’impact de la situation.

Pour résumer, la prédiction de charge à bord des trains doit à la fois tenir compte en premier lieu des caractéristiques spatiales d’une ligne de métro. Pour le cas de la ligne 9, les dynamiques de charges peuvent être évaluées localement par stations ou sur l’ensemble de la ligne où le sens de circulation est le principal facteur pour la dynamique de charge. Pour les stations, on observe la présence de différents profils variés pour les stations entre les hubs avec de nombreuses correspondances tel Havre-Caumartin et les stations simples sans correspondance comme Jasmin. Dans un second temps, la charge et part extension leur prédiction est dépendant des dynamiques temporelles de la ligne. En dehors des dynamiques des heures de pointes et jour ouvrées et weekend, on note à travers ces dernières années, on note l’évolution des mobilités suite à la crise du Covid et l’essor du télétravail. Enfin, avec l’exemple des situations atypiques, la prédiction de charges doit pouvoir généraliser sur diverses situations opérationnelles impactant l’offre et la demande de la ligne tout en intégrant les décisions opérationnelles lors de ces situations. Ainsi pour

4. Porte de Montreuil et Porte de Saint-Cloud sont également considérées comme des terminus

intégré au mieux ces situations dans le cadre de prédictions, les trains représentent l'échelle de prédiction la plus pertinente et fine pour observer et prédire les dynamiques de charges à quoi est ajoutée les caractéristiques spatiales et temporelles de la ligne.

1.4 Prédiction court terme d'un réseau de transport : les contraintes et objectifs

La prédiction de l'état d'un réseau de transport est définie par des critères opérationnels émis par l'opérateur (prédiction court/moyen terme, focus sur les hubs du réseau) ainsi que par des critères techniques relatifs au système d'information et aux données disponibles. On choisit généralement des approches de prédiction compatibles avec les données mises à disposition. Proposer par exemple une prédiction court-terme impose que les sources de données soient accessibles dans la fenêtre temporelle de prédiction choisie. On parle alors d'approches orientées par les données (ou data-driven).

En plus du cadre de prédiction (court, moyen ou long terme), les approches choisies vont être fortement orientées par les axes spatial et temporel. Dans le cadre d'un réseau de transport, la prédiction de charge peut être réalisée de diverses manières selon si l'on considère la prédiction à l'échelle d'une station, d'un quai ou d'un espace contrôlé. L'échelle temporelle peut être régulière avec un échantillonnage constant ou irrégulière avec un échantillonnage temporel qui suit l'offre de transport.

Enfin pour l'ensemble des modèles de prédiction, il est nécessaire d'étudier leur robustesse et leur capacité de généralisation en particulier lors de situations atypiques. Cette tâche impose de s'intéresser aux situations où les modèles ne présentent pas de bonnes performances ou de labelliser et/ou détecter les situations atypiques en amont pour évaluer les modèles de façon différenciée sur ces cas particuliers.

1.4.1 Granularité spatiale

On peut définir pour une tâche de prédiction de charge trois niveaux d'agrégation spatiale :

- une station
- une ligne
- un réseau global

Ordonné du plus unitaire (la station) au plus large (le réseau complet), la complexité de la tâche de prédiction augmente avec la taille de l'espace considéré. Une échelle spatiale plus grande permet bien sûr de tenir en compte de corrélations spatiale et temporelle qui traduisent la réalité d'un réseau de transport. Dans ce qui suit, nous considérons les trois granularités spatiales en faisant référence à des travaux de l'état de l'art relatifs à chacune d'elle.

1.4.1.1 Station

Une station représente le plus petit espace d'un réseau de transport⁵. Elle représente un noeud d'un réseau de transport et on peut y associer des volumes de passagers et des matériels roulants (ferré ou de surface) qui passe par cet espace. La prédiction de

5. On peut encore descendre jusqu'au quai mais les données sont pour la majorité agrégées à la station ou à l'espace contrôlé.

flux passagers d’une station consiste le plus souvent à étudier les séries temporelles à cette station sans tenir compte d’autres caractéristiques spatiales. En limitant l’exercice de prédiction de charge à une station, on réduit la complexité et l’implémentation en vue de prédire plusieurs stations.

La simplicité spatiale d’une station permet l’usage d’un panel important de méthode de régression. Une des premières approche pour prédire la charge passagers dans une station sont les modèles statistiques avec la famille des modèles à Moyenne Mobile Auto-Régressive Intégrée [9]. Cette famille propose une linéaire Il peut être décliner selon les caractéristiques des séries temporelles. [2] utilise un ARIMA saisonnier (SARIMA) pour prendre en compte les caractéristiques saisonnières des séries temporelles. Pour traiter les caractéristiques non linéaires, [33] étend l’ARIMA avec des réseaux neuronaux (NN) afin de gérer à la fois les variables linéaires et non linéaires dans les problèmes de prévision. [48] a obtenu de meilleurs résultats de prévision en fusionnant deux modèles, le filtrage de Kalman et le k-voisin le plus proche (KNN). D’autres stratégies classiques d’apprentissage automatique pour prévoir les séries temporelles d’un réseau de transport public peuvent également être mentionnées, comme le Gradient boosting [81, 25], le réseau bayésien dynamique [67] ou les machines à vecteurs de support (SVM) [57]. Enfin, depuis ces dernières, l’essor de l’apprentissage profond a imposé le modèles de réseau à longue mémoire à court terme ou Long Short Term Memory en anglais (LSTM, introduit par [36]) comme la structure de référence pour la prédiction de charge ou de réseau de transport. Ces structures construits sur le principe de réseau de neurones récurrents sont efficaces pour traiter des corrélations spatiales non linéaires.

Prédire une variable à l’échelle d’une station ne permet pas de considérer les dynamiques spatiales d’un réseau de transport sans fournir de données complémentaires sur l’état du réseau. Pour extraire des informations à la fois sur les dynamiques temporelles ou spatiales, il faut pouvoir à la fois représenter de manière pertinente un ensemble de stations mais également de les prédire toutes à la fois.

1.4.1.2 Ligne

Une deuxième granularité spatiale est la ligne du réseau de transport, qui est commune à plusieurs stations. Une ligne est le trajet réalisé par les matériels roulants ou ferrés auxquels sont associés la majorité des données et la dynamique des flux de passagers. L’approche au niveau d’une ligne est moins documentée dans la littérature que l’approche par station.

On retrouve des approches probabilistes telles que l’approche proposée dans [18] qui prédit les temps de parcours d’une ligne de bus en construisant un modèle probabiliste bayésien pour chaque paire de bus adjacent de la ligne. [80] mélange un gradient boosting et un Scaled Stacking Process afin de sélectionner les descripteurs (ou features) les plus pertinents pour la prédiction des flux de passagers de ligne de bus. Pour les modèles de deep learning, on peut mentionner les travaux de [54] qui utilisent un biLSTM pour considérer les corrélations temporelles dans les deux sens de circulation d’une ligne de bus afin de prédire des volumes de passagers par station. Enfin, certaines approches change de paradigme afin de faire abstraction des contraintes temporelles. [34] qui transforment la prévision de la charge des transports publics en une classification supervisée. L’approche n’opère plus de régression mais propose d’inférer des classes de niveaux de charge. Cependant, certaines caractéristiques ne peuvent pas être considérées dans une prédiction d’une ligne de transport public. Les inter-connexions ou les trajets origine-destinations ne peuvent pas être représentées par une seul ligne.

1.4.1.3 Réseau de transport public

La dernière approche spatiale est celle qui vise à considérer l'ensemble du réseau pour la tâche de prédiction. Le but est de traduire fidèlement le réseau avec l'ensemble des lignes et inter-connexions de ce dernier. Cette approche repose principalement sur des modèles de deep learning hybrides décomposés en une brique spatiale et une brique temporelle. Les modèles explorent les features spatiales du réseau par l'usage d'images et de réseaux convolutionnels (CNN) [87, 32], ou de réseaux convolutionnels à base de graphes [76, 89]. La brique temporelle via l'usage de modèles de deep learning récurrent, unités récurrentes fermées GRU, [23] LSTM [88, 40], extrait en aval de la brique spatiale les corrélations temporelles du réseau. Dans ce cadre de modélisation, [45] a proposé un modèle spécifique appelé DDP-GCN (Distance, Direction, and Positional relationship Graph Convolutional Network) afin d'incorporer trois relations spatiales dans le réseau de prédiction pour la prévision du trafic. [32] a utilisé des convolutions de graphe pour capturer les modèles spatiaux et une convolution commune pour les caractéristiques temporelles avec un réseau neuronal convolutionnel de graphe spatio-temporel (CNN) en y intégrant des mécanismes d'attention. Pour la prévision du trafic multi-voies à court terme, [56] a remplacé un LSTM standard par un LSTM convolutif pour traiter les grands réseaux de transport. Ces approches permettent de tenir compte de l'ensemble des dynamiques d'un réseau de transport mais nécessitent de disposer des données consolidées pour l'ensemble du système de transport.

Dans le cadre de ces travaux, les données comme introduites dans ce chapitre ne permettent pas d'utiliser des méthodes de prédiction à l'échelle du réseau complet. Les données de charge ne sont collectées que sur 3 lignes métro (parmi 14) pour un réseau aussi dense que le métro parisien.

1.4.2 Échantillonnages temporels

Le second axe à considérer pour la prédiction d'une variable d'un réseau de transport est l'échantillonnage temporel des données. Cette question est souvent liée au moyen d'acquisition des données. Dans le domaine de transport en général, la prédiction est faite de manière régulière (pas d'échantillonnage constant) car elle s'accompagne de données de comptage ou de vitesse sur des segments. Cependant dans le cadre du transport en commun, la variable à prédire comme la charge ou les temps d'attente ou de parcours, sont à la fois dépendantes des segments comme pour les transports routiers mais elles sont principalement dépendantes des véhicules (bus, métro).

1.4.2.1 Échantillonnage régulier

La grande majorité des modèles repose sur une approche régulière de la prédiction de variables d'un système de transport en commun. Les données et prédiction sont prises en compte à intervalle régulier selon les horizons de prédictions. Des modèles de prédictions allant de modèles classiques auto-régressifs [9, 33] aux architectures plus complexes d'apprentissage profond [23, 56] se basent sur des agrégations régulières des données. Ce choix d'échantillonnage fait l'impasse de certaines caractéristiques réelles d'un système de transport comme les heures réelles de départ de trains ou de bus.

1.4.2.2 Échantillonnage irrégulier

Les données brutes sont rarement régulières [79] de par les procédés qu'elle décrivent : le déplacement de métro ou bus. Les données de la ligne 9 sont un parfait exemple du déplacement fortement irrégulier des trains le long d'une ligne. De plus, à travers l'exemple des situations atypiques, une agrégation temporelle n'est pas efficace pour tenir compte de contextes de charge complexes liés aux mouvements et gestion des trains. Il est alors possible de s'intéresser aux véhicules qui pour un système de transport public porte la charge passagers. De plus, dans un contexte de prédiction de charge, l'information du train s'avère pertinente car elle traduit réellement les flux entre stations du réseau. Les séries temporelles à prédire suivent dans ce cas les mouvements de train. Pour résoudre cette contrainte, nous pouvons noter principalement deux travaux de prédiction. Les travaux de [63] ont utilisé un encodeur-prédicteur LSTM pour prendre en compte la table horaire réelle dans la prédiction de charges de train au niveau de la station. Au niveau d'une ligne, [18] construit un modèle de mélange Gaussien Bayésien pour prédire les temps de parcours en capturant les interactions entre deux bus successifs le long d'une ligne.

1.4.3 Prédiction de situations atypiques

Dans ces travaux, la prédiction de charge ou d'un autre indicateur de l'état d'un réseau de transport public doit proposer une analyse des performances sur des contextes nominaux et atypiques. On ne doit pas se limiter aux heures de pointe qui représentent la plupart du temps la seule analyse différenciée sur les résultats que l'on trouve dans les travaux de l'état de l'art (cf. section 3). Cette faiblesse est souvent due à la faible profondeur temporelle des données utilisées dans de nombreux travaux construits sur quelques mois de données. L'évaluation des modèles dans des contextes sous représentés et variés nécessite l'usage d'un historique de données large, ce qui justifie le peu de travaux dans la littérature (ces données souvent sensibles pour les opérateurs de transport).

- Échantillonnage par train
- Prédiction d'une ligne
- plusieurs pas de temps (Multi-step)
- Focus sur les situations atypiques

1.5 Conclusion

Ces travaux reposent sur des données réelles de la ligne 9 du métro parisien qui depuis 2019 reflètent des contextes de transport variés et atypiques induits par la crise du Covid 19 et la généralisation du télétravail. Bien que l'offre soit quasi constante au cours de ces dernières années, la demande associée a connu de grandes variations. En considérant l'année 2019 comme situation nominale, la fréquentation des transports en commun et les charges à bord des trains ne sont pas revenus à la situation nominale. La prédiction de charge doit pouvoir ainsi s'accommoder à des contextes temporels variés ainsi qu'à des situations atypiques futures.

La thèse s'attache à développer des modèles de prédiction de charge passagers pour une ligne de transport. La prédiction est une tâche tributaire des données et des systèmes d'information des opérateurs de transport. En effet ce sont les données qui orientent le choix des modèles. L'approche de prédiction orientée données (data-driven) est mise en place pour répondre à la question de prédiction sur un horizon temporel court tout en

tenant compte des contraintes des données et de la portée de la prédiction. Nous visons une prédiction court-terme de charge en situation nominale et en situations atypiques.

Deux axes majeurs sont à considérer pour construire un modèle de prédiction, l'agrégation spatiale et temporelle. Ces deux choix de modélisation sont déterminants pour les modèles de prédiction. Dans le cadre de ces travaux et en vue de l'analyse des situations atypiques, prédire la charge par train (et non à l'échelle d'une station) semble être la solution la plus pertinente car elle traduit le trafic réel opéré par les trains. Nous prenons ainsi en compte l'impact du contexte atypique sur la marche des trains sans perdre aucune information qu'une agrégation des données aurait engendrée. Pour l'aspect spatiale, au regard du réseau parisien très dense, une approche "réseau" serait à envisager afin de considérer toutes les inter-connexions entre stations. Cependant, la disponibilité des données de charge uniquement sur 3 lignes a orienté notre choix vers une modélisation "ligne". Notre objectif sera de fournir la prédiction de charges à chaque station de la ligne 9 pour les quatre prochains départs.

Chapitre 2

Formalisation du problème de prévision multiple comme une tâche de traitement d'image

2.1 Introduction

La prédiction des variables d'état d'un réseau de transport doit prendre en compte divers aspects et caractéristiques du-dit système. Les contraintes spatiales et temporelles sont les premières à considérer pour construire une méthodologie de prévision. Elles sont pour la plupart issues des données elle-mêmes. La disponibilité ou l'absence de données pour des stations ou des lignes priorisent l'usage de certaines approches au détriment d'autres. Une approche complète de prédiction à l'échelle d'un réseau de transport ne peut être réalisée que si les données disponibles sont définies pour l'ensemble du réseau.

La prévision de la charge à bord ou d'une autre variable d'état d'un système de transport, telle que le retard par exemple, repose sur l'analyse des séries temporelles la définissant. Les choix de modélisation et les approches doivent être adaptés aux jeux de données. La plupart des travaux se basent sur des agrégations temporelles régulières (au sens échantillonnage temporel), celles-ci, in-fine font abstraction des trains.

La majorité des modèles de prévision ne tiennent pas compte des matériels ferrés et réduisent la prédiction des flux passagers à l'inférence du nombre de voyageurs pendant une période de temps [67, 14]. Cependant, prédire la charge en nombre de passagers par train est plus pertinent en sachant la charge des trains pour les arrêts précédents. Une telle approche semble plus intéressante mais elle n'est pas sans difficultés. En effet, s'intéresser aux trains impose dans le cadre d'une ligne ou d'un réseau de transport de pouvoir encoder tout les trains y circulant comme le montre la figure 2.1, leur nombre et leur position relatives changeant de plus au cours du temps. L'intérêt de se concentrer sur les trains permet ainsi de mieux visualiser les dynamiques de charges et de représenter ainsi de manière pertinente les situations atypiques.

Un prédiction par véhicule et sur l'ensemble d'une ligne est rare [18]. De plus, dans la plupart des travaux, la prédiction est construite localement à partir des informations sur les stations et les trains proches et ne tient pas compte de l'état complet de la ligne. Dans ce chapitre, nous allons voir, comment une modélisations des données dans un format de type image permet de représenter l'état complet d'une ligne de métro, et comment cette approche peut être mobilisée pour résoudre le problème de la prédiction simultanée de l'ensemble des charges des trains parcourant la ligne.

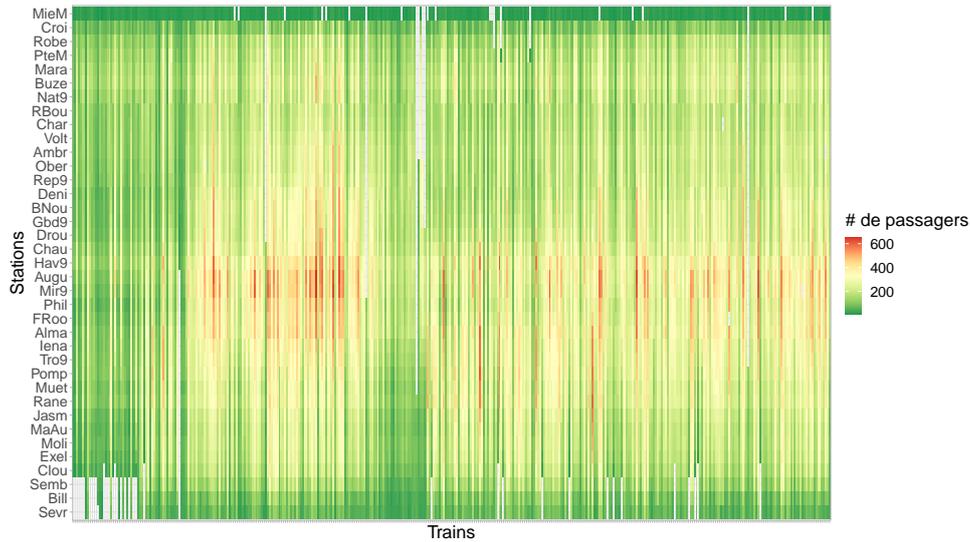


FIGURE 2.1 – Parcours de l’ensemble des trains de la ligne 9 de la station Pont de Sèvres à Mairie de Montreuil. La couleur est associée au nombre de passagers à bord des rames.

2.2 Le format image

Une image est un format de données qui vise à reproduire dans un plan une information spatiale. Dans ce travail, nous nous intéresserons uniquement aux images en 2D. L’utilisation d’image en 3D ou comportant plus de dimensions sera abordée dans les perspectives de ce travail. Une image est composée de deux dimensions, une hauteur h , une largeur l et des pixels. Le nombre de pixels (ou la résolution d’une image) est défini en fonction du produit de h et l . Par exemple, une image en haute définition possède un nombre minimum de pixels de 1280 par 720 (on parle de 720p). Un pixel est l’élément unitaire d’une image et comprend c canaux où chaque canal représente l’axe des variables. Ainsi, la taille d’une image I est définie selon :

$$|I| = h \times l \times c \quad (2.1)$$

Pour une image en noir et blanc, un pixel correspond à une valeur de nuance de gris. Pour une image en couleur (RVB : Rouge, Vert, Bleu) par exemple, chaque pixel comprend 3 variables pour chacune des couleurs primaires (synthèse additive des couleurs). Ainsi, pour une image de 16 pixels de côté, on notera une taille de $16 \times 16 \times 3$.

2.3 Représentation de l’état d’une ligne de métro dans un format image

Le trafic d’une ligne de métro est défini en fonction de trois éléments :

- Les points d’arrêt ordonnés de la ligne
- L’heure de départ des trains
- les trains (missions)

Pour illustrer ces 3 dimensions, la Figure 2.2 représente les trajets réalisés par les trains de la ligne 9, du Pont de Sèvres à la Mairie de Montreuil, pendant plus de 3 heures le 15 février 2020. Les trains représentés par les lignes dans ce diagramme espace-temps (aussi appelé diagramme de Marey) parcourent les différents arrêts au cours de leurs missions.

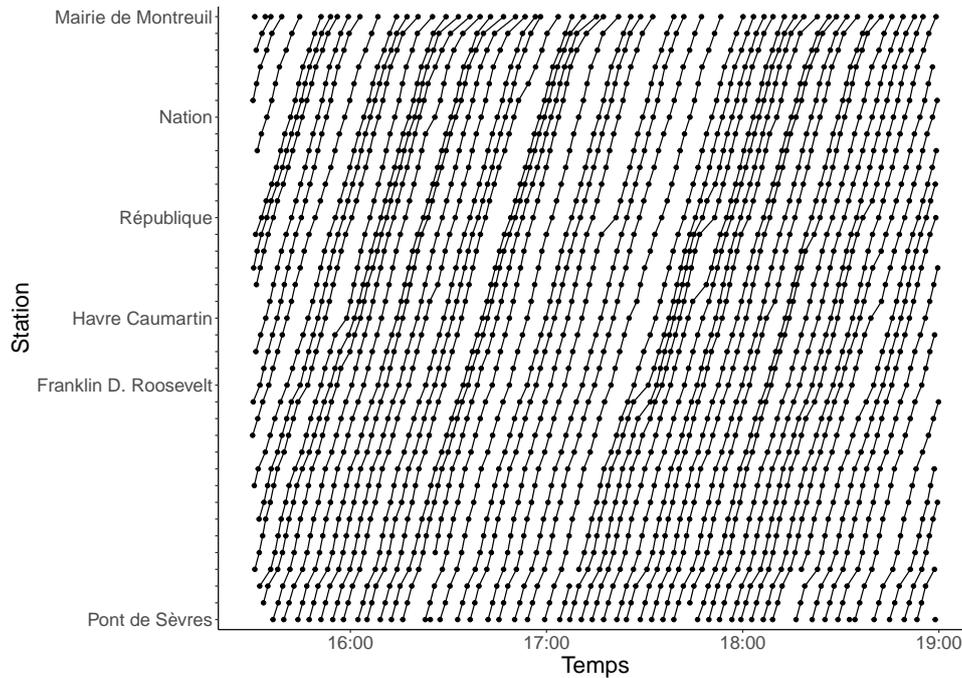


FIGURE 2.2 – Courses des trains entre Pont de Sèvres et Mairie de Montreuil entre 15 :30 et 19 :00 le 15 février 2020. Une ligne représente le parcours d’un train et les points un arrêt.

L’objectif est de considérer le maximum de ses éléments dans la représentation des données pour la prédiction. En effet, de nombreux travaux ne tiennent pas compte toujours de l’ensemble de ces caractéristiques, soit à cause de l’agrégation temporelle des données qui supprime les heures effectives des départs de train [51, 76], soit parce que la prévision de charge ne considèrerait qu’une seule station [63]. Pour répondre à ces contraintes, nous mettons les données sous la forme d’une image, ce qui permet d’encoder ces trois variables simultanément.

Dans le cadre de ces travaux, une image prend en compte comme largeur et hauteur deux dimensions : les points d’arrêt et les trains. Ainsi, un arrêt de train défini par un couple train-stations est représenté par un pixel. La Figure 2.3 schématise la transposition d’un ensemble de 4 trains sous la forme d’une image. À gauche, il s’agit d’un exemple de graphe de Marey qui est transposé en une image à droite. Les flèches représentent le lien entre le départ d’un train et les colonnes de l’image. ainsi, chaque pixel fait référence au statut d’un train à une station données. Les informations sur les trajets des trains sont conservées dans la structure de l’image, ainsi que l’ordre des trains. Les différents canaux de l’image peuvent être utilisés pour encoder différentes informations, heure de départ, temps de parcours, charges à bord, ...

De plus, la valeur des pixels peut être utilisée pour encoder d’autres variables, comme la charge par exemple (introduite dans la section 2.2). Par conséquent, une image est pertinente pour encoder les données d’un réseau de métro. Elle permet en effet d’encoder les trains et les stations indépendamment de leur échantillonnage temporel sous un format standard. Les hypothèses nécessaires à une telle mise en oeuvre (en particulier l’existence d’un ordonnancement complet des trains, i.e pas de dépassement) sont de plus raisonnables dans le contexte d’une ligne de métro, qui est fortement contrainte dans son exploitation.

Une image I est une structure de données qui prend en compte les relations d’ordre à

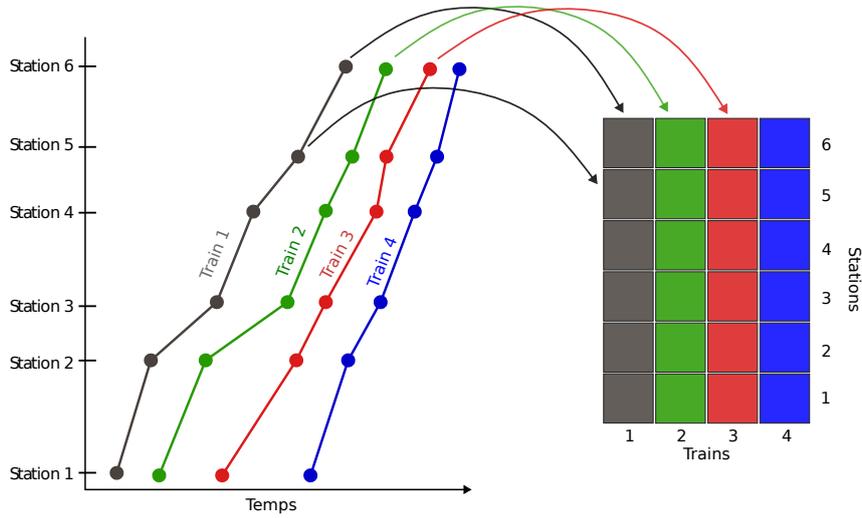


FIGURE 2.3 – Schéma présentant la transposition d'un ensemble de quatre trains en image, les couleurs encodent les trains.

la fois entre les stations d'une ligne et les trains qui circulent sur celle-ci. La hauteur et la largeur de l'image ne sont plus définies par une résolution de pixels, mais par une ligne de métro et les courses des trains. Ces deux ensembles sont des espaces ordonnés par l'ordre des stations et l'ordre des trains sur la ligne.

Nous noterons $I : \mathbb{R}^S \rightarrow \mathbb{R}^C$ l'image créée à partir d'un espace univarié ordonné S et d'un ensemble de séquences ordonnées $c \in C$ associées respectivement aux stations de métro et aux courses des trains. Le pixel noté $x_{s,c}$ est un vecteur de variables d'entrée associées à la fois au train c et à la station s . On peut alors noter I comme la matrice suivante :

$$I = \begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,3} & \dots & x_{1,|C|} \\ x_{2,1} & x_{2,2} & x_{2,3} & \dots & x_{2,|C|} \\ \dots & \dots & \dots & \dots & \dots \\ x_{|S|,1} & x_{|S|,2} & x_{|S|,3} & \dots & x_{|S|,|C|} \end{bmatrix} \quad (2.2)$$

La figure 2.4 représente l'image de la charge de passagers sur la ligne de métro de Paris résultant de la matrice 2.2. La valeur du pixel correspond dans ce cas à la charge à bord des trains lors du départ de chaque station. On remarque dans cette image l'évolution de la charge à la fois pour les trains mais aussi pour les stations centrales de la ligne dont la charge est la plus importante.

À l'intérieur de l'image, les trains circulent toujours dans la même direction. Ainsi, chaque ligne est définie par deux images pour chaque direction possible. Il est important de diviser la ligne en deux images afin de conserver la relation d'ordre de chaque pixel. En effet, les coordonnées d'un pixel dans l'image sont définies selon les inégalités d'ordre suivantes, en notant $t(x_{s,c})$ le temps de départ associé au pixel x où c et s sont respectivement la mission et la station encodée par le pixel :

$$\begin{aligned} t(x_{s,c}) &> t(x_{s-1,c}) \\ t(x_{s,c}) &< t(x_{s,c-1}) \end{aligned} \quad (2.3)$$

La figure 2.5 illustre visuellement les deux inégalités de 2.3. D'un point de vue temporel, la même relation d'ordre peut être établie avec les heures de départ ou d'arrivée des trains. De plus, la flèche indique la direction du temps entre ces pixels.

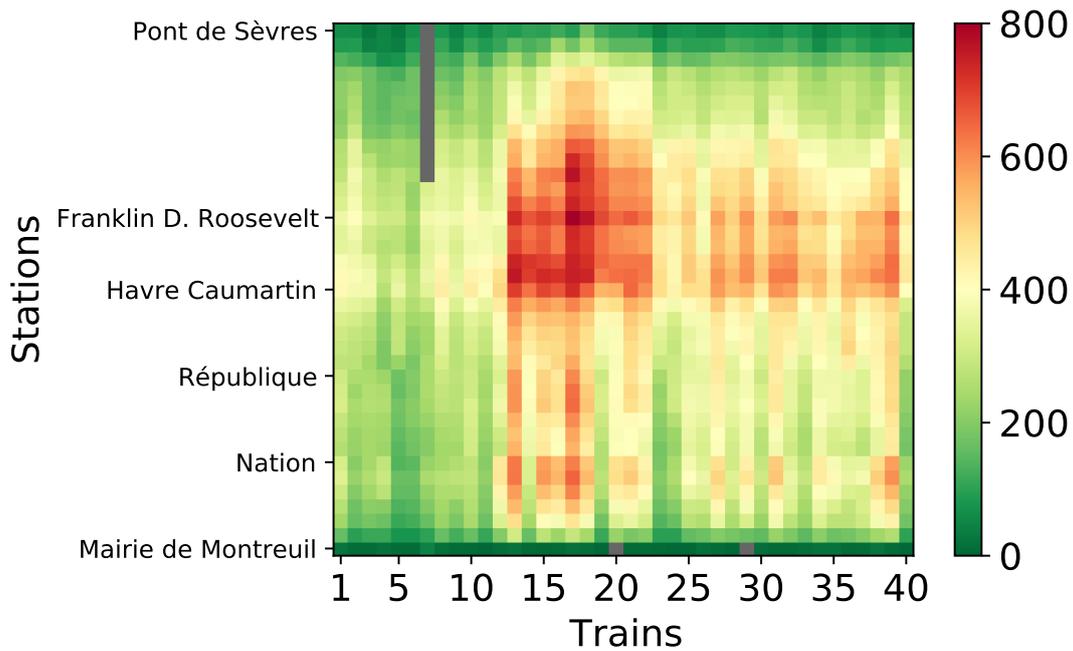


FIGURE 2.4 – Image typique de charge

Avec cette approche, on peut désigner une colonne de l'image comme la série temporelle associée à un train c : $(x_{1,c}, x_{2,c}, \dots, x_{s,c}, \dots, x_{|S|,c})$. De même, une ligne dans l'image est la série temporelle associée à une station s : $(x_{s,1}, x_{s,2}, \dots, x_{s,c}, \dots, x_{s,|C|})$. A partir des deux ensembles S et C , on peut représenter l'image I d'une ligne de métro sous la forme d'une matrice de pixels.

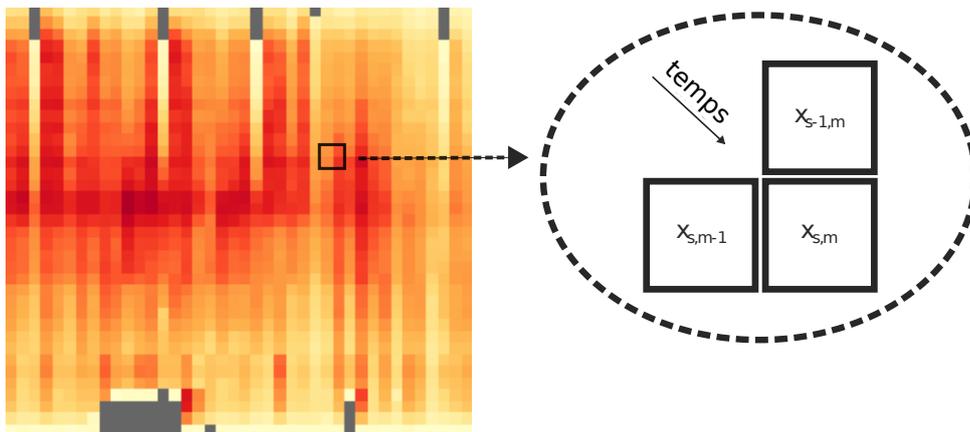


FIGURE 2.5 – Relation d'ordre entre les différents pixels de l'image.

Comme évoqué précédemment, chaque pixel $x_{s,c}$ encode, en plus de la position du train dans le réseau et son séquençage par rapport aux autres trains, des variables qui peuvent être associées au train, à la gare ou à des données contextuelles (par exemple, grèves, incidents). Ces variables représentent les canaux de l'image. Les matrices 2.4 et 2.5

représentent par exemple les canaux I_d et I_l associés aux variables l et d où $x_{s,c} = (l_{s,c}, d_{s,c})$.

$$I_l = \begin{bmatrix} l_{1,1} & l_{1,2} & l_{1,3} & \dots & l_{1,|C|} \\ l_{2,1} & l_{2,2} & l_{2,3} & \dots & l_{2,|C|} \\ \dots & \dots & \dots & \dots & \dots \\ l_{|S|,1} & l_{|S|,2} & l_{|S|,3} & \dots & l_{|S|,|C|} \end{bmatrix} \quad (2.4)$$

$$I_d = \begin{bmatrix} d_{1,1} & d_{1,2} & d_{1,3} & \dots & d_{1,|C|} \\ d_{2,1} & d_{2,2} & d_{2,3} & \dots & d_{2,|C|} \\ \dots & \dots & \dots & \dots & \dots \\ d_{|S|,1} & d_{|S|,2} & d_{|S|,3} & \dots & d_{|S|,|C|} \end{bmatrix} \quad (2.5)$$

Ainsi, l'ajout d'une nouvelle variable équivaut à l'ajout d'un nouveau canal à l'image. La figure 2.6 montre les quatre canaux obtenus à partir des données de la ligne 9, à savoir le nombre de passagers, le temps d'attente, le temps de trajet et la validation. Le format image permet d'ajouter facilement de nouvelles sources de données sans pour autant changer la structure de données utilisée.

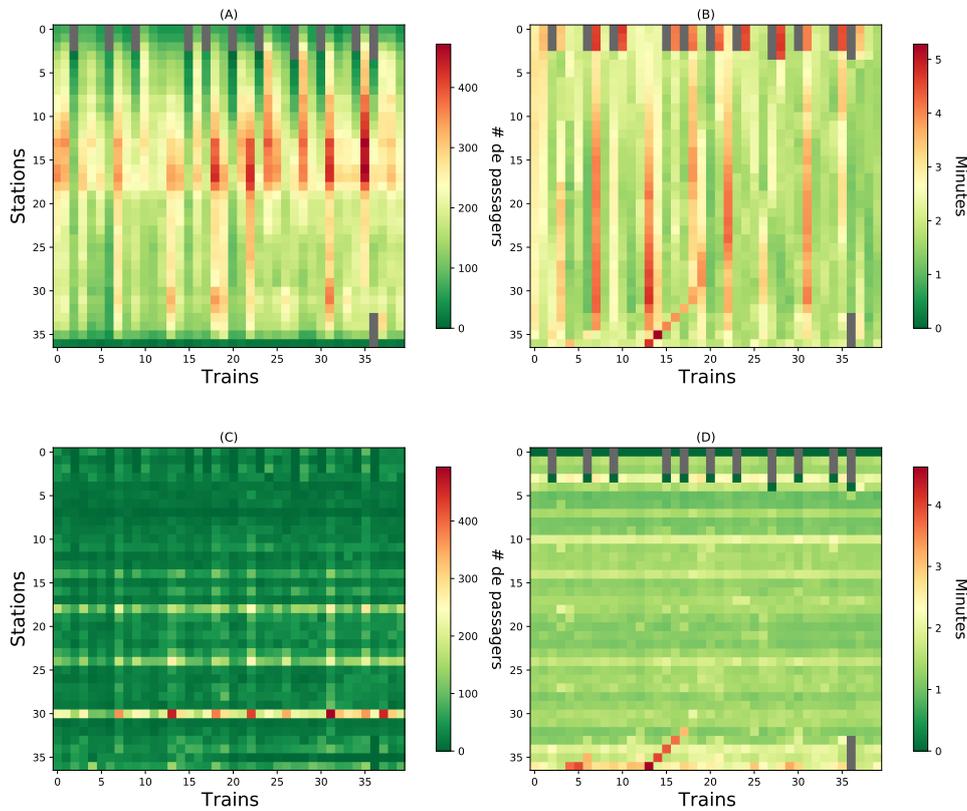


FIGURE 2.6 – Canaux d'une image de trafic de la ligne 9. (A) canal de charges passagers, (B) canal des intervalles de départ, (C) canal des validations télé-billettiques, (D) canal des temps de parcours

Pour effectuer l'apprentissage d'un modèle de prédiction (des charges ou des temps de parcours), il est nécessaire de limiter les images à une taille standard afin de faciliter leur traitement et leur génération en fonction des objectifs opérationnels. Il n'est pas pertinent de créer une image de toutes les missions des trains pour une journée entière si l'objectif est une prédiction à court terme. Nous optons alors pour générer l'image en fonction du temps d'acquisition de l'image. L'objectif est de sélectionner des missions de

C temporellement cohérentes avec l'instant d'acquisition t et les tâches opérationnelles (c'est-à-dire la détection des situations atypiques, la prédiction de la charge) que nous noterons C_t . Une fois cet ensemble de trains défini lequel rassemble les missions à utiliser pour la création de l'image à l'instant t , l'image est simplement construite à partir de l'ordonnancement des trains de C_t .

Dans le contexte de la prédiction de la charge, deux paramètres doivent être pris en compte pour définir la taille de cet ensemble. Nous devons en effet disposer d'un historique de trains suffisamment important compte tenu du mouvement des trains et de l'horizon de prédiction. On peut définir $|C_t|$ en fonction d'un nombre minimum de trains à considérer pour la génération de l'image. Ce minimum peut être borné par la capacité q de la ligne et par l'horizon de prédiction h comme suit :

$$|C_t| > q + h \quad (2.6)$$

La capacité q fournit la quantité maximale de trains qui peuvent circuler simultanément sur une ligne de métro. Pour l'horizon de prédiction h , il est relatif à l'objectif de prédiction et se réfère au nombre total de trains futurs entrant dans le réseau mais n'ayant pas encore fait d'arrêt.

On peut noter qu'il existe des effets de bord en début et en fin de journée. En début de journée, il n'y a pas assez de trains sur la ligne pour respecter 2.6. Nous remplaçons alors les trains manquants par de fausses missions de trains vides. Cette stratégie s'apparente à un traitement de type padding avec des zéros. Pour la fin de journée, c'est l'horizon de prédiction h qui tend vers 0 au fur et à mesure que l'on se rapproche de la fin du service. Aucune nouvelle mission n'est prévue pour la ligne et ainsi seuls les trains encore en mouvement sont à prédire. Contrairement au début de service, nous n'ajoutons pas de padding afin de compléter l'image.

2.3.1 Extension possible à des réseaux

Jusqu'à maintenant, le format image introduit ne traite que de lignes simples. Cependant, la prédiction de la charge pourrait être effectuée globalement à l'échelle d'un réseau complet. Ce choix dépend des données qui peuvent limiter l'expérimentation à une seule ligne ou station, mais une approche prenant en compte un réseau complet pourrait permettre de mieux modéliser les phénomènes de report entre lignes lors d'incidents par exemple.

Ainsi, nous souhaitons mentionner la possibilité de représenter l'ensemble d'un réseau de transport par une image. En effet, l'approche image peut être étendue pour tirer parti de l'encodage des trains à l'échelle de l'ensemble du réseau de transport. Cependant, cela introduit de nouvelles contraintes à la fois dans la prédiction et la représentation des flux de passagers. Pour ce faire, il serait possible d'utiliser une image 3D avec 3 axes principaux :

- Origine de l'inter-station : Point d'arrêt de départ d'un déplacement unitaire (*i.e.* une inter-station)
- Destination de l'inter-station : Point d'arrêt d'arrivée d'un déplacement unitaire (*i.e.* une inter-station)
- Trains : séquence ordonnées des trains **de chaque ligne**

Cette image peut être présentée d'une autre manière comme une superposition de matrices Origine-Destination (OD) des trains circulant sur les lignes. Chaque matrice OD contient la trajectoire déjà réalisées d'un train de chaque ligne. La première matrice

contient la trajectoire du dernier train parti du terminus de chacune des lignes, la seconde de l'avant dernier train de chaque ligne et ainsi de suite jusqu'à une profondeur fixée. Pour le réseau de métro parisien complet, une telle matrice contiendrait donc les trajectoires réalisées par 14 trains, un par ligne.

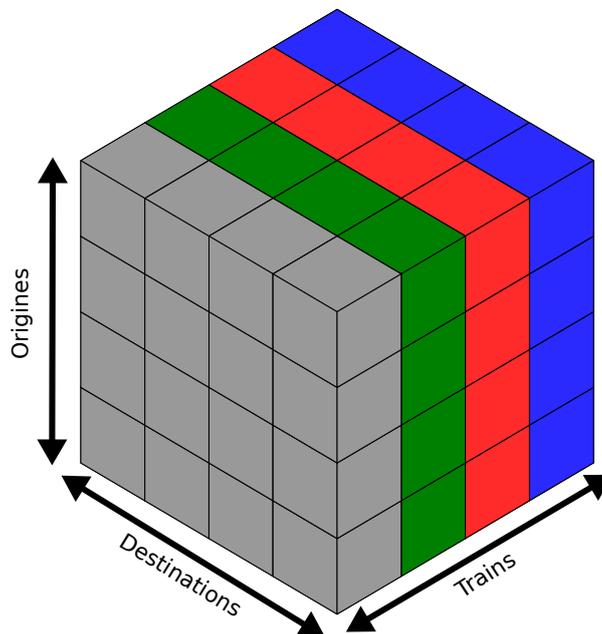


FIGURE 2.7 – Schéma d'une image représentant un réseau de transport complet.

La figure 2.7 montre un exemple d'image de réseau. Chaque couleur représente une matrice OD pour un train de chaque ligne. Il s'agit d'imiter les colonnes des images précédemment décrites avec un train par colonne mais dans le cas d'un réseau complet, il s'agira d'un train par ligne de métro. L'ordre des matrices OD est similaire aux images 2D en tenant compte de l'ordre des trains dans chaque ligne. L'usage de matrices OD permet de mieux prendre en compte les correspondances au sein d'une station en encodant pour une station l'ensemble des lignes et départs de trains possible à la station.

Une telle image présente l'avantage d'encoder les connexions entre les lignes de métro en ne séparant pas les stations par ligne. Mais la contrepartie est la création d'une image creuse où la plupart des pixels sont vides, car un train ne parcourt pas l'ensemble des couples origines/destinations. Enfin, dans le cas d'une ligne simple telle que celle que nous traitons dans la thèse, il existe un ordre naturel aussi bien pour les trains que pour les stations. Dans le cas de l'approche globale, ce n'est plus le cas en ce qui concerne les origines et les destinations et un ordonnancement doit donc être choisi.

2.4 Encodage des différentes variables

La structure d'image sur laquelle repose notre approche permet d'encoder sous forme de canal toutes les variables associées à un train et/ou à une gare. Un canal peut ainsi encoder des données communes à une gare, un train ou une période de temps (i.e. ouverture de la gare, type de mission ou période de grève) ou des données relatives à un couple train/gare (i.e. charge, temps d'arrêt). Cependant, chaque donnée doit être adaptée au format de canal selon le type de données d'une variable. On dénombre trois types de variables cibles : quantitative, qualitative et temporelle.

2.4.0.1 Données quantitatives

Les données quantitatives sont encodées de manière triviale dans une image exactement comme les couleurs d’une image standard. On associe à chaque pixel la valeur de la variable cible. Dans le cas de la charge, on transpose la charge pour chacun des couples train/station (voir Figure 2.4).

Cependant, il peut être nécessaire de pré-traiter les données pour s’adapter au format image comme par exemple pour les données de comptage ou de validation, qui sont unitaires. Il s’agit alors d’agrèger les données pour chacun des pixels de l’image. Par exemple, dans ces travaux, nous avons choisi d’agrèger les validations en associant à chaque couple train/gare le nombre de personnes entrant dans cette gare entre le train à quai et le précédent. Il est toutefois possible de tenir compte d’autres stratégies : agrèger sur un intervalle de temps (validation à 15 minutes) ou utiliser comme entrée des données agrégées issues de procédés industriels annexes.

2.4.0.2 Données temporelles

Les données temporelles nécessitent un traitement particulier. Il est en effet nécessaire de se poser la question de l’encodage de l’heure de départ en tenant compte également de l’heure de la génération de l’image.

Un approche naïve consisterait à encoder directement l’heure de départ dans chaque pixel ou l’intervalle de temps entre ce pixel et l’instant t de la génération de l’image. Cependant, les valeurs de pixels ne seraient pas normalisées entre les différentes régions de l’image. Pour obtenir une donnée cohérente dans toute l’image, nous avons choisi d’encoder les heures de départ en les décomposant entre le temps de parcours des trains et les intervalles de départ à chacune des stations. Le premier correspond à la durée totale du trajet d’un train d’une gare à l’autre. La seconde variable est similaire, mais elle est liée à la gare et indique le temps écoulé entre deux trains. Ces durées sont calculées localement entre deux pixels. Cet encodage permet de garder une cohérence dans l’image et d’une image à une autre.

2.4.0.3 Données catégorielles

Le traitement des données qualitatives constitue la dernière catégorie de données à prendre en compte. Lorsqu’il s’agit d’une variable qualitative binaire, il est possible de la représenter facilement dans un canal. Cela implique la création d’un masque lié à la variable cible. Par exemple, la figure 2.8 illustre le masque binaire de la fermeture des stations, indiqué en noir sur la figure.

Pour les données qualitatives non binaires, deux possibilités peuvent être envisagées. La première est similaire au One-hot encoding, pour chaque valeur de la variable (sauf une), un canal binaire est ajouté à l’image. Cependant, la dimension de l’image peut être fortement augmentée et avoir un impact sur le coût d’apprentissage des modèles de traitement d’image (c’est-à-dire les ressources nécessaires à l’apprentissage). La deuxième possibilité consiste à intégrer la variable par le biais d’auto-encodeurs ou d’approches de réduction de la dimension.

Ainsi, le format image peut tenir compte de données qualitatives binaires ou non. Dans le second cas, un pré-traitement est nécessaire afin d’éviter que l’image présente un grand nombre de canaux. Dans ces travaux de thèse, aucune donnée qualitative n’a été utilisée comme entrée.

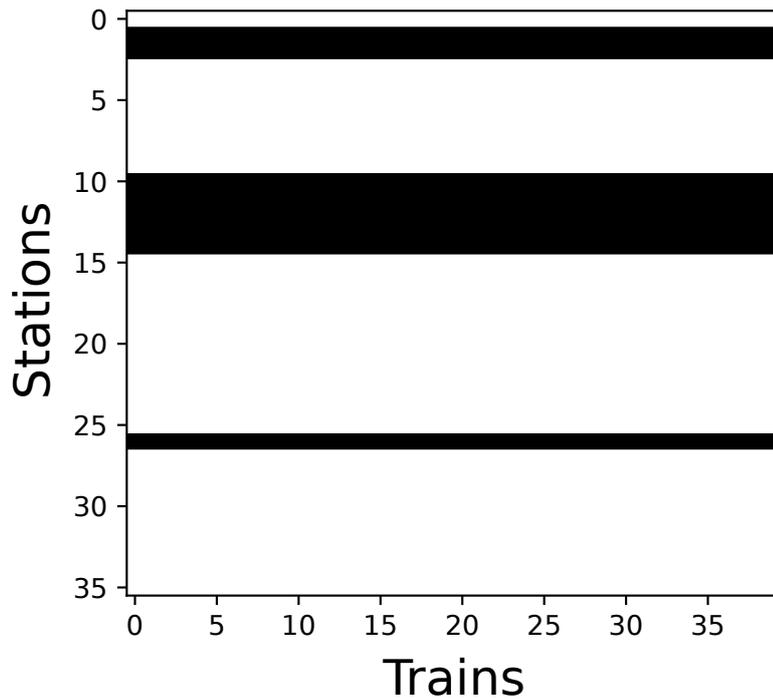


FIGURE 2.8 – Masque indiquant en noir les stations fermées.

2.5 Données manquantes

Pour encoder une ligne de métro dans une image, nous devons prendre en compte les données manquantes. Il existe deux origines possibles pour les données manquantes pour une ligne de métro. La première catégorie concerne les données manquantes aléatoires dues à une erreur de mesure. La seconde catégorie de pixels manquants correspond à des données manquantes par construction et donc non aléatoires. Les pixels manquants se réfèrent respectivement aux données manquantes lors de la génération de l'image et aux situations sur la ligne (par exemple, gare fermée, itinéraire de train raccourci). La figure 2.9 illustre la présence de pixels manquants (en gris).

2.5.1 Données manquantes aléatoires

Les données manquantes aléatoires sont les plus courantes. Elles sont dues à des problèmes de l'outil de mesure. Elles se traduisent principalement dans l'image par des pixels morts "solitaires" dans l'image. Dans la figure 2.9, plusieurs pixels morts aléatoires peuvent être trouvés en bas à droite de l'image (en noir dans l'image). Ils diffèrent des autres pixels mixtes qui représentent un événement opérationnel survenant sur la ligne.

On attribue également à cette catégorie de données manquantes les courses de trains 6 et 7 en noir dans l'image de la figure 2.9. Bien que ces pixels représentent une opération de régulation (il s'agit de deux trains ayant effectué un demi-tour), ce type de modification de l'offre est non prévue et ne peut pas être anticipée comme une fermeture de stations. On les traite alors comme des données manquantes aléatoires car elle sont liées à des événements non prévus.

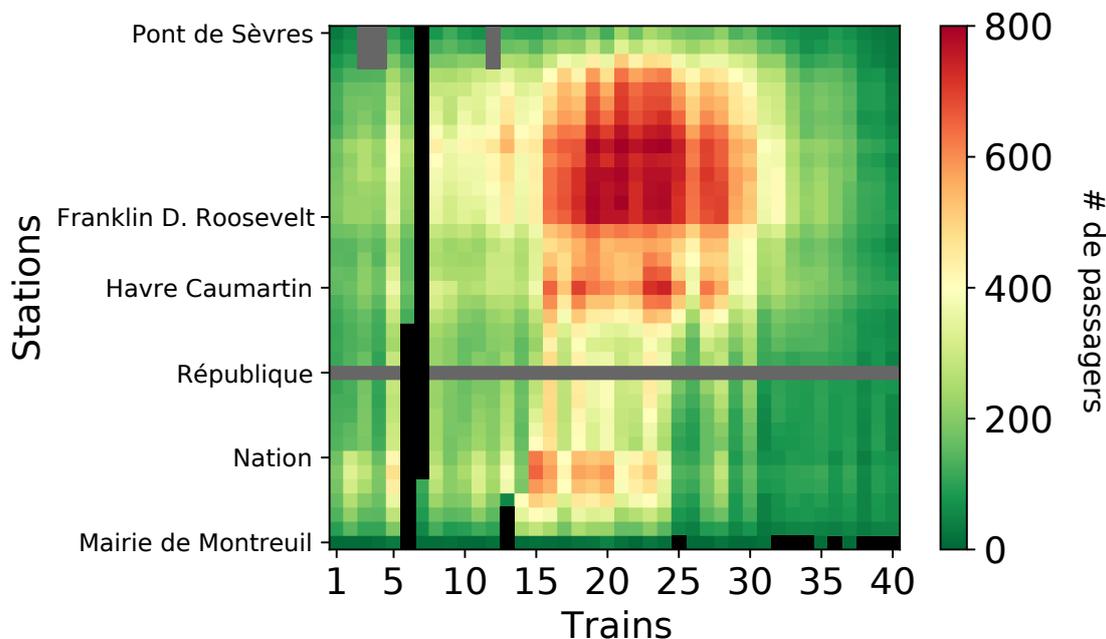


FIGURE 2.9 – Image de charge avec différents cas de données manquantes en gris

2.5.2 Données manquantes non aléatoires

La deuxième catégorie de pixels manquants représente la majorité des pixels et des données manquants. Les pixels manquants apparaissent lors de la génération des images et représentent des événements planifiés (stations fermées pour travaux, départs de terminus intermédiaires). Ces derniers sont mentionnés en gris dans la Figure 2.9. La figure 2.9 illustre plusieurs cas tels que des stations fermées planifiées qui sont visibles par des lignes grises dans les images (stations République dans l'exemple). Dans l'image, les pixels manquants situés en haut à gauche de l'image représentent également des pixels morts définies selon l'offre de transport. Il correspondent à des départs de trains au terminus intermédiaire de porte de Saint-Cloud. Ainsi, la totalité des pixels manquants prévus découlent de choix opérationnels pris lors de la construction de l'offre de transport. Il apparaissent par la construction de l'image et peuvent représenter des contextes atypiques prévus comme la fermeture de stations pour travaux. Il est possible alors de différencier les pixels manquants selon leur origine lors de la génération de l'image. Dans le cadre de ces travaux, aucune distinction n'est faite entre ces deux situations. Les données manquantes sont transcrites dans l'image par une valeur négative (-1 par exemple). Il s'agit de différencier les données manquantes des charges de trains nuls.

2.5.3 Le cas de la ligne 9 du métro Parisien

Compte tenu des données présentées dans le premier chapitre, les images seront basées sur la ligne 9 du métro parisien. Cette ligne fait partie des trois lignes de métro pour lesquelles des mesures de charge sont disponibles. Elles ne prendront en compte que les trains allant du Pont de Sèvres à la Mairie de Montreuil.

Les images des trains de la ligne 9 comportent 36 stations et 40 trains. Bien que la ligne 9 ait 37 stations, comme aucun départ commercial n'est effectué à la fin de la ligne, nous omettons la dernière station, ce qui diminue le nombre de stations de 1. Pour le nombre de trains, nous utilisons la capacité q_9 de la ligne 9 estimée à 36 trains maximum

circulant sur la ligne. Pour la prédiction, le travail est basé sur la prédiction interne du temps d'attente de la RATP fournissant les 4 prochains départs de trains par station.

En ce qui concerne les variables disponibles, une image de la ligne 9 est composée de 4 canaux numériques présentés dans la figure 2.6 :

- Charge de passagers : le nombre de passagers à l'intérieur d'un train au moment de son départ.
- Temps d'attente : le nombre de minutes entre le départ du train actuel et le départ précédent dans la même gare.
- Validations télé-billettiques : Le nombre de passagers entrant dans le réseau dans une gare pendant la période entre le train actuel et le précédent.
- Temps de parcours : la durée en minutes du voyage d'un train de la gare précédente à la gare actuelle.

2.6 Masque de prédiction

Jusqu'à présent, les images présentées sont complètes, tous les arrêts des trains en station représentés sont observés. En temps réel, les images prennent une toute autre forme car les trains circulent toujours sur la ligne et certains arrêts n'ont par conséquent pas encore été observés. Une image est alors un instantané révélant la position des trains sur la ligne et leur dynamique passée.

Une image peut être générée à la fin du service pour observer et analyser la dynamique qui a eu lieu pendant la journée, mais dans une perspective de temps réel, la forme des images est différente. À un instant t , des trains circulent sur la ligne et leur position change en continu.

Une région manquante peut alors être observée à l'intérieur de l'image. Dans la figure 2.10, les pixels manquants en gris renvoient à la connaissance du réseau au moment de la génération. Le nombre de pixels en gris est corrélé au contexte de l'image. Ils faut toutefois ne pas les confondre avec les pixels manquants en noir qui représentent des départs de trains précédant à t . Une image des heures de pointe aura une zone manquante plus importante que celle des heures creuses où moins de trains circulent sur le réseau. On parle alors d'une image temps réel que l'on note $I^{in}(t)$ et qui désigne une version dégradée de $I(t)$ prenant en compte uniquement les données connues à l'instant t .

Nous notons l'importance de choisir une taille d'image basée sur la capacité maximale de la ligne pour inclure tous les trains qui peuvent circuler sur la ligne.

À l'intérieur d'une image temps réel, un pixel est considéré comme manquant en fonction de l'heure de départ du train $t(x_{s,c})$ dénotée par l'équation suivante :

$$si \quad t(x_{s,c}) > t \quad \Rightarrow \quad x_{s,c} = \emptyset \quad (2.7)$$

où $t()$ est une fonction donnant l'heure de départ du train associé au pixel. Les pixels futurs sont des pixels morts et prennent comme valeurs les valeurs nulles de chaque variable.

Ainsi, dans une perspective temps réel, le format image fait apparaître une région manquante de départ de trains. La prévision du nombre de passagers dans les trains doit être effectuée en complétant tous les pixels de l'image. Dans le domaine de la vision par ordinateur, cela s'appelle l'inpainting et est discuté dans le chapitre suivant de la thèse.

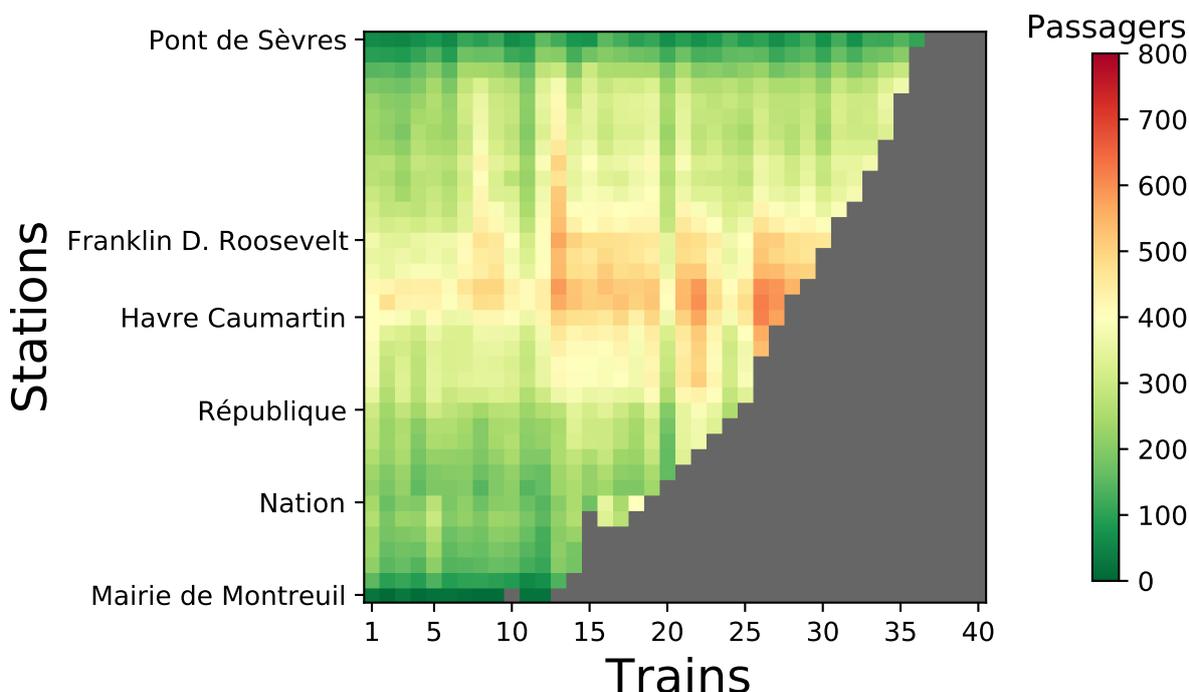


FIGURE 2.10 – Image temps réel $I^{in}(t)$ de la ligne 9 du métro. Les pixels gris indiquent les départs de trains futurs au moment t de sa génération. En noir les données manquantes de l’image.

2.6.1 Définition des masques

Pour aider à la prédiction et à la connaissance de l’image, il est possible d’utiliser un masque binaire pour coder plusieurs informations sur les données. La figure 2.8 en est un bon exemple et encode la fermeture de certaines stations.

Mais, la principale utilisation des masques concerne les prévisions. Toutes les images $I^{in}(t)$ combinent l’horizon passé, l’ensemble des valeurs connues $x_{s,c}$ à t , et l’horizon futur qui constitue la partie manquante de l’image. Pour coder ces informations liées à l’image observée, nous créons des masques équivalents à l’horizon futur m_y et à l’horizon passé m_x . Les masques sont établis en fonction de (2.8) et (2.9), respectivement.

$$m_x = m(x_{s,c}, t) = \begin{cases} 1 & \text{si } x_{s,c} \neq \emptyset; \\ 0 & \text{Sinon.} \end{cases} \quad (2.8)$$

$$m_p = m(x_{s,c}, t) = \begin{cases} 1 & \text{si } x_{s,c} \text{ est à prédire;} \\ 0 & \text{Sinon.} \end{cases} \quad (2.9)$$

Le masque de l’horizon passé peut être différent pour chaque variable de l’image. Il reflète la connaissance d’une variable à t , c’est-à-dire en particulier sa disponibilité au moment où l’image a été générée. La taille et le nombre de canaux de m_x sont égaux à l’image équivalente $I^{in}(t)$. Chaque canal de m_x se réfère à la disponibilité des variables d’entrée $x_{s,c}$. Cependant, m_x ne se concentre que sur les données passées, que les pixels identifiés aient ou non des valeurs manquantes.

L’horizon de prédiction n’est pas nécessairement égal à la totalité de la région manquante de l’image, comme le montre la figure 2.11. Il peut être limité à une région prédéfinie par un masque m_p basé sur l’horizon de prédiction h . La figure 2.11 présente un

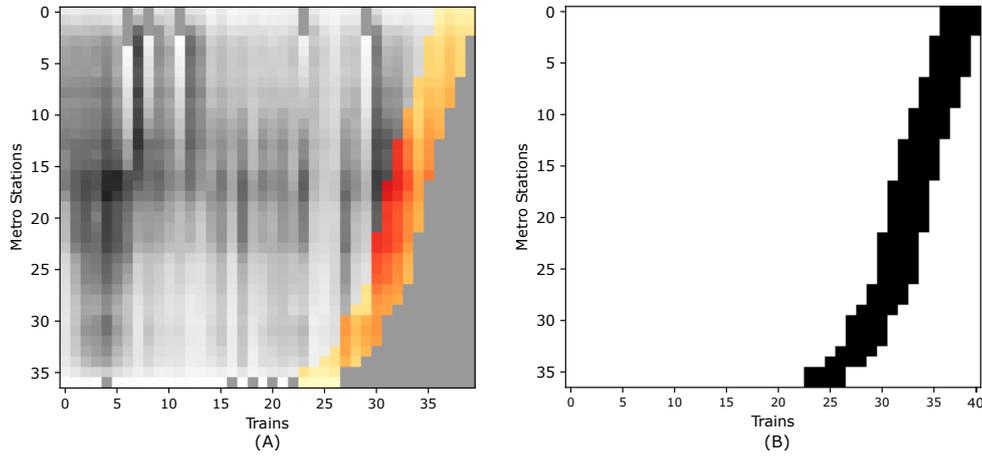


FIGURE 2.11 – À gauche : Une image avec l’horizon passé en gradient de gris avec la cible de prévision en couleur. À droite, le masque associé qui code la position de chaque pixel à prédire. Les pixels gris à droite de l’image correspondent aux pixels non prédits.

masque de prédiction utilisé dans l’expérimentation pour extraire les pixels de prévision (pixels en couleur) de $I^{in}(t)$. La zone de prévision sélectionnée se concentre uniquement sur les 4 prochains départs de chaque station ($h = 4$). Ainsi, m_p et h permettent de choisir et de se concentrer sur la période d’intérêt à court terme pour la prévision ; d’autre part, m_p réduit l’erreur de prévision en privilégiant les pixels de l’image pour la prévision à court terme, c’est-à-dire que les pixels de la charge passée sont négligés.

2.7 Utilisation des images en prédiction

La définition des images dans le temps révèle des zones manquantes qui dénotent tous les départs non encore réalisés dans l’image. La prévision d’une variable cible prend la forme d’une tâche d’inpainting qui consiste à compléter les pixels manquants dans l’image. Nous définirons ici le cadre permettant d’effectuer des prévisions à partir d’images. Les modèles ou les techniques de vision par ordinateur seront abordés dans le chapitre suivant.

2.7.1 Prédiction multiple à court terme par inpainting

L’inpainting est le nom d’une technique bien connue en vision par ordinateur qui vise à compléter la partie manquante d’une image à partir de son contexte (*i.e.* à partir de la partie qui elle est observée). Elle s’applique au trafic métro en temps réel $I^{in}(t)$ afin de compléter tous les départs de trains manquants. Cependant, la tâche d’inpainting ne prévoit pas les pixels ciblés pour l’opérateur. Elle applique une prédiction globale sur tous les pixels.

Notre cadre d’inpainting comprend deux étapes principales : (i) la reconstruction de la partie manquante de l’image $I^{in}(t)$ désignée par $\hat{I}(t)$ et (ii) l’extraction des futures données cibles de $\hat{I}(t)$ illustrée dans la figure 2.12. Dans la première étape, nous avons appliqué une fonction d’inpainting f à une image d’entrée multicanal (quatre canaux) afin de reconstruire les pixels de charge manquants. La deuxième étape vise à extraire les pixels pertinents pour la prédiction en appliquant un masque. En fonction de l’étendue de la prévision, la zone de prédiction varie. Par conséquent, nous générons un masque m_p par image de manière dynamique afin de sélectionner les pixels correspondants dans

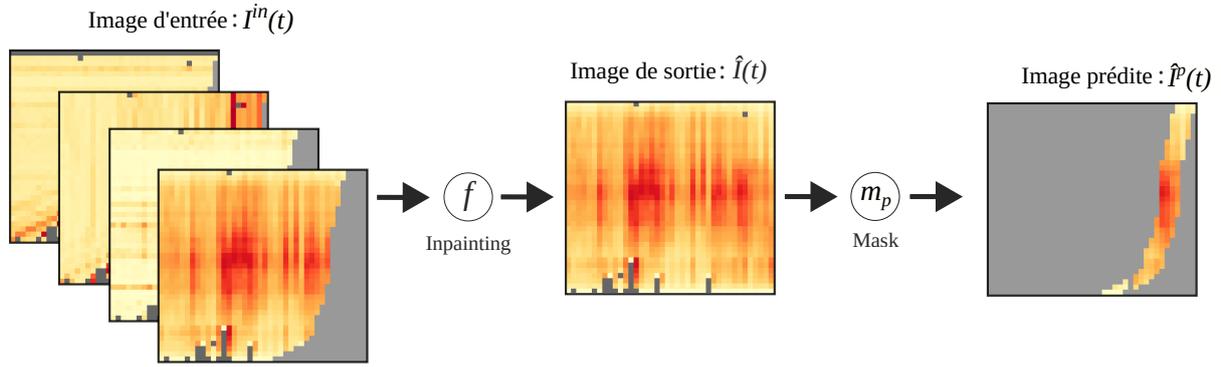


FIGURE 2.12 – Schéma de principe du fonctionnement de la tâche de prédiction multiple.

$I(t)$ et $hat{I}(t)$. L'idée est ici d'extraire la future variable cible désignée par $I^{in}(t)$ pour l'image d'entrée et $\hat{I}^p(t)$ pour l'image de sortie. Les équations suivantes résument les tâches réalisées pour obtenir l'image de prédiction :

$$\hat{I}_y(t) = f(I_x(t), \theta) \quad (2.10)$$

où f représente la fonction de prévision et son ensemble de paramètres θ , par un exemple un réseau de neurones profond. Les modèles doivent être construits en fonction de la tâche d'inpainting. L'image résultante $\hat{I}(t)$ est traitée par la deuxième équation :

$$\hat{I}_{pred}(t) = \hat{I}_y(t) \odot m_y \quad (2.11)$$

où \odot est le produit de Hadamard. Cette équation exclut les pixels non pertinents de l'image reconstruite, notez que ceux-ci varient en fonction du contexte temporel et spatial du réseau mais correspondent généralement à une diagonale de quatre pixels de large.

Cette proposition de traitement correspond à un traitement image à image qui définit la prévision d'une ligne de métro entière comme une tâche de vision par ordinateur. Cette proposition peut prendre en compte plusieurs contextes de transport (différents nombres de trains, nombre de pixels à prédire).

2.7.2 Images d'entrée et de sortie

Le cadre proposé est uniquement basé sur les images d'entrée et de sortie. L'entrée du modèle est toujours la même et comprend les images générées en fonction des données disponibles $I^{in}(t)$. Le choix principal concerne l'image de sortie qui dépend de l'objectif demandé par l'opérateur. L'image de sortie peut être limitée à une seule variable cible, mais elle peut également être multi-canal et contenir à la fois des informations sur la charge et les temps de parcours par exemple. Dans ce travail, seule la charge est utilisée comme cible, ce qui donne une image mono-canal. La prévision de plusieurs données peut être abordée en ayant une image multicanal. Le choix de la sortie est flexible en fonction des besoins de prévision.

Un autre point concerne la génération des images d'entrée et de sortie du modèle. Il est nécessaire de définir comment générer les images pour l'apprentissage des paramètres de la fonction d'inpainting. Deux stratégies sont possibles pour la génération d'images :

- Dans le premier cas, nous générons les images à intervalles réguliers. Une fréquence doit être définie (toutes les minutes, 30 secondes, etc...) qui permet d'observer les variations du réseau. La fréquence doit être suffisamment courte pour permettre

d’observer tous les mouvements de trains à l’intérieur du réseau. Elle assure la même quantité d’images par jour.

- La seconde stratégie utilise les données de trafic en générant une image pour chaque arrêt de train. C’est la méthode la plus précise pour générer des images. Toute l’évolution est observée en contrepartie d’un nombre total d’images générées par jour plus important. Pour le cas de la ligne 9 du métro un jour ouvrable et pour une direction donnée, cela représente une moyenne d’environ 10000 images par jour. A titre de comparaison, si l’on échantillonne une image par minute cela représente 1200 images par jour.

La première stratégie est à privilégier pour réduire les coûts de stockage et de ressources. Un nombre plus élevé d’images (10 fois dans ce cas) prend plus de temps à être traité pour les phases de génération et d’apprentissage.¹

2.7.3 Horizon de prédiction

Une image est donc composée d’une région passée connue et d’une seconde région à prédire. Cependant, la région réelle à prédire doit être définie en fonction des objectifs fournis par les opérateurs de transport. La prédiction de la charge peut prendre plusieurs formes en fonction du moment de la journée ou du nombre de pixels à prédire. Selon une vision à court ou à long terme, le nombre de pixels à prédire dans l’image varie. Une approche à court terme se limitera à quelques pixels par station alors qu’une approche à long terme s’intéressera à l’ensemble de l’image même si les départs des trains sont éloignés de plusieurs heures après l’acquisition de l’image. Dans le cadre de ces travaux, l’objectif est porté sur la prévisions à court terme en prédisant les quatre prochains départs de chaque point d’arrêt. Cette contrainte est faite selon des processus industriels propres à la RATP, lesquels sont liés au temps d’attente dans les points d’arrêt de la ligne de métro 9.

2.8 Conclusion

Ce chapitre a présenté la construction d’un format original de données en image pour représenter l’état d’une ligne de métro. Ce format permet d’encoder de manière triviale les différents trains et stations d’une ligne en tenant compte des choix opérationnels (stations fermées, départs intermédiaires). Un pixel définit un couple train-station et on y associe un vecteur stockant différentes variables du système pour ce couple train-station. On parle alors d’image multi-canal où un canal représente une donnée du système (temps de parcours, intervalle de départ).

La construction d’image en temps réel introduit une problématique d’inpainting afin de reconstruire les prochains départs de trains manquants lors de la génération de l’image.

Ces images peuvent être utilisées à diverses fins opérationnelles, pour la prédiction dans le cadre de ces travaux, mais elles peuvent également servir de représentation du trafic ferroviaire en temps réel afin d’offrir un historique riche avec la région des départ déjà opérés de l’image et une vision temps réel avec la région dégradée des futures départ de trains.

1. Pour information, la quantité totale d’images générées avec une période d’échantillonnage d’une minute est proche d’un million et nécessite 100 Go de stockage.

Chapitre 3

Inpainting et architectures profondes pour la prédiction de charge

L’Inpainting est un terme qui a été introduit pour la première fois par [8]. C’est une technique de traitement d’image qui vise à reconstruire, débruiter ou encore modifier des images. Il existe deux grandes familles d’algorithmes : (i) les modèles traditionnels se basant sur la diffusion [8] et une approche séquentielle du problème. Ces approches reconstruisent l’image pixel par pixel de manière séquentielle, (ii) les modèles à base d’apprentissage profond qui proposent une génération rapide ("fast generation") de l’image en la complétant directement en une seule passe.

Il peut paraître étrange d’avoir recours au traitement d’image dans une problématique de prédiction de séries temporelles, telle que celle qui nous occupe dans ce manuscrit, *i.e.* la prédiction de charges à bord dans les transports en commun. Cependant, nous avons vu, dans le chapitre précédent, que le format image est pertinent pour encoder les différentes séries temporelles décrivant une ligne de métro. De plus, ce format est usuellement employé dans de nombreuses modélisations ayant trait au transport car il permet de représenter spatialement les données ou encore de décomposer un espace urbain [76, 40]. Dans ce cadre particulier, l’image spatio-temporelle fusionne à la fois une ligne de métro et les déplacements des trains le long de la ligne, le but étant d’utiliser les spécificités des images en temps réel pour effectuer la prédiction.

Dans ce chapitre, nous présentons d’abord la méthodologie mise en place pour la prédiction de charges. Nous présentons ensuite les différentes architectures d’inpainting issues de la littérature pour finalement nous focaliser sur un sous-ensemble de modèles mis en place dans ces travaux ainsi que sur les modèles que nous considérerons pour la comparaison en termes de performances de prédiction.

3.1 Tâche d’Inpainting

Cette section introduit en détail la notion d’inpainting ainsi que les applications y étant associées. Nous ne serons pas exhaustifs sur les modèles et usages de l’inpainting. Nous introduisons seulement les grandes familles d’applications les plus récentes.

3.1.1 Applications de l’approche Inpainting

L’inpainting est un terme qui définit un ensemble de techniques de computer vision qui ont toutes pour but de reconstruire ou d’éditer une partie manquante d’une image. Pour



FIGURE 3.1 – Exemple d’inpainting dans une image. La région en blanc dans l’image représente la cible à combler par la tâche d’inpainting

la reconstruction d’image, on peut citer comme tâche courante la restauration d’images anciennes, la suppression des nuages d’images satellites... L’édition d’image quant à elle consiste à ajouter, supprimer ou remplacer des éléments de l’image, i.e. On parle alors d’occlusion ou de disocclusion.

La figure 3.1 illustre un exemple de tâche d’inpainting. L’objectif de cet exemple est la complétion de la partie du bus manquante dans l’image. On peut utiliser le même procédé pour retirer des objets de l’image, auquel cas on parle de disocclusion.

Comme nous venons de le dire, l’inpainting est une technique de computer vision utilisée pour la reconstruction ou l’édition d’image. A partir d’une image source qui présente à la fois des régions connues et des portions dégradées (départs de trains non observés ou futures), l’objectif est de proposer une solution qui est perçue comme plausible d’un point de vue humain. Le problème est souvent formaliser de la manière suivante soit une image $I : \mathbb{R}^n \rightarrow \mathbb{R}^m$ de taille (n, m) constituée de deux régions : une région source S et une région manquante U (la région en blanc dans l’image 3.1), avec $I = S \cup U$.

L’inpainting consiste alors à reconstruire U à partir des données issue de S . Les méthodologies peuvent s’aider d’information liée aux textures ou à la structure de l’image, i.e. les géométries de l’image par exemple.

3.1.2 Adaptation à notre contexte de prédiction

Notre problème de prédiction se distingue légèrement de cette tâche classique car certains pixels (ceux correspondant aux 4 prochains départs de trains dans toutes les stations) nous intéressent plus particulièrement ; *i.e* dans notre cas les pixels à prédire sont un sous ensemble de U suivant $\hat{I}_{pred}(t) \subseteq U$.

A partir de l’image initiale I , on définit la fonction f qui reconstruit la partie dégradée de l’image U par :

$$\hat{I} = f(I, \theta) \quad (3.1)$$

où θ représente les paramètres de l’hypothèse choisie quant à l’inpainting et \hat{I} représente l’image prédite dont la partie U a été complétée. L’objectif est de générer une image \hat{I} qui soit la plus proche possible de la réalité terrain ("ground truth").

Dans le cadre de la prédiction de charges par inpainting, la fonction f ne représente que la première étape du protocole. En effet, l’inpainting a pour mission de reconstruire l’image dans sa globalité mais dans la mesure où nous visons la prédiction des pixels de charges correspondant aux 4 prochains départs, nous utiliserons des masques de prédiction m introduits précédemment afin de focaliser la prédiction sur cette partie de l’image qui nous intéresse particulièrement d’un point de vue applicatif.

Par conséquent, l'apprentissage des paramètres θ de la fonction d'inpainting f doit tenir compte à la fois des pixels à prédire et de la tâche d'inpainting à proprement parlé. Pour prendre en considération ces deux aspects, nous avons utilisé dans toutes nos approches une fonction de perte correspondant à la somme pondérée de l'erreur de \hat{I} et $\hat{I}_{pred}(t)$.

$$Loss = \alpha Error(I, \hat{I}) + \beta Error(I_{pred}(t), \hat{I}_{pred}(t)) \quad (3.2)$$

où α et β sont des hyper-paramètres qui permettent de favoriser l'une des sorties par rapport à l'autre. On souhaite différencier la tâche d'inpainting et celle de prédiction à l'instar du compromis exploration-exploitation. Pour les modèles à sortie unique, le paramètre α est de 0 car l'image totale \hat{I} n'est pas prise en compte. In-fine ces deux paramètres nous permettent de pénaliser et de faire un compromis entre deux sorties.

L'apprentissage des paramètres θ de la fonction f revient à minimiser l'erreur introduite dans l'équation 3.2.

3.1.3 Les tâches de type image vers image

En traitement d'image, les structures des algorithmes varient selon les tâches et les objectifs. Dans le cadre de l'inpainting, on parle de traitement d'image à image car l'image d'entrée est similaire à l'image en sortie. Ceci n'est pas le cas lorsqu'il s'agit de classification d'images car l'image n'est utilisée qu'en entrée des modèles.

Nous travaillerons uniquement sur des approches images à images. Il est nécessaire toutefois de mentionner que dans le cas de nos prédictions de charge, les entrées et sorties des algorithmes diffèrent car d'un point de vue opérationnel, seule la valeur de charge nous intéresse. Le nombre de canaux de l'image de sorties ne sera donc pas forcément égal à celui de l'image d'entrée, cette différence ne pose pas de problème particulier et les architectures que nous allons présenter peuvent très bien traiter ce genre de cas (elles sont par exemple utilisées en colorisation d'image ou le nombre de canaux de l'image de sorties est plus élevé que celui de l'image d'entrée).

3.2 Résolution par diffusion

Cette section traite des méthodologies traditionnelles d'inpainting basées sur la diffusion. La première notion de l'inpainting est apparue avec le principe de diffusion. L'objectif est de compléter les régions manquantes d'une image en propageant l'information connue de celle-ci de manière itérative pixel par pixel. On dénote plusieurs approches qui tiennent compte des caractéristiques a priori connues de l'image.

L'inpainting par diffusion [8] est une technique de traitement d'images qui consiste à remplir de manière itérative la région dégradée de l'image en tenant compte de ses propriétés structurales et texturales. La méthode consiste à propager l'information de la région source S vers les pixels dégradés à partir de la frontière vers le centre des parties manquantes.

À l'instar de la mécanique de la propagation de chaleur, la diffusion définit un système d'équations différentielles partielles (PDE) pour modéliser la propagation de l'information dans la région manquante de l'image :

$$\begin{cases} I_0 = S \cup U \\ \frac{\partial I}{\partial n} = \Delta I \end{cases} \quad (3.3)$$

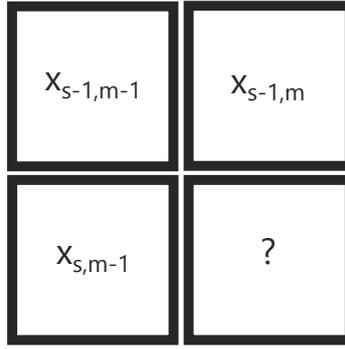


FIGURE 3.2 – Schéma de la position des pixels dans le cadre d'une diffusion

ou I_0 est l'image d'entrée avec des régions manquantes, ΔI le Laplacien de l'image, n représente l'itération de la diffusion. Cette approche introduit des effets de flou pour des régions dégradées larges et principalement sur les contours des objets et les bords de l'image.

Plusieurs approches introduisent un coefficient de diffusion pour orienter la propagation de l'information en tenant compte de la structure de l'image. Ce coefficient nommé g est un scalaire compris entre $[0, 1]$ renseignant sur les contours de l'image agissant soit comme un isolant ($g = 0$) dans le cas d'un bord, soit comme un conducteur dans le cas de pixels similaires d'un même élément de l'objet (aplat de couleur).

L'équation 3.3 est mise à jour en ajoutant un coefficient calculé à partir du gradient de l'image afin de détecter les différents bords de l'image :

$$\frac{\partial I}{\partial n} = \text{div}(g(|\nabla I|^2)\nabla I) \quad (3.4)$$

où div est l'opérateur de divergence et ∇ est le gradient. La reconstruction de l'image est faite de manière progressive jusqu'à l'obtention d'une solution possible de la tâche d'inpainting. Cette approche a fait ces preuves pour les tâches de débruitage d'image ou d'inpainting sur de petites surfaces. Elle souffre toutefois de plusieurs défauts comme l'apparition de flou, de texture ou encore de bordures non connectées. On note que de nombreuses évolutions ont été mises au point pour contourner ces différentes faiblesses comme la décomposition texture/structures [10, 78] ou le matching de patches [61, 15]. En résumé, les méthodes par diffusion ont représenté les premières techniques d'inpainting avant l'avènement des architectures de type deep learning. Elles proposaient une approche itérative de l'inpainting, et ont conduit à de bonnes performances pour les tâches de débruitage sur de petites surfaces.

3.2.1 Modèle naïf

Une approche naïve de la prédiction de charge sous la forme d'image reprend l'idée de diffusion afin de compléter pixel par pixel les départs de trains manquants. En tenant compte de la structure des données au sein de l'image, le modèle dit "naïf" effectue une opération locale qui a pour but de mettre à jour la charge du train en tenant compte de la tendance de la station considérée.

Nous pouvons définir la règle naïve comme suit : $y_{s,c} = x_{s-1,c} + \Delta x_{s,c-1}$ où $y_{s,c}$ représente la valeur prévue, $x_{s-1,c}$ désigne l'état actuel (nombre de passagers ou distance) du train à la dernière gare, $\Delta x_{s,c-1}$ désigne la variation de la valeur cible du dernier train pour

la gare actuelle par rapport à la précédente et c et s représente respectivement les trains et les gares. Cette approche sera utiliser comme base-line pour la prédiction de charge.

3.3 Approches à base d'apprentissage profond

La seconde famille de méthodes utilisée pour l'inpainting est basée sur l'apprentissage profond. Le deep learning a en effet connu un large essor ces dernière années avec principalement les modèles de convolution qui ont montré de bonnes performances dans de nombreuses tâches de traitement d'image telle que la classification, le clustering, la super-résolution, ou encore la détection d'objet. On note en particulier les modèles de réseaux de neurones convolutionnels qui permettent de traiter de manière efficace et pertinente les images et les vidéos.

3.3.1 Réseau de neurones

Un réseau de neurones est une architecture de machine learning qui imite le fonctionnement des neurones humains. Il est composé de plusieurs couches de neurones interconnectées qui traitent et transforment des données d'entrée pour produire des résultats en sortie. On peut également le résumer en une couche d'entrée, au minimum d'une ou plusieurs couches de neurones cachés, et une couche de sortie. Les couches cachées sont appelées ainsi car elles ne sont pas directement connectées aux entrées ou aux sorties, mais à d'autres couches de neurones, permettant ainsi au réseau de traiter des informations complexes et d'en construire une représentation non linéaire. Ces modèles peuvent être construite afin de réaliser des tâches de classification, de prédiction ou de reconnaissance de formes. Un neurone est une unité fondamentale d'un réseau de neurones qui prend en entrée un vecteur (x_1, x_2, \dots, x_n) et les transforme en une sortie y à l'aide d'une fonction d'activation a et d'un ensemble de poids, permettant de combiner les différentes entrées, comme cela est résumé dans la Figure 3.3 et décrit par l'équation 3.5. Les sorties y de chaque neurones servent alors de données d'entrée pour les neurones de la couche de neurone suivante jusqu'à la couche de sortie du modèle. Pour permettre au réseau de disposer de plusieurs représentations d'un même ensemble de données en entrée, chaque neurone est associé à un ensemble de poids $(w_0, w_1, w_2, \dots, w_n)$, où w_0 représente le biais du neurone.

$$y = a(z) = \text{where } z = \sum_{i=1}^n w_i x_i + w_o \quad (3.5)$$

Ces poids sont ajustés lors de l'apprentissage du réseau afin de minimiser l'erreur de prédiction ou de classification et d'améliorer les performances globales du réseau. L'as-

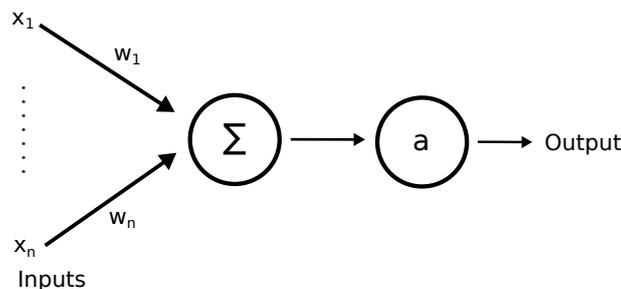


FIGURE 3.3

sociation d'un nombre important de neurones permet de construire une représentation général de problème complexes pour de nombreux domaines d'application. Cependant dans le cadre de traitement d'image, les réseau de neurones ne tiennent pas compte de la position dans l'image des pixels et de leur voisinage.

La fonction d'activation calcule la sortie du modèle et peut être perçue comme un switch entre éteint et allumé. Elle est pour la plupart du temps non linéaire et différentiable pour rendre possible l'usage de la descente de gradient lors de l'apprentissage. La fonction d'activation est à choisir selon les problématiques et données des expérimentations. En effet, certaines fonctions d'activation seront plus adéquates pour la régression (Rectified Linear Unit ou ReLU, Sofplus) alors que d'autres seront dédiées à des tâches de classification (Sigmoid, Softmax). La principale différence entre ces différentes fonctions d'activation est la plage de valeurs prises par ces fonctions en sortie. Par exemple, les fonctions Sigmoid et SoftMax ont une range $[0, 1]$ alors que les fonctions ReLU et Sofplus sont définies pour $[0, \infty[$. Dans un cas, les fonctions permettent d'estimer une probabilité d'appartenance à une classe et dans un second cas, elles sont équivalentes à une régression pour une variable définie sur \mathbb{R}^+ . Pour titre d'exemple, l'équation 3.6 définit la fonction ReLU qui est la plus couramment utilisée dans les architectures de deep learning pour réduire l'effet de vanishing gradient et son implémentation et calcul efficace .

$$a = Relu(z) = max(0, z) = max(0, \sum_{i=1}^n w_i x_i + w_o) \quad (3.6)$$

Il est important de noter que les modèles peuvent avoir différentes fonctions d'activation entre les différentes couches du réseau. Ceci est principalement mis en oeuvre pour les neurones de sortie de modèles qui présentent une fonction d'activation adaptée à la tâche à résoudre.

3.3.1.1 Rétro-propagation du gradient

Dans un cadre supervisé, l'apprentissage d'un réseau de neurones est réalisé en ajustant les poids du modèles selon un jeu de données qui présente pour chaque entrée la sortie attendue. Ce procédé repose sur la descente de gradient. Dans un premier temps, on calcul la sortie du modèle pour une instance de données. On calcule l'erreur selon la sortie attendue puis l'erreur est propagée dans le réseau à rebours pour pouvoir mettre à jour les poids de chacun des neurones. Ainsi, à chaque itération, les poids du réseau sont mis à jour selon le gradient de l'erreur du neurone selon l'équation :

$$w = w - \eta \frac{\partial E}{\partial w} \quad (3.7)$$

où E est l'erreur du réseau et η le taux d'apprentissage (learning rate). Le learning rate permet de contrôler l'évolution des paramètres lors de l'apprentissage. Pour mettre à jour chaque neurone du modèle, l'erreur est propagée à partir de la couche de sortie et repose sur l'usage des dérivés des fonctions d'activation des modèles d'où l'importance qu'elle soit différentiable. De plus, le choix de fonction dont les dérivés ont des coûts de calcul faibles comme par exemple la ReLU (cf equation 3.8) permet de simplifier le calcul des gradients. On note cependant qu'en 0, la dérivée de la ReLU n'est pas définie. Une valeur entre 0 et 1 est choisie arbitrairement à 0.

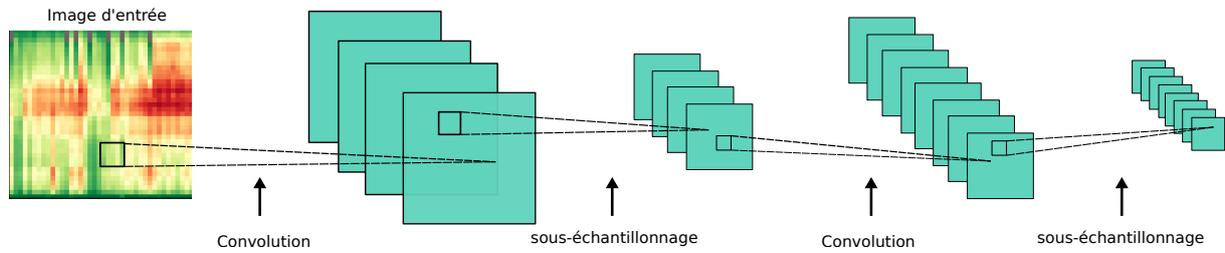


FIGURE 3.4 – Schéma de principe d'une architecture type de CNN

$$a' = \begin{cases} 0 & \text{si } x < 0 \\ 1 & \text{si } x > 1 \\ \text{nondéfini} & \text{si } x = 0 \end{cases} \quad (3.8)$$

La retropropagation (back propagation) est de manière courante couplée à un algorithme d'optimisation de type gradient stochastique, les données étant présentées par petit groupe appelé *bacth* et les paramètres sont mis à jour après chacun d'entre eux. Ces algorithmes de type gradient stochastique tel que l'algorithme Adam (Adaptive Moment estimation)[43] couramment utilisés sont très efficaces d'un point de vue complexité et permettent de traiter des volumes très importants de données.

3.3.2 Les modèles convolutionnels

Le traitement d'une image par réseaux de neurones (ANNs) peut s'avérer coûteux car chaque pixel est pris en compte comme entrée du modèle pouvant ainsi augmenter de manière significative les paramètres des modèles. De plus, les ANNs ne tiennent pas compte de la structure et de la position des pixels dans l'image. Afin de traiter l'image non pas comme un vecteur uni-dimensionnel, les architectures de traitement d'image reposent sur le principe de convolution, on parle alors de réseau de neurones convolutif, ou Convolutional Neural Network (CNN). Ces derniers font usage de filtres (kernel) sur des patches de l'image, c'est à dire un pixel et son voisinage afin d'extraire des informations localisées dans l'image. Dans le cadre des convolution, le filtre est appliqué à l'ensemble de l'image afin d'obtenir une feature map de l'image. La figure 3.4 présente un schéma de convolution 3x3. Les neurones d'un ANN sont alors remplacés par les filtres, l'idée étant que chaque filtre va extraire une information spécifique de l'image. A titre d'exemple, des filtres peuvent par exemple permettent d'extraire des structures horizontale ou verticale d'une image. Les architectures de type CNN peuvent utiliser des filtres de différentes tailles (3x3, 5x5) selon les dimensions des images. Dans la plupart des cas, le filtre se présente sous la forme d'un carré autour d'un pixel cible mais il est possible de proposer des filtres spécifiques. Nous pouvons citer les filtres dilatés [83] dont la convolution est faite non pas sur les pixels adjacents au pixel cible mais sur des pixels espacés dans son voisinage. Les CNN se composent à l'instar des modèles ANN de couches consécutives de convolutions souvent accompagnées de couches de réduction de dimensions (pooling) afin d'extraire des features d'image à différentes échelles de l'image. Les structures de CNN sont de plus construites selon la tâches à accomplir. Initialement les CNN sont utilisés comme encodeur des informations visuelles de l'image et sont accompagnés de structure ANN pour les sorties de modèles comme dans le cadre de classification d'image. Dans le cadre d'inpainting, les structures doivent répondre à une problématique d'image à image où les images sont à la fois une entrée et une sortie du modèle. Pour cela, les modèles

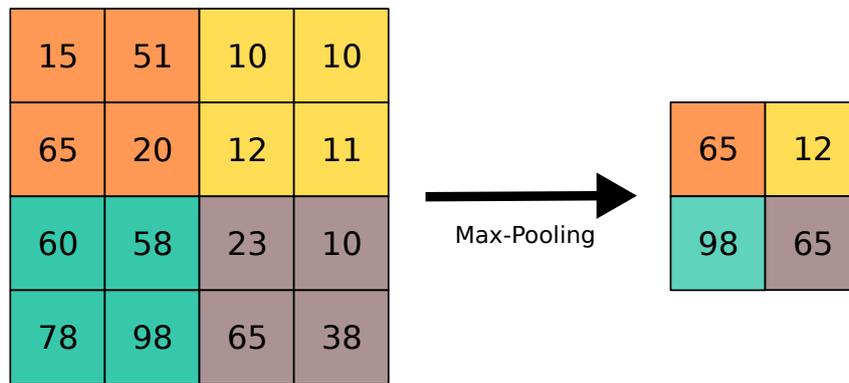


FIGURE 3.5 – Schéma de principe du fonctionnement d’une couche de maximum pooling de taille 2.

convolutions ont évolué en utilisant l’opération inverse des convolution (Transpose convolution) afin de reconstruire une image à partir de la représentation de l’image par le modèle.

3.3.2.1 Pooling

Une image est par définition un format de données de grande taille qui peut constituer un frein à l’apprentissage de modèles et l’extraction de features pertinentes. Afin de réduire la dimension des images, les réseaux utilisés en traitement d’image mobilisent très souvent la technique du pooling qui correspond à un sous échantillonnage localisé de l’image. Le pooling consiste à agréger les différentes valeurs d’un patch de l’image (grâce a un opérateur d’agrégation généralement l’opérateur max ou l’opérateur de moyenne), afin à de réduire la dimension de l’image (selon la taille des patches choisie : on parle en majorité de pooling 2x2 ou 3x3). On parle alors de Max Pooling, Average Pooling pour préciser l’opérateur d’agrégation choisi lors du pooling.

Le pooling permet ainsi de réduire la dimension des images et features maps des modèles permettant de proposer un temps d’apprentissage plus faible et un coût en ressource moindre. Il existe cependant d’autres méthodes de traitement d’image réalisant la même tâche comme le stride [4] qui consiste à réaliser les convolutions sur un nombre réduit de pixels. Par exemple un stride 2 réalise une convolution tout les deux pixels ce qui revient à réduire la dimension de l’image par 2.

3.3.2.2 Padding

En traitement d’image, il est courant que les données d’entrées aient des tailles différentes. Pour les images de transport comme introduits dans ces travaux, des images issues de lignes différentes en sont un parfait exemple. De plus, en bordure d’image, il est impossible de réaliser les convolutions car une partie des pixels n’est pas définie.

Pour y remédier, le padding consiste en l’ajout de pixels supplémentaires en bordure de l’image afin de pouvoir réaliser la convolution dans les mêmes dispositions que dans le reste de l’image. Cela permet de conserver les mêmes dimensions spatiales entre les différentes entrées. La méthode dite du Zero padding consiste par exemple à compléter l’image de pixels nuls. Mais il est possible d’étendre l’image à la place de 0. Le padding peut être également utilisé dans d’autres domaines comme les séries temporelles afin d’avoir une taille constante de séries à traiter.

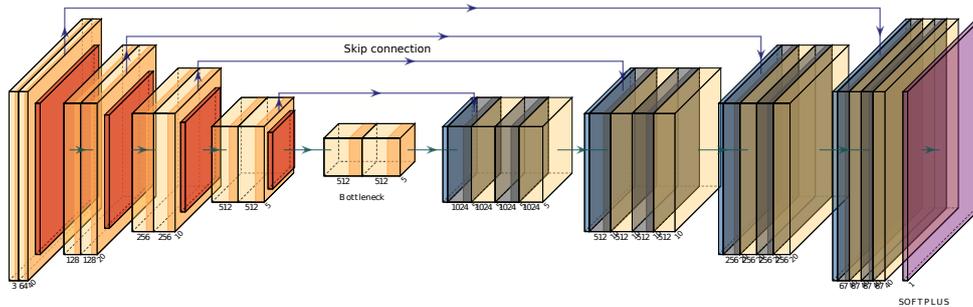


FIGURE 3.6 – Schéma de principe d'une architecture de type U-net.[6]

Suite à la présentation des techniques classiques de traitement d'image que l'on retrouve dans les architectures de type réseau de neurones profonds, nous allons détailler les principales approches et architectures de traitement d'image et d'inpainting.

3.3.3 U-net

L'architecture U-net (figure 3.6) a été initialement introduite dans les travaux de segmentation d'images par [66]. Ce modèle est un réseau neuronal symétrique composé d'une étape de convolution (côté gauche de la figure 3.6) et d'une voie d'expansion (côté droit de la figure 3.6) pour la reconstruction d'image. La première phase du modèle est constituée de couches répétées de convolution et de pooling. La seconde phase du modèle est constituée de couches de convolution inverse (up-convolution) et de sur-échantillonnage/augmentation (up-sampling). Cette architecture permet d'extraire des caractéristiques pertinentes à différentes échelles qui peuvent être réparties dans la partie du modèle consacrée à la génération d'images. En fait, chaque "up-convolution"[66] est liée à la convolution équivalente dans les premières couches du U-net. Ce mécanisme est appelé "skip connections" et permet de propager les caractéristiques des différentes convolutions et entrées à leurs étapes équivalentes de reconstruction de l'image. L'une des raisons de l'utilisation de ces skip connections réside dans ces chemins directs qui permettent une propagation plus directe du gradient et atténuent le phénomène de "gradient qui s'évanouit (vanishing gradient)" observé dans les méthodes d'apprentissage profond récurrentes. En outre, dans notre cas (inpainting), étant donné qu'une grande partie de la sortie est déjà connue, ces chemins sont tout à fait naturels. La figure 3.6 montre que les couches consécutives sont conçues de manière symétrique afin que la taille de l'entrée et de la sortie finale restent identiques.

3.3.4 Modèle génératif adversarial

Le modèle génératif adversarial (ou GAN, de l'anglais Generative Adversarial Network)[31] est une architecture de réseau de neurones artificiels utilisée pour générer des données synthétiques qui ressemblent à des données réelles. Le modèle est composé de deux réseaux neuronaux qui s'affrontent dans un jeu de compétition : un générateur et un discriminateur notée respectivement G et D . Dans le cadre de cette architecture, le discriminateur a pour but de classifier les images selon leurs origines *i.e.* factice ou réelle alors que le générateur a pour but de tromper le discriminateur, les images générées étant fausses. Ce type de modèle repose sur une approche pseudo-supervisée car le générateur n'est pas entraîné directement à partir des images mais via l'erreur du discriminateur.

$$\min_G \max_D \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [1 - \log D(G(z))] \quad (3.9)$$

L'apprentissage des GAN est fait selon sur le principe de policier et du voleur où l'on cherche à la fois à maximiser les performances du discriminateur D tout en minimisant l'erreur obtenue via les images créées par le générateur G . Pour résumer, l'équation 3.9 présente la fonction de coût d'un GAN. $D(x)$ représente la sortie du discriminateur pour les images réels. $D(G(z))$ dénote la sortie du discriminateur pour les images générées. z représente un vecteur d'entrée aléatoire tiré au hasard généralement selon une distribution probabiliste. Le but est alors de minimiser les performances de $D(G(z))$ tout en maximisant $D(x)$.

Du point de vu de l'inpainting, les GANs proposent un type d'apprentissage pertinent car ces modèles peuvent fournir plusieurs solutions pour une même image d'entrée. Il est alors possible de choisir parmi plusieurs solutions l'image la plus pertinente. De plus, l'usage comme générateur de structure établies comme les U-net [84, 82] ont été utiliser dans le cadre d'approches d'image à image.

3.3.5 Les modèles de type transformers

Ces dernières années, le deep learning a connu une évolution majeure avec l'essor d'une nouvelle architecture nommée transformer [72], basée sur les mécanismes d'attention. D'un point de vue informatique, les mécanismes d'attention tentent de mettre en valeur les relations entre les pixels de l'image ou éléments d'une même séquence. En outre, ils peuvent gérer les dépendances temporelles sans architecture récursive. Les architectures de type Transformer ont d'abord été utilisées à l'origine dans le traitement du langage naturel (NLP) pour cartographier les dépendances entre les mots d'une phrase ou d'un texte sous la forme d'un modèle encodeur-décodeur [72]. De plus, cette structure est devenue l'état de l'art par une structure séquence à séquence plus rapide à entraîner. Elle s'oppose au modèle récursif (RNN, LSTM, GRU) qui était alors privilégié pour le traitement de séries ou de séquences (Traitement de texte) mais imposant de long temps d'apprentissage.

En computer vision, l'approche par attention est rendue possible en construisant une séquence de mots visuels appelée patch (où un mot visuel est un groupe de pixels voisins). Une des premières implémentations est Le Vision Transformer (ViT) [22] qui utilise ce principe de phrase de patch et d'encodeur transformer à la place de modèles convolutifs dans un cadre de classification d'image. Le transformer permet de construire diverses représentations de dépendance de tous les pixels indépendamment de leurs coordonnées. Les transformers sont désormais considérés comme l'état de l'art dans de nombreuses tâches de vision par ordinateur, telles que la classification [85, 41, 52], la segmentation [27, 11], la détection d'objets [7, 12] et la génération d'images [49].

Suivant [72], un transformer encodeur, figure 3.7, est composé de trois éléments principaux : un encodage positionnel des pixels pour propager l'information basée sur la coordonnée d'un patch ou d'un pixel, une couche multi-têtes pour appliquer des mécanismes d'attention et un perceptron multi-couche (MLP).

L'encodage positionnel est utilisé afin d'indiquer l'ordre initial des éléments de la séquence ou des patches dans le cas des images. Au contraire des architectures récursives ou convolutives qui incorporent naturellement la position des éléments d'une séquence, il est nécessaire de transmettre cette information au modèle de type transformer qui n'y a pas accès naturellement. Cette information de position est transmise tout le long du modèle par l'usage de skip connections entre les composants du transformer qui se

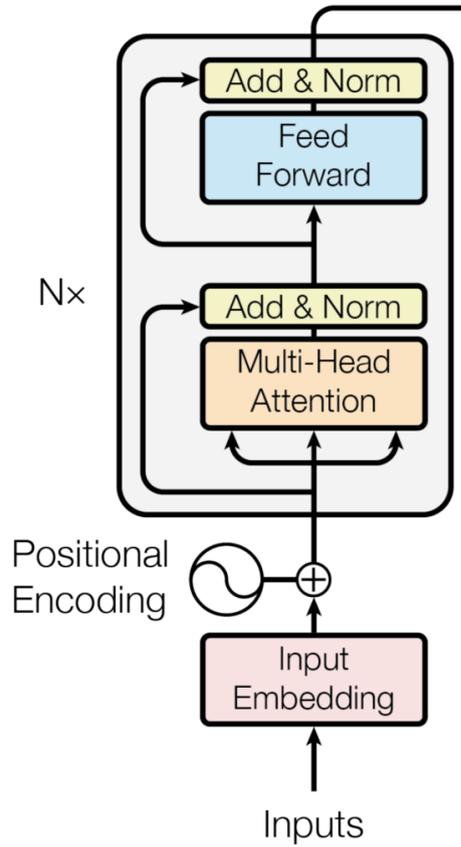


FIGURE 3.7 – Transformer Encoder [72]

traduisent par les couches "add" dans la figure 3.7. Dans le cadre du traitement d'image, un encodage 1D $((1, 2, 3, \dots, N))$ est utilisé.

Le cœur du transformer est la couche d'attention multi-têtes qui met en oeuvre la self-attention celle-ci est décrite par l'équation (3.10) suivante :

$$Attention(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (3.10)$$

où d_k représente la dimension de projection du transformer, $Q \in \mathbb{R}^{n \times d_k}$ est le vecteur "query", $K \in \mathbb{R}^{n \times d_k}$ est le vecteur "key", et $V \in \mathbb{R}^{n \times d_k}$ est le vecteur de valeur (Value). L'objectif d'un mécanisme d'attention est de calculer la similarité entre le vecteur "query" Q et le vecteur K . Ensuite, les caractéristiques des pixels passent par le vecteur de valeurs V . Les vecteurs Q , K et V sont calculés à partir des patches de pixels de l'image $I(t)$ au moyen de matrices de projection désignées par W^Q , W^K et W^V . On parle alors de selfs attention car les vecteurs Q , K et V sont issue de la même séquences d'entrée.

A l'instar des filtres pour un modèle de convolution et afin d'extraire plusieurs représentations de l'image, un transformer multiplie les représentations d'attention des images d'entrée. On parle alors d'une couche multi-tête où chacune des têtes *head* est dédiée à une représentation de l'image. Chaque tête $head_i$ a alors son propre set de paramètres W_i^Q , W_i^K et W_i^V 3.11.

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \quad (3.11)$$

Le multi-head réfère à la concaténations des têtes du modèle 3.12.

$$MultiHead(Q, K, V) = Concat(head_1, \dots, head_h)W^O \quad (3.12)$$

Ainsi, un transformer encodeur construit pour une même image un ensemble de représentations à l’instar des features d’un modèle convolutif. Cependant, les features de l’image ne sont pas estimées localement comme les filtres d’un CNN mais sur la globalité de l’image donnant à la fois une représentation locale et globale de l’image car, pour rappel, la self-attention est calculée entre tous les patches de l’image.

Enfin, concernant les transformers, il est important de préciser l’impact computationnel de ce type d’architecture [68]. L’architecture a en effet été initialement introduite, en partie, pour se substituer aux architectures récursives courantes en NLP et réduire ainsi le temps d’apprentissage (des modèles récurrents pouvaient nécessiter plusieurs semaines d’entraînement avec des structures de LSTM en NLP). L’approche basée sur les transformers travaillant directement d’une séquence à une autre (séquence à séquence), les traitements peuvent en partie être parallélisés *i.e.* le coût en temps d’apprentissage est transposé en un coût en ressources informatiques (unité de traitement graphique, GPU), ce qui permet de réduire le temps d’apprentissage si l’on dispose des ressources nécessaires. Ceci qui peut être très intéressant, mais le calcul des scores d’attention étant quadratique par rapport à la taille de la séquence d’entrée, cela induit a contrario une augmentation de la quantité de ressources mémoire devant être mobilisée. Il existe néanmoins des approches tentant de réduire le coût de l’attention comme l’attention restreinte ou sparse [19] où le but est de ne pas calculer l’attention entre tous les éléments de la séquence d’entrée. La question du coût en ressources des modèles d’attention sera discutée dans le prochain chapitre lors de la présentation de l’expérimentation, cette question a son importance lors de la mise en production d’un tel modèle.

3.3.5.1 U-Transformer

Ces travaux de thèse mettent en oeuvre deux architectures basées sur les transformers mais adaptées à la tâche d’inpaiting. Le premier, le U-transformer est une évolution du U-net afin d’intégrer les mécanismes d’attention dans la fonction d’inpaiting f . Initialement le U-net extrait localement les features via la convolution selon différentes échelles (chaque couche de convolution). Le but est d’ajouter au U-net une compréhension globale de l’image via l’attention tout en gardant cette approche. Dans le cas de notre image de trafic ferroviaire, l’image incorpore des composantes spatiales et temporelles. L’ajout des transformers permet de tenir compte des corrélations long terme dans l’image (pixels éloignés) qui n’étaient pas initialement permises par la convolution. Ainsi, on prend en compte les informations locales et globales pour chaque départ de train.

L’architecture du U-transformer est basée sur le Swin-Unet proposé par [11] pour la segmentation d’images. L’architecture proposée est une variante d’un U-net standard, où les blocs encodeur transformer sont utilisés à la place des couches convolutives du modèle U-net. Dans le cas du modèle Swin-Unet, les transformers sont conçus pour extraire des représentations hiérarchiques des caractéristiques pertinentes dans le cas de segmentation d’image. Le bloc Swin se compose de deux sous-blocs, un multi-head fenêtré et un swin multi-head. Le multi-head applique la self-attention à une portion de l’image définie par une fenêtre de sélection de patch. Le bloc Swin se compose de deux sous-blocs, un multi-head fenêtré et un multi-head fenêtrés décalées, ce qui augmente les ressources informatiques nécessaires à l’apprentissage, même si le mécanisme de self-attention est limité à de petites fenêtres.

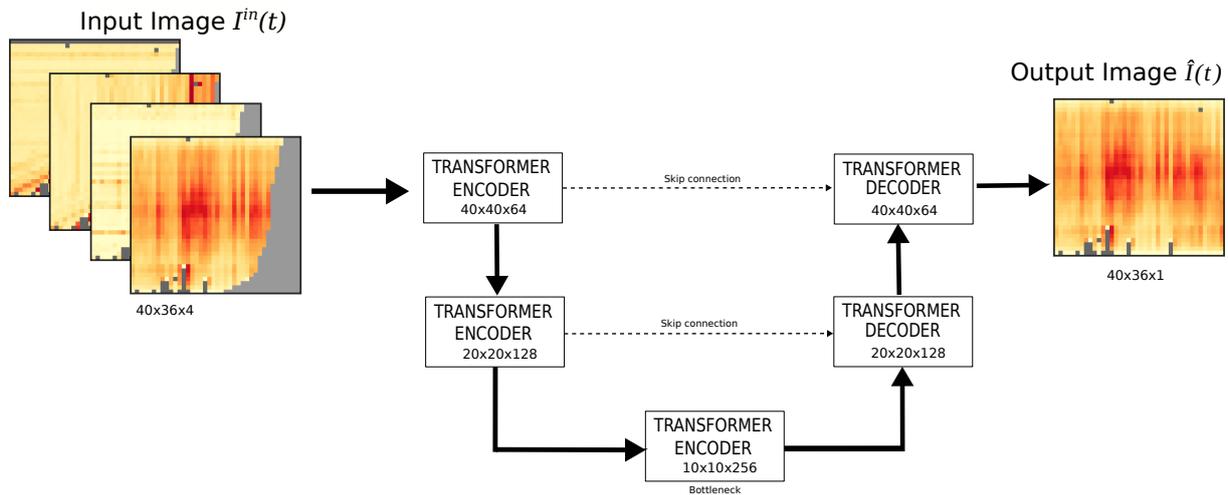


FIGURE 3.8 – Schéma de principe de l'architecture de type U-Transformer.[5]

Dans le modèle que nous proposons, nous avons conservé la structure globale du réseau et l'approche de la fusion et de l'extension des patches, mais nous avons mis en œuvre ces caractéristiques en utilisant des encodeurs de type transformers standards au lieu des blocs Swin. Le processus de fusion de patches implique la concaténation de patches suivie d'une couche linéaire pour réduire la dimension de l'image et augmenter la dimension de la caractéristique d_k d'un facteur 2. Pour la reconstruction de l'image, l'opération inverse est effectuée, augmentant la dimension de l'image tout en réduisant la dimension de la caractéristique d_k d'un facteur 2. Cette modification nous permet de conserver les avantages de l'architecture du réseau en U tout en remplaçant les couches convolutives par des transformers.

Il convient de mentionner que d'autres stratégies de fusion entre les transformers et des modèles U-net ont été élaborées, comme l'approche proposée par [16], dans laquelle un mécanisme d'attention est appliqué au niveau du bottleneck du modèle afin d'améliorer les caractéristiques visuelles, ou l'approche présentée par [64], qui utilise un mécanisme d'attention uniquement dans le décodeur du U-net.

3.3.5.2 Channel Vision Transformer

Les images d'une ligne de métro comportent à la fois des variables relatives à l'offre de transport, aux temps de parcours, temps d'attente à quai¹ et également à la demande soit en stations avec les validations télé-billettique ou à bord des rames avec la charge passagers. Le précédent modèle applique l'attention sans faire de distinction entre les différentes variables d'entrée. Il serait intéressant de pouvoir observer pour chaque variable ou indicateur soit de l'offre ou de la demande, les scores d'attention et de noter l'impact de chaque variable dans la prédiction. On pourrait ainsi découpler les variables observées et mettre en évidence quelle information est apportée par chacune des variables à la tâche de prédiction.

Pour investiguer cette piste, nous avons construit un modèle noté Channel Vision Transformer (Channel ViT) selon l'idée des méthodologies par canal [69, 17]. L'encodage des images est décomposé en une série d'encodeurs pour chaque canal, cette opération est suivie par un décodeur devant reconstruire l'image complète. Chaque transformer capture

1. Il s'agit de l'offre réalisée et non théorique

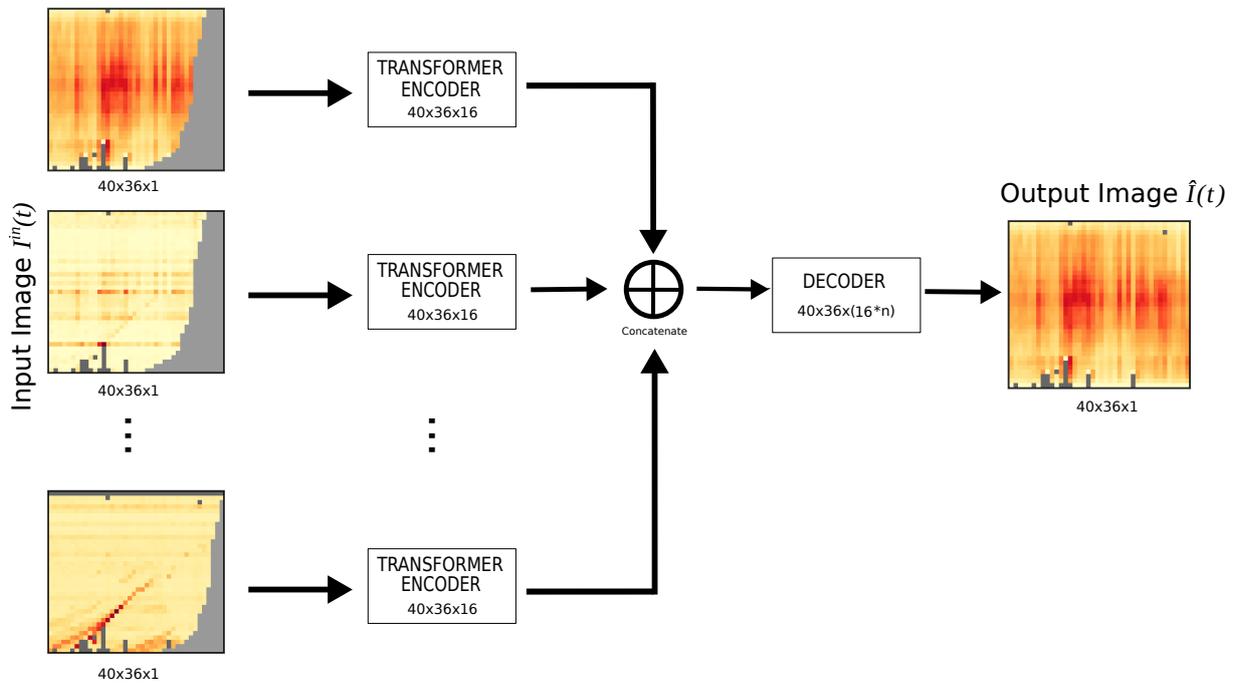


FIGURE 3.9 – Schéma de principe de l’architecture de type Channel Vision Transformer .[5]

les caractéristiques et le contexte de circulation pour une variable dans $I^{in}(t)$, comme le montre Figure 3.9. Le nombre d’encodeurs d est égal aux nombre de canaux de l’image. Les résultats de chaque canal sont concaténés avant d’être transmis à un décodeur. Le décodeur reconstruit l’image sur la base des caractéristiques extraites par la première partie du modèle comme le montre Figure 3.9. Le décodeur peut être soit une couche convolutive de taille 1×1 afin de prédire localement la charge ou un transformer décodeur. Dans le cadre de ces travaux, afin de réduire le coût en ressource le décodeur convolutif est choisi.

Pour profiter au mieux de l’analyse de l’attention dans le cadre du modèle encodant chaque canal indépendamment, nous avons décidé d’utiliser des patches de taille 1×1 . En effet, contrairement aux images standards où le voisinage porte la même information dans les différentes directions, les pixels d’une image d’une ligne de transport ne portent pas toujours la même information car pour deux pixels voisins, nous avons soit deux trains distincts, soit deux points d’arrêts avec des caractéristiques et dynamiques différentes. De plus la réduction de taille des patches améliore les performances du modèle mais implique un coût de calcul plus important car aucune réduction de dimensions n’est opérée.

3.3.6 Fonction de coût

En computer vision, l’apprentissage des modèles de deep learning ne se limite pas à prédire localement la couleur d’un pixel. Il faut également tenir compte de la cohérence globale de l’image. Pour des traitements d’inpainting, il faut que l’image générée par le modèle soit humainement réaliste. C’est à dire que la perception de l’image semble similaire à des images réelles. Des fonctions de coût ont été mises au point pour avoir un rendu pertinent tant au niveau de la texture (pixel-wise) que de l’aspect global ou la perception de l’image (high-level features).

3.3.6.1 Pixel Wise loss

- Pixel Wise loss : L'erreur par pixel est une mesure intuitive de la différence entre deux images. Elle est calculée localement pixel par pixel à l'aide de la différence entre les pixels prédits et réels de l'image. Elle nous informe de la performance locale de chacun des pixels. De par son aspect local, cette métrique ne permet pas de considérer la perception de l'image car elle n'estime l'erreur que localement. De plus des effets tel que des changements de couleur ou de luminosité de l'image peuvent impacter l'erreur par pixel.
- Perceptual loss : Pour traiter du rendu d'une image (les contours les textures, les formes), il faut pouvoir juger de la qualité d'une image et de sa vraisemblance. Contrairement à l'erreur pixel par pixel qui mesure la différence de valeur des pixels entre deux images, la computer vision se base sur l'erreur de perception (perceptual loss en anglais) afin de prendre en compte la qualité de l'image. Cette erreur peut être estimée directement entre l'image prédite et réelle. On peut également utiliser des fonctions d'erreur tenant compte des caractéristiques de l'image : formes, textures, couleurs, l'objectif étant de copier la perception humaine d'une image.

Ces différents métriques sont généralement combinées afin de traiter différentes échelles de l'image. Dans le cadre de ces travaux, nous utiliserons l'erreur par pixel car nous concentrons l'effort de prédiction sur un nombre réduit de pixels. L'usage d'une erreur de perception serait à privilégier dans le cadre de prédiction long terme afin de tenir compte de la cohérence des mission de trains.

3.4 Limitation de comparaison

Le choix de construire la prédiction de charge par un procédé d'image à image présente une contrainte de comparaison entre les modèles de la littérature et les architectures introduites dans ce chapitre. En effet, la prédiction de charge des réseaux de transport en commun ou dans les transports en général est construite selon une méthodologie de séquence à séquence (LSTM) qui suivent un échantillonnage régulier des données. C'est à dire que pour une séquence de données relatives à une station ou un réseau, on prédit la suite de la séquence pour l'ensemble des stations du réseau. Cette approche est souvent mise en place dans le cadre de données échantillonnées de manière régulière. Nous allons alors énumérer ci-dessous les contraintes induites par le format image.

Dans un premier temps, la question des données est la première contrainte. En effet, la grande majorité des modèles de prédiction de charge sont bâtis selon un échantillonnage régulier, c'est à dire une mesure toutes les n minutes. Ce format de données est couramment employé dans le cadre d'architectures récurrentes, de graphes ou modèles ensemblistes. La comparaison est alors non triviale entre un format avec des séquences régulières à un format irrégulier de données. De plus, cette difficulté de comparaison est amplifiée avec l'aspect spatial des images où l'irrégularité de l'échantillonnage de données n'est pas seulement liée à un train ou une station mais à l'ensemble d'une ligne. Plusieurs modèles tentent de tenir compte de cette caractéristique mais à une plus petite échelle, une station pour les travaux de [63] ou entre deux trains [18]. Ces modèles peuvent alors représenter des modèles de comparaison pertinents car ils intègrent les spécificités de notre méthodologie mais seulement localement. Il faudrait alors construire pour chaque station un modèle de prédiction local qui multiplierait les temps d'apprentissage selon la taille du

réseau. D'un point de vue industrielle, dupliquer des modèles sur l'ensemble des stations représente un surcoût pour l'usage de ces modèles.

Enfin, nous pouvons nous poser la question quant à la comparaison avec des modèles avancés comme les modèles de graphe qui représentent l'état de l'art [32, 28, 45] dans la prédiction de flux passagers de transport en commun. Ces architectures permettent de tenir compte de la réalité physique du réseau : correspondance, report de passagers sur des stations ou ligne et prennent tout leur intérêt pour des réseaux denses et inter-connectés. Le métro Parisien est un candidat idéal pour l'application de telles architectures de par sa densité (presque 3 stations par km²) et ses stations avec des correspondances au nombre de 60 pour 308 stations. Cependant les approches par graphes supposent l'accès aux données de l'ensemble du réseau de transport ou d'une partie importante de celui-ci. Dans le cadre du métro parisien seules trois des 14 lignes du système de transport (métro/rer) sont équipées de capteurs de charge. De plus, les trois lignes sont très peu connectées les unes avec les autres. Ainsi, une approche basée sur des lignes individuelles semblent plus appropriées pour le système de métro parisien qui ne permet pas de profiter pleinement des architectures de graphes.

En conclusion, bien que l'approche de la prédiction par traitement d'image semble pertinente pour la représentation des données, sa faible application dans la littérature rend sa comparaison avec les travaux de l'état de l'art plus difficile.

3.5 Conclusion

Ce chapitre a introduit la technique d'inpainting issue du traitement d'image et sa transposition à un problème concret de prédiction de charge. Il s'agit de reconstruire les départs de trains manquants dans les images de charge en temps réel tel que nous l'avons expliqué dans le chapitre précédent. La création de ces images a permis de transposer la tâche de prédiction en un traitement de type image à image.

L'inpainting est aujourd'hui dominé par les techniques d'apprentissage profond avec de manière chronologique, les modèles convolutifs, les modèles génératifs et les modèles intégrant le mécanisme d'attention. Il faut toutefois tenir compte de la spécificité de la tâche de prédiction qui ne rend pas la comparaison avec d'autres modèles de la littérature triviale (échantillonnage des données, ligne de transport).

Cette méthodologie est atypique et originale au regard de la littérature dans le domaine de prédiction des transport en commun. Celle-ci comporte majoritairement des modèles LSTM et à base de graphes. On note également l'essor des modèles d'attention et leur application aux graphes [73]. De plus, il ne s'agit pas d'utiliser les images comme support de l'information spatiale [40] mais de transposer la problématique sous une approche de computer vision et de voir cette dernière comme une boîte à outils pour les transports en commun tant dans les modélisations que les applications. L'objectif est alors mettre à profit les images pour la construction de modèles interprétables et pertinents au regard des données et des contraintes opérationnelles pour la prédiction.

Chapitre 4

Une approche de type image pour la prédiction des charges d'une ligne de métro

4.1 Introduction

Dans les chapitre 2 et 3, nous avons présenté comment traduire une tâche de prévision de charges en une tâche de traitement d'images (Inpainting). Le protocole mis en avant a pour but d'utiliser le format image à la fois comme outil de visualisation et de support pour la prédiction. La tâche d'Inpainting pour la prédiction de charges doit s'accommoder de plusieurs contraintes techniques et opérationnelles de la ligne étudiée. Ce procédé est mis en place dans le cadre d'une expérimentation sur la ligne 9 du métro Parisien, avec comme objectif de prédire pour chaque point d'arrêt de la ligne 9 du métro les charges passagers des 4 prochains départs en tenant compte du fait que les temps d'attente sont disponibles pour chaque station pour les 4 prochains trains. Enfin, nous allons comparer les performances des modèles proposés à d'autres modèles de l'état de l'art. Cette comparaison sera accompagnée d'une analyse spatiale et temporelle des performances de prédiction. Le but est de mettre en évidence les heures et stations critiques de la ligne de transport où la prévision pose plus de difficultés.

4.2 Expérimentation sur la ligne 9 du métro Parisien

Nous avons mené une expérimentation sur la ligne 9 du métro parisien à partir de données collectées entre 2019 et 2022. Cette section présente le cadre mis en place pour valider et évaluer les méthodes de prédiction par inpainting. Nous fournissons également les ressources mises à disposition pour nos travaux ainsi que les contraintes rencontrées lors de la mise en place des modèles. Pour rappel, les données concernent la ligne 9 du métro parisien, de Pont de Sèvres à Mairie de Montreuil, sur une période allant de 2019 à 2023. L'objectif de cette expérimentation est de prédire les charges des 4 prochains départs de train pour chaque point d'arrêt de la ligne. Il s'agit de reproduire une tâche de prévision de charge basée sur les processus internes de l'opérateur de transport, en se basant principalement sur les informations des voyageurs pour la ligne 9.

4.2.1 Architectures des modèles de prédiction

L'évaluation et la comparaison des performances de modèles de prédiction ont été fortement orientées par le fait de suivre une approche "traitement d'images". Nous souhaitons évaluer à la fois la pertinence de l'utilisation du traitement d'image et les choix d'architectures, nous utiliserons pour cela, les modèles décrits dans le chapitre précédent, en ajoutant des structures de base telles que des réseaux de neurones standards et convolutionnels. Maintenant que les grandes familles d'algorithmes ont été mentionnées, nous allons nous intéresser à leur usage dans la prédiction de charge au sein de transport en public. Il s'agira de regarder dans la littérature l'usage des images dans la prédiction à court terme mais également de présenter les méthodologies pouvant s'apparenter à de l'inpainting utilisé dans des domaines externes à la computer vision.

4.2.2 Modèles de réseaux de neurones profonds

Les méthodes d'apprentissage profond sont également largement utilisées dans la prédiction de la charge des trains en raison de leur capacité à traiter les caractéristiques non linéaires. Par exemple, [26] a déployé un réseau de neurones (ANN) au niveau des gares. [33] a combiné les modèles ARIMA avec des réseaux neuronaux (ANN) afin de gérer caractéristiques linéaires et non linéaires dans les problèmes de prédiction. On retrouve les ANN principalement utilisés comme des blocs de décision soit dans des architectures GAN sous la forme de discriminateur ou pour la classification d'image en sortie de convolution.

Dans le cadre du benchmark conduit durant cette thèse, le modèle de comparaison sera un réseau de neurones (ANN) ayant comme entrées chaque pixel de l'image et le même nombre de sorties. Il se composera de plusieurs couches cachées afin d'extraire les caractéristiques de l'image.

4.2.3 Modèles de réseaux de neurones Convolutifs

Les modèles CNN sont couramment utilisés en prédiction à court terme comme encodeur de dépendances spatiales de l'espace de prédiction. Ils sont utilisés dans des structures hybrides en amont pour extraire les caractéristiques spatiales soit à partir d'un découpage géographique des données (quadrillage d'une ville)[38, 39], soit à partir d'une structure ordonnée de stations [56]. On retrouve [76] qui ont développé un réseau LSTM convolutionnel (ConvLSTM) pour extraire des informations spatio-temporelles afin de prédire les flux de passagers pour la ville de Chengdu en Chine. La structure ConvLSTM permet de traiter à la fois les dépendances spatiales et temporelles contrairement aux modèles hybrides cités plus haut.

Dans une optique "computer vision", contrairement aux modèles précédemment introduits, peu de travaux utilisent une approche strictement visuelle pour la prédiction de charge [39]. Cependant, on peut considérer certains travaux [55] où les auteurs traitent une image spatio-temporelle du trafic d'un axe routier comme entrée et construisent les prochaines valeurs de flux sur cet axe. Il ne s'agit pas de modèles d'image à image comme un réseau de type U-net par exemple. Dans d'autres domaines, l'inpainting [53] est utilisé pour prévoir les performances des systèmes d'énergie électrique en utilisant des séries temporelles comme images 3D. Nous utiliserons le modèle introduit par [55] comme benchmark. Nous l'accompagnerons d'un modèle strictement convolutionnel à l'instar du U-net mais sans les skip connections. Le but est de juger de l'impact de ces connections dans une tâche de prédiction qui diffère de l'usage naturel du U-net qui porte sur la segmentation.

Son architecture est similaire en termes de paramètres et couches de pooling et fonction d'activation.

4.2.4 Modèles d'attention et Transformer

L'attention prend une part importante dans de nombreux domaines tant en traitement d'images qu'en transport. L'application première de l'attention dans la prédiction à court terme est liée au modèle de graphe [86, 77]. L'attention remplace dans ces modèles les couches de convolution de graphes [74]. Ces modèles reposent en général sur trois composants, un graphe pour encoder la physique du réseau de transports, l'attention pour mettre en valeur les relations et dépendances entre les noeuds du réseau et enfin un bloc temporel (LSTM, GRU) pour extraire les dépendances temporelles.

Nous nous concentrerons sur les modèles de Vision Transformer car ils représentent les modèles pouvant être appliqués dans le cas de notre protocole d'image. Le modèle revient à un encodeur transformer comme introduit dans [22]. Le décodeur sera une couche de convolution pour reconstruire localement chaque pixel à partir des features d'attention extraites. Ce choix de convolution en sortie est aussi lié au coût de calcul des transformers. Un modèle entièrement construit avec des transformers serait plus coûteux. Les modèles de graphes ne sont pas pertinents dans le cadre d'une ligne de transport comme nous l'avons précisé. De plus, le coût d'apprentissage nous freine également dans l'implémentation de ces architectures.

4.2.5 Modèle U-net

Le modèle U-net mis en oeuvre dans cette expérimentation se compose d'une convolution 3x3 avec comme fonction d'activation une ReLU suivie d'un max pooling 2x2. Cette étape de réduction de dimensions est accompagnée d'une augmentation du nombre de features channels doublant à chaque étape. Le résultat est un bottleneck de 512 features. La reconstruction de l'image se fait via des couches de up-convolution symétrique à l'étape de convolution. La seconde étape consiste en des convolution de taille 3x3 suivie d'un up-sampling de taille 2x2 pour la reconstruction des images. Le choix d'un pooling 2x2 et de convolution 3x3 sont faits selon les dimensions de l'image de trafic ferroviaire. Le pooling permet de conserver les mêmes dimensions d'image en phase de convolution et de dé-convolution sans opérer de traitement supplémentaire (cropping par exemple). Dans une même optique, la faible taille de l'image (40 par 36 pixels) ne présentent pas l'intérêt de convolution de taille supérieure. En sortie du modèle, une dernière couche de convolution de taille 1x1 permet de réduire localement chaque pixel en une image d'un seul canal qui représente la cible de prédiction. La fonction d'activation de la dernière couche est remplacée par la fonction softplus pour s'accorder à la charge qui est positive. La figure 3.6 du précédent chapitre est le modèle utilisé dans l'expérimentation. On note que les modèles de comparaison construits à partir de convolution ont des structures similaires par rapport au nombre de couches de convolution et de pooling.

4.2.6 Modèle U-transformer

Le modèle U-transformer est construit de manière analogue au modèle U-net en remplaçant chaque étape de convolution par des blocs transformer. Le modèle consiste alors en trois blocs de transformer de 64 têtes. Le premier block utilise une taille de patch

3x3 et une dimension de projection de 16. Entre chaque transformer-encoder, le nombre de patch est divisé par deux et la dimension de projection est quant à elle doublée. Ces étapes de réduction de dimension sont suivies par une phase de up-sampling. En sortie du modèle, une couche de convolution 1x1 avec comme fonction d'activation "softplus" est utilisée pour reconstruire la charge.

4.2.7 Modèle Channel ViT

Le channel ViT se compose de 4 transformers encoders, un par canal de l'image. Chaque bloc de transformer se compose de 16 têtes, d'une dimension de projection de 16 et d'une taille de patch de 1x1. La sortie de chaque transformer est concaténée et utilisée comme entrée d'un décodeur convolutionnel similaire au précédent modèle. Une taille de patch 1x1 a été utilisée pour deux raisons, la première est un gain de performance en comparaison avec des patches de plus grande taille. La seconde est une interprétation des choix des modèles par l'obtention des scores d'attention pour chacun des pixels. Cependant, ce choix augmente le coût en ressources (en particulier en ce qui concerne l'usage de la RAM de la carte graphique) des modèles car il ne profite pas de la réduction de dimension offerte par des patches de plusieurs pixels. Ce choix n'a pas pu être fait pour le U-transformer pour des contraintes techniques liées au coût d'apprentissage du modèle, mais devenait possible avec une approche séparant le traitement de chacun des canaux de l'image.

4.2.8 Benchmark de prédiction

Pour résumer, le benchmark de l'expérimentation consiste en l'usage de modèles adaptés à la structure d'inpainting. Ainsi, les modèles utilisés à des fins de comparaison sont les suivants :

- Modèle Naïf
- Neural Network (NN) [55]
- CNN + NN
- Fully Convolutionnal NN
- Unet
- Vision Transformer
- Channel ViT
- U-Transformer

Nous faisons le choix d'une approche globale qui vise à prédire les charges pour l'ensemble de la ligne (stations, trains) car celle-ci limite grandement le nombre de modèles à développer. Il aurait été envisageable de modéliser chaque départ de train à prédire par un modèle unitaire ou récurrent par station ou train mais cela revient à multiplier le nombre de modèles de prédiction pour chacune des stations et sens de circulation des trains. Le tableau 4.1 montre les différents modèles ainsi que les différents paramètres utilisés pour leur conception et apprentissage. Nous allons définir en détail les 3 principales architectures à savoir le modèle U-net, le modèle U-transformer et le modèle Channel Transformer (Channel ViT).

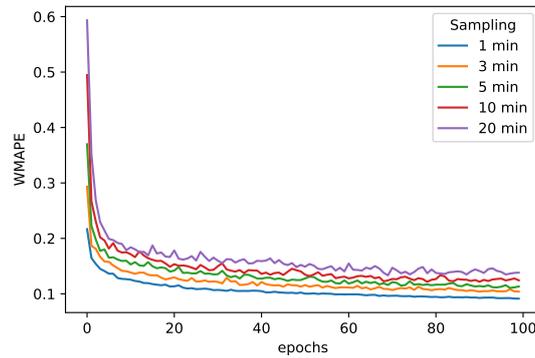
4.2.9 Taux d'échantillonnage

Avant tout apprentissage ou construction de base d'apprentissage ou test, une question à résoudre porte sur l'échantillonnage des images tant pour la base d'apprentissage que

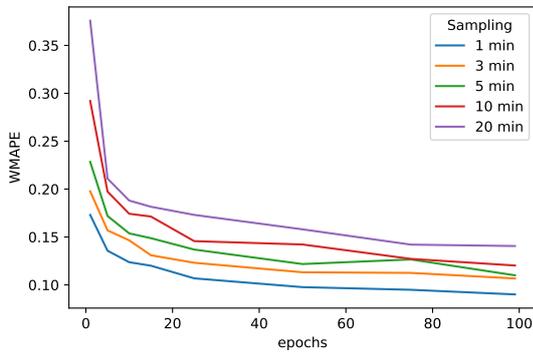
TABLE 4.1 – The settings of all the benchmarked models.

| Models | #parameters | encoding | decoding | Batch size | Head | patch size |
|------------------|-------------|------------|------------|------------|------|------------|
| Naïve | - | - | - | - | - | - |
| NN | 27,354,528 | 24,403,968 | 2,950,560 | 128 | - | - |
| CNN | 20,254,465 | 11,173,632 | 9,080,833 | 128 | - | - |
| CNN+NN [55] | 25,418,272 | 22,467,712 | 2,950,560 | 128 | - | - |
| ViT [22] | 232,449 | 232,400 | 49 | 8 | 64 | 3 |
| U-net [6] | 16,552,065 | 4,112,512 | 12,439,553 | 128 | - | - |
| U-Transformer | 1,911,985 | 618,128 | 1,293,857 | 8 | 64 | 3 |
| C-ViT | 588,417 | 117,664 | 470,753 | 4 | 16 | 1 |

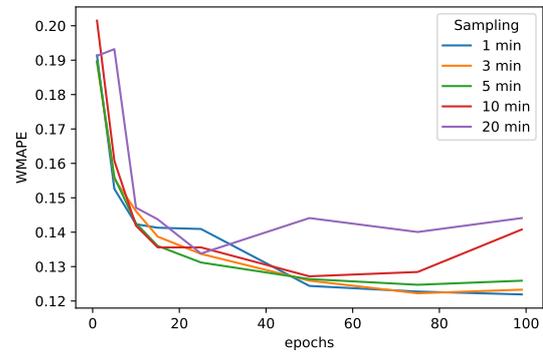
pour celle de test. En effet, les images étant des instantanées de la ligne, chaque image peut être générée selon une fréquence choisie soit par des contraintes techniques (toutes les X minutes pour s'accorder avec le système d'information de la ligne) , soit pour chaque nouveau départ de trains (apparition d'un nouveau pixel dans l'image). Et cette question d'échantillonnage se pose à la fois pour l'apprentissage et pour le test ; l'échantillonnage utilisé peut être différent dans les deux phases. En effet, la tâche de prévision peut selon son horizon temporel être faite toutes les minutes pour une prédiction court-terme ou toutes les 20 minutes pour une prédiction moyen terme.



(a) Jeu de données d'entrainement.



(b) Jeu de données de test (échantillonnage à 5 min)



(c) Jeu de données de test (échantillonnage à 1 min)

FIGURE 4.1 – Évolution des erreurs du modèle de prévision de type U-net en fonction de différents taux d'échantillonnage

Afin d'illustrer l'impact de la fréquence d'échantillonnage des images sur la prédiction, nous avons réalisé plusieurs expériences préliminaires avec des bases d'apprentissage construites selon des temps d'échantillonnage allant de 1 minute à 20 minutes (cf. Figure 4.1). A partir de ces différentes bases de données, un modèle de prévision¹ est construit pour chacune des fréquences d'échantillonnage. Il en résulte la figure 4.1a qui montre l'évolution de l'erreur de prédiction en apprentissage des différents modèles pour chaque fréquence. On note que l'erreur d'apprentissage diminue lorsque la fréquence d'échantillonnage augmente.

Ces modèles sont testés ensuite sur différentes bases de test avec un échantillonnage de 1-minute pour la figure 4.1c et de 5-minutes pour la figure 4.1b. L'objectif est de mettre en évidence le gain de performance d'un échantillonnage type selon que la visée applicative de prédiction soit de court ou de moyen terme. Sur la figure 4.1b, on constate la même tendance qu'en apprentissage, à savoir que l'erreur de prédiction est plus faible pour les fréquences d'échantillonnage hautes. Cependant, la figure 4.1c met en lumière le fait que la fréquence d'échantillonnage de test doit être homogène avec celle utilisée en apprentissage, faute de quoi on constate un sur-apprentissage. On conclut que l'échantillonnage de test doit être à minima inférieur à l'échantillonnage cible (celui de la prévision voulue pour le système). Ainsi, dans le cadre de ces travaux, afin d'avoir la prédiction à court terme la plus pertinente et fine possible, une fréquence d'échantillonnage de 1-minute est choisie tant pour le test que pour l'apprentissage.

4.2.10 Bases d'apprentissage et de test

Avec plus de trois ans de données, la sélection d'une base d'apprentissage et d'une base de test pertinentes pour évaluer de façon robuste les modèles est une étape importante. Au regard du trafic ferroviaire qui a connu une évolution majeure avec la crise du COVID, la généralisation du télé-travail et la présence de périodes de grèves, autant de d'événements qui ont impacté la demande et l'offre dans les transports publics. Une première solution aurait consisté à choisir des bases de test sur une fenêtre glissante pour ne se concentrer que sur les données les plus récentes. Mais cette solution n'aurait pas permis de tester les modèles sur les différentes situations complexes observées sur la ligne depuis 2019 (grève, confinement, télétravail).

Ainsi l'ensemble des données est composé d'images générées toutes les minutes depuis le démarrage du service, à 5h30 du matin, jusqu'à la fin du service à 1h30 du matin le jour suivant, pour un total de 20 heures. Le nombre total d'images par jour est donc de 1200. Le nombre total d'images obtenues de janvier 2019 à décembre 2022 s'élève à plus de 1,000,000 images. L'année 2022 ne présente que 106 jours au total réparties sur l'année complète². La base d'apprentissage est définie sur les 3 premières années de 2019 à 2021 et représente environ 700,000 images. Cependant, afin de couvrir le plus grand nombre de situations opérationnelles de transport, plusieurs mois ont été sélectionnés dans ces trois ans dans la base d'apprentissage (2 pour 2019, 2 pour 2020, et 1 pour 2021). Ils regroupent à la fois des mois nominaux et atypiques incluant une grève, un confinement ou des vacances. On ajoute à ceci l'année 2022 comme base de test pour un total de 400,000 images en test.

Nous tenons à préciser que des mois entiers ont été choisis pour éviter toute fuite de données entre les bases de test et d'apprentissage. Cette contrainte est faite afin que deux

1. U-net pour avoir un faible temps de calcul

2. Les processus de sauvegarde des données sont arrêtés plusieurs fois dans l'année.

images successives ne soit pas séparées entre les deux bases de par leur forte similitude. De plus le fait de choisir des mois spécifiques permet de garantir une base de test pertinente au vu des événements ayant impacté les transports parisiens sur ces dernières années.

4.2.11 Contraintes lors de l’entraînement et plan d’expérience

L’entraînement est réalisé sur un processeur graphique (GPU) NVIDIA RTX 3090 avec 24 Go de RAM en utilisant le framework TensorFlow et python 3.8. La descente de gradient est basée sur l’optimiseur Adam [43]. La Table 4.1 présente les paramètres des neuf modèles considérés, avec leur nombre de paramètres et la taille du patch utilisés pour la formation de tous les modèles. Les modèles basés sur les transformers comportent deux colonnes supplémentaires, le nombre de têtes et la taille du patch utilisés pour construire les modèles. L’ensemble des modèles est entraîné sur la même base d’apprentissage et un nombre constant d’époques fixé à 10.

Enfin, on note dans le tableau Table 4.1 que les tailles des batchs diffèrent de manière significative entre les modèles purement convolutionnelles et ceux utilisant des transformers. La raison est l’utilisation de l’attention par la RAM. Le calcul quadratique et le coût de la mémoire GPU de l’attention ont un impact important sur la taille des modèles basés sur les transformers par rapport aux modèles CNN ou NN. Cette contrainte entraîne un compromis entre la taille du patch, le nombre de têtes, la taille du batch et la dimension de la projection. Cette contrainte impose des tailles de batch plus petites et un nombre de têtes faible. Malgré ces valeurs plus petites, il est à noter que le temps d’entraînement des modèles de type transformer est tout de même plus important. Il paraît donc naturel de se poser la question de l’apport de telles architectures dans un cadre industriel. En tout état de cause, le coût d’entraînement est largement en faveur des modèles convolutionnels.

4.3 Résultats de prédiction

Nous présentons maintenant les résultats de prédiction de l’ensemble des 4 prochains départs de train sur la base de test introduite précédemment. Les résultats se concentreront sur la performance globale des modèles en proposant des analyses par pixel ainsi que des analyses différenciées selon les caractéristiques temporelles et spatiales. Afin de garantir une évaluation robuste, la base de test se compose de différents contextes regroupant un large panel de situations atypiques. Nous tiendrons compte de cette spécificité dans le prochain chapitre qui sera dédié aux situations atypiques.

4.3.1 Résultats globaux

Les résultats du Tableau 4.2 présentent les performances de prédiction pour les deux étapes de notre protocole : inpainting et prédiction, à l’aide de l’écart quadratique moyen (RMSE) basé sur la fonction de coût d’apprentissage et pour l’erreur moyenne pondérée en pourcentage (Weighted Mean Average Percentage Error WMAPE). L’erreur RMSE permet de rendre compte de l’erreur moyenne en nombre de passagers par pixel. À l’opposé, l’erreur WMAPE (eq 4.2) permet de tenir compte de la charge prédite afin d’éviter les effets de bord rencontrés pour les faibles et/ou fortes charges.

$$RMSE = \sqrt{\frac{1}{|S| \cdot |C_t|} \sum (I(t) - \hat{I}(t))^2}. \quad (4.1)$$

$$WMAPE = \frac{\sum |\hat{I}(t) - I(t)|}{\sum |I(t)|}. \quad (4.2)$$

A propos de la tâche Inpainting, les architectures avancées comme le VIT ou celles proposées dans ces travaux présentent un gain en reconstruction d’image d’environ deux fois les performances du modèles naïf en RMSE. On note même que en WMAPE, les modèles utilisant des réseaux de neurones sont moins performants que le modèle naïf et que ce sont les architectures transformer et CNN qui offrent les meilleurs résultats. Enfin, c’est le modèle U-net qui présente les meilleurs performances pour l’inpainting. Cette architecture présente une meilleure capacité pour la reconstruction de l’image dans sa globalité. Elle est à privilégier pour la prévision à moyen ou long terme qui visent à prédire efficacement l’entièreté de l’image et non pas seulement les 4 prochains pixels par station.

Cependant dans le contexte de l’expérimentation, les modèles se basant sur des transformers présentent de fortes contraintes techniques qui impactent à la fois l’apprentissage des modèles et l’espace de leurs features. Ainsi, en utilisant plus de ressources, il serait envisageable d’améliorer les performances de ces architectures.³

TABLE 4.2 – Résultats de l’inpainting et de la prévision des prochains départs en pourcentage d’erreur absolue moyenne pondérée (WMAPE) et en erreur quadratique moyenne (RMSE).

| Models (reference) | Inpainting | | Forecasting | |
|----------------------------|------------|------------|-------------|-------------|
| | RMSE | WMAPE | RMSE | WMAPE |
| Naïve | 63 | 14.2 | 50 | 19.9 |
| NN | 60 | 22.4 | 48 | 18.9 |
| CNN | 41 | 12.3 | 33 | 13.1 |
| CNN + NN [55] | 44 | 15.9 | 36 | 14.8 |
| VIT [22] | 30 | 8.3 | 43 | 18.2 |
| U-net [6] | 29 | 6.6 | 31 | 12.3 |
| U-Transformer | 34 | 9.9 | 31 | 11.2 |
| Channel Vision Transformer | 32 | 9.2 | 32 | 12.1 |

Pour illustrer le fonctionnement du modèle en prédiction, nous présentons dans les figures 4.2 et 4.3, un exemple de prédiction du modèle U-transformer à la station Havre-Caumartin le 25 Juin 2020 pour des départs de trains en heure creuse et en heure de pointe, la ligne bleu représente les prédictions du modèle pour le prochain passage de trains avec une mise a jour de la prédiction toutes les minutes, (la valeur prédite qui serait affichée à un voyageur sur le quai); les lignes oranges quand à elles présentent la charge réelle du train à prédire lorsque celui-ci est arrivé a quai. Enfin, des lignes verticales gris claires délimitent les différents trains prédits.

La figure 4.2 montre également qu’un même départ de train peut être prédit plusieurs fois par le modèle comme pour le train de 11 heures. En effet l’échantillonnage des images à la minute peut dans le cadre de temps d’attente long (5 minutes) revient à prédire plusieurs fois le même départ sans garantir que la valeur de charge prédite soit la même.

3. Nous tenons à préciser que dans nos premières expérimentations [6], les modèles avaient de meilleurs performances que les modèles cités ici. En effet, le modèle U-net avait une WMAPE de 6,2% (contre 11,2%) en raison du cadre expérimental qui englobait l’année 2019 sans situations atypiques majeures (grèves, confinements). La grande taille du jeu de données diminue les performances des modèles qui se trouvent confrontés à un plus large panel de contextes de transport.

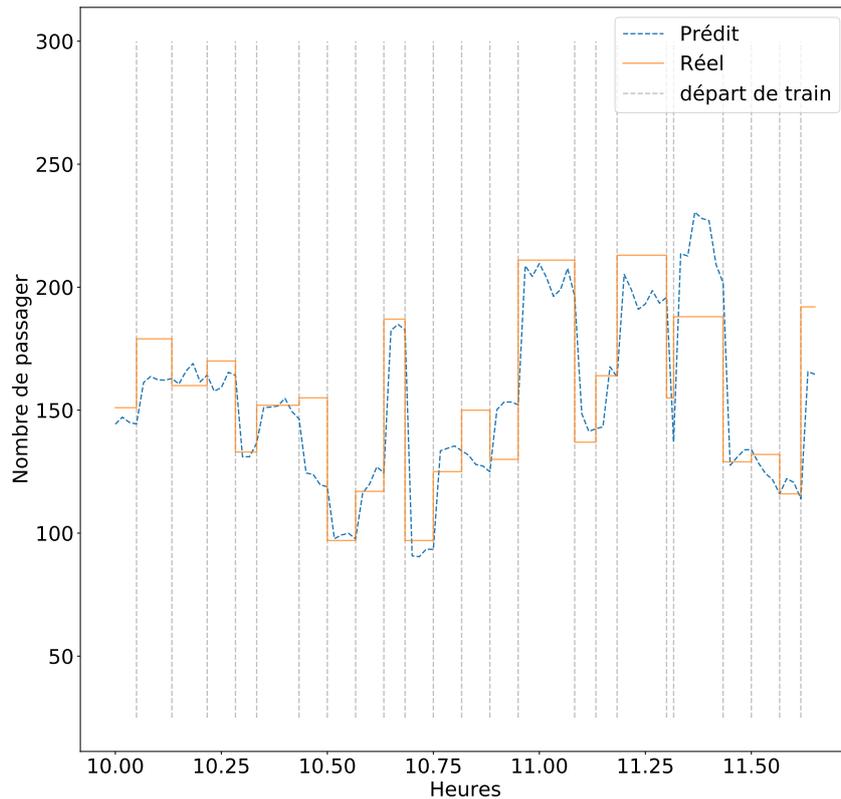


FIGURE 4.2 – Prédiction du modèle U-Transformer pour le 25 juin 2020 pour la station de Havre Caumartin en heure creuse. Les lignes verticales représentent les départs de trains à la station.

La mise à jour de pixels dans l’image (nouveau départ de trains dans la ligne) peut changer les résultats de prédiction. On observe ce comportement principalement pour les heures creuses du fait d’un plus faible volume de trains.

En comparaison, pour la figure 4.3, la fréquence des départ de trains augmentent en réponse à la demande. On observe que l’échantillonnage des images à la minute permet de considérer tous les départs de trains pour la prédiction. Ainsi, le format image avec un échantillonnage pertinent, permet de tenir compte des variations entre la fréquence des trains aux différentes heures de la journée. De plus, cela permet de tenir compte des irrégularités des départs de trains.

4.3.2 Résultats par pixels

Sur l’ensemble des images prédites en test et en apprentissage, le rendu d’une image et la position des pixels à prédire dans l’image varient entre les heures de la journée et entre les jours de la semaine, s’accordant selon l’offre et la demande des périodes étudiées. Ainsi, le nombre de trains variant selon l’heure de la journée introduit des formes d’images différentes les unes des autres. Il est pertinent alors de vouloir noter les régions de l’image les plus prédites ainsi que de localiser les performances du modèles. Il s’agit de visualiser directement dans l’image les aires principales de prédictions et leur répartition dans l’image.

Pour ce faire la Figure 4.4 présentent les résultats sous la forme de heatmaps selon le nombre de fois où le pixel est considéré comme à prédire et les performances de prédiction

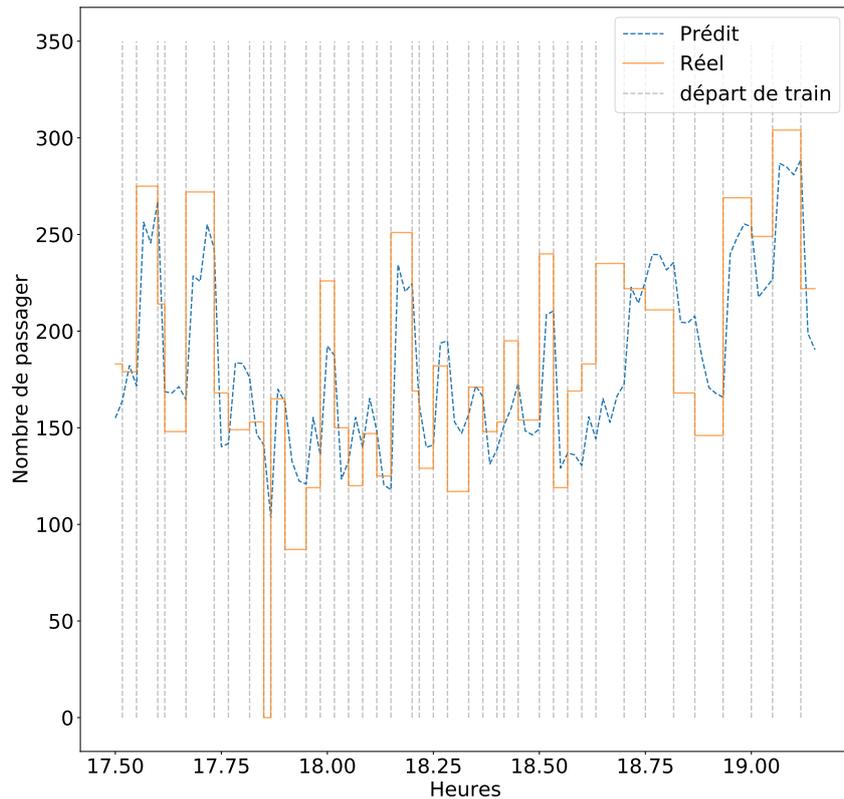


FIGURE 4.3 – Prédiction du modèle U-Transformer pour le 25 juin 2020 pour la station de Havre Caumartin en heure de pointe. Les lignes verticales représentent les départs de trains à la station.

par pixel. Elle permet de juger des performances de prédiction des modèles pixel par pixel. Elle se décompose en trois sous-figures. La première, la figure 4.4a présente la fréquence de prédiction de chaque pixel, elle nous renseigne sur le nombre de fois où chaque pixel est prédit. La seconde figure 4.4b présente les performances de prédiction par pixel sur la base de test. Enfin la dernière figure 4.4c est le produit des deux figures précédentes. Il s’agit de pondérer les performances par pixel selon la fréquence de prédiction.

Premièrement on note pour chacune des images une région en gris dans l’image qui correspond aux pixels qui n’ont jamais été la cible d’une prédiction. On note que cette zone est réduite dans la partie gauche de l’image qui représentent temporellement les départ de trains les plus vieux. Toutefois, on retrouve des départs de trains prédit tôt dans l’image (dans la partie gauche) qui sont étonnant et traduisent des cas atypiques (grève, début de service). Hormis ces cas spécifiques, la zone de prédiction principale se situe comme le montre la figure 4.4a sous la forme d’un axe entre les trains en haut à droite de l’image qui sont presque toujours à prédire et les trains entre le 10^{ieme} et 35^{ieme} trains. Ces régions dont le nombre de prédiction est supérieur à 40% dénotent les fronts des trains de la ligne dans le cas nominal de circulations de ces derniers. La prédiction est ainsi effectuée sur une région spécifique de l’image dans la majorité des cas.

On peut poursuivre l’analyse en s’intéressant à l’erreur par pixel dans nos images de manière analogue à la figure 4.4a. Le but est de juger si la position d’un pixel impacte sa prédiction. La figure 4.4b représente l’erreur WMAPE de chacun des pixel. On note dans un premier temps que le modèle parvient difficilement à prédire les pixels des derniers trains de l’image et certains pixels dans la partie gauche de l’image. La région principale

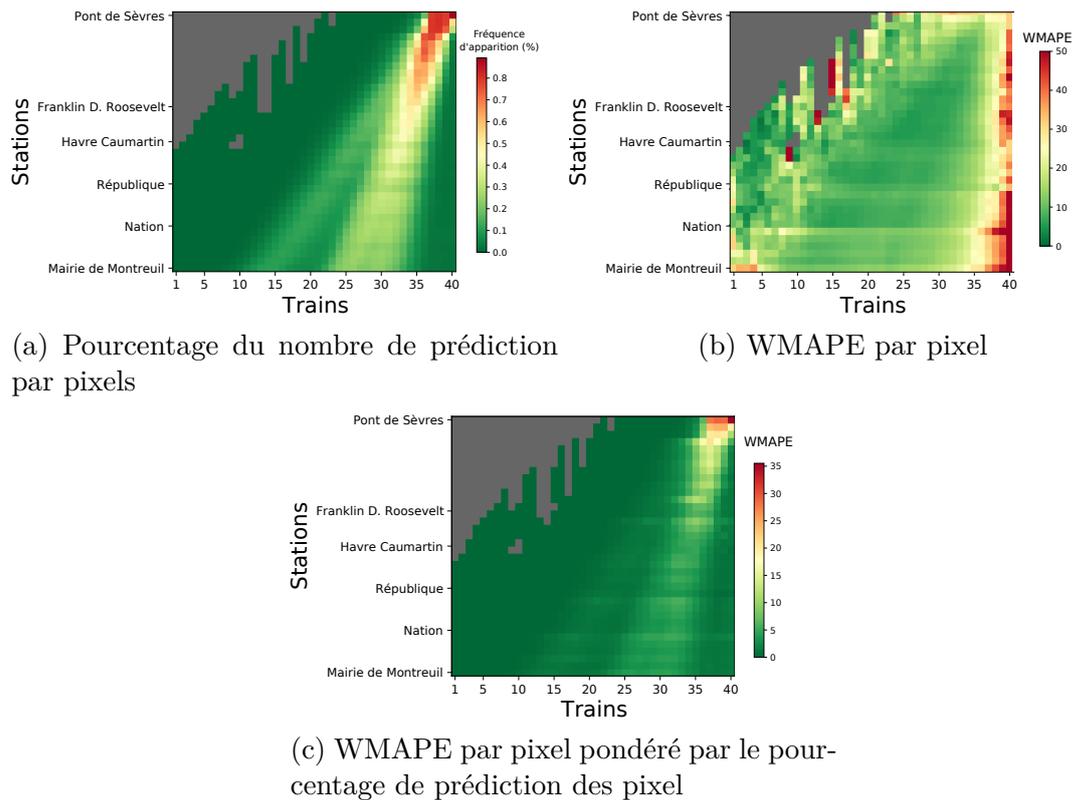


FIGURE 4.4 – Visualisation par pixel de la tâche de prédiction

de prédiction présentent quant à elle les meilleurs résultats. Comme le montre la figure 4.4c, il faut toutefois préciser que les résultats de chaque pixel sont pondérés par leur nombre de prédiction et que les pixels mal prédits représentent des cas rares du trafic et donc faiblement représentés dans les données. La figure 4.4c montre bien que ce sont les pixels en début de ligne qui sont les moins bien prédits et d'autre part les moins prédits prédit dans le cadre du format par image.

Ainsi, on peut noter que la prédiction par inpainting se concentre sur des régions spécifiques de l'image. On note également que le modèle prédit mal certaines régions de l'image mais qui sont très peu représentées dans les données. Ainsi en tenant compte de la fréquence à laquelle un pixel est à prédire, ce sont les départs de train en début de ligne qui présentent les moins bons résultats. Ils correspondent à des départs de train où nous ne possédons que peu d'information quant à la dynamique propre des trains.

4.3.3 Analyse spatio-temporelle des résultats

En commençant par l'analyse temporelle, la figure 4.5 met en évidence l'erreur en prédiction des modèles selon l'heure de la journée. Le but est de détecter les heures critiques en prédiction avec une erreur WMAPE plus élevée. On note dans un premier temps que ce sont les heures en début et fin de services (5h30 et 23h30) qui présentent les moins bonnes performances. Ces périodes correspondent à des charge de trains faibles. On note que l'erreur augmentent à partir de 20 : 30 après l'heure de pointes du soir (16 :30, 20 :30).

Concernant l'aspect spatial, la figure 4.6 présente les mêmes conclusions que lors de

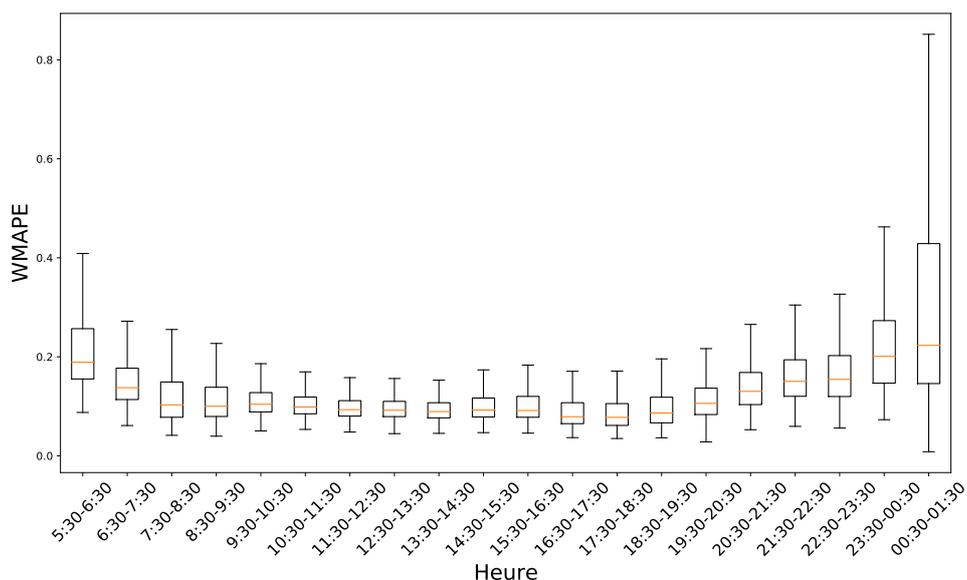


FIGURE 4.5 – Erreur en WMAPE par heure de la journée

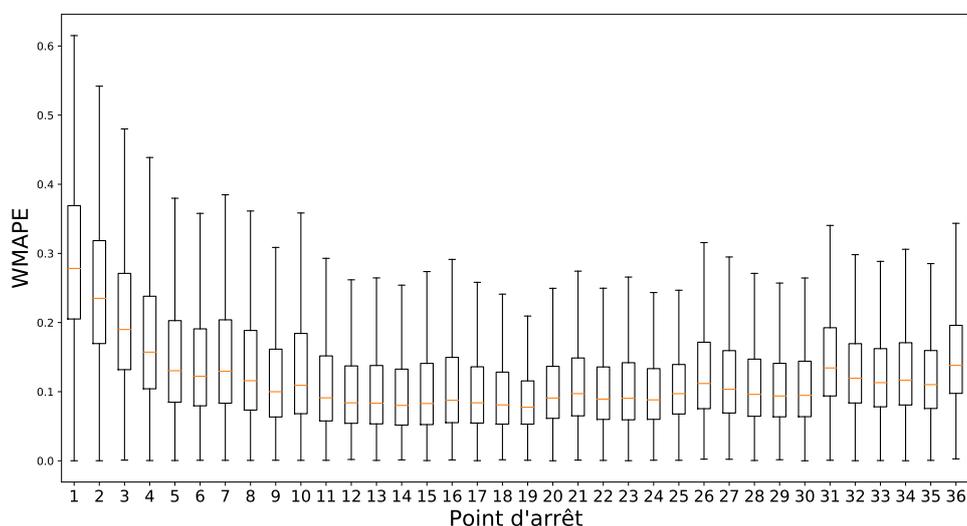


FIGURE 4.6 – Erreur en WMAPE par station de la ligne (sens de circulation de gauche à droite)

l'analyse par pixel (cf figure 4.4c) . Ce sont les point d'arrêt en début de parcours qui présentent l'erreur médiane la plus grande ainsi qu'une plus grande dispersion de l'erreur. Ce phénomène s'explique par le fait qu'en début de parcours, les modèles n'ont que peu d'information ou de contexte concernant les trains. Pour les premières stations, la grande majorité des trains n'ont pas fait encore un seul arrêt. Il est alors plus difficile de prédire la charge alors qu'en fin de ligne, les modèles ont une plus grande source d'information concernant les trains à prédire.

On note également que certaines stations comme la station 31 et 26 (respectivement Nation et République) présentent une erreur médiane et une variance plus importante. Ces point d'arrêt ont pour particularité de posséder des connexions avec d'autres lignes du réseau. Cette remarque peut être étendue aux autres stations de la ligne ayant des correspondances. Cette augmentation de l'erreur est principalement due au fait que les

flux de passagers entre les lignes au sein d'un même point d'arrêt (on parle alors d'espace contrôlé) ne peuvent pas être mesurés. Ces passagers ne valident pas leur titre de transport dans cette situation. Ainsi Les correspondances influent sur les performances de prévisions.

Ainsi, les faiblesses du modèle se situent d'une part lors de la mise en service de la ligne et en fin d'exploitation de la ligne qui se traduisent par des périodes de faibles charge suppléées d'une fréquence de trains faible. D'autre part, on note également que la prédiction de charge est meilleure pour des départs de trains en aval de la ligne car on profite de l'ensemble des départs du trains aux stations précédentes, avec cependant de moins bonnes performances pour les stations avec des correspondances, qui comme attendu sont plus difficiles à prédire.

4.3.4 Analyse des scores d'attention

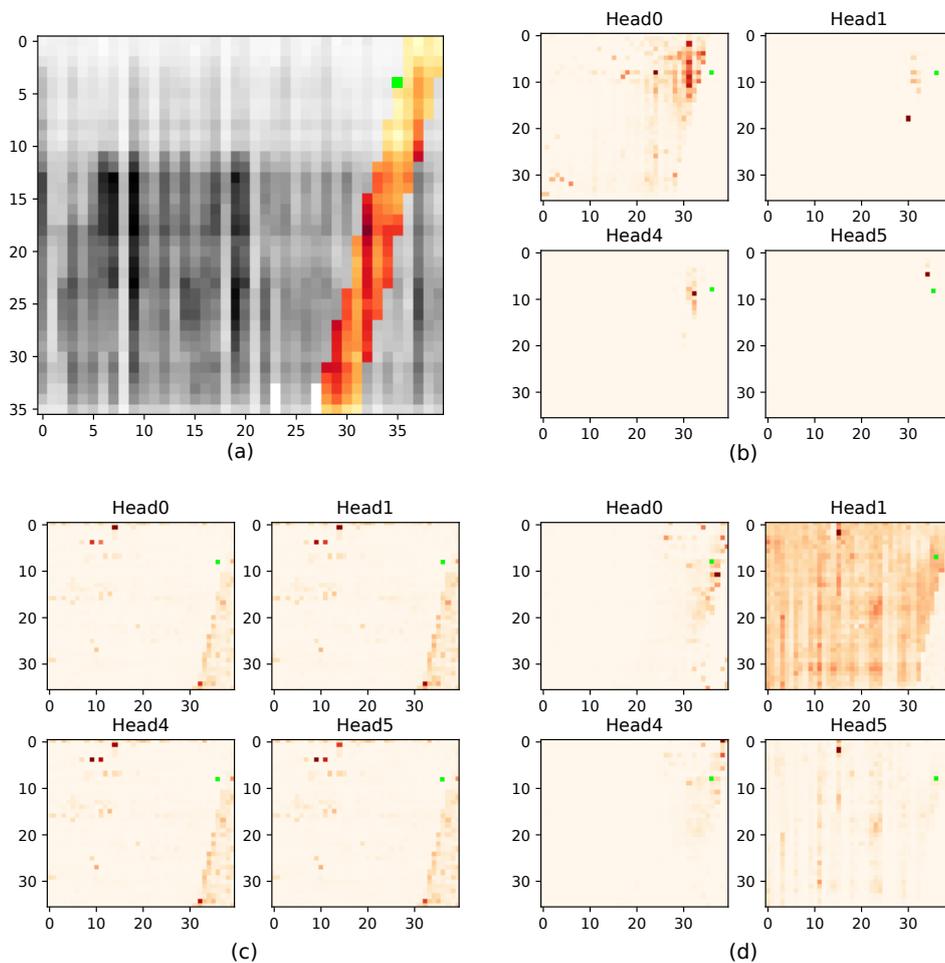


FIGURE 4.7 – Visualisation des scores d'attention pour un seul pixel dans 4 têtes du modèle de transformation du canal de vision utilisé pour la prédiction de la charge des passagers : (a) $I(t)$ étudié avec le pixel cible ; (b) scores d'attention du canal de charge des passagers ; (c) scores d'attention du canal de temps de trajet ; (d) scores d'attention du canal de temps d'attente ; [une couleur rouge plus intense indique un score d'attention plus élevé]

Les modèles basés sur les Transformers fournissent des scores d'attention qui peuvent améliorer l'interprétabilité de nos modèles de prédiction en visualisant les relations entre

les pixels utilisés pour générer les prévisions. Cette sous-section se concentre sur l’interprétabilité du modèle *channel vision transformer* basé sur les scores d’attention calculés avec l’équation 3.10 pour les 4 variables d’entrée. Notez que ce modèle a été choisi à la place du modèle U-transformer car (i) il obtient des résultats similaires au modèle U-transformer et a un encodage de patch de taille 1 (ce qui permet de visualiser facilement les scores d’attention); (ii) il utilise une approche par canal qui nous permet de déterminer le score d’attention pour chaque variable. Nous calculons les scores d’attention pour chaque pixel dans $\hat{I}^p(t)$ et pour toutes les têtes de l’encodeur. Le score d’attention est une valeur comprise dans la plage $[0, 1]$ qui indique la pertinence de l’information apportés par un pixel pour en prédire un second (le pixel cible). Une attention élevée est indiquée par un score proche de 1 (c’est-à-dire représentée en rouge dans la Figure 4.7) et implique une forte relation entre ces deux pixels, tandis qu’un score d’attention proche de 0 indique que ce pixels n’a pas apportés d’information.

La Figure 4.7 présente les scores d’attention calculés pour un pixel (en vert) dans $\hat{I}^p(t)$ par rapport aux autres pixels. La Figure 4.7 (a) montre l’emplacement du pixel cible (pixel vert) dans $\hat{I}^p(t)$. Nous avons limité la présentation à un seul pixel et à la tête ayant des scores d’attention élevés (informations pertinentes) ou des corrélations entre les différents pixels dans les images sélectionnées. La Figure 4.7 (b) représente les scores d’attention mesurés entre le pixel vert et tous les autres pixels pour quatre têtes dans l’encodeur transformer pour le canal de charge de passager. Les Figures 4.7 (c) et (d) montrent la même représentation pour les canaux de temps de parcours et des intervalles de départ. Une tête est une partie de l’architecture qui capture un contexte spécifique. La tête est activée si la situation qu’elle cherche à détecter est présente dans une image ou entre deux pixels. Ce comportement est illustré dans la Figure 4.7 (b).

Dans les images présentées, les scores d’attention basés sur le canal de charge dans la Figure 4.7 (b) montrent que le modèle transformer se concentre sur les départs de train les plus récents (par exemple, Head0) et les stations voisines de la ligne. De plus, les scores d’attention indiquent les trains corrélés avec le pixel actuel, comme Head0 ou Head4. Dans Head0, nous observons que certaines attentions ne sont pas corrélées spatialement avec les départs de train voisins. Certains pixels à l’extrémité de la ligne ont des scores d’attention significatifs. Le modèle recherche des informations soit temporellement et spatialement proches du pixel vert, soit des départs de train qui se produisent plus loin le long de la ligne. De plus, certaines têtes (par exemple, Head1 ou Head5 dans la Figure 4.7 (b)) n’ont montré aucun score d’attention significatif, ce qui implique qu’il n’y avait aucune corrélation entre les pixels et l’image qui a activé la tête correspondante. Pour le canal de temps de parcours représenté dans la Figure 4.7 (c), l’accent est davantage mis sur la position du pixel par rapport aux futurs départs de train que sur les trajets passés ou les départs passés. Cette approche aide le modèle à obtenir des informations sur le profil du trafic (*i.e.* le nombre de pixels manquants peut donner des informations sur l’horodatage de l’image). Pour le canal d’intervalle de départ (voir la Figure 4.7 (d)), les têtes 1 et 5 montrent que l’attention se porte principalement sur les départs de train passés pour obtenir le contexte de l’image de trafic du métro.

4.4 Conclusion

Dans ce chapitre, nous avons utilisé le cadre de la ligne 9 afin de tester le protocole de prédiction par inpainting et les différentes architectures de réseau de neurones introduits dans le Chapitre 3. Les modèles ont été comparés en considérant plusieurs architectures de

deep learning convolutionnel et d'attention et en ayant comme baseline un modèle naïf de diffusion sur plus de 3 ans de données. Premièrement, malgré les contraintes techniques, le U-Transformer présente de bien meilleurs résultats en prédiction que la baseline. Le modèle U-net quant à lui surpasse l'ensemble des modèles pour la tâche d'inpainting. Il offre une prédiction moyen terme plus robuste que ces concurrents construits sur la base de l'attention. Ces résultats ont mis en avant la pertinence du format image pour traiter l'irrégularité des départs de trains ainsi que la gestion de différents contextes de charges. On tire également profit d'une base de test et d'apprentissage construite sur les 3 ans de données permettant de couvrir un large panel de contexte et situation de charge. Les modèles présentent les meilleures performances pour les images en journée y compris pendant les deux heures de pointe de la ligne. Par contre, ce sont les images en début et fin de service qui présentent les moins bons résultats en terme de WMAPE, cette remarque est à moduler par le fait que ces créneaux présentent généralement des charges assez faibles et présentent des enjeux applicatifs moindres. Les modèles montrent également des résultats plus faibles en début de ligne là où nous disposons de peu d'informations concernant la dynamique des trains. De plus, on a pu voir qu'il existe des régions de l'image mal prédites par les modèles mais que celles-ci représentent une portion faible des données, ces situations atypiques montrent les faiblesses des modèles utilisés dans ce chapitre. On peut alors se poser la question sur l'évaluation des modèles de prédiction. L'analyse présentée jusqu'alors ne met pas en valeur les divers contextes pouvant impacter la charge (incident, grève, confinement). Une analyse robuste des modèles de prédiction doit être étendue aux situations atypiques afin de juger de manière précise les forces et faiblesse de tels modèles pour la prévision de charges, ce qui sera l'objet du chapitre suivant.

Chapitre 5

Labellisation d'images et amélioration des prédictions en situations atypiques

5.1 Introduction

Le vrai challenge de la prévision de charge d'un réseau ou d'une ligne de transport public ne réside pas uniquement dans les performances globales car les opérateurs possèdent une bonne connaissance des dynamiques nominales de charge résultant d'une offre adaptée (heure de pointe, heure creuse) et de telles performances de prédiction renseignent peu sur les capacités de généralisation des modèles dans des cas atypiques. Le challenge réside bien dans la prédiction des cas spécifiques du réseau (introduits dans le chapitre 1) qui sont définis par leur récurrence et la possibilité de les anticiper ou non (prévisible ou non). A noter que la plupart des modèles de prédiction fournissent une prédiction agrégée ne mettant pas en relief la réalité des dynamiques d'une ligne ou d'un réseau de transport public. Dans le cas de la région parisienne, l'apparition de nouvelles lignes de transport avec le grand Paris et la tenue d'événements spécifiques tels que les JO ou la coupe du monde de rugby va influencer et modifier les dynamiques des charge au sein du réseau de transport géré par l'opérateur RATP. Ce réseau déjà impacté par le télétravail et la crise du COVID présente des contextes de charge nouveaux ou atypiques dans les données historisées.

La prédiction doit tenir compte de situations non prévues et observées dans les données. Ces situations que l'on considère comme atypiques représentent le réel enjeu de prédiction de charge car rares et difficiles à anticiper pour certaines. Pour évaluer de façon robuste les modèles dans ce contexte applicatif, il est important d'effectuer une analyse différenciée des performances, qui consiste à évaluer les modèles pour la prédiction de ces cas atypiques en plus d'une évaluation globale.

Se pose alors la question de la gestion de ces données atypiques dans la modélisation. La première possibilité consiste à améliorer les modèles dans ces cas atypiques en équilibrant la base de données ou en construisant des modèles spécialisés dans ces rares situations. Ceci demande de les détecter en amont via une classification des contextes [47] de la ligne. Une autre possibilité est de fournir une prédiction ne se réduisant pas à l'inférence d'une valeur de charge mais en intégrant une incertitude sur les prévisions [18].

Nous proposons ici de traiter les situations atypiques en trois temps. Premièrement, il s'agit d'extraire les cas atypiques présent dans notre base d'apprentissage et de les

grouper par catégories d’images et de contexte. Dans un second temps, nous évaluerons les modèles au regard des différents contextes atypiques précédemment identifiés. Ces cas atypiques sont estimés de différentes manières, à partir de statistiques simples calculées sur chacune des images ou au travers de l’analyse de l’espace latent de modèle de prédiction déjà développé. Cette catégorisation va nous permettre de mieux connaître les forces et les faiblesses des modèles proposés dans cette thèse. Enfin, l’objectif est d’améliorer les performances des modèles via diverses méthodes, sur-échantillonnage, génération d’images, afin de faire progresser les modèles pour les contextes de transport les moins bien prédits.

5.1.1 Protocole pour la gestion des cas atypiques

La démarche que nous proposons pour traiter les cas atypiques se décompose en trois phases :

1. Labellisation d’images et détection des groupes atypiques. Cette première étape a pour but de créer des sous ensembles de données relatives à certains contextes de la ligne. L’objectif est d’extraire des groupes d’images rares ou pouvant décrire des situations à anticiper dans une ligne de métro.
2. Évaluation des performances de prédiction en situations atypiques. Cette analyse différenciée et fine des résultats nous permet de mettre en lumière les faiblesses des modèles ainsi que les situations atypiques pour lesquelles les erreurs de prédiction sont importantes. Cette évaluation est à la fois pertinente pour connaître la généralisation du modèle mais aussi pour garantir la fiabilité des prédictions, tant pour les gestionnaires de ligne que pour la planification de trajets.
3. Amélioration des modèles pour les groupes identifiés. A travers des mécanismes comme l’augmentation des données ou le sur-échantillonnage, on cherche à compenser la faible représentation des images atypiques dans la base d’apprentissage et améliorer ainsi leur prédiction.

Ainsi avant toute chose, il est nécessaire d’étiqueter les images selon des critères opérationnels et de performance. En computer vision, labelliser ou partitionner des images en groupes homogènes (clustering) est une tâche courante. Elle permet de rendre semi-automatique la tâche de labellisation qui face à la quantité importante d’images générées (issue de divers mediums : réseau sociaux, caméras de sécurité), ne peut plus être seulement manuelle. L’automatisation du partitionnement et de la labellisation d’images est rendue possible grâce à des modèles de classification d’images n’ayant pas recours à la supervision humaine [60]. Ce type de méthode se base sur des approches de deep clustering [59] qui ont montré une supériorité significative par rapport aux algorithmes de clustering traditionnels. Initialement, le clustering d’images repose sur des approches non-surveillées ou semi-surveillées via des auto-encodeurs et l’analyse de leur espace latent par des méthodes de clustering traditionnelles [46] comme l’algorithme des K-means, la classification ascendante hiérarchique ou l’analyse spectrale. On peut également ne pas labelliser les images mais caractériser les features des images à partir d’approches non supervisées [13] afin d’obtenir une labellisation plus fine (par exemple par rapport à l’offre d’une part et la demande d’autre part).

Cependant, il s’agit ici d’images spatio-temporelles s’étalant sur plus de trois ans de données. Avant d’utiliser des méthodes avancées, il est nécessaire d’utiliser les connaissances externes (date, évènements) sur nos images ou encore de chercher des comportements opérationnels considérés comme atypiques (charges de trains ou temps de parcours extrêmes).

5.2 Données calendaires et évènements atypiques

Depuis 2019, la région parisienne a connue plusieurs évènements ayant eu un fort impact sur la fréquentation de la ligne du métro 9 comme la grève de décembre 2019 ou les divers confinements relatifs à la crise du Covid. On retrouve également sur la ligne 9, l'accès au Parc des Princes (stade de foot), les Jardins d'Auteuil avec Roland Garros ou encore la Place de la République avec la station éponyme où la plupart des manifestations parisiennes ont lieu, tout ces contextes et évènements différents ont un impact sur les images de trafic de la ligne.

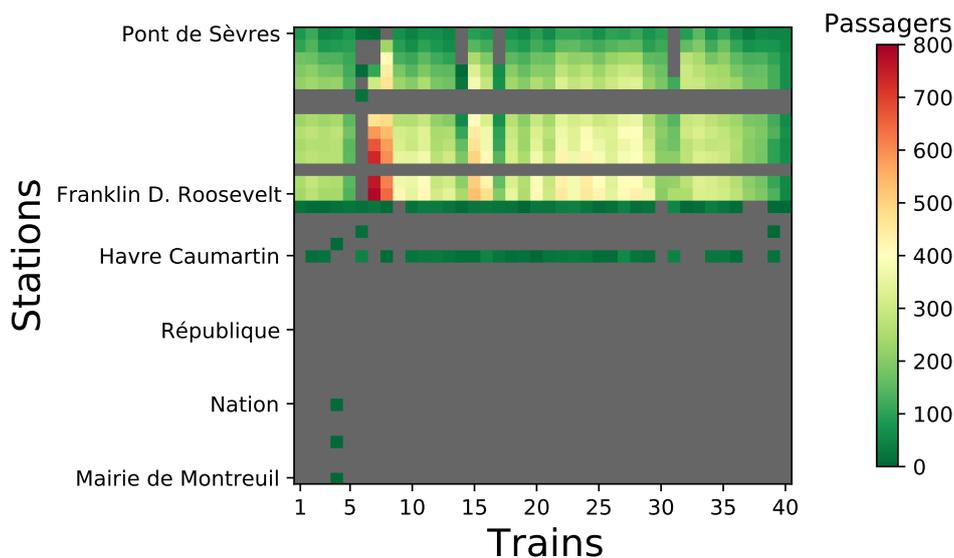


FIGURE 5.1 – Image de Grève

Une manière triviale de labelliser les données consiste à sélectionner les dates et heures où ont lieu les évènements particuliers. Cette approche a été utilisée pour construire deux groupes d'images : les images de grèves et celles de confinement. On les nomme respectivement *Grève* et *Confinement*. La figure 5.1 dépeint une image de la grève de décembre 2019 où l'on observe que seulement les premières stations sont ouvertes en répercussion du mouvement social. L'avantage de la profondeur des données dont nous disposons est l'accès à des évènements s'étendant sur des périodes supérieures à 1 mois de données. On peut également cité pour un critère temporelle les évènements récurrents comme les matchs de foot du parc des Princes mais nous les traiterons plus tard via les espaces latents des modèles de prévision afin de s'intéresser au type de comportement de la charge qui n'est pas limité à ces jours précis.

5.3 Détection statistique des cas atypiques

Il est également possible de chercher au sein de la base de données des images ayant des statistiques descriptives bien spécifiques relevant d'une situation atypique. Ces variables peuvent être construites selon divers critères allant de l'analyse des charges individuelles de l'image aux temps de parcours totaux des trains. Cette section montre l'extraction de groupes atypiques selon des critères à la fois statistiques en étudiant la distribution de nos images et selon des critères issus des différentes variables du système. On peut penser

aux images ayant un parcours total plus long que la moyenne ou un nombre important de pixels manquants. De même, la recherche de "patterns" relatives à des observations des opérateurs de transport permet d'extraire des situations atypiques non observées avec les méthodes précédentes (i.e. alternance de trains faiblement et fortement chargés en heure de pointe, par exemple).

5.3.1 Statistiques de charges

Dans un premier lieu, il est évident de s'intéresser aux caractéristiques des images selon la charge qui est la première variable d'intérêt. Deux catégories d'images seront mises en valeur à propos de la charge des trains, les contextes de fortes charges et ceux correspondant à de fortes transitions de charges. La première catégorie a pour but de considérer la charge comme un critère a posteriori de situations atypiques. Une forte charge serait la conséquence d'évènements rares sur la ligne. Cette catégorie peut regrouper des images lors des pics d'affluence de la ligne en cherchant en particulier les situations extrêmes. La seconde catégorie concerne les fortes variations de charge qui vise à détecter les successions de trains vide et plein. Ce critère met en lumière une mauvaise gestion de la demande par l'offre de transport qui impacte la qualité à bord des trains.

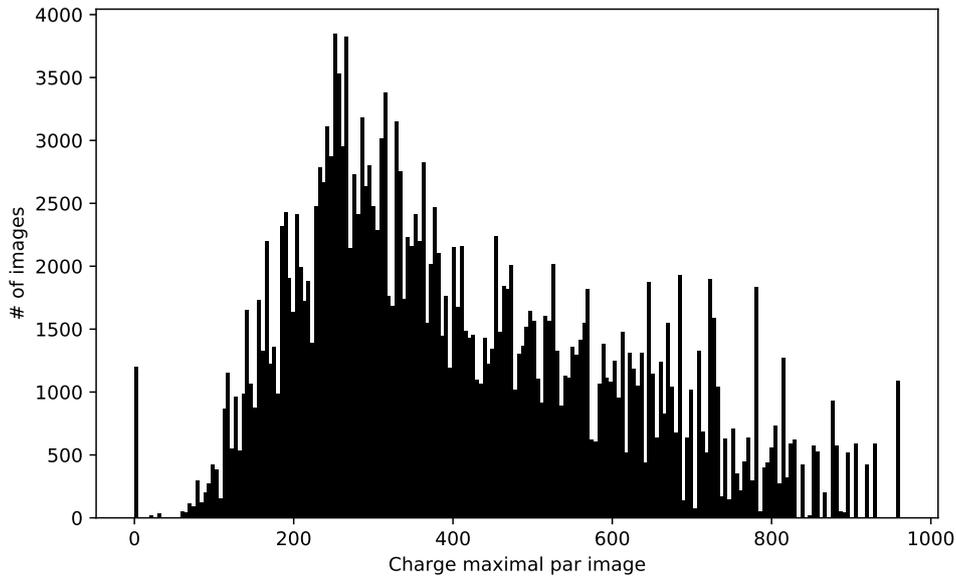


FIGURE 5.2 – Distribution de la charge maximale notée cm

Afin de sélectionner un set d'image pertinent, nous nous intéressons à la distribution des images selon différents critères et métriques (médiane de la charge par image, maximum de la charge cumulé par train, ou station, variation maximale de charge). La figure 5.2 montre la distribution de la charge maximale à bord des trains dans une image.

$$cm = \max_{s,c}(x_{sc}),$$

avec x le canal de charge de l'image, c le train et s la station. D'autres métriques comme la médiane ou la moyenne auraient pu être utilisées mais le fait de traiter les images par pixel permet de détecter à la fois les contextes de congestion du réseau et les fortes charges ponctuelles sur quelques trains dans l'image. Un seuil est appliqué ensuite pour extraire les images avec une charge maximale d'au moins 650 passagers par train.

Ce groupe nommé *Forte charge* a ainsi pour but de détecter les profils d'image avec au moins un train présentant des volumes importants de passagers sur la ligne.

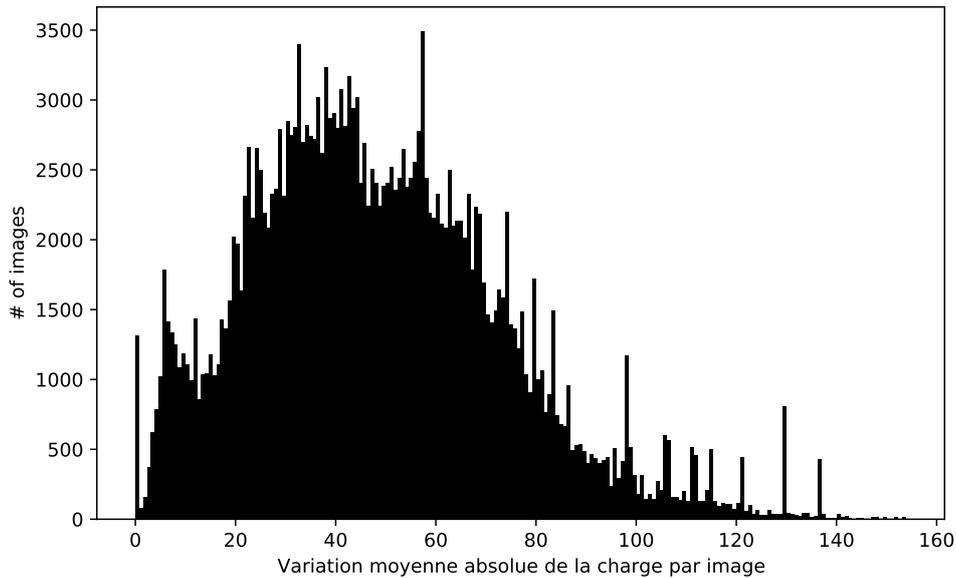


FIGURE 5.3 – Distribution de la variation absolue moyenne de la charge entre deux départs de train vc .

Pour la seconde catégorie, la figure 5.3 présente la distribution des images selon les variations entre la charge d'un train et celle du précédent train à la même station, en valeur absolue et moyennées sur l'ensemble de l'image :

$$vc = \frac{1}{|S|(|C_t| - 1)} \sum_{s,c} |x_{s,c} - x_{s,c-1}|,$$

avec x le canal de charge de l'image, c le train et s la station, S l'ensemble ordonnées des stations et C_t , l'ensemble des trains de l'image à l'instant t . On cherche à détecter les successions entre un train avec une forte charge et ceux avec une faible charge. D'un point de vu de computer vision, il s'agit de détecter les contours d'objet dans une image. De manière similaire au précédent groupe, on considère comme atypique les images avec un différence de charge moyenne entre deux trains supérieure à 80 passagers. Nous nommons le groupe *variation*.

Les figures 5.4 et 5.5 sont deux images de chacun des groupes atypiques respectivement pour les *Forte Charge* et *Variation*. On observe dans l'image "forte charge" un pic d'affluence sur l'ensemble de la ligne pour les trains 21 à 24 puis une surcharge dans les trains en amont de la station Franklin Roosevelt. Le pic d'affluence est issu d'une charge anormale en début de ligne qui se répercute sur le reste de la ligne. La seconde image présente un taux de charge plus faible que la précédente mais on y observe une alternance entre des trains faiblement et fortement chargés.

Ainsi, à partir des images, nous créons deux jeux de données atypiques relatifs à deux contexte de charges spécifiques. Il faut toutefois préciser qu'une image peut appartenir aux deux groupes. Une image peut en effet avoir une charge maximale supérieure à 650 passagers par train et présenter une alternance de train chargé et non chargé conduisant à sa détection par le critère de *Variation*.

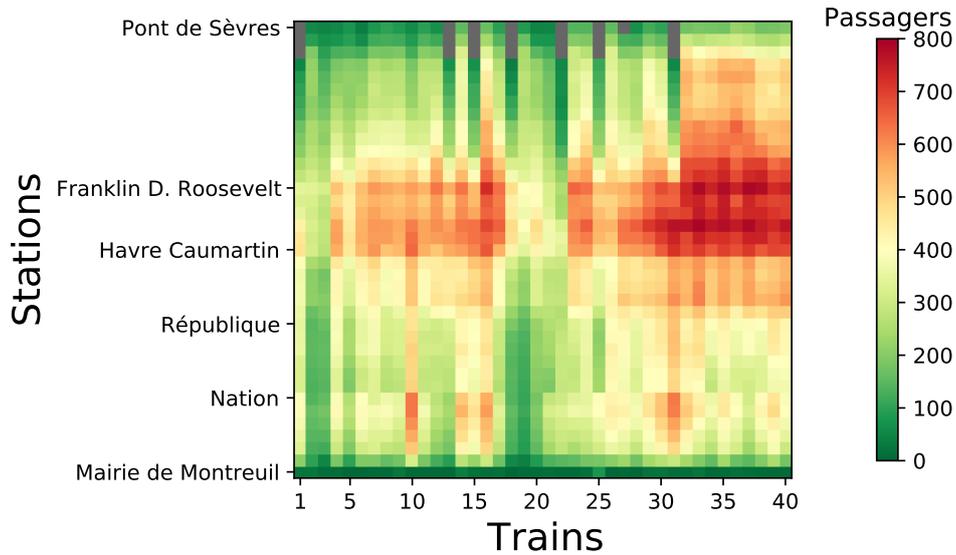


FIGURE 5.4 – Image de forte Charge

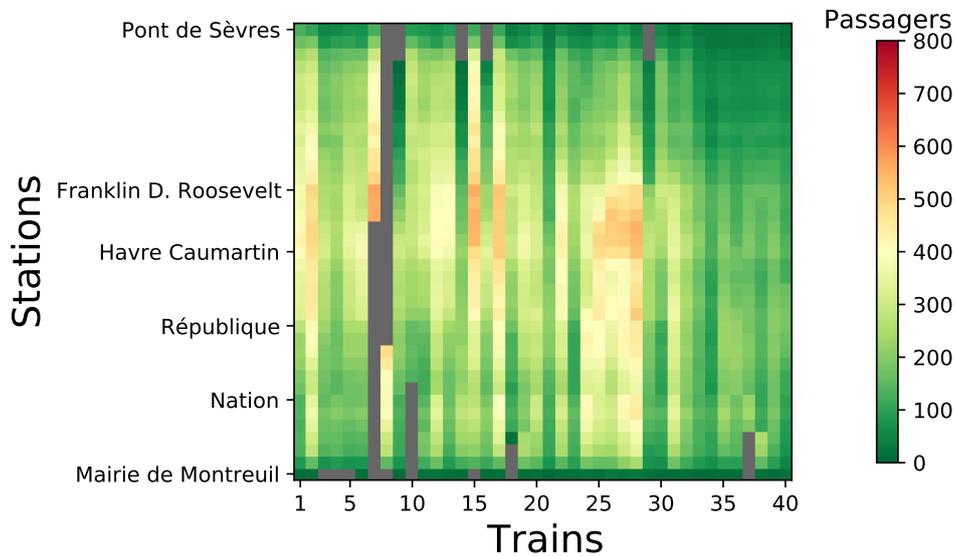


FIGURE 5.5 – Image du groupe atypique *Variation*

5.3.2 Statistiques sur les temps de parcours

La même approche est construite pour les composantes temporelles de l'image. Le but est de mettre en évidence les images avec des temps de parcours ou d'inter-stations importants qui, comme pour le cas de la charge, peuvent indiquer une situation atypique. Nous nous concentrons ici sur les temps de parcours. La figure 5.6 montre la distribution des images selon le temps de parcours maximal par image tpm :

$$tpm = \max_c \sum_s x_{c,s},$$

avec x le canal de temps de parcours de l'image, c le train et s la station.

Le but est de considérer les trains ayant un temps de parcours nettement supérieur au temps de parcours moyen de la ligne (52 minutes). On note sur la figure 5.6 que la grande

majorité des temps de parcours de la ligne sont inférieurs à 100 minutes. Les valeurs extrêmes supérieures à 200 minutes représentent la plupart du temps des images en début de journée à l'ouverture du service (mise en service des trains et déplacement des trains entre les terminus). Ces images sont exclues car elles représentent des effets de bord qui ne sont pas pertinents pour la prévision de charge. Les images atypiques sélectionnées que l'on nomme *Délais*, correspondent à des images dans lesquelles au moins un train possède un temps de parcours supérieur à 90 minutes.

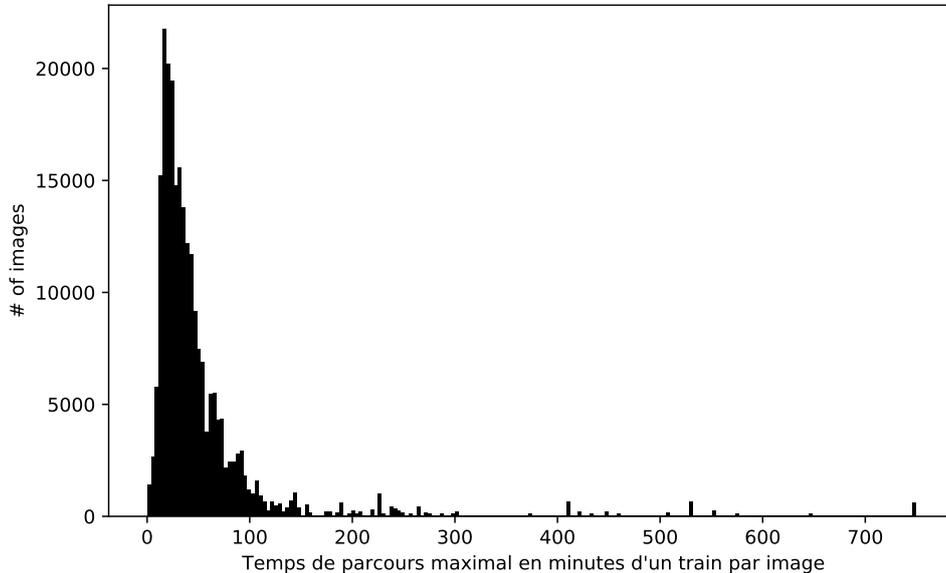


FIGURE 5.6 – Distribution des temps de parcours maximaux par image

Les forts retards détectés grâce à cette métrique se traduisent visuellement dans nos images sous la forme "d'escaliers" comme dans la figure 5.7 où un évènement en fin de ligne pour le train 14 a impacté l'ensemble des trains compris entre ce dernier et le trains numéro 30. Ces pixels représentent les stations où les trains ont stationné en attente de la reprise du trafic.

5.3.3 Pixels manquants

Une dernière information issue de l'usage spécifique des images par les modèles prédictifs peut être utilisée ; il s'agit du nombre de pixels manquants dans les images d'entrée et à prédire (qui varie en fonction de la fréquence des trains). Pour rappel, une donnée manquante traduite comme un pixel mort indique soit une erreur de mesure ou bien un train ne faisant pas d'arrêt à cette station. Nous engloberons sous l'appellation "pixels inconnus" l'ensemble de ces pixels.

L'objectif est d'extraire les images atypiques en tenant compte du nombre de pixels inconnus tout en ciblant les perturbations importantes de la ligne (fermeture partielle de la ligne, station fermée). Les images de grève seront exclues du groupe pour éviter une redondance avec le groupe déjà sélectionné précédemment. La Figure 5.8 représente une image comprise dans les images du groupe atypique *Pixels* où l'on note la fermeture de 8 stations entre République et Franklin Roosevelt.

La figure 5.9 présente la distribution des images selon le nombre de pixels inconnus. On retrouve trois pics vers 300, 450 et 650 pixels inconnus qui représentent respectivement les

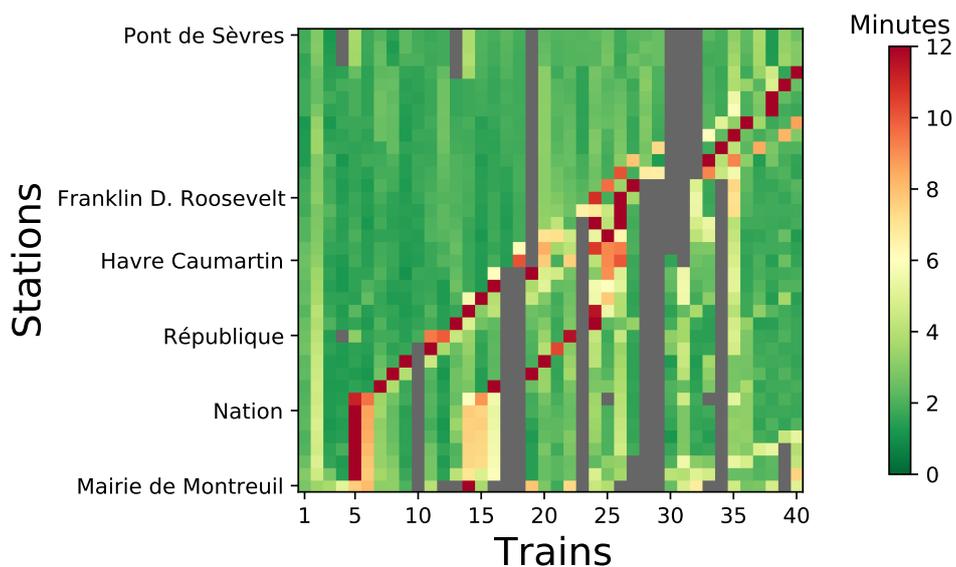


FIGURE 5.7 – Image des temps de parcours lors de forts délais

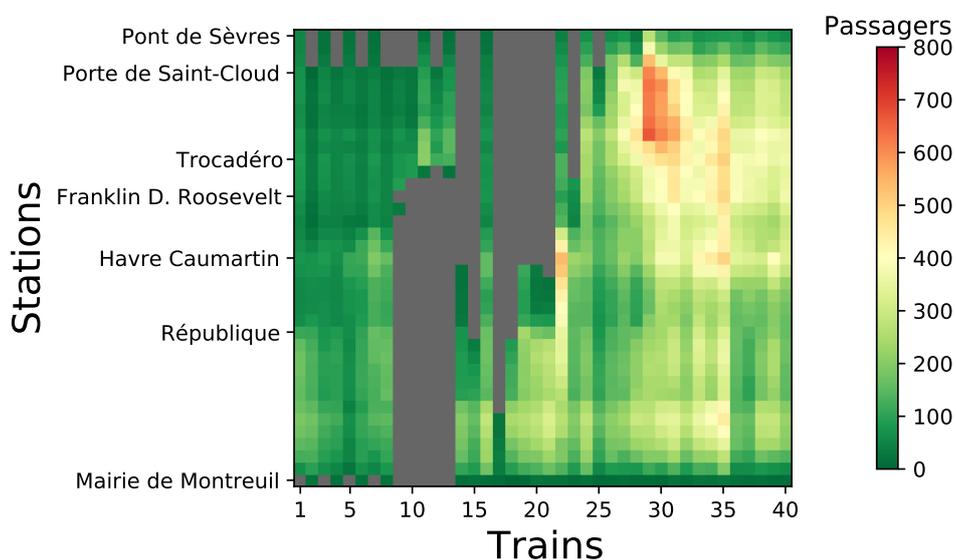


FIGURE 5.8 – Image avec une forte proportion de données manquantes

images de week-end, d'heures creuses et d'heures de pointe du réseau. Ce constat fait écho à la figure 4.4a du chapitre 4 où l'on a observé deux zones de prédiction liées au heures de pointe et aux heures creuses. Nous considérerons comme atypiques les images ayant plus de 850 pixels manquants dans l'image d'entrée des modèles. Il s'agit principalement d'images d'heures de pointe avec des contextes spécifiques (stations fermées, perturbations). Nous appellerons ce groupe *Pixels*.

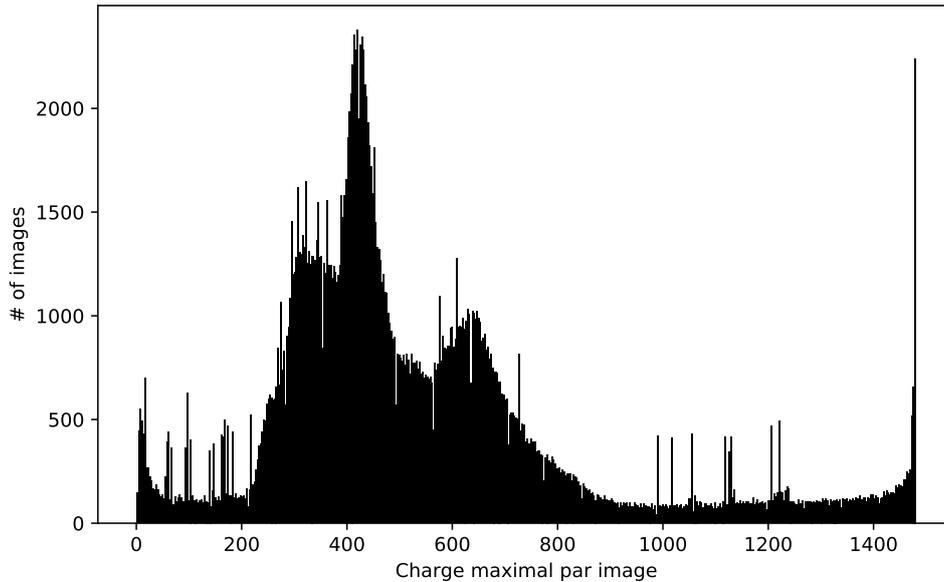


FIGURE 5.9

5.4 Exploration et annotation semi-automatique de l'espace latent

Une dernière approche pour la détection d'images atypiques repose sur l'usage de techniques issues de la computer vision pour la labéllisation [3] ou le regroupement automatique en clusters des images [58, 1]. L'objectif est de projeter nos images dans un espace latent afin de déterminer des classes sous-jacentes non détectées par les deux premières méthodes de filtrage mises en place. Habituellement, ce type d'approche utilise des architectures de deep learning nommées auto-encodeur [70]. Il s'agit de modèle de réduction de dimension qui projettent dans un espace dit latent les caractéristiques d'une image d'entrée afin de la reconstruire uniquement à partir de l'espace latent. Cette projection est équivalente à construire une représentation des données de taille réduite qui nous permet dans un second temps de faciliter la caractérisation des images [70].

En nous appuyant sur les travaux des chapitres précédents, il semble naturel de tirer profit des modèles U-net qui peuvent s'apparenter à des auto-encodeur où les bottleneck sont considérés comme leur espace latent. Cette représentation des données via l'espace latent des modèles de prévision permet d'une part d'extraire des groupes d'images atypiques et d'autre part d'interpréter le modèle et les choix du modèle par rapport à la prédiction. La figure 5.11 propose une première visualisation de l'espace latent selon l'heure de la journée. La projection t-distributed stochastic neighbor embedding (t-SNE) [35] est utilisée ici pour réduire l'espace latent en un plan plus facile à visualiser. La méthode t-SNE sert uniquement pour la visualisation.

Dans la figure 5.10, nous retrouvons une composante principale qui représente le cas nominal pour les images. Il s'agit de l'ensemble des points situés au centre de la figure. Ce groupe d'images est organisé selon l'heure de la journée qui est donc retrouvée par le modèle. En effet, seuls les délais entre les trains et stations sont fournis au modèle. Il reconstruit l'heure de la journée à partir des indicateurs d'offre et de demande dans l'image d'entrée. Les informations pour chaque année sont indiquées dans une couleur différente.

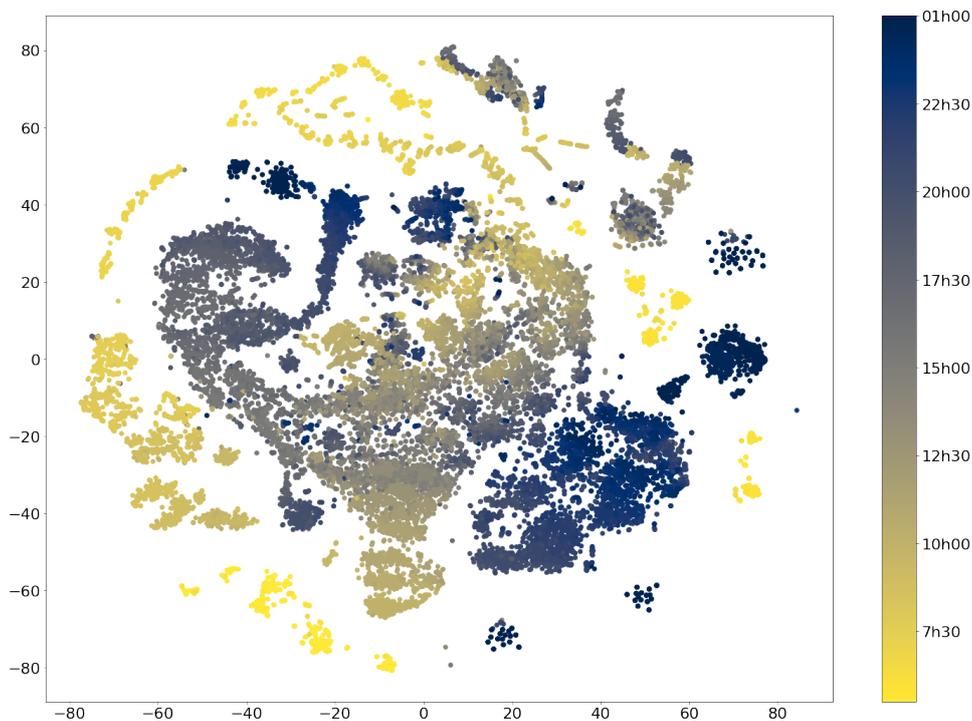


FIGURE 5.10 – Représentation t-SNE de l’espace latent du modèle U-transformer, les couleurs encodent l’heure de la journée

5.4.0.1 Clustering de l’espace latent

A partir de l’espace latent, nous cherchons à extraire et détecter des sous-ensembles de groupes atypiques. Ces groupes doivent présenter des situations et contextes non détectés à l’aide des précédentes approches. L’espace latent se compose de 512 features map et est issu du modèle de prédiction ne permettant pas de réduire la dimensionalité de l’espace contrairement à des méthode d’auto-encodeur. Pour ce faire, nous appliquons l’algorithme de clustering k-means pour partitionner l’espace latent en un ensemble de groupes. Le choix de k-means est fait de manière à pouvoir être facilement implémenté. De plus, il permet de décomposer les images faisant partie du cas nominal créant par la même occasion des groupes par tranche horaire. De manière empirique, nous avons choisi 20 groupes. Ce nombre de groupe permet d’avoir suffisamment de groupes nominaux tout en détectant les groupes non-nominaux de l’espace latent.

Les groupes se situant sur les périphéries de la figure représentent des contextes de charge spécifiques à la ligne neuf ne faisant pas partie du cas nominal telles que les perturbations et les grèves. On dénote également les contextes de charge spécifiques au format image comme les débuts et fin de service qui se trouvent toujours en périphérie du cas nominal. On retrouve également des groupes d’images précédemment extraits comme le confinement et la grève comme indiqué grâce aux labels sur la figure 5.11.

L’objectif est alors de labelliser les groupes atypiques qui n’ont pas été extraits de manière supervisé à l’aide des traitements définies dans la section précédente. Nous avons exclu les clusters *Confinement* et *Grève* qui sont visibles dans cet espace latent. Enfin, trois groupes ont été sélectionnés, le groupe 0, le groupe 13 et le groupe 19, représentant des situations atypiques d’un point de vue opérationnel, soit environ 10 % des données. Les Figures 5.12, 5.13 et 5.14 illustrent une image de chacun des groupes *événements Correspondances* et *perturbation*. Le groupe 19, désigné par *Correspondances*, désigne une

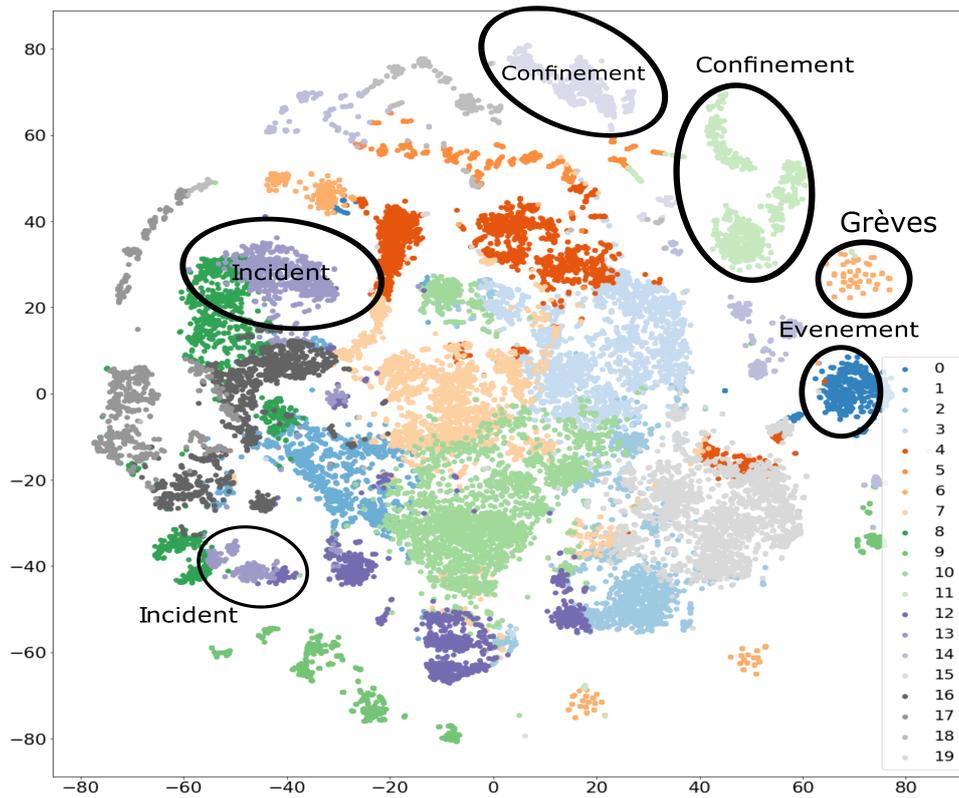


FIGURE 5.11 – Représentation t-SNE de l’espace latent de la charge de passagers correspondant au modèle U-Transformer. Des couleurs différentes distinguent les 20 classes obtenues à partir du regroupement par K-means, des labels manuels permettent de situer les différents groupes extraits.

correspondance des passagers d’autres lignes de métro vers la ligne actuelle généralement à partir de la station Trocadéro (voir 1.2). Les images de Correspondances représente des images où l’on note des lignes horizontales dans l’image comme pour les stations Franklin D. Roosevelt et Havre Caumartin de la figure 5.13. Les charges varient selon le contexte de charge. Ces lignes sont observées uniquement pour les stations avec correspondance. Le nombre de passagers augmente chaque fois que les trains passent par une station de correspondance. Ce groupe contient donc a priori des images ou des perturbation sur d’autres lignes qui génèrent une affluence anormale sur une partie de la ligne 9. Le groupe 13, appelé *perturbation*, représente les images perturbées ou les images avec des gares fermées. Dans la figure 5.14 où des trains (16 et 39) ont effectué un demi-tour sur la ligne. Cette action fait écho à des choix opérationnels pour traiter les fortes charges passagers du réseau. Le groupe 0, appelé *événements*, représente les images avec des charges élevées au début de la ligne correspondant à une affluence induite par un événement (par exemple, les événements sportifs). L’image 5.12 de l’évènement représente la charge en sortie d’un match de foot du Parc des princes. On note que la charge se concentre sur un nombre réduit de trains entre les stations Porte de Saint-Cloud et Havre-Caumartin (Saint Lazare). L’algorithme k-means entraîné sur les données de la base d’apprentissage a ensuite été utilisé avec les images de l’ensemble de test pour labelliser celles-ci.

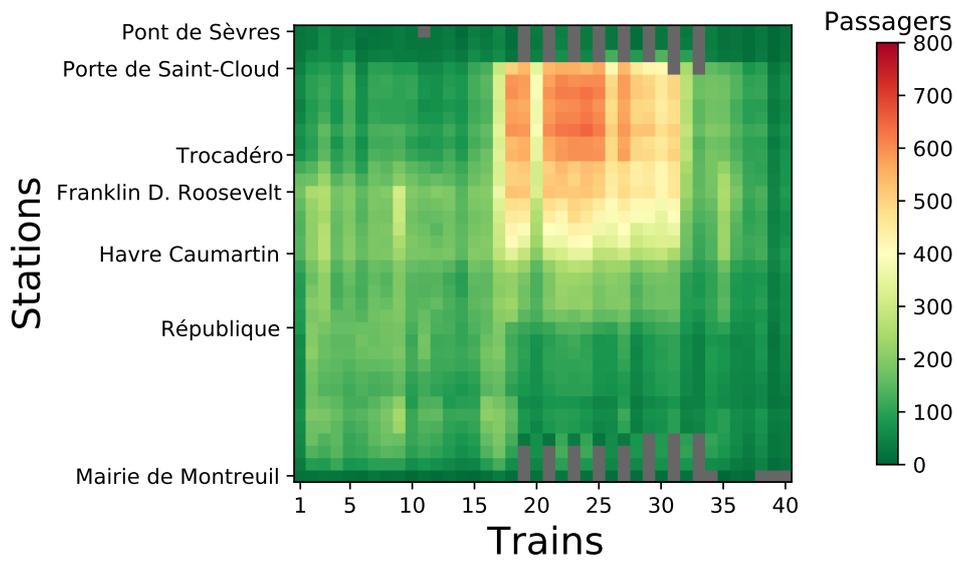


FIGURE 5.12 – Image issues du groupe *Evènements*

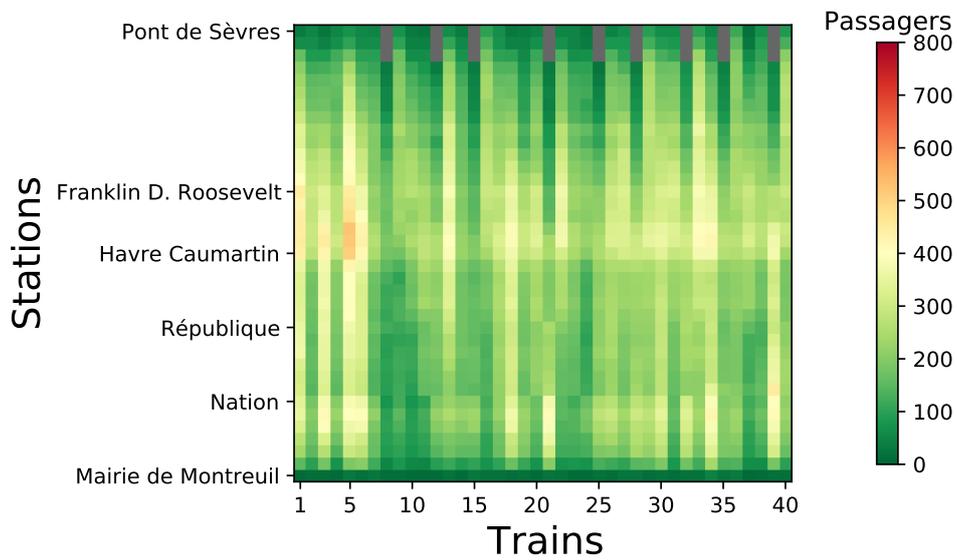


FIGURE 5.13 – Image issues du groupe *Correspondances*

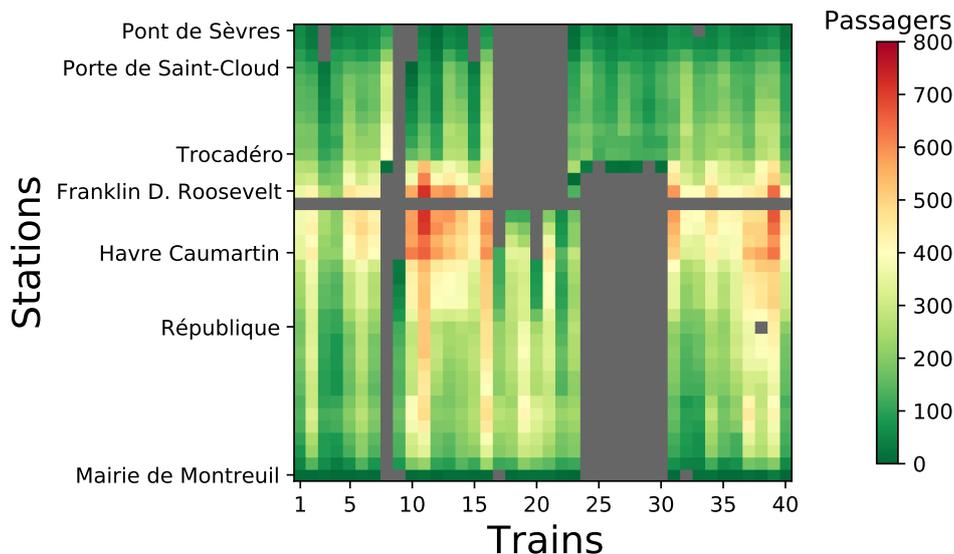


FIGURE 5.14 – Image issues du groupe *Perturbations*

5.4.1 Outliers au sein des groupes nominaux

Une dernière spécificité de cet espace latent est relative aux groupes nominaux issus du clustering où l'on retrouve des outliers pouvant être attribués aux diverses classes atypiques déjà présentées. Ces images représentent des cas atypiques observés sur le réseau mais ayant un plus faible impact sur la charge. On note en particulier la présence d'image en forme "escalier" où l'ensemble des trains de la ligne a du marquer un arrêt plus long dans les différentes stations et des images avec des charges ponctuelles comme dans la figure 5.7.

Nous créons ainsi un dernier groupe de situations atypiques nommé *Outliers* qui englobe les outliers des groupes nominaux de l'image. Pour automatiser le processus, les images atypiques sont extraites à l'aide de la méthode Local Outlier Factor (LOF)¹ qui calcule l'écart de densité locale d'un point par rapport à ses voisins. Les images ayant une faible densité locale dans l'espace latent sont considérées comme atypiques. Il est important d'appliquer cette méthodologie groupe par groupe car sinon cette méthode détecterait des images pour la plupart déjà considérées comme atypiques lors de l'étape de clustering.

5.5 Impact des différentes situations sur les performances

Pour comprendre les enjeux liés aux situations atypiques au regard de la prédiction, il est intéressant de comparer la représentativité de chaque groupe dans chacune des bases d'apprentissage et de test. En effet, sur plus de trois ans de données, l'occurrence de cas atypiques sur la ligne peut varier et changer entre les données utilisées pour la construction du modèle et les données cibles de test. Par contre, il se peut qu'une même image soit dénombrée plusieurs fois. En effet, certaines situations atypiques peuvent être une accumulation de contextes divers dans la période de temps définie de l'image. Ainsi

1. La LOF utilise une contamination de 0.01 par groupe et un nombre de voisin de 40

on retrouve des images de grève à la fois dans les situations de fortes charges et de retard (délai).

Le tableau 5.1 présente pour chacune des catégories le nombre d’images et leur proportion dans la base associée. La première observation pouvant être faite est la proportion supérieure des groupes atypiques dans la base de test. Seules les situations de Correspondances sont moins courantes dans la base de test. Les plus grandes différences sont observées pour les groupes *fortes charges* et *Confinements*. Toutefois, concernant les *Confinements*, il faut préciser que les bases de test et celle d’apprentissage car elles ne définissent pas les mêmes confinements. La région Parisienne a connu trois confinements au totale entre 2020 et 2022 dont les dynamiques de charge étaient différentes (cf figure 1.1). Les contraintes sanitaires ayant évolué entre chaque confinement, le nombre d’usagers des transports en commun est passé d’une charge presque nulle pour le confinement de mars à avril 2020 à un trafic plus élevé et fortement dégradé pour les deux confinements suivants.

Les situations atypiques les plus rares tant en apprentissage qu’en test sont les forts retards *i.e.* constitués des groupes *Délais*, *Grèves* et *Outliers* avec des taux inférieurs à 1%. Concernant les outliers et les images de retard, leur faible représentation est due au choix quant à leur sélection : le seuil pour les images de retard et le pourcentage d’outlier choisi pour les groupes nominaux. Pour les grèves, les données présentent l’avantage de couvrir la grève de décembre 2019 mais de nombreux jours sont manquants sur le mois car aucun train ne circulait.

TABLE 5.1 – Nombre et proportion des images de groupes atypiques à la fois pour la base d’apprentissage et la base de test.

| Cas atypiques | Apprentissage | | Test | | Prévu | Récurrent |
|------------------------|---------------|------|----------|------|-------|-----------|
| | # images | % | # images | % | | |
| Base complète | 705600 | 100 | 235200 | 100 | | |
| Base nominale | 531994 | 75 | 152880 | 65 | ✓ | ✓ |
| <i>Fortes Charges</i> | 57504 | 8.1 | 32723 | 14.3 | ✓ | ✓ |
| <i>Correspondances</i> | 56844 | 7.9 | 15587 | 6.8 | ✓ | ✓ |
| <i>Évènements</i> | 22139 | 3.1 | 14775 | 6.4 | ✓ | ✓ |
| <i>Confinements</i> | 50400 | 7.0 | 36000 | 15.7 | ✓ | |
| <i>Grèves</i> | 8400 | 1.1 | 9600 | 4.2 | ✓ | |
| <i>Variations</i> | 45638 | 6.4 | 22119 | 9.7 | | |
| <i>Perturbations</i> | 26247 | 3.6 | 6896 | 3.0 | | |
| <i>Retards</i> | 4596 | 0.6 | 2294 | 1.0 | | |
| <i>Pixels</i> | 74242 | 10.5 | 27115 | 12.1 | | |
| <i>Outliers</i> | 4641 | 0.6 | 1520 | 0.6 | | |

5.5.1 Résultats

Pour résumer, nous sommes en possession de 10 groupes de situations atypiques qui caractérisent divers contextes de trafic ferroviaire.

La Table 5.2 résume les résultats des prévision obtenus sur l’ensemble des groupes atypiques précédemment présentés. Les performances de prévision sont représentées par la métrique WMAPE. Le modèle U-Transformer présente les meilleurs résultats pour la plupart des cas de test, à l’exception de deux ensembles de données : les, *Perturbations*

TABLE 5.2 – Résultats de la prévision de la charge de passagers sur la base du pourcentage d’erreur absolue moyenne pondérée (WMAPE) pour l’ensemble des modèles et les situations atypiques.

| Base | Naïf | NN | CNN | CNN+NN | VIT-3 | U-net | U-Trans | C-Vit |
|------------------------|-------------|------|------|--------|-------|-------------|-------------|-------|
| Test | 19.9 | 18.9 | 13.1 | 14.8 | 12.3 | 12.3 | 11.4 | 12.1 |
| <i>Délais</i> | 27.3 | 28.4 | 24.2 | 24.2 | 20.3 | 19.3 | 19.1 | 19.9 |
| <i>Fortes Charges</i> | 20.1 | 20.1 | 13.0 | 14.3 | 12.1 | 11.6 | 11.2 | 12 |
| <i>Variations</i> | 20.3 | 23.1 | 19 | 14 | 13.5 | 12.5 | 12.8 | 13.4 |
| <i>Pixels</i> | 31.2 | 31.7 | 29.8 | 26.8 | 22.9 | 21.4 | 21.5 | 21.9 |
| <i>Grèves</i> | 38.2 | 40.2 | 32.2 | 32.4 | 31.1 | 28.1 | 29.6 | 30.0 |
| <i>Confinements</i> | 21.1 | 26.8 | 18.6 | 25.9 | 18.2 | 16.8 | 13.7 | 14.2 |
| <i>Correspondances</i> | 28.9 | 28.5 | 21.4 | 25.9 | 20.9 | 19.5 | 18.4 | 19.5 |
| <i>Perturbations</i> | 29.8 | 60.5 | 34.2 | 40.2 | 52.8 | 30.4 | 58.9 | 51.6 |
| <i>Événements</i> | 22.3 | 18.7 | 14.6 | 15.6 | 13.1 | 13.9 | 12.5 | 13.1 |
| <i>Outliers</i> | 42 | 34 | 19.1 | 20.1 | 21.6 | 17.6 | 24.5 | 22.8 |

et les *Grèves*. Pour les images *Perturbations*, l’approche naïve obtient la meilleure performance, avec une valeur WMAPE de 29.8%, suivie par U-net, alors que les transformers sont beaucoup plus impactés par ce type d’image avec des valeurs WMAPE supérieures à 50%. Pour l’ensemble de données *Grèves*, U-net présente la meilleure performance, avec une différence de 1% par rapport aux deux meilleurs modèles suivants. Ces deux contextes de trafic (*Perturbations* et *Grèves*) présentent les pires résultats parmi tous les cas atypiques. Ils représentent des scénarios dans lesquels le trafic est fortement impacté par des événements externes tels que la fermeture de gares, l’arrêt prématuré de trains ou des interruptions de trafic. En ce qui concerne les autres scénarios atypiques, les ensembles *Fortes Charges*, *Confinement* et *Événement* sont les mieux prédits, avec des erreurs proches de celles obtenues pour l’ensemble de test global. Il est important de souligner que *Fortes Charges* et *Confinements* sont deux cas atypiques qui présentent des conditions de circulation opposées. Le premier cas comprend des images de trains surchargés, tandis que *Confinements* se compose principalement d’images de trains très peu chargés. Il convient de noter que l’erreur de prédiction n’augmente pas proportionnellement au nombre de passagers, mais dépend davantage du contexte du trafic, comme pour les ensembles *Perturbations* et *Grèves* mentionnés précédemment. Enfin, de bons résultats de prévision sont obtenus pour des contextes atypiques tels que *Événements* (c’est-à-dire des manifestations sportives se déroulant à proximité de plusieurs gares).

Pour les deux situations atypiques restantes, *Délais* et *Correspondances*, les performances en matière de prévision sont inférieures à celles de l’ensemble de test global, avec des valeurs WMAPE de 19.1% et 18.4% pour le modèle de U-transformer. Pour le contexte *Délais*, les images sont considérées comme contenant un trafic perturbé lorsque des régulations ont été utilisées pour gérer les flux de passagers. Ces situations sont plus difficiles à prévoir car elles impliquent une gestion plus complexe du trafic, des retards et un nombre élevé de passagers. Pour les images *Correspondances*, l’erreur peut être expliquée par le manque d’information sur les correspondances : les informations sur les correspondances des passagers entre deux lignes ne sont pas mesurées par la billetterie et le modèle dispose donc de peu d’informations pour anticiper l’affluence de passagers provenant de tels flux à la station Trocadéro par exemple. Nous touchons ici à l’une des limitations de l’approche par ligne de métro que nous proposons. L’image d’une seule ligne ne prend pas en compte

les transferts de passagers d'autres lignes.

Enfin, la tenue d'évènements de portée internationale pour la fin d'année 2023 avec la coupe du monde de rugby d'une part et les Jeux Olympique d'été en 2024 dans la capitale vont enrichir cette analyse en offrant de nouvelles situations et contextes pour l'heure non observés dans les données. Ainsi, les bonnes performances dans le contexte de fortes charges peut être mis au crédit des modèles que nous proposons, ces images sont en effet, parmi les cas atypiques, celles qui sont le mieux prédites.

5.6 Augmentation de données et sur-échantillonnage

L'évaluation des modèles par classe d'images atypiques permet de juger si une approche, un modèle, présente de meilleurs résultats dans certains contextes d'offre ou de demande. Elle nous permet de déterminer les faiblesses des modèles de prédiction et ainsi d'en tenir compte lors de leur usage pour les passagers et opérateur. Cette évaluation différenciée constitue une métrique de la capacité de généralisation des modèles sur des cas opérationnels concrets. On peut alors se demander si nous ne pouvons pas chercher à faire progresser les modèles pour les situations les plus difficiles ou des situations spécifiques choisies par des opérateurs.

Les résultats mis en valeur dans ce chapitre montrent que les classes sous représentées peuvent être sujettes à des performances de prédiction inférieures à celles du cas nominal. Ce phénomène s'accroît avec les situations atypiques extrêmes comme la grève et les fortes perturbations car le trafic est alors difficile à prédire.

Une possibilité pour améliorer les performances des modèles sur ce type de situation, peut consister à sur-représenter les classes d'images atypiques rares lors de l'apprentissage. Toutefois, il n'est pas garanti que le sur-échantillonnage des images rare soit suffisant pour améliorer les modèles du fait du décalage entre l'apprentissage et le test.

Une seconde approche, peut être envisagée dans l'esprit des techniques de data-augmentation couramment utilisée en traitement d'image. En vision par ordinateur, différentes techniques (regroupées sous le terme de data-augmentation) sont utilisées pour élargir la base de données d'images, afin d'aider les modèles à gérer différents sujets ou objets. L'augmentation artificielle des images d'entrée est principalement utilisée, car contrairement à l'homme, les modèles d'apprentissage profonds ne sont pas capables d'abstraire les caractéristiques d'un objet ou sa position dans une image. Les modèles d'apprentissage parviennent à extraire ces informations en multipliant les images, en variant les points de vue et en effectuant des changements de couleur par exemple.

Selon les tâches de vision par ordinateur, différentes méthodes peuvent être utilisées. Dans le cas de la classification d'images, il est courant d'appliquer des rotations, des déformations et des transpositions aux images afin de créer plusieurs variations à partir d'une seule image. En revanche, pour la détection d'objets, afin de permettre aux modèles de traiter des objets à différentes échelles ou positions dans l'image, l'apprentissage est réalisé en créant des images artificielles qui combinent différents objets et les placent dans diverses positions dans l'image. L'objectif est de reproduire des associations d'objets ou de formes qui peuvent ne pas être présentes initialement dans la base de données d'entraînement.

Dans le cadre de ces travaux, il ne s'agit pas à proprement parler d'image, ainsi certaines méthodes de data-augmentation comme des rotations, transpositions d'image n'ont pas de sens pour les parcours de trains. Les axes de l'image, stations et trains sont fixes et communs à toutes les images. Il est toutefois possible de créer des images pseudo-aléatoires

en conservant un des deux axes de l'image fixe. Il s'agirait de mélanger soit les stations soit les trains afin de garder les diverses dynamiques liées à l'un des deux axes. Nous choisissons ici de traiter les trains et de garder l'axes des stations fixes pour conserver les dépendances spatiales de l'image.

Ainsi nous nous retrouvons avec deux axes possibles pour améliorer les modèles dans le cadre des situations atypiques :

- Sur-échantillonnage d'image atypique
- Data-augmentation par introduction d'images pseudo-aléatoires

Le but in-fine est d'améliorer la généralisation des modèles de prédiction dans le cadre de contextes de transport nominaux et atypiques. Cependant, concernant les situations atypiques, la principale difficulté est le décalage entre les bases de test et d'apprentissage, c'est à dire que les situations observées dans la base d'apprentissage ne sont pas complètement représentatives des cas atypiques de la base de test. Comme mentionné dans le tableau 5.1, les différentes classes peuvent être sous ou sur représentées en test. De même des cas atypiques peuvent apparaître en test et ne pas être détectés comme tels par nos différents critères.

Avant de mettre en oeuvre un sur-échantillonnage des images, plusieurs approches associées à de la data-augmentation ont été implémentées afin de tenter de générer de nouvelles images atypiques. Le premier est l'usage de GAN pour chacune des classes atypiques. L'idée était de générer des images possibles de charge pour les diverses situations atypiques et ainsi augmenter la nombre d'image par classes à l'instar de traduction d'image à image [37]. La seconde méthode est similaire au méthodologie de style et contenu (content) de traitement d'image qui ont pour but de décomposer l'image selon son style (par exemple, réaliste, impressionniste) et son contenu (par exemple un train, un vélo) ou de cross-domain disentanglement [30]. Le style et le contenu sont perçus ici comme l'offre et la demande². L'objectif de cette décomposition par offre et demande aurait été de traduire la demande d'une première image pour l'offre d'une seconde image. A l'instar des modèles de style content, il s'agissait de créer des images directement à partir de la base d'apprentissage. Cependant ces méthodes n'ont pas offert de résultats concluants dans le cadre de l'amélioration des performance de prévision. Nous tenions à mentionner l'usage de ce type d'approche permis par l'usage d'image.

Ainsi il s'agira de discuter de la prévision de charge dans les cas atypiques et de la robustesse des modèles et des processus de généralisation.

5.6.1 Sur-échantillonnage des cas atypiques

Le début de ce chapitre a mis en évidence la relative faiblesse des modèles de prédiction en réponse à certaines situations atypiques. Pour faire face à ce constat de la représentation de certains groupes d'image, il est possible de ré-équilibrer notre base d'apprentissage pour que les différents groupes nominaux et atypiques soient présents dans des proportions semblables. La première méthodologie mise en place est le sur-échantillonnage d'image afin de sur représenter les situations atypiques par rapport aux situations nominales.

Pour procéder au sur-échantillonnage, deux stratégies sont possibles :

- Un équilibrage des classes d'images tant atypiques que nominales
- Un sur-échantillonnage simple des classes atypiques sans tenir compte de leur représentativité dans la base de données

2. Cette approche avait été aussi un candidat pour la création d'espace latent pour la labellisation d'image

La première approche cherche à construire une base équilibrée entre les images nominales et celles atypiques dont on souhaite améliorer les performances. Il s’agit de mettre sur un pied d’égalité les images atypiques en procédant à un sur-échantillonnage des groupes sous représentés et de sous-échantillonner les images les plus représentées, principalement nominales. Dans le second cas, nous créons deux bases d’apprentissage, une base complète comprenant toutes les images et une seconde base uniquement construite à partir des cas atypiques précédemment définis qui sera alors sur-échantillonnée. La base d’images atypiques est issue des groupes présents dans le tableau 5.2. Mais ces images peuvent varier selon les métriques et clusters choisis pour la sélection des images atypiques. Le but est d’observer les performances de prédiction selon la représentation de ces données dans la base d’apprentissage dans diverses configurations.

5.6.1.1 Expérimentation

Pour tester cette approche, le choix est fait d’utiliser uniquement les modèles U-net car moins coûteux que les modèles construits avec de l’attention. Nous ne cherchons pas ici à obtenir les meilleures performances mais à évaluer s’il est possible d’améliorer les performances des modèles pour les différentes situations atypiques extraites.

La première expérience tente de construire une base d’apprentissage équilibrée entre d’une part la base nominale et d’autre part les situations atypiques cibles. Il s’agira ici de s’intéresser aux images du groupe *Perturbations* et du groupe *Pixels* qui présentent parmi les situations atypiques les moins bien prédites (respectivement 30.4 et 21.4). Elles présentent également l’avantage d’être fortement représentées avec respectivement 8% et 10% des images. Pour chaque groupe, nous utilisons un nombre fixe d’image fixé à partir du groupes atypiques le plus représenté, c’est à dire le groupe *Pixels*. Ainsi, la base est construite en sous-échantillonnant la base nominale et en sur-échantillonnant la base de perturbation. Dans le cadre du sous-échantillonnage de la base nominale, de par l’écart du nombre d’image avec les bases atypiques (plus de 500000 contre environ 75000 pour les images avec des données manquantes), nous proposons soit d’avoir un tirage aléatoire d’image changeant d’une époque à une autre au cours de l’apprentissage soit d’avoir un tirage fixe d’image avant l’apprentissage avec par conséquent une base d’image nominales constantes pour toutes les époques. Concernant le sur-échantillonnage, nous ne testerons pas cette approche sur les groupes très peu représentés car le risque de sur-apprentissage des modèles pour les cas atypiques cibles nous paraît bien trop important.

Pour l’approche par simple sur-échantillonnage, il s’agira de sur-représenter l’ensemble des cas atypiques ou un sous ensemble sans équilibrer les classes. On entraîne plusieurs modèles avec des taux d’images atypiques différents. Nous choisissons d’utiliser un facteur multiplicatif qui représentera le nombre de fois où une image atypique est présente lors de l’apprentissage. De plus, pour chaque configuration de données, plusieurs modèles U-net sont entraînés afin d’avoir plusieurs tirages par sur-échantillonnage (au minimum 3 par configuration). Il s’agit de réduire les effets liés à l’aléatoire utilisée lors de l’apprentissage qui pourrait conduire à des résultats non-robustes. De plus, les différents modèles auront au total le même nombre d’itérations et d’images par itération pour ne pas favoriser les modèles avec des facteurs multiplicatifs élevés (à partir de 10, le nombre d’image totale peut dépasser le million d’images). Deux jeux de données atypiques seront utilisés pour le sur-échantillonnage, le premier utilisera uniquement les groupes atypiques les moins bien prédits alors que le second sur-échantillonnera toutes les classes d’images atypiques.

5.6.1.2 Résultats pour les classes équilibrées

Cette section présente les résultats des modèles lorsque la base d'apprentissage est construite en équilibrant les classes d'images comme présenté précédemment. Le tableau 5.3 présente les résultats pour les deux stratégies d'équilibrage (avec tirage aléatoire ou tirage fixe des images nominales). On note tout d'abord qu'équilibrer les classes ne permet pas une amélioration notable des performances globales ce qui semble logique. De plus, seul le modèle avec un sous-échantillonnage aléatoire à chaque époque obtient des performance proches du modèle initial. Ce phénomène est principalement du au nombre d'image composant la base nominale dont on extrait seulement 15 % à chaque époque de l'apprentissage.

Pour les cas atypiques, on note en premier lieu que seule deux classes sont légèrement améliorées : les images des groupes *Perturbations* et *Correspondances* avec une amélioration autour de 2% en WMAPE.

A contrario, nous pouvons également observer que les images du groupe *Données manquantes* qui a été utilisé pour la création de la base voit au contraire ses performances se dégrader avec une baisse de plus de 5%. Le sur-échantillonnage de ses images très particulier semble donc avoir conduit à un sur-apprentissage, et les performance en généralisation ne sont donc pas bonnes.

Les cas atypiques qui n'ont pas été sur-échantillonnés voient leurs performances diminuer. Pour les images à forte charge, on note une perte de 4 % en WMAPE. Ce constat est le même avec différentes configurations quant aux classes sélectionnées et au nombre d'image par classes. La méthodologie utilisée ici ne permet pas d'améliorer toutes les situations atypiques voulues mais en plus elle dégrade les classes déjà correctement prédites par le modèle comme la classe *Fortes Charges*.

TABLE 5.3 – Résultats des modèles par groupes d'images atypiques en équilibrant le nombre d'images par classes

| Base | U-net | Base équilibrée | Base équilibrée par époque |
|------------------------|-------------|-----------------|----------------------------|
| Base complète | 11.4 | 19.3 | 13.7 |
| <i>Fortes Charges</i> | 12.0 | 23.5 | 16.4 |
| <i>Variations</i> | 12.8 | 21.4 | 15.2 |
| <i>Délais</i> | 19.1 | 25.2 | 20.1 |
| <i>Pixels</i> | 21.5 | 28.7 | 26.6 |
| <i>Grèves</i> | 28.1 | 38.9 | 38.2 |
| <i>Correspondances</i> | 18.4 | 19.9 | 17.5 |
| <i>Événements</i> | 7.3 | 17.4 | 8.8 |
| <i>Perturbations</i> | 30.4 | 36.9 | 28.7 |
| <i>Outliers</i> | 17.6 | 23.3 | 18.5 |

5.6.1.3 Résultats par sur-échantillonnage

La seconde expérimentation a pour but de proposer un sur-échantillonnage des situations atypiques ayant les performances de prédiction les plus faibles (*Outliers*, *Perturbation*, *Délais*). Les autres situations atypiques ne sont pas sur-échantillonnées et sont donc présentes dans leurs proportions initiales dans la base de données d'apprentissage . L'objectif est d'améliorer les prédictions dans les contextes de charge les plus complexes. Le Tableau 5.4 présente les résultats pour chaque jeu d'images atypiques. La base de test

TABLE 5.4 – Résultats des modèles par groupes d’images atypiques selon le sur-échantillonnage des images de *Outliers*, *Perturbations*, *Délais*

| Base | ×1 | ×2 | ×3 | ×5 | ×10 | ×15 | ×20 |
|----------------------|-------------|------|-------------|-------------|-------------|-------------|------|
| Base complète | 11.4 | 12.3 | 12.7 | 12.0 | 12.1 | 11.7 | 11.9 |
| <i>Variations</i> | 12.8 | 12.5 | 12.8 | 11.9 | 12.1 | 12.1 | 12.4 |
| <i>Délais</i> | 19.1 | 17.6 | 17.7 | 16.8 | 17.0 | 18.1 | 18.8 |
| <i>Pixels</i> | 21.5 | 22.2 | 23.1 | 21.1 | 21.6 | 21.5 | 22.2 |
| <i>Grèves</i> | 28.1 | 34.3 | 34.1 | 35.0 | 34.0 | 32.2 | 34.8 |
| <i>Confinements</i> | 13.7 | 14.5 | 15.3 | 14.0 | 14.7 | 13.7 | 13.8 |
| <i>Perturbations</i> | 30.4 | 24.4 | 23.3 | 25.8 | 25.1 | 25.2 | 27.0 |
| <i>Outliers</i> | 17.6 | 16.7 | 16.8 | 15.9 | 15.7 | 16.0 | 16.5 |

est la même pour chacun des facteur multiplicatif. Ce dernier n’ayant un impact que pour l’apprentissage des modèles. Certains groupes (*Evènements* et *Fortes charges* par exemple) sont omis dans le tableau car on ne note pas de changement significatif par rapport au U-net initial. Pour les situations ciblées par le sur-échantillonnage, on observe un gain pour 3 situations : *Outliers*, *Perturbations*, *Délais*. Par contre, les images de type *Grèves* sont moins bien prédites par les modèles. On note un gain minime de performances pour les autres cas atypiques comme *Evènements*, *Variations* et *Correspondances*, ces dernières n’ayant pas fait l’objet de sur-échantillonnage pour la configuration présentée.

De plus, on observe dans le tableau 5.4, que les meilleurs performances sont obtenues avec des facteurs multiplicateurs différents. Ainsi, le sur-échantillonnage ne permet pas de garantir les meilleures performances pour un facteur multiplicatif donné. Dans le cadre de l’expérimentation, le facteur multiplicatif de 5 semble être un compromis acceptable. Il améliore la majorité des classes cibles hormis les grèves sans dégrader de manière importante les autres.

Le Tableau 5.5 représente la troisième expérimentation de sur-échantillonnage en utilisant toutes les situations atypiques pour le sur-échantillonnage. Dans cette configuration, aucun gain notable n’est observé pour les différentes classes. Seulement trois situations atypiques voient leurs performances s’améliorer avec un facteur multiplicateur de 2 : *Délais*, *Correspondances* et *Outlier*. Comme pour le tableau précédent certaines situations atypiques sont omises, car on ne note pas de changement significatif par rapport au modèle U-net initial.

TABLE 5.5 – Résultats des modèles par groupes d’images atypiques selon le sur-échantillonnage de toutes les images atypiques

| Base | ×1 | ×2 | ×3 | ×5 | ×10 | ×15 | ×20 |
|------------------------|-------------|-------------|------|------|------|------|------|
| Base complète | 11.4 | 12.3 | 12.6 | 12.8 | 13.2 | 13.0 | 13.2 |
| <i>Fortes Charges</i> | 12.0 | 14.2 | 14.4 | 14.5 | 14.7 | 15.2 | 14.8 |
| <i>Délais</i> | 19.1 | 18.6 | 19.2 | 19.8 | 20.5 | 19.8 | 20.2 |
| <i>Confinement</i> | 13.7 | 14.4 | 14.7 | 14.9 | 15.0 | 15.3 | 15.4 |
| <i>Correspondances</i> | 18.4 | 14.9 | 15.0 | 15.1 | 15.4 | 15.6 | 15.7 |
| <i>Perturbations</i> | 30.4 | 28.4 | 31.1 | 29.7 | 29.4 | 30.4 | 30.2 |

A ce stade, on note que le choix des classes atypiques à sur-échantillonner peut induire des résultats et des gains différents comme le montrent les tableaux 5.4 et 5.5. De même, on ne peut pas fixer de facteur multiplicatif optimal pour les performances des modèles. Nous

avons présenté ici seulement deux configurations mais ces conclusions restent valables pour d'autres configurations (sélections des groupes atypiques à sur-échantillonner). Nous pouvons cependant noter que contrairement à la première méthodologie qui proposait d'équilibrer la base d'apprentissage selon les différentes classes d'images, les performances globales des modèles sur l'intégralité de la base de test restent proches de la baseline (modèle U-net).

Le sur-échantillonnage peut être utilisé pour améliorer les performances des modèles de prédiction sur certaines classes cibles mais les résultats ne semblent pas complètement robustes comme nous avons pu le voir à travers les expérimentations. Un gain en performance est possible mais dépend de divers facteurs en commençant par la sélection des images atypiques. Il semble en particulier qu'il est difficile d'améliorer certaines classes avec une telle approche comme la classe *Grèves*, pour laquelle le sur-échantillonnage semble conduire presque systématiquement à un sur-apprentissage du fait des différences importantes entre les images du jeu de données d'apprentissage et de test dans ce groupe.

Enfin, le choix de limiter les paramètres d'apprentissage (batch, nombre d'époch ou encore modélisations) pour tous les modèles peut également impacter les résultats de chaque sur-échantillonnage.

Une alternative serait de cibler des situations atypiques à améliorer plutôt que de vouloir procéder de manière globale sur l'ensemble des cas atypiques. Une autre approche serait donc non pas d'améliorer les modèles mais de construire à partir de notre labellisation des modèles spécifiques pour chaque cas atypique ou des familles de situations atypiques. Cette démarche suppose que l'on soit capable de :

- détecter : anticiper les situations atypiques
- labelliser : classifier les images atypiques
- prédire : sélectionner les modèles adaptés à la situation détectée.

Cette démarche nous semble plus coûteuse en terme de développement et de maintenance (suivi des performances des modèles dans le temps) et n'a pas donc pas été évaluée dans le cadre de cette thèse.

5.6.2 Génération d'images pseudo-aléatoires

Une autre possibilité étudiée dans le contexte de ces travaux concerne l'enrichissement de la base de données à l'aide d'images obtenues par pseudo-randomisation. Une image pseudo-aléatoire est une image réaliste du trafic ferroviaire construite à partir d'images réelles. Comme première approche, nous avons simplement permuté les parcours de trains des images d'entrée. La figure 5.15 illustre pour une image donnée 5.15a, une variante de cette image 5.15b. Il s'agit de permuter des départs de trains dans une fenêtre temporelle réduite. Il n'est pas pertinent de mélanger des trains de début de journée avec des trains de fin de journée. La permutation est bien sûr la même entre les différentes variables de l'image. Nous souhaitons conserver la dynamique de chaque train. Bien que ce procédé ne conserve pas l'ordre réel des trains de l'image et donc l'évolution de la charge entre les trains à une station donnée, il nous paraissait tout de même être intéressant en introduisant une sorte régularisation à la manière d'un drop out. Dans le même esprit, d'autres approches de perturbation telle que la suppression aléatoire de certains pixels auraient pu être testées.

L'idée est de créer des profils de charge réalistes ne tenant pas forcément compte des tendances de la dynamique de charge de la ligne. En effet, l'ordre des trains de l'image ne respecte pas l'offre et la demande initiale de l'image mais conserve toutefois les dyna-

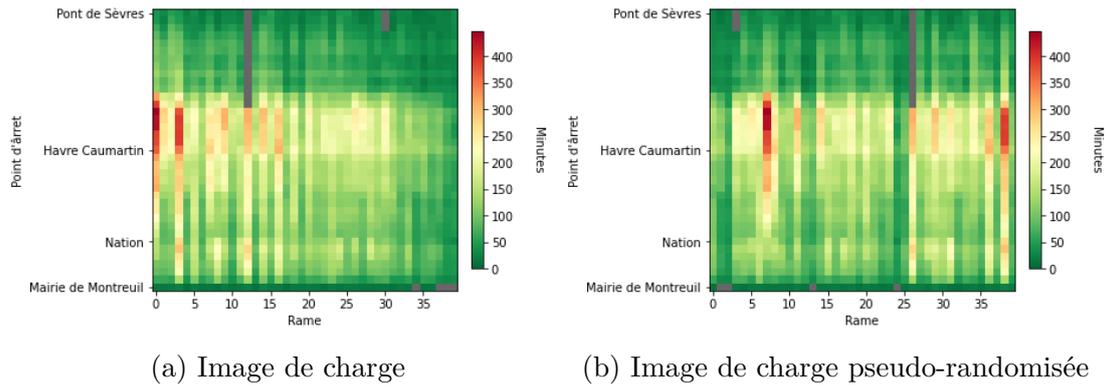


FIGURE 5.15 – Exemple d’une image de charge se variante pseudo-randomisée

miques intrinsèque des trains. Cette approche est rendue possible par l’échantillonnage des données par les trains et par le format image d’autre part.

5.6.2.1 Expérimentation

Pour tester l’impact de l’utilisation des images pseudo-aléatoires sur les performances des modèles, on établit un modèle selon le pourcentage d’image généré dans la base d’apprentissage. Nous entraînons un modèle U-net comme introduit dans la section 3. Il s’agit alors de construire la base d’apprentissage avec 50% d’images pseudo-aléatoires. Le modèle sera nommé U-net 50%. Nous nous limitons à 50% d’images générées car au delà aucun gain significatif de performances en situations atypiques n’est observé.

La génération d’images pseudo-aléatoires est faite pour chaque batch de données. On complète chaque batch d’images générées soit à partir du batch actuel soit avec des images d’un autre batch de données. La génération des images pseudo-aléatoires est effectuée à la volée. Ainsi pour une même image, plusieurs images pseudo-aléatoires peuvent être créées. Enfin, nous utiliserons un modèle U-net pour mener à bien l’apprentissage afin d’avoir une taille de batch suffisante pour mélanger les images pseudo aléatoires et réelles.

5.6.2.2 Résultats

Le tableau 5.6 présente le comparatifs entre le modèle U-net, le modèle U-Transformer et un modèle U-net utilisant 50% d’images pseudo aléatoires. On constate que le modèle utilisant des images pseudo-aléatoires présente des résultats proches, même si légèrement moins bons globalement, de ceux obtenus par les modèles du benchmark avec 12.8% de WMAPE sur le jeu de test.

Dans le détail, nous notons deux grandes différences entre les modèles avec et sans images pseudo-aléatoire dans le cadre des situations atypiques. La première concerne les situations atypiques impactant l’offre de transport où le modèle U-net 50% surpasse les autres modèles. Les groupes : *Délais* , *Outliers* ou *Perturbations* sont mieux prédits avec l’usage d’images pseudo-aléatoires avec en particulier un gain de 4% pour les images de *Délais* .

Au contraire, dans le cadre de demandes atypiques, le modèle U-net 50% est moins performant que les modèles entraînés normalement. Ce phénomène est clairement visible avec les images de *Fortes Charges* avec 6 % de différences. Toutefois, les images du groupe atypiques *Correspondances* sont mieux prédites par le U-net 50% .

TABLE 5.6 – Comparaison des performances pour les cas atypiques entre les modèles U-net, U-Transformer et Unet 50% (Unet avec une base d’apprentissage comprenant 50% d’image généré)

| Base de test | U-net | U-transformer | U-net 50% |
|---------------------------|-------------|---------------|-------------|
| Base complète | 12.3 | 11.4 | 12.8 |
| <i>Délais</i> | 19.3 | 19.1 | 15.9 |
| <i>Fortes charges</i> | 11.6 | 11.2 | 17.1 |
| <i>Variations</i> | 12.5 | 12.8 | 14.6 |
| <i>Données manquantes</i> | 21.4 | 21.5 | 22.5 |
| <i>Grèves</i> | 28.1 | 29.6 | 38.5 |
| <i>Confinement</i> | 16.8 | 13.8 | 14.4 |
| <i>Correspondances</i> | 19.5 | 18.4 | 8.0 |
| <i>Perturbations</i> | 30.4 | 58.9 | 28.0 |
| <i>Événements</i> | 13.9 | 12.5 | 15.3 |
| <i>Outliers</i> | 17.6 | 24.5 | 16.5 |

L’usage d’images pseudo-aléatoires permet d’améliorer les performances des modèles pour les cas atypiques relatifs à l’offre. Concernant les situations atypiques liés à la charge et à la demande, le fait de ne pas tenir compte de l’évolution de la demande en station et de se concentrer uniquement sur la dynamique des trains impacte les performances du modèle. Avec une telle approche, on se retrouve à spécialiser le modèle pour un sous ensemble de cas atypiques comme les images avec un fort retard.

Ainsi, nous avons vu qu’il est possible d’améliorer les prédictions des charges par images en utilisant des techniques de data-augmentation ici classique. Il s’agit de tester s’il est possible d’utiliser la toolbox de la computer vision dans le cadre de la prévision de charge. La difficulté alors est de garantir une robustesse dans les procédés et méthodologies mis en place pour améliorer la généralisation des modèles. Dans le cas du sur-échantillonnage, nous avons vu que la composition des images atypiques à sur-représenter peut conduire à un gain de performance mais souffre d’un manque de robustesse. Pour ce qui est de la data-augmentation basée sur des images pseudo aléatoires, le gain de performances est limité aux situations atypiques relative à l’offre.

5.7 Conclusion

Pour conclure, la performance d’un modèle de prédiction de charge des transport en commun doit être observé non seulement dans le cas général ou sur un échantillon restreint d’images comme les heures de pointes, mais elle doit être menée en tenant compte des cas atypiques observés sur la ligne. L’évaluation des performances de généralisation des modèles est ainsi plus robuste, plus fine, et permet de mieux cerner les limitations attendues lors du déploiement opérationnel d’une telle approche. Dans ce chapitre, nous avons d’abord exploité l’espace latent et d’autres indicateurs statistiques ou calendaires pour identifier des situations atypiques. Nous avons ensuite cherché à savoir dans quelle mesure nous pouvions améliorer les performances de prédiction sur ces cas atypiques en augmentant leur présence dans la base de données d’apprentissage.

Nous avons pu extraire 10 groupes d’images atypiques. On retrouve dans ces groupes d’images particulières les contextes atypiques relatifs à la demande comme les images de

Fortes Charges, de *Correspondance* et de *Variations*. On trouve également des cas atypiques qui ont trait à l'offre avec les situations de *Délais* et de *Perturbations*. On peut également ajouter les cas atypiques liés à des événements spécifiques connus à l'avance pouvant impacter tant l'offre théorique que la demande comme les *Grèves*, les *Confinements* et les *Evènements* (principalement les match de foots pour la ligne 9). Enfin, on retrouve les *Outliers* issus de l'analyse de l'espace latent et ceux découlant du format images des données (*Données manquantes*).

Deux procédés d'augmentation des données ont été implémentés, le premier est basé sur le sur-échantillonnage tandis que le second a recours à l'usage de la génération d'images pseudo-aléatoires. La première méthode peut permettre un gain en performance mais son manque de stabilité ne permet pas de garantir une meilleure généralisation des modèles dans tous les cas. Certaines classes, semblent présenter des différences significatives entre données d'apprentissage et données de test (en particulier les *Grèves*), ce qui empêche d'obtenir des gains de performances pour l'ensemble des situations.

La seconde approche a permis d'opérer un gain de performance sur un sous-ensemble d'images atypiques relatives à l'offre au dépend des images de fortes affluences passagers.

Ainsi, la méthodologie visant à mettre en place un seul modèle pour tenir compte de l'ensemble des cas atypiques n'est pas concluante au regard des résultats que nous avons obtenus. Il serait envisageable de mettre en place des modèles spécialisés via du transfer learning (spécialiser les modèles à partir des modèles nominaux) ou en découpant l'apprentissage par catégorie d'image. Cependant, cette approche induit alors une diversification des méthodologies et modèles (en associant détection d'atypique, classification et prédiction) qui peuvent être contraires à une optique de prédiction à court terme et poser des difficultés lors de la mise en oeuvre opérationnelle d'une telle approche.

Conclusion and Perspectives

Conclusion

Les charges de passagers ont connu plusieurs évolutions au cours de ces dernière années au sein de la capitale Parisienne. L'irruption du Covid 19 et le développement du télétravail en réponse, impactent encore aujourd'hui les dynamiques de charges et les comportements des usagers. Ainsi une prédiction de charge doit tenir compte de nouveaux contextes tout en ayant en mémoire les cas nominaux et atypiques de la ligne.

Dans cette thèse, nous avons proposé une méthodologie de prédiction de charge construite à partir d'image définissant finement les déplacement des trains où il s'agit d'associer chaque départ de train à un pixel. Une telle approche permet de prendre en compte la dynamique réelle des flux de passagers en encodant l'ensemble des mouvement de trains le long d'une ligne. Les images fournissent *in fine* un format ajustable selon la taille des lignes de métro ou le nombre de variable disponible (charge, temps de stationnement, heure d'arrivée, etc), elle permettent également de tenir compte des contextes de transport atypiques sans pour autant devoir modifier ou adapter la structure des images.

Les images sont donc à la fois un outil de visualisation de la dynamique d'une ligne de métro mais elles ouvrent également la possibilité d'utiliser l'ensemble des méthodologies de traitement d'image. Dans cette thèse nous avons abondamment mobilisé celles-ci pour dans un premier temps, prédire la charge des quatre prochains départs de trains sur l'ensemble de la ligne et dans un second temps, faciliter la labellisation de situations atypiques.

La prédiction de charge que nous avons proposée utilise des techniques d'inpainting pour prédire efficacement les charges d'une ligne de métro entière. L'inpainting est rendu possible par la structure des images en temps réel où se dessine une région manquante de l'ensemble des futurs départ de trains. Ces travaux ont servi à valider l'usage de telles images pour une tâche de prédiction. Nous pensons que l'introduction d'approches de ce type et leur évaluation détaillée telle que réalisée dans cette thèse constitue un apport dans le domaine de la prédiction de données spatio-temporelles dans un contexte de données de transports en commun.

Cette thèse a également mis en lumière l'importance des cas atypiques pour la prédiction de charge et en particulier l'importance d'une évaluation différenciée des performances des modèles de prédiction dans ces situations. Réduire les résultats d'outils de prédictions aux seuls cas nominaux ou heures de pointe d'une ligne de métro n'est pas significatif pour évaluer pleinement une approche. L'enjeu n'est plus ici d'améliorer la performance globale d'un modèle mais de s'assurer de sa généralisation au regard des situations atypiques ou modification des comportements des usagers comme au cours de ces dernières années avec le Covid et le télétravail. Dans ce contexte, nous pensons que l'approche hybride proposée basée sur des statistiques 'métier' et l'usage d'outils semi-automatiques (clustering / détection d'outliers) pour aider à l'interprétation et à la labellisation des données dans

l'espace latent de réseau de neurones profonds, constitue également une contribution intéressante et mobilisable pour d'autres type de problématique de prédiction dans le domaine du transport.

Perspectives

Pour prolonger cette thèse, certains travaux de recherche pourraient être développés. Je dégage de ces trois années de thèse trois perspectives qui me semblent particulièrement pertinentes en ce qui concerne la suite de ces travaux :

1. Une première perspective naturelle aux travaux présentés concerne la problématique de détection et de labellisation des situations atypiques. Dans ces travaux, la catégorisation des images a été réalisée via des techniques de clustering et de projection dans des espaces latents assez classiques. Il existe dans la littérature des méthodes de computer vision avancées sur ces questions de labellisation. D'un point de vue opérationnel, de telles approches pourraient être mobilisées pour aller vers une détection préventive.
2. La seconde perspective concerne également l'aspect de labellisation des images mais en mobilisant des méthodologies qui s'éloignent du traitement d'image, par exemple venant du traitement du langage naturel, comme les approches de type word2vec. L'objectif serait de construire à l'instar de ces méthodes, un espace latent adapté non pas de mots mais de trains qui soit pertinent. Pour ce faire, le corpus de texte nécessaire à l'apprentissage d'un modèle d'embedding de type word2vec serait remplacé par les images, chaque colonne représentant un élément d'une séquence (mot du corpus), et chaque image une séquence complète (document). L'objectif serait alors de projeter nos trains pour les grouper à l'instar des mots afin de détecter des patterns de trains pouvant permettre un clustering des trains et un multi-labellisation des images selon les trains présents dans l'image.
3. La dernière perspective se raccroche à la prédiction de charge. Ces travaux se sont limités à une seule ligne et un seul sens de circulation pour conserver une cohérence spatiale entre nos images. C'est à dire qu'une station se situe toujours au même endroit dans l'image. La question du transfert d'un modèle appris sur une ligne à une autre ligne / sens est une autre piste naturelle d'extension et de poursuite des travaux initiés ici. Cette perspective est pertinente d'un point de vue applicatif puisqu'elle concorde avec la fiabilisation des données de charge de deux nouvelles lignes de métro (lignes 1 et 14). Elle pourrait aussi permettre de mettre en commun les situations atypiques faiblement représentées et par conséquent améliorer les capacités de généralisation des modèles de prédiction dans ces situations. Cette problématique est cependant complexe du fait des spécificités spatiales des différentes lignes.

Résumé détaillé de la thèse en français

*Prévision de charge dans les transports en commun
en situation atypique par traitement d'image*

Bibliographie

- [1] P AJAY et al. « Unsupervised hyperspectral microscopic image segmentation using deep embedded clustering algorithm ». In : *Scanning* 2022 (2022).
- [2] Qinhe AN et al. « Analysis of adding-runs strategy for peak-hour regular bus services ». In : *Transportation Research Part E : Logistics and Transportation Review* 143 (2020), p. 102100. ISSN : 1366-5545. DOI : <https://doi.org/10.1016/j.tre.2020.102100>. URL : <https://www.sciencedirect.com/science/article/pii/S1366554520307493>.
- [3] Matthias ARZT et al. « LABKIT : labeling and segmentation toolkit for big image data ». In : *Frontiers in computer science* 4 (2022), p. 10.
- [4] Riadh AYACHI et al. « Strided Convolution Instead of Max Pooling for Memory Efficiency of Convolutional Neural Networks ». In : *Proceedings of the 8th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT'18), Vol.1*. Sous la dir. de Med Salim BOUHLEL et Stefano ROVETTA. Cham : Springer International Publishing, 2020, p. 234-243. ISBN : 978-3-030-21005-2.
- [5] Thomas BAPAUME et al. « Forecasting passenger flows and headway at train level for a public transport line : Focus on atypical situations ». In : *Transportation Research Part C : Emerging Technologies* 153 (2023), p. 104195. ISSN : 0968-090X. DOI : <https://doi.org/10.1016/j.trc.2023.104195>. URL : <https://www.sciencedirect.com/science/article/pii/S0968090X23001845>.
- [6] Thomas BAPAUME et al. « Image Inpainting and Deep Learning to Forecast Short-Term Train Loads ». In : *IEEE Access* 9 (2021), p. 98506-98522. DOI : [10.1109/ACCESS.2021.3093987](https://doi.org/10.1109/ACCESS.2021.3093987).
- [7] Josh BEAL et al. « Toward transformer-based object detection ». In : *arXiv preprint arXiv :2012.09958* (2020).
- [8] Marcelo BERTALMIO et al. « Image inpainting ». In : *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. 2000, p. 417-424.
- [9] George.E.P. BOX et Gwilym M. JENKINS. *Time Series Analysis : Forecasting and Control*. Holden-Day, 1976.
- [10] Aurélie BUGEAU et Marcelo BERTALMIO. « Combining Texture Synthesis and Diffusion for Image Inpainting. » In : *VISAPP 2009 - Proceedings of the Fourth International Conference on Computer Vision Theory and Applications*. Portugal, 2009, p. 26-33. URL : <https://hal.science/hal-00551587>.
- [11] Hu CAO et al. « Swin-Unet : Unet-like Pure Transformer for Medical Image Segmentation ». In : *arXiv preprint arXiv :2105.05537* (2021).

- [12] Nicolas CARION et al. « End-to-end object detection with transformers ». In : *European conference on computer vision*. Springer. 2020, p. 213-229.
- [13] Mathilde CARON et al. *Deep Clustering for Unsupervised Learning of Visual Features*. 2019. arXiv : [1807.05520](https://arxiv.org/abs/1807.05520) [cs.CV].
- [14] Changlu CHEN et al. « Bidirectional Spatial-Temporal Adaptive Transformer for Urban Traffic Flow Forecasting ». In : *IEEE Transactions on Neural Networks and Learning Systems* (2022).
- [15] Geng CHEN et al. « XQ-NLM : Denoising Diffusion MRI Data via x-q Space Non-local Patch Matching ». In : *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2016*. Sous la dir. de Sebastien OURSELIN et al. Cham : Springer International Publishing, 2016, p. 587-595. ISBN : 978-3-319-46726-9.
- [16] Jieneng CHEN et al. « Transunet : Transformers make strong encoders for medical image segmentation ». In : *arXiv preprint arXiv :2102.04306* (2021).
- [17] Long CHEN et al. « Sca-cnn : Spatial and channel-wise attention in convolutional networks for image captioning ». In : *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, p. 5659-5667.
- [18] Xiaoxu CHEN et al. *Probabilistic forecasting of bus travel time with a Bayesian Gaussian mixture model*. 2022. DOI : [10.48550/ARXIV.2206.06915](https://doi.org/10.48550/ARXIV.2206.06915). URL : <https://arxiv.org/abs/2206.06915>.
- [19] Rewon CHILD et al. « Generating Long Sequences with Sparse Transformers ». In : *arXiv e-prints*, arXiv :1904.10509 (avr. 2019), arXiv :1904.10509. DOI : [10.48550/arXiv.1904.10509](https://doi.org/10.48550/arXiv.1904.10509). arXiv : [1904.10509](https://arxiv.org/abs/1904.10509) [cs.LG].
- [20] Ernesto CIPRIANI, Stefano GORI et Marco PETRELLI. « Transit network design : A procedure and an application to a large urban area ». In : *Transportation Research Part C : Emerging Technologies* 20.1 (2012). Special issue on Optimization in Public Transport+ISTTT2011, p. 3-14. ISSN : 0968-090X. DOI : <https://doi.org/10.1016/j.trc.2010.09.003>. URL : <https://www.sciencedirect.com/science/article/pii/S0968090X10001397>.
- [21] Journal du DIMANCHE. *La carte des sites olympiques de Paris 2024*. Online ; accessed Aout 15, 2023. 2017. URL : <https://www.lejdd.fr/Sport/jo-a-paris-la-carte-des-sites-olympiques-3435365>.
- [22] Alexey DOSOVITSKIY et al. *An Image is Worth 16x16 Words : Transformers for Image Recognition at Scale*. 2021. arXiv : [2010.11929](https://arxiv.org/abs/2010.11929) [cs.CV].
- [23] Shengdong DU et al. « A Hybrid Method for Traffic Flow Forecasting Using Multimodal Deep Learning ». In : *International Journal of Computational Intelligence Systems* 13 (mars 2018). DOI : [10.2991/ijcis.d.200120.001](https://doi.org/10.2991/ijcis.d.200120.001).
- [24] Ministère de la transition ÉCOLOGIQUE. *Chiffres clés du transport*. [Online ; accessed 25 août, 2023]. 2022. URL : <https://www.statistiques.developpement-durable.gouv.fr/edition-numerique/chiffres-cles-transports-2022/partie1-economie-des-transports>.
- [25] Oscar EGU et Patrick BONNEL. « Medium-term public transit route ridership forecasting : What, how and why ? A case study in Lyon ». In : *Transport Policy* 105 (mars 2021). DOI : [10.1016/j.tranpol.2021.03.002](https://doi.org/10.1016/j.tranpol.2021.03.002).

- [26] Mariano GALLO et al. « Artificial Neural Networks for Forecasting Passenger Flows on Metro Lines ». In : *Sensors* 19 (août 2019), p. 1-14. DOI : [10.3390/s19153424](https://doi.org/10.3390/s19153424).
- [27] Yunhe GAO, Mu ZHOU et Dimitris N METAXAS. « UTNet : a hybrid transformer architecture for medical image segmentation ». In : *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2021, p. 61-71.
- [28] X. GENG et al. « Spatiotemporal Multi-Graph Convolution Network for Ride-Hailing Demand Forecasting ». In : *AAAI*. 2019.
- [29] Le Parisien/Delphine GOLDSZTEJN. *Rames bondées, incidents... Notre test pour savoir combien vous attendez vraiment votre métro à Paris*. [Online; accessed august 25, 2023]. 2022. URL : <https://www.leparisien.fr/info-paris-ile-de-france-oise/transports/la-galere-du-metro-parisien-decouvrez-les-vrais-temps-dattente-sur-votre-ligne-16-12-2022-FFLIRB72FBFHXEFB32PHQAXX5M.php>.
- [30] Abel GONZALEZ-GARCIA, Joost van de WEIJER et Yoshua BENGIO. *Image-to-image translation for cross-domain disentanglement*. 2018. arXiv : [1805.09730](https://arxiv.org/abs/1805.09730) [cs.CV].
- [31] Ian GOODFELLOW et al. « Generative Adversarial Networks ». In : *Advances in Neural Information Processing Systems* 3 (juin 2014). DOI : [10.1145/3422622](https://doi.org/10.1145/3422622).
- [32] S. GUO et al. « Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting ». In : *AAAI*. 2019.
- [33] Jiuran HE et Bingfeng SI. « The application of ARIMA-RBF model in urban rail traffic volume forecast ». In : *Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013)*. Atlantis Press, 2013, p. 1662-1665. ISBN : 978-90-78677-61-1. DOI : <https://doi.org/10.2991/iccsee.2013.416>. URL : <https://doi.org/10.2991/iccsee.2013.416>.
- [34] Léonie HEYDENRIJK-OTTENS et al. « Supervised learning : Predicting passenger load in public transport ». In : *CASPT Conference on Advanced Systems in Public Transport and TransitData*. 2018, p. 30-32.
- [35] Geoffrey E HINTON et Sam ROWEIS. « Stochastic neighbor embedding ». In : *Advances in neural information processing systems* 15 (2002).
- [36] Sepp HOCHREITER et Jürgen SCHMIDHUBER. « Long Short-term Memory ». In : *Neural computation* 9 (déc. 1997), p. 1735-80. DOI : [10.1162/neco.1997.9.8.1735](https://doi.org/10.1162/neco.1997.9.8.1735).
- [37] Phillip ISOLA et al. *Image-to-Image Translation with Conditional Adversarial Networks*. 2018. arXiv : [1611.07004](https://arxiv.org/abs/1611.07004) [cs.CV].
- [38] W. JIANG et L. ZHANG. « Geospatial Data to Images : A Deep-Learning Framework for Traffic Forecasting ». In : *Tsinghua Science & Technology* 24 (2019), p. 52-64.
- [39] Dohyoung JO et al. « Image-to-Image Learning to Predict Traffic Speeds by Considering Area-Wide Spatio-Temporal Dependencies ». In : *IEEE Transactions on Vehicular Technology* 68.2 (2019), p. 1188-1197. DOI : [10.1109/TVT.2018.2885366](https://doi.org/10.1109/TVT.2018.2885366).

- [40] Jintao KE et al. « Short-term forecasting of passenger demand under on-demand ride services : A spatio-temporal deep learning approach ». In : *Transportation Research Part C : Emerging Technologies* 85 (2017), p. 591-608. ISSN : 0968-090X. DOI : <https://doi.org/10.1016/j.trc.2017.10.016>. URL : <https://www.sciencedirect.com/science/article/pii/S0968090X17302899>.
- [41] Salman KHAN et al. « Transformers in vision : A survey ». In : *ACM Computing Surveys (CSUR)* (2021).
- [42] Kyung Min KIM et al. « Does crowding affect the path choice of metro passengers ». In : *Transportation Research Part A : Policy and Practice* 77 (2015), p. 292-304. ISSN : 0965-8564. DOI : <https://doi.org/10.1016/j.tra.2015.04.023>. URL : <https://www.sciencedirect.com/science/article/pii/S0965856415001093>.
- [43] Diederik KINGMA et Jimmy BA. « Adam : A Method for Stochastic Optimization ». In : *International Conference on Learning Representations* (déc. 2014).
- [44] LE PARISIEN. *Métro parisien : comment est calculé le temps d'attente affiché sur les quais ?* [Online; accessed august 25, 2023]. 2022. URL : <https://www.leparisien.fr/info-paris-ile-de-france-oise/transports/metro-parisien-comment-est-calculé-le-temps-dattente-affiché-sur-les-quais-16-12-2022-ON6ZXXOBTVGYNCC57IUSQZMUDQ.php>.
- [45] K. LEE et Wonjong RHEE. « Graph Convolutional Modules for Traffic Forecasting ». In : *ArXiv abs/1905.12256* (2019).
- [46] Fengfu LI et al. *Discriminatively Boosted Image Clustering with Fully Convolutional Auto-Encoders*. 2017. arXiv : [1703.07980](https://arxiv.org/abs/1703.07980) [cs.CV].
- [47] Gen LI et Jason J JUNG. « Deep learning for anomaly detection in multivariate time series : Approaches, applications, and challenges ». In : *Information Fusion* (2022).
- [48] S. LIANG et al. « Short-Term Passenger Flow Prediction in Urban Public Transport : Kalman Filtering Combined K-Nearest Neighbor Approach ». In : *IEEE Access* 7 (2019), p. 120937-120949. DOI : [10.1109/ACCESS.2019.2937114](https://doi.org/10.1109/ACCESS.2019.2937114).
- [49] Chen-Hsuan LIN et al. « St-gan : Spatial transformer generative adversarial networks for image compositing ». In : *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, p. 9455-9464.
- [50] Dong LIN, Wout BROERE et Jianqiang CUI. « Metro systems and urban development : Impacts and implications ». In : *Tunnelling and Underground Space Technology* 125 (2022), p. 104509. ISSN : 0886-7798. DOI : <https://doi.org/10.1016/j.tust.2022.104509>. URL : <https://www.sciencedirect.com/science/article/pii/S0886779822001493>.
- [51] M. LIPPI, M. BERTINI et P. FRASCONI. « Short-Term Traffic Flow Forecasting : An Experimental Comparison of Time-Series Analysis and Supervised Learning ». In : *IEEE Transactions on Intelligent Transportation Systems* 14.2 (2013), p. 871-882. DOI : [10.1109/TITS.2013.2247040](https://doi.org/10.1109/TITS.2013.2247040).
- [52] Lu LIU et al. « A universal representation transformer layer for few-shot image classification ». In : *arXiv preprint arXiv :2006.11702* (2020).
- [53] Yanzhu LIU et al. « An Image Inpainting Approach to Short-term Load Forecasting ». In : *IEEE Transactions on Power Systems* (2022).

- [54] Sohani LIYANAGE et al. « AI-based neural network models for bus passenger demand forecasting using smart card data ». In : *Journal of Urban Management* (2022). ISSN : 2226-5856. DOI : <https://doi.org/10.1016/j.jum.2022.05.002>. URL : <https://www.sciencedirect.com/science/article/pii/S2226585622000280>.
- [55] Xiaolei MA et al. « Learning Traffic as Images : A Deep Convolutional Neural Network for Large-Scale Transportation Network Speed Prediction ». In : *Sensors* 17 (avr. 2017), p. 818. DOI : [10.3390/s17040818](https://doi.org/10.3390/s17040818).
- [56] Y. MA, Z. ZHANG et A. IHLER. « Multi-Lane Short-Term Traffic Forecasting With Convolutional LSTM Network ». In : *IEEE Access* 8 (2020), p. 34629-34643. DOI : [10.1109/ACCESS.2020.2974575](https://doi.org/10.1109/ACCESS.2020.2974575).
- [57] Meng MENG et al. « Short-Term Travel-Time Prediction using Support Vector Machine and Nearest Neighbor Method ». In : *Transportation Research Record* 2676.6 (2022), p. 353-365. DOI : [10.1177/03611981221074371](https://doi.org/10.1177/03611981221074371). eprint : <https://doi.org/10.1177/03611981221074371>. URL : <https://doi.org/10.1177/03611981221074371>.
- [58] Marriam NAWAZ et al. « Skin cancer detection from dermoscopic images using deep learning and fuzzy k-means clustering ». In : *Microscopy research and technique* 85.1 (2022), p. 339-351.
- [59] Chuang NIU, Hongming SHAN et Ge WANG. « SPICE : Semantic Pseudo-Labeling for Image Clustering ». In : *IEEE Transactions on Image Processing* 31 (2022), p. 7264-7278. DOI : [10.1109/tip.2022.3221290](https://doi.org/10.1109/tip.2022.3221290). URL : <https://doi.org/10.1109/tip.2022.3221290>.
- [60] Foivos NTELEMIS, Yaochu JIN et Spencer A. THOMAS. « Information maximization clustering via multi-view self-labelling ». In : *Knowledge-Based Systems* 250 (2022), p. 109042. DOI : [10.1016/j.knosys.2022.109042](https://doi.org/10.1016/j.knosys.2022.109042). URL : <https://doi.org/10.1016/j.knosys.2022.109042>.
- [61] Ozan ÖZDENIZCI et Robert LEGENSTEIN. « Restoring Vision in Adverse Weather Conditions With Patch-Based Denoising Diffusion Models ». In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.8 (2023), p. 10346-10357. DOI : [10.1109/TPAMI.2023.3238179](https://doi.org/10.1109/TPAMI.2023.3238179).
- [62] Ville de PARIS. *Bilan des déplacements à Paris en 2021 : Les transports en commun*. [Online ; accessed 28 août, 2023]. 2021. URL : <https://www.paris.fr/pages/bilan-des-deplacements-a-paris-en-2021-21167>.
- [63] Kevin PASINI et al. « LSTM Encoder-Predictor for Short-Term Train Load Forecasting ». In : *ECML/PKDD*. 2019.
- [64] Olivier PETIT et al. « U-net transformer : Self and cross attention for medical image segmentation ». In : *International Workshop on Machine Learning in Medical Imaging*. Springer. 2021, p. 267-276.
- [65] Institut Paris RÉGION. *Le mass transit à l'heure du télétravail et de la sobriété énergétique*. [Online ; accessed 28 août, 2023]. 2013. URL : <https://www.institutparisregion.fr/nos-travaux/publications/le-mass-transit-a-lheure-du-teletravail-et-de-la-sobriete-energetique/>.
- [66] Olaf RONNEBERGER, Philipp FISCHER et Thomas BROX. *U-Net : Convolutional Networks for Biomedical Image Segmentation*. 2015. arXiv : [1505.04597](https://arxiv.org/abs/1505.04597) [cs.CV].

- [67] J. ROOS, S. BONNEVAY et G. GAVIN. « Short-Term Urban Rail Passenger Flow Forecasting : A Dynamic Bayesian Network Approach ». In : *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*. 2016, p. 1034-1039. DOI : [10.1109/ICMLA.2016.0187](https://doi.org/10.1109/ICMLA.2016.0187).
- [68] Or SHARIR, Barak PELEG et Yoav SHOHAM. « The Cost of Training NLP Models : A Concise Overview ». In : *arXiv e-prints*, arXiv :2004.08900 (avr. 2020), arXiv :2004.08900. DOI : [10.48550/arXiv.2004.08900](https://doi.org/10.48550/arXiv.2004.08900). arXiv : [2004.08900](https://arxiv.org/abs/2004.08900) [[cs.CL](#)].
- [69] Hualian SHENG et al. « Improving 3d object detection with channel-wise transformer ». In : *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, p. 2743-2752.
- [70] Zhenhao TANG et al. « Auto-encoder-extreme learning machine model for boiler NOx emission concentration prediction ». In : *Energy* 256 (2022), p. 124552.
- [71] Agence de la TRANSITION ECOCOLOGIQUE. *ELETRAVAIL (IM)MOBILITE ET MODES DE VIE*. [Online ; accessed 28 août, 2023]. 2021. URL : <https://bibliothèque.ademe.fr/cadic/315/teletravail-modes-de-vie-2020-synthese.pdf?modal=false>.
- [72] Ashish VASWANI et al. *Attention Is All You Need*. 2017. arXiv : [1706.03762](https://arxiv.org/abs/1706.03762) [[cs.CL](#)].
- [73] Petar VELIČKOVIĆ et al. *Graph Attention Networks*. 2018. arXiv : [1710.10903](https://arxiv.org/abs/1710.10903) [[stat.ML](#)].
- [74] Petar VELIČKOVIĆ et al. « Graph attention networks ». In : *arXiv preprint arXiv :1710.10903* (2017).
- [75] Eleni I. VLAHOGIANNI, Matthew G. KARLAFTIS et John C. GOLIAS. « Short-term traffic forecasting : Where we are and where we're going ». In : *Transportation Research Part C : Emerging Technologies* 43 (2014). Special Issue on Short-term Traffic Flow Forecasting, p. 3-19. ISSN : 0968-090X. DOI : <https://doi.org/10.1016/j.trc.2014.01.005>. URL : <https://www.sciencedirect.com/science/article/pii/S0968090X14000096>.
- [76] Dongjie WANG, Yan YANG et Shangming NING. « DeepSTCL : A Deep Spatio-temporal ConvLSTM for Travel Demand Prediction ». In : *2018 International Joint Conference on Neural Networks (IJCNN)*. 2018, p. 1-8. DOI : [10.1109/IJCNN.2018.8489530](https://doi.org/10.1109/IJCNN.2018.8489530).
- [77] Yi WANG et al. « Attention based spatiotemporal graph attention networks for traffic flow forecasting ». In : *Information Sciences* 607 (2022), p. 869-883. ISSN : 0020-0255. DOI : <https://doi.org/10.1016/j.ins.2022.05.127>. URL : <https://www.sciencedirect.com/science/article/pii/S0020025522005679>.
- [78] Zhendong WANG et al. *Patch Diffusion : Faster and More Data-Efficient Training of Diffusion Models*. 2023. arXiv : [2304.12526](https://arxiv.org/abs/2304.12526) [[cs.CV](#)].
- [79] Philip B. WEERAKODY et al. « A review of irregular time series data handling with gated recurrent neural networks ». In : *Neurocomputing* 441 (2021), p. 161-178. ISSN : 0925-2312. DOI : <https://doi.org/10.1016/j.neucom.2021.02.046>. URL : <https://www.sciencedirect.com/science/article/pii/S0925231221003003>.

- [80] Jianqing WU et al. « Towards attention-based convolutional long short-term memory for travel time prediction of bus journeys ». In : *Sensors* 20.12 (2020), p. 3354.
- [81] Weitiao WU, Yisong XIA et Wenzhou JIN. « Predicting bus passenger flow and prioritizing influential factors using multi-source data : Scaled stacking gradient boosting decision trees ». In : *IEEE Transactions on Intelligent Transportation Systems* 22.4 (2020), p. 2510-2523.
- [82] Zhaoyi YAN et al. *Shift-Net : Image Inpainting via Deep Feature Rearrangement*. 2018. arXiv : [1801.09392 \[cs.CV\]](https://arxiv.org/abs/1801.09392).
- [83] Fisher YU et Vladlen KOLTUN. *Multi-Scale Context Aggregation by Dilated Convolutions*. 2016. arXiv : [1511.07122 \[cs.CV\]](https://arxiv.org/abs/1511.07122).
- [84] Jiahui YU et al. *Generative Image Inpainting with Contextual Attention*. 2018. arXiv : [1801.07892 \[cs.CV\]](https://arxiv.org/abs/1801.07892).
- [85] Yongtao YU et al. « CapViT : Cross-context capsule vision transformers for land cover classification with airborne multispectral LiDAR data ». In : *International Journal of Applied Earth Observation and Geoinformation* 111 (2022), p. 102837. ISSN : 1569-8432. DOI : <https://doi.org/10.1016/j.jag.2022.102837>. URL : <https://www.sciencedirect.com/science/article/pii/S1569843222000395>.
- [86] Tianqi ZHANG et Ge GUO. « Graph attention LSTM : A spatiotemporal approach for traffic flow forecasting ». In : *IEEE Intelligent Transportation Systems Magazine* 14.2 (2020), p. 190-196.
- [87] Tianhong ZHAO et al. « Coupling graph deep learning and spatial-temporal influence of built environment for short-term bus travel demand prediction ». In : *Computers, Environment and Urban Systems* 94 (2022), p. 101776. ISSN : 0198-9715. DOI : <https://doi.org/10.1016/j.compenvurbsys.2022.101776>. URL : <https://www.sciencedirect.com/science/article/pii/S0198971522000205>.
- [88] Z. ZHAO et al. « LSTM network : a deep learning approach for short-term traffic forecast ». In : *Iet Intelligent Transport Systems* 11 (2017), p. 68-75.
- [89] Yan ZHENG et al. « Urban road traffic flow prediction : A graph convolutional network embedded with wavelet decomposition and attention mechanism ». In : *Physica A : Statistical Mechanics and its Applications* 608 (2022), p. 128274. ISSN : 0378-4371. DOI : <https://doi.org/10.1016/j.physa.2022.128274>. URL : <https://www.sciencedirect.com/science/article/pii/S0378437122008329>.

