

**Université de BLIDA 1**  
Faculté des Sciences  
Département d'informatique



**MEMOIRE DE FIN D'ETUDES POUR L'OBTENTION DU DIPLOME DE  
MASTER EN INFORMATIQUE**

**Option : Spécialité « Ingénierie du Logiciel »**

**Etude et implémentation des techniques de modélisation de la trajectoire  
d'un objet mobile dans une ville intelligente en vue de prédiction de  
mobilité.**

**Réalisé par : BENSEBTI Riadh, HADJ KACEM Ahmed**

**Prometteur : M.KAMECH Abdellah**

**Encadreur : M.BOUKHEDOUMA Hocine**

Année universitaire 2020/2021

# Remerciements

Nous commencerons par remercier nos familles pour nous avoir soutenus et encouragés durant la réalisation de ce projet et pour toute leur contribution pendant notre cursus universitaire, pour leurs efforts inconditionnels et pour nous mettre dans la situation agréable qui nous a permis d'atteindre cette réalisation.

Nous adressons notre reconnaissance tout particulièrement à M.KAMECH Abdellah pour avoir été notre promoteur de projet.

Nous le remercions pour sa patience, sa disponibilité et surtout ses judicieux conseils, qui ont contribué à alimenter notre projet.

Nous tenons tout aussi à remercier notre Encadreur M.BOUKHEDOUMA Hocine qui nous a fait part de son savoir et de son expérience dans le domaine de la recherche et dans le domaine de la mobilité, aussi pour l'apport de ses compétences mis dans notre travail et son enthousiasme pour réaliser notre travail.

Nous tenons à remercier les membres du jury pour avoir pris le temps d'examiner et évaluer notre travail.

Nous n'oublierons pas les professeurs qui nous ont suivi tout au long de notre parcours universitaire, nous leur sommes reconnaissants pour leurs efforts et le savoir qu'ils nous ont transmis.

.

## ***Résumé***

L'émergence du concept des villes intelligentes, des technologies de l'information et de la communication (TIC) et de l'Internet des objets (IoT), observés durant ces dernières années, a permis d'améliorer le quotidien du citoyen tout en l'impliquant dans ce concept.

Parmi les données utilisées, celles liées à la mobilité des individus sont nécessaires pour des solutions visant l'amélioration de plusieurs domaines tels que la recommandation des services, le transport, les réseaux cellulaires, etc.

Les données de mobilité sont dans un état brut, afin de les utiliser dans un processus de prédiction, un filtrage de données est nécessaire, ces données filtrées seront utilisées par la suite pour la modélisation de la trajectoire, qui signifie dans ce cas la manière avec laquelle la trajectoire est décrite.

Dans ce mémoire nous allons proposer un modèle de description de trajectoire pour faire une prédiction basée sur le machine Learning qui sera évaluée et comparée à d'autres modèles en utilisant des jeux de données réelles de mobilité humaine.

**Mots-clés** : Ville intelligente, technologie de l'information, internet des objets, mobilité humain, prediction, Intelligence Artificielle.

## ***Abstract***

The emergence of the concept of smart cities, information and communication technologies (ICT) and the Internet of things (IoT), observed in recent years, has made it possible to improve the daily life of citizens while involving him in this task.

Among the data used, those related to the mobility of individuals are necessary for solutions aimed at improving several areas such as service recommendation, transport, cellular networks, etc.

Mobility data is in a raw state. In order to use them in a prediction process, data filtering is used subsequently for the modeling of the trajectory, which here means the way in which the trajectory is described.

In this thesis, we will propose a trajectory description model, to make a prediction based on deep learning which will be evaluated, and compared to other models, using real human mobility datasets.

**Keywords**: Smart city, information technology, internet of things, human mobility, prediction, Artificial Intelligence.

## ملخص

إن ظهور مفهوم المدن الذكية وتقنيات المعلومات والاتصالات (ICT) وإنترنت الأشياء (IoT)، الذي لوحظ في السنوات الأخيرة، قد أتاح تحسين الحياة اليومية للمواطنين مع إشراكه في هذه المهمة.

من بين البيانات المستخدمة، تلك المتعلقة بتنقل الأفراد ضرورية للحلول التي تهدف إلى تحسين عدة مجالات مثل توصية الخدمة، والنقل، والشبكات الخلوية، إلخ.

بيانات التنقل في حالة أولية. من أجل استخدامها في عملية التنبؤ، يتم استخدام تصفية البيانات لاحقًا لنمذجة المسار، مما يعني هنا الطريقة التي يتم بها وصف المسار.

في هذه الأطروحة، سوف نقترح نموذجًا لوصف المسار، لعمل تنبؤ قائم على التعلم العميق الذي سيتم تقييمه ومقارنته بالنماذج الأخرى باستخدام مجموعات بيانات حقيقية للتنقل البشري.

الكلمات المفتاحية: المدينة الذكية ، تكنولوجيا المعلومات ، إنترنت الأشياء ، التنقل البشري ، التنبؤ ، الذكاء الاصطناعي

# Table des matières

<i>Introduction</i> .....	2
Chapitre 1 Apprentissage automatique .....	错误!未定义书签。
1.1 Introduction .....	4
1.2 Apprentissage automatique.....	4
1.3 Les Réseaux de neurones.....	5
1.4 Conclusion.....	17
Chapitre 2 La mobilité dans les villes intelligentes .....	19
2.1 Introduction .....	19
2.2 Ville intelligente .....	19
2.3 Mobilité dans les villes intelligentes .....	22
2.4 Le MCSC et sa relation avec les villes intelligentes .....	25
2.5 La géolocalisation dans les villes intelligentes.....	26
2.6 Conclusion.....	27
Chapitre 3 Modélisation de la trajectoire : et travaux liées à la prédiction.....	28
3.1 Introduction .....	28
3.2 Définition d'une trajectoire .....	28
3.3 Technique de description d'une trajectoire .....	29
3.4 Modèles de trajectoire existants : mentionner les différents travaux .....	30

3.5	Etude comparative des modèles .....	37
3.6	Travaux liés à la prédiction de trajectoire .....	39
3.7	Conclusion .....	44
Chapitre 4 Spécification des besoins et conception .....		45
4.1	Introduction .....	45
4.2	Conception global.....	45
4.3	Conception détailler.....	46
4.4	Conclusion .....	55
Chapitre 5 Implémentation de la solution .....		56
5.1	Introduction .....	56
5.2	Environnement et outils de développement utilisés .....	56
5.3	Données utilisé pour la solution .....	64
5.4	Implémentation et fonctionnement.....	65
5.5	Evaluation des résultats .....	74
5.6	Conclusion .....	79
<i>Conclusion et perspective .....</i>		<i>80</i>

# Table des figures

Figure 1. 1 architecture CNN.....	8
Figure 1. 2 architecture RNN.....	9
Figure 1. 3 architecture LSTM .....	13
Figure 1. 4 architecture GRU.....	14
Figure 1. 5 architecture autoencoder.....	16
Figure 1. 6 architecteur GAN .....	17
Figure 3. 1 représentation du mouvement et la trajectoire .....	28
.Figure 3. 2 Un chemin spatio-temporel .....	29
Figure 3. 3 Modélisation de trajectoire .....	32
Figure 3. 4 l'Aube et Al. ....	32
Figure 3. 5 diagramme de classe trajectoire.....	34
Figure 3. 6 architecture du modèle multicouche.....	36
Figure 4. 1 Schéma de la solution proposé .....	46
Figure 4. 2 architecture du modèle avec LSTM .....	51
Figure 4. 3 architecture du modèle avec GRU.....	51
Figure 4. 4 la fonction de MAE .....	52
Figure 4. 5 la fonction de MSE.....	53

Figure 4. 6 fonction de Huber Loss .....	53
Figure 5. 1Logo Pythons.....	56
Figure 5. 2Logo colab.....	57
Figure 5. 3Logo TensorFlow .....	58
Figure 5. 4Logo Keras .....	59
Figure 5. 5 Logo Pandas .....	60
Figure 5. 6 Logo Matplotlib.....	61
Figure 5. 7Logo scikitmobility .....	62
Figure 5. 8Logo mobilityDB .....	63
Figure 5. 9Logo mobilityDB .....	64
Figure 5. 10 importation de jeu de donne .....	65
Figure 5. 11 installation de scikitmobility .....	65
Figure 5. 12 méthode stop detection.....	65
Figure 5. 13Resample par minute .....	66
Figure 5. 14 Resample par jour.....	67
Figure 5. 15 instalation des module .....	68
Figure 5. 16 adapté le dataframe au mobilityDB.....	68
Figure 5. 17 le type du dataframe .....	68
Figure 5. 18 construction de la table de la bd et enregistrement de la trajectoire.....	69
Figure 5. 19 importation des module d'apprentissage .....	70



Figure 5. 20 division du jeu de donne.....	70
Figure 5. 21 Scaling.....	71
Figure 5. 22 methode split series .....	71
Figure 5. 23 utilisation de la méthode split séries.....	72
Figure 5. 24 construction du modèle LSTM.....	72
Figure 5. 25 construction du modèle GRU .....	72
Figure 5. 26 entrainement du modèle LSTM.....	73
Figure 5. 27 entrainement du modèle GRU .....	73
Figure 5. 28 méthode pour la prédiction.....	73
Figure 5. 29 graph de la fonction loss GRU du modèle proposé.....	75
Figure 5. 30 graphe représente training accu et validation accu GRU du modèle proposé..	75
Figure 5. 31 graph de la fonction loss LSTM du modèle proposé.....	76
Figure 5. 32 graphe représente training accu et validation accu LSTM du modèle proposé	76
Figure 5. 33 graph de la fonction loss GRU du modèle d'évaluation.....	77
Figure 5. 34 graphe représente training accu et validation accu GRU du modèle d'évaluation.....	77
Figure 5. 35 graph de la fonction loss LSTM du modèle d'évaluation .....	78
Figure 5. 36 graphe représente training accu et validation accu LSTM du modèle d'évaluation.....	78



## ***Introduction***

L'émergence du concept des villes intelligentes, des technologies de l'information et de la communication (TIC) et de l'Internet des objets (IoT), observés durant ces dernières années, a permis d'améliorer le quotidien du citoyen tout en l'impliquant dans cette tâche. En effet, grand nombre de solutions proposées dans les différents domaines et qui sont offertes au citoyen utilisent les données circulant dans la ville et provenant de sources diverses. Le citoyen, utilisateur de ces solutions, joue un rôle important car il est à la fois fournisseur et consommateur de ces données.

Parmi les données utilisées, celles liées à la mobilité des individus sont nécessaires pour des solutions visant l'amélioration de plusieurs domaines tels que la recommandation de service, le transport, les réseaux cellulaires, etc. Une des solutions qui puisse avoir un apport positif dans ces domaines et qui nécessite l'utilisation de ce type de données est la *prédiction* de la mobilité.

À l'origine, les données de mobilité sont dans un état brut. Afin de les utiliser dans un processus de prédiction, un filtrage de données est nécessaire [1].

Les données filtrées seront utilisées par la suite pour la modélisation de la trajectoire [1], [2], [3] qui signifie ici la manière avec laquelle la trajectoire est décrite. Cette modélisation joue un rôle important dans le processus de prédiction et influence directement sur le résultat de la prédiction : plus la modélisation est meilleure, plus les résultats de prédiction le sont.

La plupart des travaux traitant de la mobilité, en particulier la prédiction de mobilité, utilisent soit les identificateurs (IDs) des cellules des réseaux cellulaires [4] soit une solution de clustering [5] pour décrire une trajectoire. La première alternative souffre du problème de chevauchement [4] qui provoque à son tour le phénomène d'oscillation. L'autre alternative souffre du problème de perte de précision.

Les travaux présentés dans ce mémoire contribuent à la problématique, proposer une technique de modélisation de la trajectoire qui permette de surmonter les problèmes cités plus haut. Afin de faire une prédiction de la trajectoire à l'aide du deep learning.

Dans le **Chapitre 1** nous présentons succinctement l'apprentissage automatique ainsi que la différence entre l'apprentissage supervisé et le non supervisé, également les différents algorithmes de deep Learning seront cités.

Dans le **chapitre 2**, nous abordons le concept de ville intelligente, dans un premier temps, nous expliquons le fondement de ce concept ainsi que les avantages qu'il présente. Ensuite, nous abordons l'apprentissage automatique, et présentons la différence entre le supervisé et le non supervisé, enfin nous définissons les différents algorithmes de **deep Learning**.

Dans le **chapitre 3**, dans une première phase, nous abordons les différents travaux sur les techniques et modèle de description de trajectoire, ensuite nous présentons une comparaison entre les modèles présentés, et après nous citerons les travaux réalisés dans les prédictions de la trajectoire.

Dans le **Chapitre 4**, nous présentons l'architecture globale de la solution proposée ainsi que notre approche de description de trajectoire, et modèle conceptuel, orienté prédiction.

Dans le **Chapitre 5**, nous énumérons toutes les étapes de la réalisation de notre solution et nous affichons les résultats obtenus pour l'ensemble des jeux de données utilisées sur les modèles de prédictions utilisés.

# Chapitre 1 Apprentissage automatique

## 1.1 Introduction

Dans ce chapitre nous introduisons les principes du « machine Learning » qui est une filiale de l'intelligence artificielle, nous allons définir l'apprentissage automatique et cité les différences entre l'apprentissage supervisé et non supervisé, ainsi vous trouverez une brève définition des réseaux de neurones et ces modèles, ce dernier est l'outil d'exercice de l'apprentissage.

## 1.2 Apprentissage automatique

Dans le domaine de l'apprentissage automatique, il existe deux principaux types de tâches, supervisées et non supervisées.

La principale différence entre les deux types est que l'apprentissage supervisé se fait à l'aide d'une vérité terrain, ou en d'autres termes, nous avons une connaissance préalable de ce que devraient être les valeurs de sortie de nos échantillons. Par conséquent, le but de l'apprentissage supervisé est d'apprendre une fonction qui, étant donné un échantillon de données et de sorties souhaitées, se rapproche le mieux de la relation entre l'entrée et la sortie observable dans les données. L'apprentissage non supervisé, en revanche, n'a pas de sorties étiquetées, son objectif est donc d'inférer la structure naturelle présente dans un ensemble de points de données [6].

### 1.2.1 Apprentissage supervisé

L'apprentissage supervisé est généralement effectué dans le contexte de la classification, lorsque nous voulons mapper les entrées sur les étiquettes de sortie, ou la régression, lorsque nous voulons mapper les entrées sur une sortie continue.

Les algorithmes courants en apprentissage supervisé incluent la régression logistique, les Naive baye, les SVM (Support Vector Machines), les réseaux de neurones artificiels (abordés en section 2) et les forêts aléatoires. Dans la régression comme dans la classification, l'objectif est de trouver des relations ou une structure spécifique dans les données d'entrée qui nous permettent de produire efficacement des données de sortie correctes [6].

Notez que la sortie «correcte» est entièrement déterminée à partir des données d'apprentissage, donc bien que nous ayons une vérité fondamentale que notre modèle supposera vraie, cela ne veut pas dire que les étiquettes de données sont toujours correctes dans des situations réelles.

Les étiquettes de données bruyantes ou incorrectes réduisent clairement l'efficacité de votre modèle.

Lors de la conduite d'un apprentissage supervisé, les principales considérations sont la complexité du modèle et le compromis biais-variance. Notez que les deux sont interdépendants.

## **1.2.2 Apprentissage non supervisé**

Les tâches les plus courantes dans l'apprentissage non supervisé sont le clustering, l'apprentissage des représentations et l'estimation de la densité. Dans tous ces cas, nous souhaitons apprendre la structure inhérente de nos données sans utiliser d'étiquettes fournies explicitement.

Certains algorithmes courants incluent le clustering k-means, l'analyse des composants principaux et les auto-encodeurs.

Étant donné qu'aucune étiquette n'est fournie, il n'existe aucun moyen spécifique de comparer les performances du modèle dans la plupart des méthodes d'apprentissage non supervisées.

Les cas d'utilisation courants de l'apprentissage non supervisé sont l'analyse exploratoire et la réduction de la dimensionnalité [6].

## **1.3 Les Réseaux de neurones**

### **1.3.1 Définition**

Les réseaux de neurones, communément appelés des réseaux de neurones artificiels sont des **imitations simples des fonctions d'un neurone dans le cerveau humain** pour résoudre des problématiques d'apprentissage de la machine (Machine Learning), le neurone est une unité qui est exprimée généralement par une fonction sigmoïde.

### **1.3.2 Algorithme d'apprentissage**

Les **algorithmes d'apprentissage** par renforcement sont basés sur des systèmes de récompenses et de punitions. L'**algorithme** se voit assigner un objectif et cherche à s'en rapprocher pour

obtenir une récompense maximale. Il se base sur des informations limitées et apprend de ses actions précédentes.

### **1.3.3 Différents modèles**

#### **I. CNN « Convolutional Neural Network »**

CNN est un type de modèle de réseau de neurones qui nous permet d'extraire des représentations plus élevées pour le contenu de l'image. Contrairement à la reconnaissance d'image classique où vous définissez vous-même les caractéristiques de l'image, CNN prend les données de pixels brutes de l'image, entraîne le modèle, puis extrait les caractéristiques automatiquement pour une meilleure classification.

Semblable au fonctionnement du cerveau humain, CNN distingue les caractéristiques significatives d'une image afin de classer l'image dans son ensemble.

Principes de CNN :

Convolution :

Une convolution balaie la fenêtre à travers les images puis calcule ses valeurs de pixel du produit scalaire d'entrée et de filtre, ce qui permet à la convolution de mettre en évidence les caractéristiques pertinentes.

Avec ce calcul, vous détectez une caractéristique particulière à partir de l'image d'entrée et produisez des cartes de caractéristiques (entités convoluées) qui mettent l'accent sur les caractéristiques importantes.

Ces caractéristiques convoluées changeront toujours en fonction des valeurs de filtre affectées par la descente de gradient pour minimiser la perte de prédiction.

De plus, plus il y a de filtres déployés, plus CNN extraira de fonctionnalités, cela permet de trouver plus de fonctionnalités mais avec le coût de plus de temps de formation.

Essentiellement, ces couches de convolution favorisent le partage du poids pour examiner les pixels dans les noyaux et développer un contexte visuel pour classer les images.

Contrairement au réseau neuronal (NN) où les poids sont indépendants, les poids de CNN sont attachés aux pixels voisins pour extraire des caractéristiques dans chaque partie de l'image [7].

Max Pooling :

CNN utilise la mise en commun maximale pour remplacer la sortie par un résumé maximal afin de réduire la taille des données et le temps de traitement, cela vous permet de déterminer les fonctionnalités qui produisent le plus d'impact et réduisent le risque de sur apprentissage.

La mise en commun maximale prend deux hyper paramètres : la foulée et la taille. La foulée déterminera le saut des pools de valeurs tandis que la taille déterminera la taille des pools de valeurs dans chaque saut.

Fonction d'activation (ReLU et Sigmoid) :

Après chaque opération de mise en commun convolutive et maximale, nous pouvons appliquer l'unité linéaire rectifiée (ReLU).

La fonction ReLU imite nos activations neuronales sur un « stimulus suffisamment important » pour introduire une non-linéarité pour les valeurs  $x > 0$  et renvoie 0 si elle ne remplit pas la condition.

Cette méthode a été efficace pour résoudre les gradients décroissants. Les poids très petits resteront à 0 après l'activation de la fonction ReLU.

Le CNN Big Picture + Couche entièrement connectée :

Enfin, nous servirons les sorties de carte de caractéristiques convolutives et de regroupement maximal avec une couche entièrement connectée (FCL), nous aplatirons les sorties de caractéristiques au vecteur de colonne et les transmettrons à FCL.

Nous enveloppons nos fonctionnalités avec la fonction d'activation softmax qui attribue des probabilités décimales pour chaque étiquette possible qui s'ajoutent à 1,0.

Chaque nœud de la couche précédente est connecté à la dernière couche et représente quelle étiquette distincte sortir.



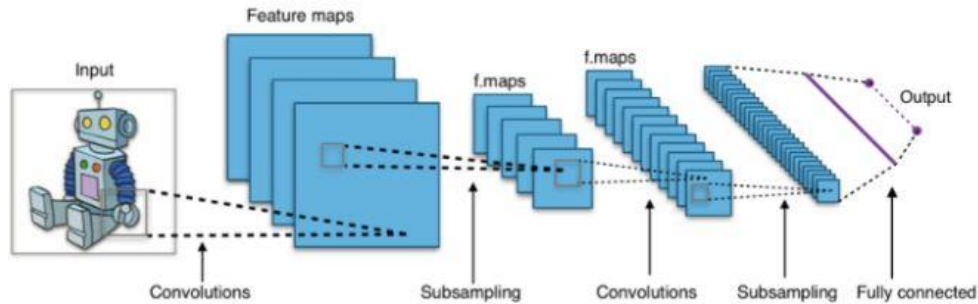


Figure 1. 1 architecture CNN

## II. RNN « recurrent neural network »

Les RNN sont un type de réseau de neurones puissant et robuste, et appartiennent aux algorithmes les plus prometteurs utilisés car c'est le seul à disposer d'une mémoire interne.

Comme beaucoup d'autres algorithmes d'apprentissage en profondeur, les réseaux de neurones récurrents sont relativement anciens. Ils ont été initialement créés dans les années 1980, mais ce n'est que ces dernières années que nous avons vu leur véritable potentiel. L'augmentation de la puissance de calcul, les quantités massives de données avec lesquelles nous devons maintenant travailler et l'invention de la mémoire à long terme (LSTM) dans les années 1990 ont vraiment mis les RNN au premier plan.

En raison de leur mémoire interne, les RNN peuvent se souvenir de choses importantes sur les entrées qu'ils ont reçues, ce qui leur permet d'être très précis pour prédire ce qui va suivre.

C'est pourquoi ils sont l'algorithme préféré pour les données séquentielles telles que les séries temporelles, la parole, le texte, les données financières, l'audio, la vidéo, la météo et bien plus encore.

Les réseaux de neurones récurrents peuvent former une compréhension beaucoup plus approfondie d'une séquence et de son contexte par rapport à d'autres algorithmes.

### Comment fonctionnent les réseaux de neurones récurrents :

Pour bien comprendre les RNN, vous aurez besoin d'une connaissance pratique des réseaux de neurones « normaux » et des données séquentielles.

Les données séquentielles sont essentiellement des données ordonnées dans lesquelles des éléments liés se succèdent. Des exemples sont des données financières ou la séquence d'ADN.

Le type de données séquentielles le plus populaire est peut-être les données de séries chronologiques, qui ne sont qu'une série de points de données répertoriés dans l'ordre chronologique.

Dans un RNN, les informations circulent en boucle. Lorsqu'il prend une décision, il considère l'entrée actuelle et également ce qu'il a appris des entrées qu'il a reçues précédemment.

Par conséquent, un RNN a deux entrées : le présent et le passé récent. Ceci est important car la séquence de données contient des informations cruciales sur ce qui va suivre, c'est pourquoi un RNN peut faire des choses que d'autres algorithmes ne peuvent pas faire.

### Deux problèmes de RNN standard :

Il y a deux obstacles majeurs auxquels les RNN ont dû faire face, mais pour les comprendre, vous devez d'abord savoir ce qu'est un gradient.

Un gradient est une dérivée partielle par rapport à ses entrées. Si vous ne savez pas ce que cela signifie, pensez-y comme ceci : un gradient mesure à quel point la sortie d'une fonction change si vous modifiez un peu les entrées.

#### 1. GRADIENTS EXPLOSIFS :

Les gradients explosifs surviennent lorsque l'algorithme, sans trop de raisons, attribue une importance stupidement élevée aux poids. Heureusement, ce problème peut être facilement résolu en tronquant ou en écrasant les dégradés.

#### 2. DÉGRADÉS ÉVASIONNÉS :

Les gradients de disparition se produisent lorsque les valeurs d'un gradient sont trop petites et que le modèle arrête d'apprendre ou prend beaucoup trop de temps en conséquence.

C'était un problème majeur dans les années 1990 et beaucoup plus difficile à résoudre que l'explosion des gradients, heureusement, il a été résolu grâce au concept de LSTM de Sepp Hochreiter et Juergen Schmidhuber [8].

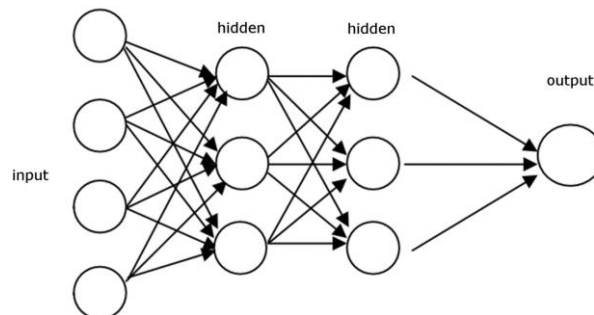


Figure 1. Zarchitecture RNN

## **LSTM (long short-term memory)**

Les réseaux de mémoire à long terme (LSTM) sont une extension des réseaux de neurones récurrents, qui étendent essentiellement la mémoire. Par conséquent, il est bien adapté pour tirer des leçons d'expériences importantes qui ont de très longs décalages entre les deux.

Les unités d'un LSTM sont utilisées comme unités de construction pour les couches d'un RNN, souvent appelé réseau LSTM.

Les LSTM permettent aux RNN de mémoriser les entrées sur une longue période de temps. En effet, les LSTM contiennent des informations dans une mémoire, un peu comme la mémoire d'un ordinateur, le LSTM peut lire, écrire et supprimer des informations de sa mémoire.

Cette mémoire peut être considérée comme une cellule fermée, ce qui signifie que la cellule décide de stocker ou non des informations (c'est-à-dire si elle ouvre les portes ou non), en fonction de l'importance qu'elle attribue aux informations.

L'attribution de l'importance se fait par le biais de poids, qui sont également appris par l'algorithme. Cela signifie simplement qu'il apprend au fil du temps quelles informations sont importantes et celles qui ne le sont pas.

Dans un LSTM, vous avez trois portes : porte d'entrée, d'oubli et de sortie, ces portes déterminent s'il faut ou non laisser entrer une nouvelle entrée (porte d'entrée), supprimer les informations car elles ne sont pas importantes (porte d'oubli) ou la laisser affecter le sortie au pas de temps courant (porte de sortie).

Les portes d'un LSTM sont analogiques sous la forme de sigmoïdes, ce qui signifie qu'elles vont de zéro à un. Le fait qu'ils soient analogiques leur permet de faire de la rétropropagation.

Les problèmes problématiques de la disparition des gradients sont résolus via LSTM car il maintient les gradients suffisamment raides, ce qui maintient la formation relativement courte et la précision élevée.

### Concept de base :

Le concept de base des LSTM est l'état de la cellule et ses différentes portes. L'état de la cellule agit comme une autoroute de transport qui transfère les informations relatives tout au long de la chaîne de séquence. Vous pouvez le considérer comme la « mémoire » du réseau.

L'état de la cellule, en théorie, peut transporter des informations pertinentes tout au long du traitement de la séquence.

Ainsi, même les informations des pas de temps antérieurs peuvent se retrouver dans les pas de temps ultérieurs, réduisant ainsi les effets de la mémoire à court terme.

Au fur et à mesure que l'état de la cellule poursuit son chemin, des informations sont ajoutées ou supprimées à l'état de la cellule via des portes. Les portes sont différents réseaux de neurones qui décident quelles informations sont autorisées sur l'état de la cellule. Les portes peuvent apprendre quelles informations sont pertinentes à conserver ou à oublier pendant la formation.

### Sigmoïde :

Gates contient des activations sigmoïdes. Une activation sigmoïde est similaire à l'activation tanh. Au lieu d'écraser les valeurs entre -1 et 1, il écrase les valeurs entre 0 et 1.

Cela est utile pour mettre à jour ou oublier les données car tout nombre multiplié par 0 est égal à 0, ce qui fait que les valeurs disparaissent ou sont « oubliées ». Tout nombre multiplié par 1 est la même valeur, donc cette valeur reste la même ou est « conservée ».

Le réseau peut apprendre quelles données ne sont pas importantes et peuvent donc être oubliées ou quelles données sont importantes à conserver.

### Forget Gate :

Premièrement, nous avons la porte d'oubli. Cette porte décide quelles informations doivent être jetées ou conservées. Les informations de l'état caché précédent et les informations de l'entrée actuelle sont transmises via la fonction sigmoïde. Les valeurs sont comprises entre 0 et 1. Le plus proche de 0 signifie oublier, et le plus proche de 1 signifie garder.

### Porte d'entrée :

Pour mettre à jour l'état de la cellule, nous avons la porte d'entrée, tout d'abord, nous passons l'état caché précédent et l'entrée actuelle dans une fonction sigmoïde.

Cela décide quelles valeurs seront mises à jour en transformant les valeurs entre 0 et 1. 0 signifie pas important, et 1 signifie important. Vous transmettez également l'état caché et l'entrée actuelle à la fonction tanh pour écraser les valeurs comprises entre -1 et 1 pour aider à réguler le réseau.

Ensuite, vous multipliez la sortie tanh avec la sortie sigmoïde, la sortie sigmoïde décidera quelles informations sont importantes à conserver de la sortie tanh.

#### État de la cellule :

Maintenant, nous devrions avoir suffisamment d'informations pour calculer l'état de la cellule. Tout d'abord, l'état de la cellule est multiplié par points par le vecteur d'oubli.

Cela a la possibilité de supprimer des valeurs dans l'état de la cellule s'il est multiplié par des valeurs proches de 0.

Ensuite, nous prenons la sortie de la porte d'entrée et effectuons une addition ponctuelle qui met à jour l'état de la cellule avec de nouvelles valeurs que le réseau de neurones juge pertinentes. Cela nous donne notre nouvel état de cellule.

#### Porte de sortie :

Enfin, nous avons la porte de sortie. La porte de sortie décide quel doit être le prochain état caché, n'oubliez pas que l'état caché contient des informations sur les entrées précédentes. L'état caché est également utilisé pour les prédictions.

Tout d'abord, nous passons l'état caché précédent et l'entrée actuelle dans une fonction sigmoïde. Ensuite, nous passons l'état de la cellule nouvellement modifié à la fonction tanh.

Nous multiplions la sortie tanh avec la sortie sigmoïde pour décider quelles informations l'état caché doit porter, la sortie est l'état caché, le nouvel état de la cellule et le nouveau caché est ensuite reporté au pas de temps suivant [9].

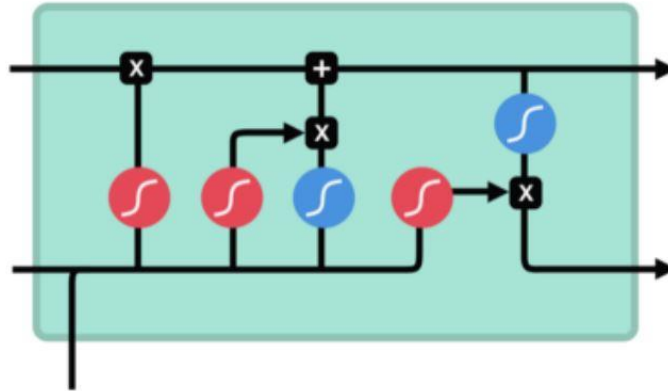


Figure 1. Architecture LSTM

### GRU (Gated Recurrent Units)

Le GRU peut être considéré comme une variante du LSTM, les deux sont conçus de la même manière et, dans certains cas, produisent des résultats tout aussi excellents.

Pour résoudre le problème de VanishingGradient d'un RNN standard, le GRU utilise ce qu'on appelle une porte de mise à jour(update gate) et une porte de réinitialisation (reset gate), ce sont essentiellement deux vecteurs qui décident quelles informations doivent être transmises à la sortie.

#### Update Gate :

La porte de mise à jour agit de manière similaire à la porte d'oubli et d'entrée d'un LSTM. Il décide quelles informations jeter et quelles nouvelles informations ajouter.

#### Reset gate :

La porte de réinitialisation est une autre porte utilisée pour décider de la quantité d'informations passées à oublier.

Les GRU ont moins d'opérations tensorielles ; par conséquent, ils sont un peu plus rapides à s'entraîner que les LSTM, il n'y a pas de gagnant clair lequel est le meilleur.

Les chercheurs et les ingénieurs essaient généralement les deux pour déterminer lequel fonctionne le mieux pour leur cas d'utilisation [10].

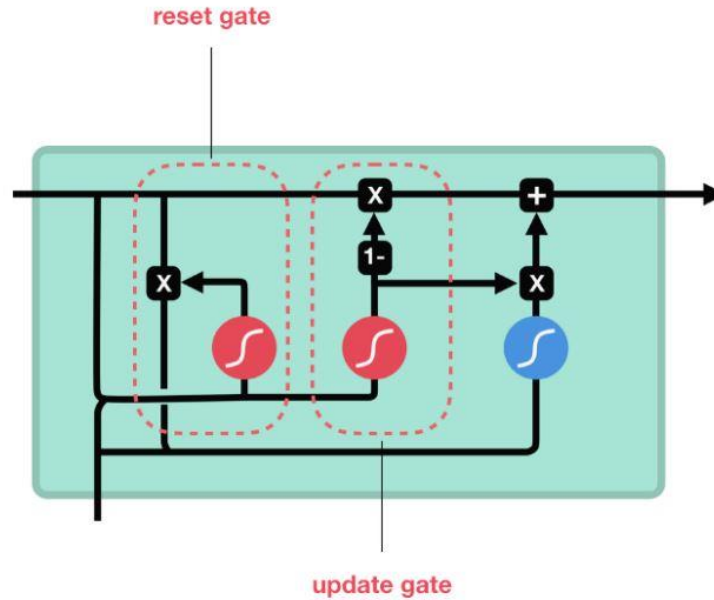


Figure 1. Architecture GRU

## Autoencoder

Les Autoencoders sont un type spécifique de réseaux de neurones à action directe où l'entrée est la même que la sortie, ils compressent l'entrée en un code de dimension inférieure, puis reconstruisent la sortie à partir de cette représentation.

Le code est un « résumé » compact ou une « compression » de l'entrée, également appelée représentation de l'espace latent.

Un auto-encodeur se compose de 3 composants : un encodeur, un code et un décodeur.

L'encodeur comprime l'entrée et produit le code, le décodeur reconstruit ensuite l'entrée uniquement à l'aide de ce code.

Les autoencoders sont principalement un algorithme de réduction de dimensionnalité (ou de compression) avec quelques propriétés importantes :

Spécifique aux données : les autoencoders ne sont capables de compresser de manière significative que des données similaires à celles sur lesquelles ils ont été formés. Comme ils apprennent des fonctionnalités spécifiques aux données d'entraînement données, ils sont différents d'un algorithme de compression de données standard comme gzip. Nous ne pouvons

donc pas nous attendre à ce qu'un encodeur automatique formé sur des chiffres manuscrits compresse des photos de paysage.

Lossy : La sortie de l'autoencodeur ne sera pas exactement la même que l'entrée, ce sera une représentation proche mais dégradée. Si vous voulez une compression sans perte, ce n'est pas la solution.

Non supervisé : pour entraîner un encodeur automatique, nous n'avons pas besoin de faire quoi que ce soit d'extraordinaire, il suffit de lui envoyer les données d'entrée brutes. Les autoencodeurs sont considérés comme une technique d'apprentissage non supervisée car ils n'ont pas besoin d'étiquettes explicites pour s'entraîner. Mais pour être plus précis, ils sont auto-supervisés car ils génèrent leurs propres étiquettes à partir des données d'apprentissage.

### Architecture :

L'encodeur et le décodeur sont tous deux des réseaux de neurones à action directe entièrement connectés.

Il s'agit d'une visualisation plus détaillée d'un auto-encodeur. L'entrée passe d'abord par l'encodeur, qui est un ANN entièrement connecté, pour produire le code.

Le décodeur, qui a la structure ANN similaire, produit alors la sortie uniquement en utilisant le code.

Le but est d'obtenir une sortie identique à l'entrée. A noter que l'architecture du décodeur est l'image miroir de l'encodeur.

Ce n'est pas une obligation mais c'est généralement le cas, la seule exigence est que la dimensionnalité de l'entrée et de la sortie doit être la même.

### Cas d'utilisation :

Réduction de dimensionnalité :

visualiser des données de grande dimension est difficile, t-SNE est la méthode la plus couramment utilisée, mais se débat avec un grand nombre de dimensions (généralement supérieur à 32).



Ainsi, les auto-encodeurs sont utilisés comme étape de prétraitement pour réduire la dimensionnalité, et cette représentation compressée est utilisée par t-SNE pour visualiser les données dans l'espace 2D, pour d'excellents articles sur t-SNE référez-vous ici et ici.

Les auto-encodeurs sont une technique de réduction de dimensionnalité très utile. Ils sont très populaires comme matériel d'enseignement dans l'apprentissage profond d'introduction [11].

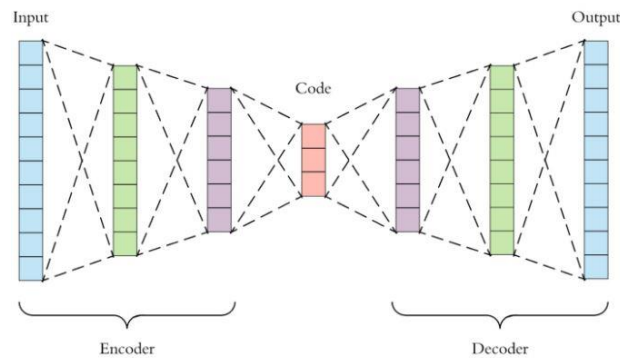


Figure 1. 5architectureautoencoder

### **GAN « Generative adversarial network »**

Le GAN fonctionne sur la base de trois principes, tout d'abord pour faire apprendre le modèle génératif, les données peuvent être générées en utilisant une représentation probabiliste.

Deuxièmement, la formation d'un modèle peut se faire dans n'importe quelle situation conflictuelle, enfin en utilisant les réseaux de neurones d'apprentissage profond et en utilisant les algorithmes d'intelligence artificielle pour former le système complet [12].

L'idée de base du déploiement du réseau GAN concerne les techniques de ML non supervisées, mais s'est également avérée être de meilleures solutions pour l'apprentissage semi-supervisé et par renforcement.

Ensemble, ces facteurs permettent aux réseaux GAN de constituer des solutions complètes dans de nombreux domaines tels que la santé, la mécanique, la banque, etc.

## Limites du GAN :

Cependant, l'architecture GAN à certaines limites, les images créées par GAN ressemblent de manière trompeuse à une photographie d'une personne réelle basée sur l'analyse des portraits.

Différentes inquiétudes de la population ont été soulevées pour l'utilisation potentielle de la synthèse d'images humaines par le GAN par des fraudes, produisant ainsi le faux, des photographies et des vidéos sans autorisation.

Sur les réseaux sociaux, les faux profils peuvent être évités en utilisant les GAN pour générer des images uniques ou pragmatiques de personnes qui n'existent pas.

Les programmes Media Forensics de la DARPA aident à lutter contre ces faux profils médiatiques produits à l'aide de GAN et, parallèlement à cela, de nombreuses lois sont adoptées et mises en œuvre d'ici 2020.

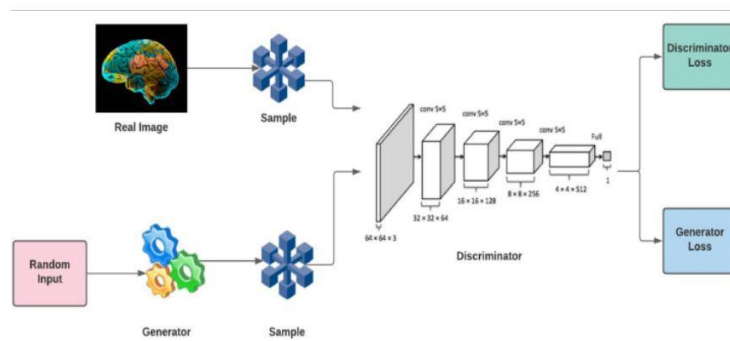


Figure 1. Architecture GAN

Les GAN sont établis de manière à couvrir le domaine complet de l'expansion des données indépendantes et résolvent les problèmes qui nécessitent une solution générative telle que la transformation d'image en image [13].

## 1.4 Conclusion

Nous avons vu les cas d'utilisation de chaque modèle des réseaux de neurones ainsi que leur limite, nous avons fait une comparaison entre l'apprentissage supervisé, non supervisé et

définit de l'apprentissage automatique. Par la définition et les divers exemples des réseaux de neurones nous avons vu l'importance de cette dernière dans l'apprentissage.

# Chapitre 2 La mobilité dans les villes intelligentes

## 2.1 Introduction

Ces dernières années, nous avons constaté une montée exponentielle des données de mobilités dans les grandes villes mondiales, ceci est dû non seulement à l'émergence des capteurs dans les outils dite intelligents comme les smartphone et tablette mais aussi à cause de l'évolution de la mobilité humaine dans des zones encombrée. Dans ce chapitre nous allons voir les principaux composants des villes intelligentes et les défis rencontrés, ensuite nous allons illustrer la mobilité dans les villes intelligentes, les techniques d'acquisition de localisation et la relation entre le MCSC et les V.I.

## 2.2 Ville intelligente

### 2.2.1 Définition

**Définition.1** : De nos jours, le concept de Smart City est encore vague, on peut saisir les limites auxquelles on se réfère (technologie, communication, durabilité) et les objectifs généraux, mais ce terme a encore des interprétations différentes.

Depuis l'an 2000, une ville est considérée comme intelligente non seulement lorsqu'elle dispose d'une forte présence technologique, mais aussi lorsque ses citoyens sont capables d'utiliser cette technologie pour obtenir une meilleure qualité de vie, en concentrant l'attention sur le rôle du capital humain et social.

Définissant le concept de Smart City, NicosKomninos affirme qu'une ville peut être considérée comme intelligente lorsqu'elle « se caractérise par un haut niveau d'innovation soutenu par la créativité des citoyens, par des institutions capables de générer des connaissances et par des services numériques pour l'administration » [14].

De plus, une recherche récente a mis en évidence six caractéristiques essentielles d'une ville à considérer comme intelligente :

1) l'utilisation « d'infrastructures en réseau pour améliorer l'efficacité économique et politique et permettre le développement social, culturel et urbain » [Hollande].

Le terme infrastructure désigne les services aux entreprises, le logement, les services de loisirs et de style de vie, et les TIC (téléphones mobiles et fixes, télévisions par satellite, réseaux informatiques, commerce électronique, services Internet).

2) le soutien aux nouvelles entreprises avec un accent particulier sur les activités entrepreneuriales orientées vers l'innovation et le développement urbain.

3) une attention particulière à l'inclusion sociale, favorisant un développement homogène tant en centre-ville qu'en périphérie.

4) une forte concentration sur le rôle de l'industrie de la haute technologie et de la créativité pour un développement urbain à long terme.

5) une attention à la participation active des citoyens à la vie publique, en soutenant les relations sociales et le capital humain.

6) enfin, un souci particulier de la durabilité comme composante essentielle d'une stratégie de développement urbain.

**Définition.2 :** Une ville intelligente est une ville qui contient différents capteurs électroniques dans tous les secteurs qui constituent cette ville pour collecter des données qui aide à la gestion des ressources et tous les services urbains de la ville intelligente.

Pour la réalisation du concept de la ville intelligente, il faut implémenter les technologies de l'information et communication(TIC), et l'internet des objets (IOT) et divers dispositifs sous forme physique connecté au réseau qui communique avec le système.

Dans notre cas, on va s'intéresser à la dernière dimension de mobilité intelligente.

Les solutions permettant de faciliter les déplacements des hommes ou des marchandises heurtent à de nouvelles contraintes, en particulier budgétaires et environnementales .dans ce contexte, et pour continuer à répondre aux besoins d'échanges, il importe d'imaginer des alternatives aux solutions basées uniquement sur une croissance extensive des réseaux routiers ou ferroviaires.

C'est à ce stade qu'intervient la notion de mobilité intelligente, expression désignant générales l'application des nouvelles technologies au domaine du transport

### **2.2.2 Les dimensions principales des initiatives dans les villes intelligentes**

Les six composantes de la ville intelligente présentées dans les schémas de Giffingerson sont illustrées de façon non hiérarchique, tel que classé dans **le tableau 1** , chacune des composantes est définie selon une hiérarchie proposée, cette hiérarchie est en fait une suite logique de processus ou étapes par lesquels une ville deviendra intelligente.

Avant le tout début d'une démarche, il est essentiel d'avoir l'appui de l'instance qui gouverne et par le fait même avoir une bonne gouvernance qui est adaptée et spécifique à la démarche et aux objectifs visés.

Ensuite, dans la ville intelligente le citoyen est une partie prenante qui participera de près ou de loin à la réussite de cette démarche.

Une fois que les deux dimensions précédentes sont ancrées et bien établies dans la démarche, les quatre autres dimensions sont d'égale importance.

C'est à-dire que l'on doit toutes les considérer si l'on veut devenir une ville intelligente, et ce, à importance égale.

Tableau 1 : dimension du concept de ville intelligentes avec leur objectif et résultat.

DIMENSIONS DE LA VILLE INTELLIGENTE	DÉFIS URBAINS ACTUELS	OUTILS DE MISE EN OEUVRE	OBJECTIFS ET RÉSULTATS VISÉS
Gouvernance intelligente	Gouvernance fermée et peu transparente.	<ul style="list-style-type: none"> <li>•Panneaux informatifs électroniques</li> <li>• Web diffusion</li> <li>• Logiciel pour le suivi des requêtes en ligne</li> </ul>	Gouvernance intégrée, transparente, ouverte et favorisant la coopération.
Citoyen intelligent	La qualité des services rendus aux citoyens, La participation citoyenne.	<ul style="list-style-type: none"> <li>• Plateforme en ligne pour favoriser la participation du public</li> </ul>	Favoriser la créativité et la flexibilité des citoyens et la participation à la communauté dans le but d'améliorer son milieu de vie.
Économie intelligente	Difficulté d'attirer de nouvelles entreprises au centre de la ville.	<ul style="list-style-type: none"> <li>• Transactions en lignes</li> <li>• Échanges de données informatisées</li> </ul>	Attirer des entreprises et de la main-d'œuvre, créer des emplois durables créant de la valeur.
Mobilité intelligente	Infrastructures routières vieillissantes, congestion du réseau routier, demande croissante en transport en commun.	<ul style="list-style-type: none"> <li>• Systèmes de géolocalisation</li> <li>•bornes de recharge électriques</li> <li>• écrans électroniques</li> <li>• Application cellulaire</li> <li>• Caméras</li> </ul>	Optimiser les différents systèmes de transport en commun et transport actif, le tout coordonné efficacement par des systèmes technologiques.
Environnement	Consommation des	<ul style="list-style-type: none"> <li>• Capteurs</li> </ul>	Utiliser les ressources

intelligent	ressources naturelles, gestion de l'eau, gestion des matières résiduelles.	<ul style="list-style-type: none"> <li>• senseurs</li> <li>• smart grids</li> </ul>	naturelles de façon durable et protéger l'environnement naturel. Favoriser la planification et l'aménagement durable du territoire. Créer un environnement sain qui favorise une vie de quartier.
Habitat intelligent	Gestion du développement urbain, étalement urbain, qualité et accessibilité aux services offerts.	<ul style="list-style-type: none"> <li>•Plateforme d'échange web pour les quartiers</li> <li>•Nouvelles technologies pour les habitations écologiques</li> </ul>	Crée un milieu de vie sécuritaire qui rassemble des composantes culturelles, des services de santé et d'éducation, qui offre des bâtiments de qualité et où il y a une cohésion sociale.

### 2.3 Mobilité dans les villes intelligentes

L'accès aux données de transport en temps réel via des écrans électroniques dans les stations, dans les wagons de métro ou dans les autobus ou encore via les téléphones intelligents personnels permettrait aux usagers de connaître une foule d'informations.

C'est-à-dire, l'état de la circulation sur le réseau routier, le temps d'attente aux arrêts et stations de transport en commun, les pannes et en somme une meilleure gestion des flux urbains. Une mobilité intelligente qui serait possible grâce aux divers centres de gestion des données, aux capteurs d'informations et aux caméras.

Ainsi, les utilisateurs des transports deviennent des producteurs de données. Une mobilité intelligente passe aussi par le développement et l'accès aux applications qui permettront aux usagers de vivre l'expérience d'une mobilité intelligente [18].

#### 2.3.1 La collecte des données de mobilité dans les VI

Dans le transport et la mobilité, trois principales sources de Big Data peuvent être identifiées : les médias sociaux (non structurés), les données de capteurs (non structurées ou semi-structurées) et

les données ouvertes (soit non structurées - texte brut -, semi-structurées - JSON/XML - soit sémantiquement structurées – Données ouvertes liées) [29].

Dans ce tableau on va présenter les différents sources de données de la mobilité et du transport :

Tableau.2 :Principales sources de données liées aux domaines du transport et de la mobilité [86].

Source traditionnelle	Source	type	format	exemple
	ITS	structurée	CSV, XML etc.	données de navigation automobile ou de contrôle des feux de circulation
	legacysystems (RDBMS)	structurée	ERM,XML, JSON	base de données des usagers du télépéage
	GISdata and maps	Semi-structurée	GIS	carte routière d'une région
Nouvelle source de données potentiel	Open Data	Non-structurée	text, CSV	Smart Cities open data
	Linked Data	Semi-structurée	RDF	données d'enrichissement sémantique de la mobilité urbaine
	Sensor Data	données brutes	text, CSV, sensor formats	De nouveaux capteurs sur les routes pour mesurer l'état de la chaussée
	Social Media	Semi-structurée	text, CSV, XML, RDF,etc.	Données des appareils mobiles de l'utilisateur

### 2.3.2 Technique d'acquisition de données

Les TIC sont la technologie de base qui permet aux données collectées à partir d'appareils de ville intelligente, tels que les appareils de maison intelligente et les téléphones mobiles, d'être transférées efficacement vers un centre de données et facilement partagées et échangées entre les appareils connectés [14].



Par exemple, la RFID (identification par radiofréquence) est une technologie utilisée pour l'identification et la localisation d'un objet en mouvement basée sur des champs électromagnétiques. Une étiquette RFID peut être intégrée dans pratiquement n'importe quel objet, allant des produits industriels aux animaux en passant par les êtres humains, à des fins de suivi à la fois extérieur et intérieur.

L'identification par radiofréquence "RFID" trouve de nombreuses applications dans les villes intelligentes, telles que la gestion des déchets en temps réel [15], le stationnement intelligent [16] et la détection et l'évitement des collisions dans les transports [17].

Le WiFi est une technologie sans fil à courte portée utilisée pour fournir aux utilisateurs finaux un accès à distance à haut débit à Internet. Par rapport aux réseaux câblés standard, un réseau de communication sans fil activé par Wifi offre une meilleure flexibilité et mobilité pour les utilisateurs des villes intelligentes.

Soundararaj et al. (sous presse) décrivent une utilisation intéressante de la sonde Wifi demandes» pour comprendre les niveaux d'activité des piétons dans les centres commerciaux à travers la Grande-Bretagne.

Les demandes de sonde Wifi sont générées à partir d'appareils mobiles en présence d'un point d'accès (AP) Wifi, tel qu'un routeur sans fil.

Les informations de localisation d'un appareil mobile peuvent être obtenues par triangulation par plusieurs points d'accès.

Dans les environnements urbains denses où il y a une couverture Wifi presque partout, cette sonde de demande de données devient une source précieuse pour collecter des données de consommation en temps réel, comme la durée d'une visite dans un magasin de détail, ou un décompte de piétons dans les foules qui affluent autour de diverses installations.

De nombreuses applications commerciales intelligentes peuvent bénéficier de ces techniques à faible coût et à haute disponibilité, telles que l'évaluation des tarifs commerciaux sur les propriétés ou l'estimation des valeurs locatives des unités [18].

La technologie de positionnement intérieur en temps réel basée sur le Wifi peut également être utilisée dans d'autres applications de ville intelligente, telles que la cartographie de la population urbaine [17], l'analyse des mouvements humains [20] et le contrôle automatique de la maison intelligente, dispositifs [22].

## 2.4 Le MCSC et sa relation avec les villes intelligentes

**Mobile CrowdSensing and Computing** : est un paradigme de détection à grande échelle basé sur la puissance des appareils associés à l'utilisateur, y compris les téléphones mobiles, les véhicules intelligents, les appareils portables, etc.

MCSC permet au nombre croissant d'utilisateurs de téléphones mobiles de partager les connaissances locales (par exemple, les informations locales, le contexte ambiant, le niveau de bruit et les conditions de circulation) acquises par leurs appareils améliorés par capteurs, et les informations peuvent être agrégées davantage dans le cloud pour de grandes la détection d'échelle et l'exploration de l'intelligence communautaire.

La mobilité des utilisateurs mobiles à grande échelle fait de MCSC une plate-forme polyvalente qui peut souvent remplacer les infrastructures de détection statiques.

Un large éventail d'applications est ainsi activé, notamment la planification du trafic, la surveillance de l'environnement, la recommandation sociale mobile, la sécurité publique, etc.

La détection et l'informatique mobiles de foule, est une technique où un grand groupe d'individus ayant des appareils mobiles capables de détecter et d'informatique (tels que les smartphones, les tablettes, les appareils portables) partagent collectivement des données et extraient des informations pour mesurer, cartographier, analyser, estimer ou déduire (prédire) tout processus d'intérêt commun.

En bref, cela signifie crowdsourcing des données de capteurs à partir d'appareils mobiles. Avec l'essor des techniques de détection par smartphone, de réseau sans fil et de réseautage social mobile, la détection et l'informatique mobiles de foule (MCSC) est devenu un paradigme prometteur pour la détection à grande échelle et dans l'espace.

MCSC étend la vision de la détection participative en exploitant à la fois les données sensorielles participatives, les données des appareils mobiles (hors ligne) et les données fournies par les utilisateurs des services de réseaux sociaux mobiles (en ligne).

En outre, il explore les rôles complémentaires et présente la fusion/collaboration de l'intelligence machine et humaine dans les processus de détection de foule et de calcul [28].

## 2.5 La géolocalisation dans les villes intelligentes

Les villes produisent de nombreux types de données géospatiales en temps réel, les données de flux de trafic ont été parmi les premières à être automatiquement collectées par des appareils numériques [24].

Ces dernières années, les embouteillages et la pollution liée à la circulation sont devenus un grave problème non seulement causant des pertes économiques importantes, mais affectant également négativement la qualité de vie des citoyens. Par conséquent, la construction d'un système intelligent de gestion du trafic devient essentielle pour augmenter l'efficacité et la sécurité des transports urbains.

Les sources de données spatio-temporelles typiques sur la mobilité humaine comprennent les enregistrements détaillés des appels (CDR), les enregistrements Wifi AP, les traces GPS, les médias sociaux géolocalisés et les enregistrements de cartes à puce des transports en commun.

Le CDR capture la communication entre le téléphone mobile et les tours cellulaires à proximité par rapport à d'autres sources, il est massif en raison du grand nombre d'utilisateurs de téléphones mobiles, mais éparpillés dans le temps car un enregistrement n'est généré que lorsqu'il y a une utilisation active du téléphone. Le GPS, quant à lui, offre un positionnement plus précis et une meilleure continuité temporelle pour les agents en mouvement. Ces dernières années, le GPS à bord des véhicules publics, tels que les taxis et les vélos en libre-service, permet la collecte de traces GPS pour quantifier la mobilité humaine basée sur l'activité, générant des matrices origine-destination [25], et délimiter des zones fonctionnelles urbaines, parmi de nombreuses autres applications urbaines [26].

Les données des cartes à puce fournissent une autre mesure de la mobilité urbaine en enregistrant l'heure et la gare à laquelle un passager entre et sort d'un service de transport public.

Ces données ont souvent une résolution temporelle fine, pour donner un exemple, le système de carte Oyster de Londres génère 9 millions de transactions par jour pour la seule utilisation du train [27], soit plus de 100 transactions par seconde.

## **2.6 Conclusion**

Le concept des villes intelligentes est orienté par des dimensions, chaque une avec des défis. Nous devons mettre en œuvre des outils pour atteindre les résultats et comportement voulu comme détaillé ci-dessus. Nous avons vu aussi dans ce chapitre les principales sources de données liées aux domaines du transport et de la mobilité et les techniques de géolocalisation, ainsi la relation entre le MCSC et les V.I.

# Chapitre 3 Modélisation de la trajectoire : et travaux liées à la prédiction

## 3.1 Introduction

Modéliser la trajectoire humaine a tant préoccupé les chercheurs dans ce domaine, on a défini dans ce chapitre la trajectoire et sa différence avec le mouvement, on va décrire la trajectoire par plusieurs techniques, ensuite, on a les modèles de trajectoire existant.

## 3.2 Définition d'une trajectoire

La trajectoire d'un objet est la courbe qui est dessinée par une série de points de l'objet mobile, chaque point est imprimé d'un temps  $T$  et d'une coordonnée de l'espace qui sont bornés par un premier point qui désigne le début de la trajectoire et un point final qui représente l'arrivée et la fin de la courbe, la trajectoire peut être définie aussi par l'ajout d'autres éléments tels que la vitesse et l'accélération et la distance qui a été parcourue.

Certaines positions sont particulièrement pertinentes, POI : « point of interest », différent d'un objet mobile à un autre, un lieu de travail pour un piéton, un port pour un navire et un point d'eau pour un animal, la définition d'un POI est variable selon la **technique de descriptions** basée dans le travail, on peut par exemple définir un POI par les lieux, zone ou région où l'objet en mouvement a passé plus d'une minute.

Un autre concept qui est dérivé du concept de POI : c'est que la trajectoire d'objets mobiles est la liaison entre deux POI, un qui se présente comme le POI de départ et le deuxième le POI de l'arrivée de l'objet mobile en question [1].

Pour ne pas confondre un mouvement avec une trajectoire, voici une figure 3.1 explicative :

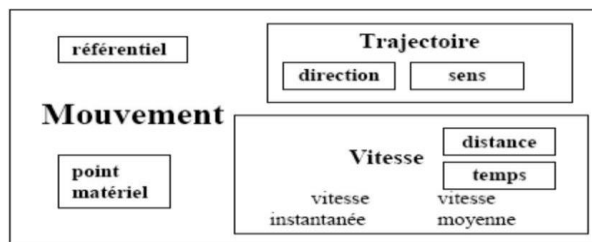
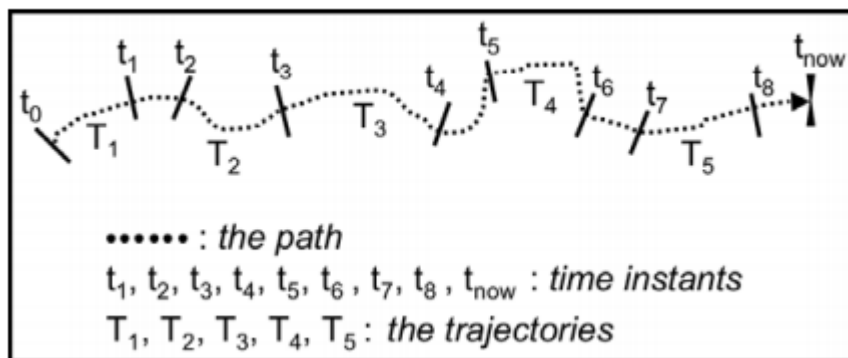


Figure 3.2 représentation du mouvement et la trajectoire[1]

Une trajectoire est l'enregistrement défini par l'utilisateur de l'évolution de la position (perçue comme un point) d'un objet qui se déplace dans l'espace pendant un intervalle de temps donné afin d'atteindre un but donné, la fonction espace-temps est définie par l'utilisateur et n'est pas nécessairement celle fourni par le mécanisme d'acquisition de données.

Ces dernières sont les données brutes, dont la forme habituellement est une séquence de paires (point d'échantillonnage, temps), Les données brutes doivent souvent subir un processus de nettoyage pour corriger les erreurs et approximations dans l'acquisition des données... Une autre différence importante entre un objet avec des trajectoires et un objet en mouvement (appelé aussi objet à géométrie variable dans le temps dans) est que, pour un objet en mouvement, la fonction d'espace-temps qui décrit sa position est définie sur toute la durée de vie de l'objet. Au lieu de cela, une trajectoire est donnée en restreignant la fonction à un intervalle de temps spécifique, [T début, T fin] inclus dans la durée de vie de l'objet.

Une trajectoire est un segment du chemin spatio-temporel parcouru par un objet en mouvement comme indique la figure 3.3ci-dessous [1] :



.Figure 3.4 Un chemin spatio-temporel pour un objet en mouvement et ses trajectoires définies par une segmentation sémantique du chemin. Dans cet exemple des parties du chemin n'appartiennent à aucune trajectoire : elles correspondent à des déplacements sans rapport avec l'application[1].

### 3.3 Technique de description d'une trajectoire

La modélisation de trajectoire est une abstraction du mouvement de l'utilisateur requis pour la prédiction de trajectoire. Selon la façon dont l'emplacement d'une unité mobile est représenté, il existe deux méthodes de modélisation : modélisation de trajectoire à base de cellules et modélisation de trajectoire à base de réseau.

Dans la modélisation cellulaire, la zone géographique est divisée en cellules de forme régulière, généralement déterminé par l'architecture du réseau cellulaire, et l'emplacement de l'unité mobile est déterminé comme la cellule dans laquelle il est contenu.

### **I. Technique basée sur les cellules**

La trajectoire est exprimée comme une série de cellules faisant le chemin de l'unité mobile, et le problème de la prédiction de trajectoire consiste à déterminer le prochain ensemble de cellules que l'unité mobile suivra.

Cependant, cette méthode ne prend pas en compte la géométrie et la topologie du chemin réel, et ne peut pas localiser avec précision un utilisateur mobile car le rayon d'une cellule typique est de 150 m et plus [31].

### **II. Technique basée sur les régions**

L'une des techniques les plus utilisées repose sur la localisation par GPS, son principe de fonctionnement est que le mobile envoie la position obtenue par GPS à sa station de base. Cette dernière détermine si le mobile est au bord de sa cellule. A chaque réception de la position du mobile, le système calcule la distance séparant le mobile des cellules voisines, la plus courte distance est sélectionnée [30].

## **3.4 Modèles de trajectoire existants : mentionner les différents travaux**

### **I. Modèle basic**

Le modèle présenté dans cette partie est une base de travail dans plusieurs travaux sur l'analyse des données de trajectoire, la modélisation et l'exploration, pour cela, on va l'identifier et d'extrapoler avec des données plus complexes Ce modèle a été introduit en 2008 par Spaccapietra et al. dans le projet européen «SemanticEnrichment of TrajectoryKnowledgeDiscovery»

En général, les travaux existants traitent des trajectoires brutes, où une trajectoire est une liste de points situés dans l'espace et le temps. On commence donc par définir un point.

**Définition 1. Point.** Un point  $p$  est un tuple  $(x,y,t)$ , où  $x$  et  $y$  sont les coordonnées spatiales qui représentent un lieu et  $t$  est l'horodatage auquel le point a été collecté. Une trajectoire brute est une liste ordonnée de points, telle que présentée dans la définition 2.

**Définition 2. Trajectoire.** Une trajectoire  $T$  est une liste ordonnée de points  $\langle p_1, p_2, p_3, \dots, p_n \rangle$ , où  $p_j = (x_j, y_j, t_j)$  et  $t_1 < t_2 < t_3 < \dots < t_n$ . Une trajectoire a plusieurs caractéristiques que nous appelons caractéristiques physiques qui peuvent être soit lié à la trajectoire soit extraites des points tels que l'accélération, la vitesse, le changement de direction, la longueur, etc. Une trajectoire peut être divisée en différentes parties appelées sous-trajectoires. Une sous-trajectoire peut avoir plusieurs définitions mais en général dans la littérature, une sous-trajectoire est définie comme un segment de la trajectoire brute. Par conséquent, une sous-trajectoire est elle-même une trajectoire brute. La définition 3 présente ce concept.

**Définition 3. Sous-trajectoire.** Une sous-trajectoire  $s$  de  $T$  est une liste de points consécutifs  $\langle p_k, p_{k+1}, \dots, p_{k+l} \rangle$ , où  $p_i \in T$ ,  $k \geq 1$  et  $k+l \leq n$ , après avoir défini des trajectoires et des sous-trajectoires brutes, il est nécessaire de définir le **contexte** pour définir plus tard des trajectoires sémantiques.

La notion de contexte est approfondie dans la littérature, en particulier concernant les appareils mobiles sensibles au contexte, comme par exemple dans le travail de Dourich (2004). Cependant, pour les besoins de cet article, nous utilisons le terme contexte dans un sens plus spécifique, où le contexte englobe les informations externes liées au déplacement objet et la trajectoire créée à partir de son mouvement.

**Définition 4. Contexte.** Le contexte est une information sur l'objet en mouvement, provenant soit du monde réel où l'objet se déplace ou plus abstraitement de la connaissance du domaine d'application. Des exemples d'informations contextuelles sur un objet en mouvement peuvent être la température de l'objet, la température à l'endroit où l'objet passe, la météo, le transport moyen utilisé par l'objet à déplacer, l'objectif du mouvement ou l'activité exercée en se déplaçant.

Un autre concept important concernant les trajectoires est le comportement de l'objet en mouvement. Le terme comportement a un sens très large et, en général, fait référence à la façon dont les organismes réagissent à l'environnement. Pour les besoins de ce travail, nous utilisons le terme de comportement appliqué aux données de trajectoire inspiré de la définition de Parent et al. (2013). [35]

Ce modèle est considéré comme étant le premier modèle sémantique introduit en 2008 :



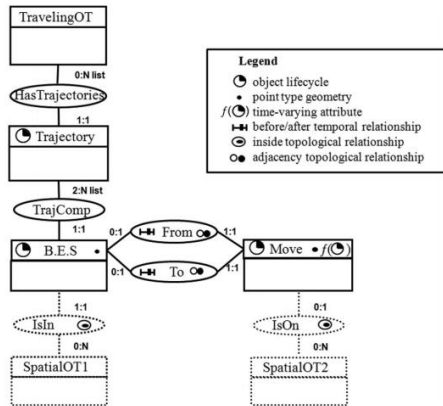


Figure 3. 5Modélisation de trajectoire[11]

## II. L'approche de Laube et al. (2005)

Définit cinq types de trajectoires de comportements collectifs : Convergence, Rencontre, Récurrence, Troupeau et Leadership, les auteurs explorent les épisodes de collocation en données spatio-temporelles.

L'objectif principal est de trouver des objets qui se déplacent ensemble pendant un certain temps et font bouger un autre objet avec eux. Par conséquent, le concept de fenêtre temporelle est utilisé, où les trajectoires sont divisées en tranches de temps, et la relation entre les objets est identifiée par la distance entre les points dans chaque fenêtre de temps. Cette méthode peut être utilisée pour découvrir les comportements d'un puma chassant une chèvre [36], ci-dessous le diagramme utilisé :

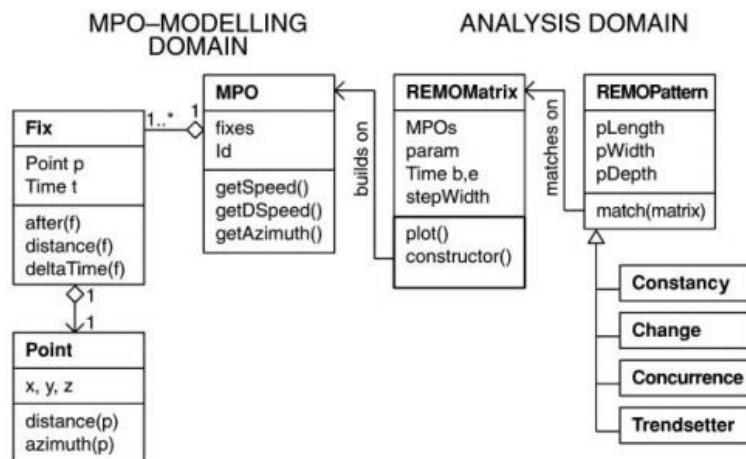


Figure 3. 6L'Aube et Al[36].

### III. Spaccapietra et al. (2008):

Certains travaux identifient des comportements communs dans des trajectoires uniques, essayant de comprendre les comportements de l'objet en analysant **les mouvements individuels**, par exemple, la méthode CB-SMOT qui utilise l'idée d'arrêts et de pour trouver des sous-trajectoires (arrêts) où l'objet se déplace lentement. Un nouveau comportement des trajectoires est introduit pour trouver des sous-trajectoires qui évitent des points/régions, comme un voleur évitant une caméra de surveillance. Il évalue également la confiance du modèle, pour s'assurer qu'il s'agissait d'un évitement intentionnel. **Rocha et al. (2010)** proposent l'algorithme DB-SMOT pour trouver des sous-trajectoires dans des trajectoires uniques où le changement de direction caractérise le comportement ; par exemple, un navire dans une région de pêche [37]

### IV. Spinsanti et al. (2010) et Baglioni et al. (2009) :

Dans les approches proposées par **Spinsanti et al. (2010) et Baglioni et al. (2009)** le terme **le comportement** est utilisé comme synonyme de but, les deux approches visent à annoter des trajectoires avec des buts déduit d'un ensemble d'objectifs prédéfinis. Dans Spinsanti et al. (2010) une trajectoire sémantique est associée à un objectif lié à un arrêt, par exemple, une trajectoire qui a une fréquence élevée de les arrêts aux points d'intérêt touristiques sont associés à un objectif « tourisme ». Cependant, cette approche n'associe pas de buts à des sous-trajectoires.

Les travaux de **Baglioni et al. (2009)** introduit un système appelé **Athena** qui annote des trajectoires et des modèles avec des objectifs prédéfinis (par ex. Devoirs, Tourisme) utilisant les connaissances du domaine encodées dans une ontologie comme le précédent travail, cependant, ils ne considèrent pas les sous-trajectoires; la section suivante présente le modèle proposé et plusieurs définitions liées à une trajectoire sémantique :

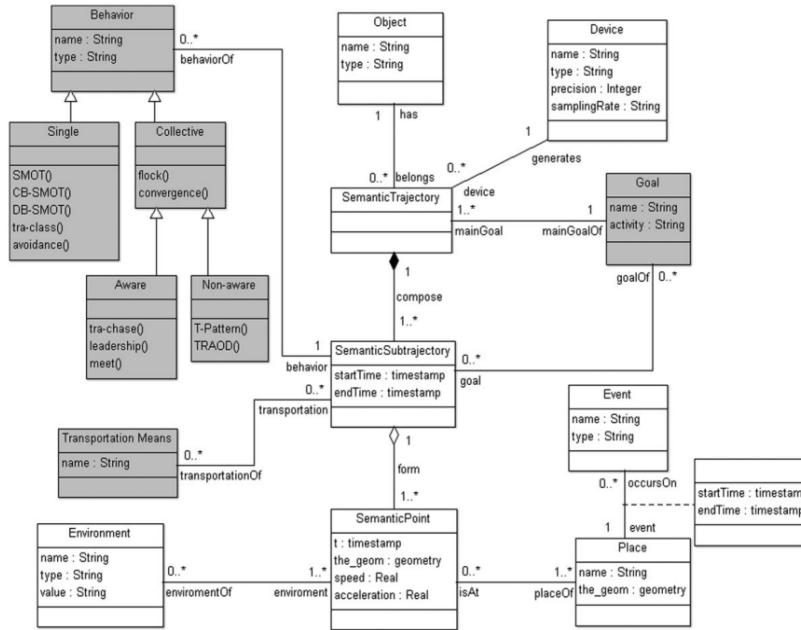


Figure 3. 7diagramme de classe trajectoire

## V. Modélisation d'objets en mouvement sur plusieurs granularités :

Le mouvement d'objets ou d'individus sur de multiples granularités. La granularité fait référence à la sélection du niveau de détail approprié pour une tâche. À des granularités fines, des informations spatio-temporelles sont révélées qui n'étaient pas connues auparavant, telles que des emplacements supplémentaires qu'un individu a visités ou des visites multiples au même emplacement. une granularité plus grossière ou une vue plus simple généralise les aspects spatiaux et temporels du mouvement permettant une meilleure compréhension du mouvement. Le mouvement est modélisé comme des lignes de vie géospatiales, des enregistrements horodatés des emplacements qu'un individu a occupés sur une période de temps. Selon la granularité souhaitée, les lignes de vie sont modélisées comme des perles de ligne de vie, des colliers ou des approximations plus générales de ces structures et cet article examine comment les différents aspects des lignes de vie deviennent pertinents à des granularités raffinées ou grossières. [38]

## VI. Modèles avancées (modèle multicouche)

Pour répondre aux exigences de modélisation, un **modèle de trajectoire autonome** doit prendre en compte différents niveaux de mouvement sémantique. Nous fournissons un modèle de trajectoire multicouche avec perspectives multiples sur les trajectoires qui intègre les trois types de données de trajectoire ensemble (RMD, GD et ADD), comme le montre [la figure3.5](#) dans la page 34.

**Définition 1** (RMD - Données brutes de mouvement).

Les données brutes de mouvement géométrique sont les données initiales qui sont généralement capturées par des appareils mobiles, enregistrant l'évolution de position où réside temporairement un objet en mouvement, RMD est formellement défini comme une séquence de tuples spatio-temporels (position, temps), par exemple, dans un système spatial 2D typique (de même en grandes dimensions),  $RMD = \{h_{xi}, y_i, t_{ii}\}$  (avec tous les  $t_i$  distinctes).

**Définition 2** (GD - Données géographiques).

Les données géographiques "GD" désigne connaissances géographiques générales (par exemple, données de forme comme les routes et bâtiments), selon les exigences d'application de la trajectoire. Il comprend également des connaissances génériques sur le statut de données environnementales telles que les indicateurs météorologiques (par exemple: la température, les précipitations, le vent) et leur variation dans le temps.

**Définition 3** (ADD - Application Domain Data).

L'application domain data "ADD" s'appuie sur les caractéristiques sémantiques des données géographiques dans GD, données de domaine d'application ADD détient les connaissances d'application (comme les bases de données d'application traditionnelles), par exemple, dans un scénario de migration d'oiseaux, ADD contient des informations relatives aux oiseaux, par ex. données d'alimentation. Dans les applications de trafic se concentrant sur véhicules en mouvement, ADD détient par exemple les connaissances liées à l'automobile comme les garages de réparations.

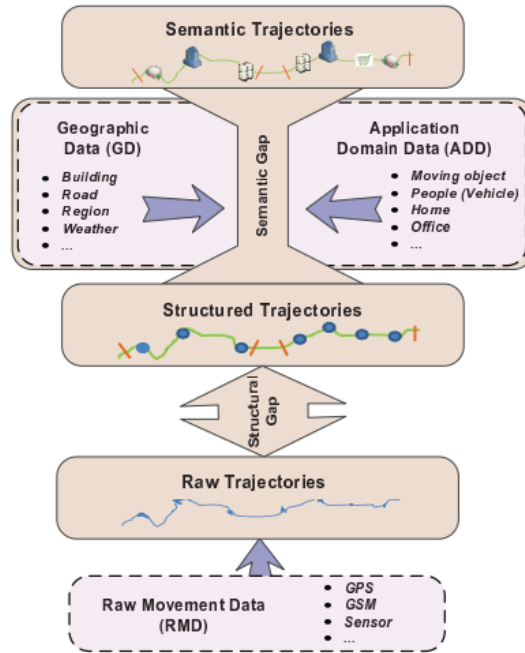


Figure 3. *Architecture du modèle multicouche*

A partir de RMD, des trajectoires brutes peuvent être définies, qui ne considèrent pas les emplacements géométriques sur le mouvement. Concrètement parlant, la trajectoire brute est une version nettoyée du sous-RMD séquence se concentrant sur un objet en mouvement spécifique.

**Définition 4** (Traw - Trajectoire brute). Une trajectoire brute Traw est une sous-séquence nettoyée de RMD pour un objet en mouvement et un intervalle de temps donné [ début, et fin]. Il est une liste de triplets  $hx, y, ti$  tels que tous les  $t$  sont différents et la liste est ordonnée par  $t$  croissant ;

$hx, yi$  est l'emplacement géométrique dans un système spatial 2D.

Pour obtenir une vue plus informative, les trajectoires brutes peuvent être structurées en segments correspondant à des étapes significatives dans la trace de la trajectoire (par exemple, voyage), nous définissons une structure trajectoire en tant que trajectoire brute augmentée d'un composant structure (début, fin, arrêts et mouvements).

**Définition 5** (Tstr - Trajectoire Structurée). Une trajectoire structurée Tstr est une trajectoire brute dont la liste des triples est partitionnée en quatre types de composants : un début B, une extrémité E et une séquence de séquences consécutives et alternées déplace M et arête S.

Un début (B) est une position spatio-temporelle où commence la trajectoire. Il est défini comme un tuplehpoint, instanti, où instant = t début ; - une extrémité (E) est une autre position spatio-temporelle où se termine la trajectoire, avec une définition similaire hpoint, instanti, où instant = t fin ;

Un mouvement (M) est défini par un intervalle de temps [tbeginMove, tendMove] pendant laquelle l'objet se déplace. Un mouvement est défini par un point variant dans le temps.

Un stop (S) est défini par un intervalle de temps [tbeginStop, tendStop] pendant laquelle l'objet ne bouge pas ou presque (ce qui signifie que le mouvement physique n'a pas d'importance).

BES est un terme générique que nous utilisons pour désigner l'union de commencer, terminer et arrêter avec l'amélioration des informations structurelles, les trajectoires structurées **Tstr** ont une meilleure signification que les trajectoires brutes **Traw** mais ignorent toujours les connaissances sémantiques des applications du monde réel. Pour de nombreuses applications, il est pertinent d'en savoir plus sur la nature de l'endroit où l'objet voyageant s'arrête, ou connaître des informations relatives à ces arrête. L'application peut également rechercher des arrêts contraignants pour que ça ne se produise qu'à un certain endroit. Par conséquent, nous introduisons davantage le concept de trajectoire sémantique.

**Définition 6** (Tsem - Trajectoire sémantique). Une trajectoire sémantique Tsem est une trajectoire structurée avec informations sémantiques : *s'arrêter, se déplacer* ainsi que Begin et End sont liés aux objets de GD et ADD.

Un mouvement (M) n'est plus un temps géométrique variable d'un point mais un chemin significatif (par exemple une séquence de segments de route) positionné dans le temps ;

### 3.5 Etude comparative des modèles

Tableau3 comparatif entre les approches mis en évidence

Modèle	Exhaustive	Mouvement a base d'objet excitatif	Système Athéna	Mouvement individuelle	Implémentation
modèle basic	1/3				3/3
L'approche de Laube et al. (2005)		x			
L'approche de Spinsanti et al. (2010) et Baglioni et al. (2009)			x		
Modélisation d'objets en mouvement sur plusieurs granularités	3/3				1/3
Modèle multicouche	3/3				1/3
Spaccapietra et al. (2008)				x	

- i/3 : Qualité d'approche par rapport au critère.
- x : satisfaction les critères.

Le critère d'exhaustivité indique si le modèle prend en considération tous les conditions de la trajectoire, par exemple on voit que le modelé basic ne prend pas en charge l'incertitude des données, le bruit, les erreurs provenant du dispositif (sensors)

Le mouvement a base d'objet excitatif et un mouvement qui ne se déclenche qu'avec une excitation, comme par exemple des passagers assis dans une gare de train, les passagers ne se lève qu'après l'arrivée du train.

Contrairement au mouvement à base d'objet excitatif, le mouvement individuel ne s'influence pas par élément déclencheur, c'est une consécution de déplacement programmé préalablement

Le critère d'implémentation indique la possibilité de programmé le modèle en question.

### **3.6 Travaux liés à la prédiction de trajectoire**

Dans cette section, nous divisons les algorithmes de prédiction en quatre catégories et les illustrons dans les sous-sections suivantes.

#### **I- Méthodes basées sur Markov**

Les méthodes basées sur Markov sont représentatives de la stochastique modèle qui illustre les probabilités de transition d'un état à un autre, pour les observations, les méthodes basées sur Markov supposent que chaque observation est un état.

Si la transition dépend de l'état actuel, le modèle est appelé modèle de Markov du 1<sup>er</sup> ordre, le modèle de Markov d'ordre 2 considère à la fois l'état actuel et l'état précédent étant donné une séquence d'observations.

Song et al. [39] se concentrent sur la prédiction du prochain emplacement de la mobilité humaine et vérifient la supériorité des méthodes de Markov par rapport aux autres méthodes de prédiction (compression, appariement partiel et appariement de motifs échantillonnés).

Qiao et al. [40] proposent un modèle de Markov caché qui a les caractéristiques de la sélection auto-adaptative des paramètres en fonction du changement de vitesse de déplacement des objets.

Les résultats expérimentaux montrent une précision de positionnement élevée soumise à des vitesses changeant de manière stochastique.

Terroso et al. [41] développent une approche multiniveau basée sur Markov pour prédire la prochaine position des personnes et vérifier son efficacité en utilisant des tweets géolocalisés.

Qiao et al. [42] conçoivent un modèle de Markov hybride tenant compte des caractéristiques spatio-temporelles de la mobilité humaine et prédisent les emplacements futurs des personnes



avec une précision d'environ 56 %. Yin et al. [43] proposent des modèles de Markov cachés pour extraire les modèles d'activité des voyageurs des CDR et fournissent une solution exploitable et modulaire pour analyser la demande de voyage des personnes.

## **II- Méthode basées sur Compression**

Les méthodes basées sur la compression incluent Lempel-Ziv (LZ), LZ améliorée, la correspondance partielle (PPM) et la correspondance de modèle d'échantillon (SPM), qui sont à l'origine utilisées dans la compression de texte.

Le type de méthodes découle d'une méthode incrémentale populaire l'algorithme d'analyse proposé par Ziv et al. [44], de plus, les méthodes basées sur la compression sont comme le kème modèle de Markov sauf que k peut être infini [45].

Certains chercheurs tirent parti des méthodes mentionnées ci-dessus pour améliorer la précision des prédictions de la mobilité humaine.

Alam et al. [46] proposent un algorithme de prédiction de découverte d'épisode amélioré pour prédire les activités des habitants et atteindre une précision de prédiction de 88,3 %.

Gopalratnam et al. [47] proposent un algorithme de prédiction LeZi actif pour prédire les événements futurs en utilisant une séquence d'événements antérieurs.

Pulliyakode et al. [48] proposer un algorithme PPM amélioré pour prédire l'emplacement futur des personnes et démontrer l'efficacité sur la base d'un ensemble de données de localisation.

### **III- Méthode Time series :**

Les méthodes de séries chronologiques sont appliquées aux données de séries chronologiques pour prédire les valeurs futures de la série.

Étant donné une série chronologique de données, ces algorithmes exploitent différentes techniques pour rendre les données de séries chronologiques stationnaires, ce qui contribue à l'amélioration de la précision des prédictions.

Ces méthodes incluent Autogressive (AR) [49], MovingAverage (MA) [50], AutoregressiveMovingAverage (ARMA) [51], AutoregressiveIntegratedMovingAverage (ARIMA) [52].

Yang et al. [53] proposent une nouvelle méthode (modèles de séries temporelles) pour prédire les futurs liens dans les réseaux de mobilité humaine et acquérir une plus grande précision.

De et al. [54] utilisent des méthodes de séries chronologiques non linéaires multivariées pour prédire les mouvements humains en les combinant avec des modèles de mobilité humaine.

Zeng et al. [55] proposent un modèle de séries chronologiques d'ordre élevé pour prédire les emplacements visités dans le futur, et étudient également la prévisibilité en analysant d'autres facteurs tels que la résolution spatio-temporelle, le rayon de giration.

### **IV- Méthode Machine-Learning**

Les méthodes d'apprentissage automatique sont largement utilisées pour la prédiction de la mobilité humaine, les méthodes d'apprentissage automatique utilisent principalement des techniques statistiques pour extraire des lois potentielles dans des données spécifiques.

En explorant les données de trajectoire des personnes, les méthodes d'apprentissage automatique peuvent prédire le volume du flux migratoire, le prochain emplacement et l'heure correspondante dans le futur.

Les méthodes d'apprentissage automatique comprennent l'apprentissage supervisé, semi-supervisé et non supervisé, et effectuent principalement la classification ou la régression des entrées.

Joseph et al. [56] proposent une approche bayésienne non paramétrique pour modéliser les modèles de mobilité humaine et résoudre les problèmes de sur-ajustement ou de sous-ajustement.

Song et al. [57] conçoivent un système intelligent de prédiction de la mobilité humaine dans des scénarios urbains basé sur l'architecture d'apprentissage profond à partir de données hétérogènes multisources.

Monreale et al. [58] proposent une méthode efficace pour prédire la prochaine localisation des personnes en utilisant un algorithme d'arbre de décision.

Song et al. [59] concevoir un système intelligent (DeepTransport) pour prédire les déplacements futurs des personnes et les conditions de déplacement sur la base de l'architecture d'apprentissage en profondeur.

Feng et al. [60] proposent un réseau récurrent attentionnel pour la prédiction de la mobilité à partir de trajectoires longues et éparées et surpassent les modèles de pointe par une série d'expériences.

Baumann et al. [61] prédisent la mobilité humaine en choisissant le modèle le plus approprié dans le domaine de l'apprentissage automatique et démontrent que différentes méthodes d'apprentissage automatique pour différentes périodes peuvent grandement améliorer les performances.

Kun et al. [62] développent un nouveau modèle génératif non paramétrique pour les trajectoires de localisation qui exploite les caractéristiques statistiques de la mobilité humaine.

Pendant ce temps, nous résumons les recherches pertinentes sur les méthodes de prédiction de la mobilité humaine dans Tableau 3.

Tablout.4 : tableau comparatif entre les méthodes de prédiction de mobilité humain.

Taxonomie	Référence	Principe des algorithmes proposés	Objective
Basée Markov	Song et al. [39]	Le prédicteur de Markov $O(2)$ suppose que le prochain emplacement peut être prédit à partir des deux derniers contextes.	Emplacement suivant

	Qiao et al. [40]	Un modèle de Markov caché pour prédire la position et le comportement des objets en mouvement.	Traces continues
	Terroso et al. [41]	Une méthode basée sur Markov à plusieurs niveaux qui utilise des ensembles de données de médias mobiles et sociaux.	Nextactivity and position
	Qiao et al. [42]	Un modèle hybride basé sur Markov qui considère la spatio-temporelle orale et la similitude des modèles de mobilité humaine.	Emplacement suivant
Basée compression	Alam et al. [43]	Un algorithme de correspondance partielle pour prédire la prochaine activité à partir de l'historique précédent.	Activité suivante
	Gopalratnam et al. [44]	Un algorithme LeZi actif qui utilise une séquence d'événements observée pour prédire le prochain événement	Événement suivant
	Pulliyakode et al. [45]	Un algorithme PPM amélioré qui suppose des arbres de fréquence de plus grande profondeur résultant en une meilleure prédiction	Emplacement suivant
Time-series	Yang et al. [53]	Une technologie de prédiction de lien qui utilise des séries temporelles de densité de réseau et une durée de vie de périphérie pour déduire les futurs liens.	Lien suivant
	De et al. [53]	Techniques de prédiction de séries chronologiques non linéaires multivariées prenant en compte les mouvements des amis et des personnes.	Interactions sociales
	Zeng et al. [54]	Une méthode basée sur des séries chronologiques utilisant l'échantillonnage de Gibbs pour prédire le mouvement de l'utilisateur.	Emplacement suivant
Machine learning	Song et al. [62]	Un système intelligent (DeepTransport) basé sur l'architecture d'apprentissage profond pour prédire la mobilité humaine dans des scénarios	-Mouvements futurs -mode de transport

		urbains.	
	Feng et al. [63]	Un réseau de neurones récurrents attentionnels qui capture la régularité de transition hétérogène et la périodicité à plusieurs niveaux.	Emplacement suivant
	Baumann et al. [64]	Un cadre générique qui explore les données de mobilité humaine et calcule à la fois des modèles de population et des modèles individuels pour prédire la mobilité humaine.	-Lieu suivant. -Emplacement suivant. -Transition vers l'emplacement suivant.

### 3.7 Conclusion

Dans ce chapitre nous avons défini la trajectoire et cité les techniques de description de cette dernière, après cela nous avons présenté les modèles de trajectoire existant et enfin nous avons parlé des algorithmes de prédictions. La trajectoire peut être décrite par plusieurs techniques selon le cas et on peut appliquer à cette technique l'algorithme adéquat pour avoir un résultat de prédiction proche de la réalité

# Chapitre 4 Spécification des besoins et conception

## 4.1 Introduction

Après avoir présenté les différents travaux dans le domaine de la modélisation de la trajectoire et les travaux de la prédiction de trajectoire, nous présentons notre solution pour la résolution de notre problématique, tout d'abord nous abordons notre solution d'une façon globale puis nous allons détailler notre conception chaque étape de la solution.

## 4.2 Conception globale

Dans cette section nous présentons une conception globale de la solution, qui se compose de sept étapes :

**Étape 1 :** consiste à acquérir les données de mobilité humaine dans un environnement urbain « ville intelligente », nous allons collecter les données de trajectoire d'un utilisateur de type GPS, latitude, longitude, avec une estampille temporelle.

**Étape 2 :** consiste à faire un prétraitement des données collectées pour les préparer aux étapes qui succèdent à cette étape.

**Étape 3 :** choix d'une technique de description de trajectoire adéquate à notre problématique

**Étape 4 :** proposition d'un modèle conceptuel de trajectoire

**Étape 5 :** prédiction de la trajectoire avec un algorithme de machine learning

**Étape 6 :** c'est une étape parallèle à la cinquième étape, qui consiste à enregistrer la trajectoire dans une base de données adaptée.

**Étape 7 :** évaluation de notre modèle proposé avec un autre modèle

Dans la figure (Figure4.1), nous présentons les étapes de notre solution :

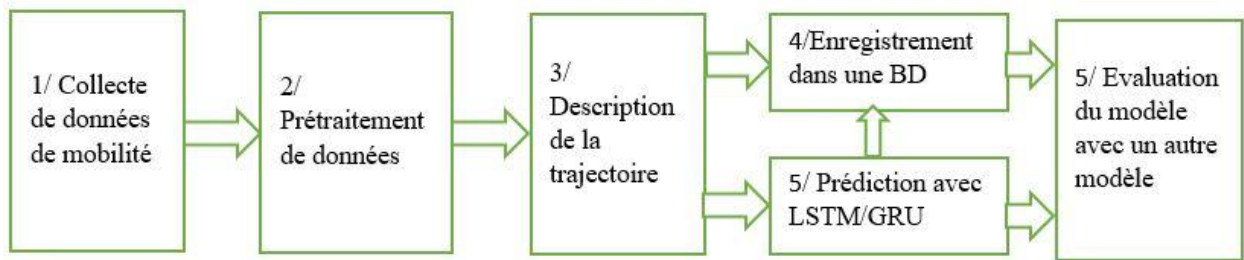


Figure 4. 1Schéma de la solution proposée

### 4.3 Conception détaillée

Dans la section précédente nous avons présenté la conception de notre solution sommairement, dans cette section nous énumérons les étapes de la solution à savoir :

#### 4.3.1 Collecte de donné de mobilité

Dans la partie **collecte** de données mobilité nous allons trouver des donné public de mobilité sous format exploitable comme « **.txt** » ou format « **rar** », nous importons les données sur notre environnement de développement.

#### 4.3.2 Prétraitement de donné de mobilité

La partie deux de notre solution consiste à analyser les données collectées avec la bibliothèque **Pandas** présente dans python 3, cette technique consiste à extraire la taille du jeu, la dimension, le nombre de colonnes et le type de données dans les colonnes du **dataframe**.

Après avoir fait une première analyse sur le jeu de données nous avons remarqué que l'intervalle de l'enregistrement de ces données GPS avec une estampille temporelle n'était pas enregistré à un intervalle régulier, c'est-à-dire de point à point, des fois il y a un intervalle de 5 seconde, 10 et nous pouvons même trouver des heures manquante d'enregistrement.

Pour utiliser les algorithmes de machine Learning (RNN) LSTM, GRU avec des données time séries, les lignes estampies doivent être à intervalle régulier.

Notre approche pour résoudre ce problème d'irrégularité de time séries dans notre jeu de données, c'est d'utiliser une méthode présentée dans la bibliothèque Pandas, la méthode « **Resample** » qui agit sur les données time séries, plus précisément les données d'où les lignes du data frame sont indexés par le temps composé par une date « **anné-mois-jour** » et « **heure-minute- seconde** ».

La première étape, c'est d'implémenter cette méthode de **resample par minute**, le faite de produire un autre data frame avec une ligne qui représente une minute.

La deuxième analyse c'est de vérifier les valeurs NAN dans les lignes après avoir appliqué la méthode **Resample** par Minute, nous constatons que beaucoup de ligne dans le data frame ne contient pas de valeurs mais des Nan « **missing data** », avec la méthode de la bibliothèque Pandas, **IsNull ().sum()**, les deux méthode concaténé nous donne le nombre de lignes qui contient les valeur Nan.

La solution proposée à ce problème consiste à utiliser une méthode existante dans la bibliothèque Pandas, c'est **Bfill ()**, cette méthode remplit les lignes trouvées comme Nan, par des valeurs cohérente du data frame.

Après cela nous vérifions avec les deux méthodes **isnull().sum()**, le nombre de lignes avec les valeurs Nan .

### **4.3.3 Technique de description proposée**

Comme nous avons vu dans le chapitre précédent, les différents technique de description de trajectoire, par apport à notre problématique celle de prédiction de trajectoire en vue de recommandation, ce qui veut dire pour recommander un service, c'est de donner l'emplacement future de l'objet mobile.

Pour décrire les emplacements passé et les emplacementsfuture, nous proposons une description basé sur les régions, pour être en phase avec notre problématique celle de travaillé avec les trajectoires humain dans une ville, les villes sont divisées administrativement donc, dans une ville nous allons trouver plusieurs régions

En vue de gagner en précisions, nous permutons la description point par point préalablement utilisée vers la technique basée sur les régions, notre approche proposé c'est de définir les régions par un rayon c'est-à-dire le rayon englobe une région, quand l'objet mobile passe par une région, nous allons comptabiliserla traversé de cet objet mobile.



#### 4.3.4 Modèle de trajectoire proposé

Dans ce modèle, nous considérons uniquement la modélisation des trajectoires spatio-temporelles générées par des objets représentés comme des points, on ces inspiré de l'approche de « Spaccapietra » basée sur les points d'arrêt :

##### Arrêt

Un arrêt est une partie d'une trajectoire, telle que :

L'utilisateur a explicitement défini cette partie de la trajectoire [**Tbeginstopx ; Tendstopx**] pour représenter un arrêt.

L'étendue temporelle [**Tbeginstopx ; Tendstopx**] est un intervalle de temps non vide, etLe mobile ne bouge pas (en ce qui concerne la vue applicative de cette trajectoire), c'est-à-dire la portée spatiale de la trajectoire pour l'intervalle [**Tbeginstopx ; Tendstopx**] est un point unique. Tous les arrêts sont temporairement disjoints, c'est-à-dire que les étendues temporelles de deux arrêts sont toujours disjointes.

##### Mouvement

Un mouvement est une partie d'une trajectoire, telle que :

La partie est délimitée par deux extrémités qui représentent soit deux arrêts consécutifs, soit le début et le premier stop, [**Tbegin, Tend** ] .

L'étendue temporelle [**TbeginmoveX;TendmoveX**] est un intervalle de temps non vide, et la portée spatiale de la trajectoire pour l'intervalle [**TbeginmoveX; TendmoveX**] est la ligne spatio-temporelle (pas un point) défini par la fonction de trajectoire (en fait, c'est la poly ligne construite sur les points d'échantillon dans l'intervalle [**TbeginmoveX; TendmoveX**].

Tableau 5 représentation de type de données utilisé

Nom	Type données	allocation	Description
ID	Int	4	Identifiant de trajectoire
Longitude	float	8	Données GPS longitude
Latitude	float	8	Données GPS latitude

Date	varchar	10	date
Temps	varchar	8	temps

Dans notre approche de modèle de trajectoire, nous avons établi que pour comptabiliser un arrêt c'est une période de 20 min dans une région avec un rayon de 1km.

Pour réaliser ce modèle, nous avons utilisé une bibliothèque présente dans python 3, **SciKitMobility**, qui fournit plusieurs méthodes pour l'étude de trajectoire humain et l'extraction de connaissances de c'est trajectoire.

Nous avons utilisé la méthode « Stop détection » pour l'extraction des arrêts.

La méthode se définit comme, Détectez les arrêts pour chaque individu dans un **TrajDataFrame**.

Un arrêt est détecté lorsque l'individu passe au moins **minutes\_for\_a\_stop** minutes à une distance **stop\_radius\_factor** \* **spatial\_radius** km d'un point de trajectoire donné.

Les coordonnées de l'arrêt sont les valeurs médianes de latitude et de longitude des points trouvés dans la distance spécifiée.

### Paramètre

- **tdf** (*TrajDataFrame*) – the input trajectories of the individuals.
- **stop\_radius\_factor** (*float, optional*) – if argument *spatial\_radius\_km* is *None*, the *spatial\_radius* used is the value specified in the *TrajDataFrame* properties (“*spatial\_radius\_km*” assigned by a *preprocessing.compression* function) multiplied by this argument, *stop\_radius\_factor*. The default is *0.5*.
- **minutes\_for\_a\_stop** (*float, optional*) – the minimum stop duration, in minutes. The default is *20.0*.
- **spatial\_radius\_km** (*float or None, optional*) – the radius of the ball enclosing all trajectory points within the stop location. The default is *0.2*.
- **leaving\_time** (*boolean, optional*) – if *True*, a new column ‘*leaving\_datetime*’ is added with the departure time from the stop location. The default is *True*.

- **no\_data\_for\_minutes** (*float, optional*) – if the number of minutes between two consecutive points is larger than *no\_data\_for\_minutes*, then this is interpreted as missing data and does not count as a stop. The default is *1e12*.
- **min\_speed\_kmh** (*float or None, optional*) – if not *None*, remove the points at the end of a stop if their speed is larger than *min\_speed\_kmh* km/h. The default is *None*.

**Returns** a `TrajDataFrame` with the coordinates (latitude, longitude) of the stop locations.

**Return type** `TrajDataFrame`

#### 4.3.5 Prédiction de la trajectoire

Dans cette partie de la solution, après avoir fait un prétraitement sur le data frame qui représente la trajectoire de l'objet humain, et propos une technique, aussi un modèle des descriptions de la trajectoire présente dans le data frame.

Nous allons commencer à construire un modèle de prédiction qui est basé sur les algorithmes de **deep learning** plus précisément les réseaux de neurones récurrent, comme vue dans le chapitre précédent qui traite des différentes méthodes de prédiction, le choix que nous avons choisie c'est une méthode de **machine Learning**, basé sur les réseaux de neurones.

L'algorithme adapté pour la prédiction de donnée time série qui est cité par des travaux dans le chapitre 2, c'est le LSTM, un Réseaux de neurones particulier qui peut apprendre des dépendances à long terme. Tous les RNN ont la forme d'une chaîne de modules répétitifs. Dans un RNN standard, ce module répétitif a généralement une structure simple. Cependant, le module répétitif des LSTM est plus compliqué. Il n'y a pas qu'une seule couche de réseau de neurones, mais quatre couches qui interagissent d'une manière spéciale. De plus, il a deux états : l'état caché et l'état cellulaire.

En 2014 un algorithme plus simple mais performant que LSTM a été introduit le GRU, la structure interne de GRU est plus simple et plus facile à former car elle implique moins de calculs.

Pour notre solution nous proposons deux modèle de RNN, un modèle avec une couche de LSTM et l'autre modèle avec une couche de GRU, les deux modèle auront une couche d'encodeur et une couche de décoder dans [la figure \(4.2\)](#) et [figure \(4.3\)](#) montrent l'architecture de ces modèles.

Notre approche pour la prédiction c'est d'entraîner les modèles construits avec un jeu de données qui décrit **les arrêts** de l'objet mobile, pour la prédiction nous allons donner comme entrées à notre modèle, **25 arrêts** pour prédire les 3 prochaines **arrêts**, nous avons choisi seulement **3 arrêts** pour ne pas réduire la précision de notre prédiction.

Le jeu de données utilisé pour l'entraînement et l'évaluation de notre modèle provient du résultat de la méthode mentionnée dans la partie modèle de trajectoire stop détection.

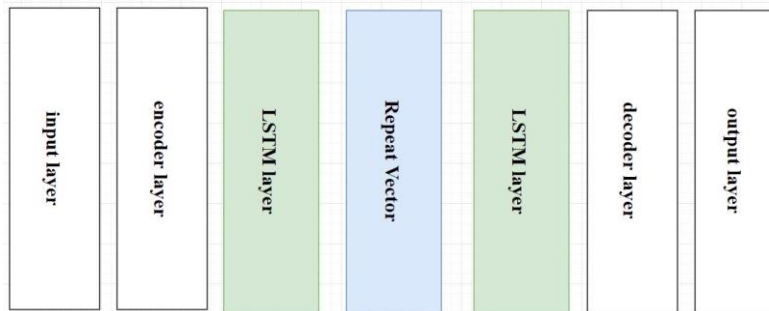


Figure 4. 2architecture du modèle avec LSTM

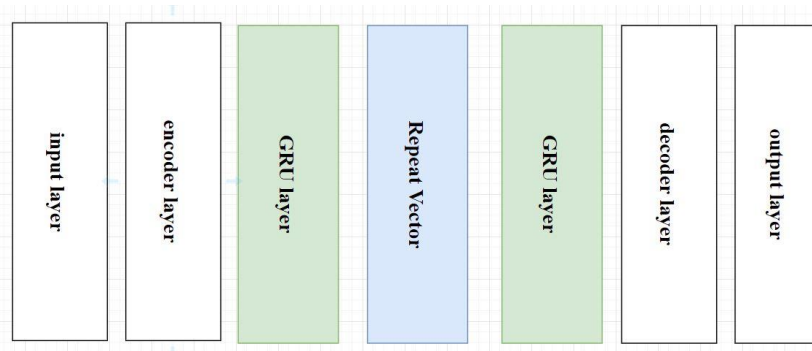


Figure 4. 3architecture du modèle avec GRU

#### 4.3.6 Enregistrement de la trajectoire dans un BD

Dans cette partie on va enregistrer les données collectées dans une base de données qui gère les données spatiales temporelles, selon notre modèle de description de trajectoire proposé.

#### 4.3.7 Evaluation du modèle proposé par un autre modèle

Dans cette partie de la solution globale proposée, après avoir implémenté la technique et le modèle de description, et implémenté la technique de prédiction basée sur le machine Learning, on va évaluer le modèle proposé, on le comparera avec un autre modèle, la comparaison se base sur les performances du modèle proposé avec le modèle de comparaison.

Les méthodes de comparaison seront la fonction de perte « **lossfunction** » et « **accuracy** , **validation accuracy** » qui seront appliquées sur le modèle proposé et au modèle d'évaluation choisie.

La méthode **loss** est une méthode mathématique, nous permet de comparer entre certaines cibles réelles et des cibles prédites, elle le fait en imposant une « perte » sur chaque prédiction si elle s'écarte des cibles réelles.

Il existe deux principaux types de problèmes d'apprentissage supervisé : la classification et la régression. Dans le premier cas, votre objectif est de classer un échantillon dans la bonne classe. Dans ce dernier cas, cependant, vous ne classez pas mais estimez plutôt un nombre réel. Ce que vous essayez de faire, c'est régresser une fonction mathématique à partir de certaines données d'entrée, c'est donc ce qu'on appelle la régression. Pour les problèmes de régression, de nombreuses fonctions de perte sont disponibles.

## I. MeanAbsoluteError

**MeanAbsoluteError** est l'une des fonctions de perte les plus simples mais robustes utilisées pour les modèles de régression.

Comme son nom l'indique, MAE prend la somme moyenne des différences absolues entre les valeurs réelles et prédites, pour un point de données  $x_i$  et sa valeur prédite  $y_i$ ,  $n$  étant le nombre total de points de données dans l'ensemble de données, l'erreur absolue moyenne est défini comme :

$$\text{MAE} = \frac{\sum_{i=1}^n |y_i - x_i|}{n} = \frac{\sum_{i=1}^n |e_i|}{n}$$

Figure 4. 4la fonction de MAE

## II. RootMeanSquaredError

**MeanSquaredError** est la moyenne des différences au carré entre les valeurs réelles et prédites. Pour un point de données  $Y_i$  et sa valeur prédite  $\hat{Y}_i$ , où  $n$  est le nombre total de points de données dans l'ensemble de données, l'erreur quadratique moyenne est définie comme :

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

Figure 4. 5 la fonction de MSE

La perte MAE est plus robuste que son homologue.

En regardant de plus près les formules, on peut observer que si la différence entre la valeur prédite et la valeur réelle est élevée, la perte MSE amplifie l'effet par rapport à la MAE. Étant donné que MSE succombe aux valeurs aberrantes, la fonction de perte MAE est la fonction de perte la plus robuste.

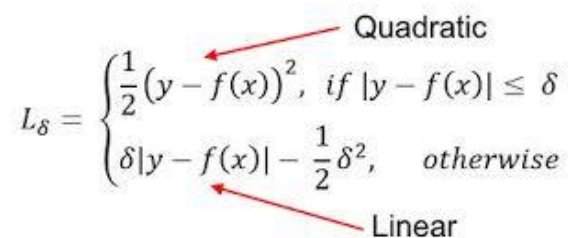
La fonction de perte MEA est moins stable que la fonction de perte MSE.

Étant donné que la perte MEA traite de la différence de distances, un petit changement horizontal peut entraîner un saut important de la ligne de régression. Un tel effet se produisant sur plusieurs itérations conduirait à un changement significatif de la pente entre les itérations.

### III. Huber Loss

**Huber Loss** combine la robustesse du **MEA** avec la stabilité du **MSE**, essentiellement le meilleur des pertes MEA et MSE. Pour les erreurs énormes, il est linéaire et pour les petites erreurs, il est de nature quadratique.

**Huber Loss** est caractérisé par le paramètre delta ( $\delta$ ). Pour une prédiction  $f(x)$  du point de données  $y$ , avec le paramètre caractérisant  $\delta$ , la perte de Huber est formulée comme :

$$L_{\delta} = \begin{cases} \frac{1}{2}(y - f(x))^2, & \text{if } |y - f(x)| \leq \delta \\ \delta|y - f(x)| - \frac{1}{2}\delta^2, & \text{otherwise} \end{cases}$$


The diagram shows the Huber Loss function defined as a piecewise function. The first part,  $\frac{1}{2}(y - f(x))^2$  for  $|y - f(x)| \leq \delta$ , is labeled 'Quadratic' with a red arrow pointing to it. The second part,  $\delta|y - f(x)| - \frac{1}{2}\delta^2$  for 'otherwise', is labeled 'Linear' with a red arrow pointing to it.

Figure 4. 6 fonction de Huber Loss

La deuxième comparaison du modèle proposé avec le modèle d'évaluation, c'est utilisé la précision des deux modèle avec le « **testing set** » et « **le validation set** ».



**Validation set :** L'ensemble de données de validation est utile lorsqu'il s'agit de régler les hyperparamètres et de sélectionner le modèle. Les exemples de validation inclus dans cet ensemble seront utilisés pour trouver les valeurs optimales pour les hyper-paramètres du modèle considéré.

**Testing set :** L'ensemble de test est utilisé pour évaluer les performances de ce modèle et s'assurer qu'il peut bien se généraliser à des nouveaux points de données invisibles.

La comparaison va se faire sur la différence entre les résultats de la précision du **testing set** avec les résultats de la **validation set**, plus la différence est réduite plus le modèle donné une bonne précision.

Le modèle d'évaluation proposé est basé sur les positions de l'objet mobile de chaque journée, son principe est de prendre la position de l'objet mobile de chaque journée, pour cela nous devons procéder par un **resampling** de minute par minute pour ne pas perdre l'information, par la suite nous procédons par le **resampling** par jour.

Contrairement au modèle qu'on a proposé, ce modèle d'évaluation décrit la trajectoire en se basant sur le temps et non sur le déplacement.

La raison de notre choix sur le modèle d'évaluation, la similarité de la représentation de la trajectoire, et aussi la taille du jeu de données qui est équivalente.

#### **4.4 Conclusion**

Dans ce chapitre nous avons d'abord rappelé la problématique à laquelle nous faisons face puis nous avons défini le plan de la solution, nous avons donné en premier lieu la description global des étapes successives puis nous avons détaillé chaque étape de la solution dont l'étape de la collecte et le prétraitement jusqu'à l'architecture des modèle de machine Learning utilisés.



# Chapitre 5 Implémentation de la solution

## 5.1 Introduction

Dans cette partie, nous présentons le matériel sur lequel nous avons développé notre modèle, les différents outils utilisés ainsi que les composantes applicatives réalisées. Ce modèle sera testé et évalué et par la suite comparé à d'autres architectures qui ont été proposées pour résoudre notre problématique.

Durant les différentes étapes d'implémentations, d'entraînements et de tests de notre modèle, nous avons exploité les ressources matérielles de notre station personnelle qui possède les spécifications suivantes :

## 5.2 Environnement et outils de développement utilisés

### Présentation du langage Python :

Python est devenu l'un des langages de programmation les plus populaires pour la recherche au cours de la dernière décennie. Sa nature gratuite et open-source et sa vaste communauté en ligne sont quelques-unes des raisons de son succès. On trouve d'innombrables exemples d'augmentation de la productivité de la recherche grâce à Python dans une pléthore de domaines en ligne, notamment la science des données, l'intelligence artificielle et la recherche scientifique[65].



Figure 5. 1 Logo Python

## Présentation de googlecolab:

Colaboratory est un projet de recherche créé pour aider à diffuser l'enseignement et la recherche en apprentissage automatique, il s'agit d'un environnement de notebook Jupyter qui ne nécessite aucune configuration pour être utilisé, Colab permet à n'importe qui d'écrire et d'exécuter le code Python de son choix par le biais du navigateur[66].



*Figure 5. 2 Logo colab*

## Présentation du Framework TensorFlow :

TensorFlow est une plate-forme open source de bout en bout pour l'apprentissage automatique. Il dispose d'un écosystème complet et flexible d'outils, de bibliothèques et de ressources communautaires qui permet aux chercheurs de pousser l'état de l'art en matière de ML et aux développeurs de créer et de déployer facilement des applications basées sur le ML.

TensorFlow a été développé à l'origine par des chercheurs et des ingénieurs travaillant au sein de l'équipe Google Brain au sein de l'organisation Machine Intelligence Research de Google pour mener des recherches sur l'apprentissage automatique et les réseaux de neurones profonds. Le système est suffisamment général pour être également applicable dans une grande variété d'autres domaines.

TensorFlow fournit des API Python et C++ stables, ainsi qu'une API rétrocompatible non garantie pour d'autres langages[66].



*Figure 5.3 Logo TensorFlow*

### Présentation du Framework Keras :

Keras est une bibliothèque open-source de composants de réseaux neuronaux écrits en Python. Keras est capable de fonctionner au-dessus de TensorFlow, Theano, PlaidML et autres. La bibliothèque a été développée pour être modulaire et conviviale, mais elle a initialement commencé dans le cadre d'un projet de recherche pour le système d'exploitation intelligent neuro-électronique ouvert ou ONEIROS. L'auteur principal de Keras est François Chollet, un ingénieur de Google qui a également écrit Xception, un modèle de réseau neuronal profond. Bien que Keras ait été officiellement lancé, il n'a été intégré à la bibliothèque centrale TensorFlow de Google qu'en 2017. Un support supplémentaire a également été ajouté pour l'intégration de Keras avec Microsoft Cognitive Toolkit.

Composée d'une bibliothèque de composants d'apprentissage automatique couramment utilisés, notamment des objectifs, des fonctions d'activation et des optimiseurs, la plateforme open-source de Keras offre également un support pour les réseaux neuronaux récurrents et convolutifs. En outre, Keras propose le développement de plateformes mobiles pour les utilisateurs ayant l'intention de mettre en œuvre des modèles d'apprentissage profond sur les smartphones, tant iOS qu'Android. En 2018, la bibliothèque compte 22 % d'utilisation à travers de ses plus de 200 000 utilisateurs[68].



Figure 5. 4Logo Keras

### Présentation du module Numpy:

NumPy est le paquetage fondamental pour le calcul scientifique en Python. Il s'agit d'une bibliothèque Python qui fournit un objet tableau multidimensionnel, divers objets dérivés (tels que les tableaux masqués et les matrices), ainsi qu'un assortiment de routines permettant d'effectuer des opérations rapides sur les tableaux, notamment des opérations mathématiques, logiques, de manipulation de formes, de tri, de sélection, d'E/S, de transformées de Fourier discrètes, d'algèbre linéaire de base, d'opérations statistiques de base, de simulation aléatoire et bien plus encore[69].

Au cœur du paquetage NumPy se trouve l'objet ndarray. Celui-ci encapsule des tableaux à n dimensions de types de données homogènes, de nombreuses opérations étant effectuées en code compilé pour des raisons de performances.

La vectorisation décrit l'absence de toute boucle explicite, d'indexation, etc., dans le code - ces choses ont lieu, bien sûr, juste "dans les coulisses" du code C optimisé et précompilé. Le code vectorisé présente de nombreux avantages, dont les suivants :

- Le code vectorisé est plus concis et plus facile à lire
- Moins de lignes de code signifie généralement moins de bogues
- Le code ressemble davantage à la notation mathématique standard (ce qui facilite, en général, le codage correct des constructions mathématiques)
- La vectorisation donne lieu à un code plus 'pythonique'. Sans la vectorisation, notre code serait parsemé de boucles for inefficaces et difficiles à lire.

### Présentation du module Pandas :

Pandas (错误!未找到引用源。) est une bibliothèque Python de structures de données riches et d'outils permettant de travailler avec des ensembles de données structurés communs aux statistiques, à la finance, aux sciences sociales et à de nombreux autres domaines. La bibliothèque fournit des routines intégrées et intuitives permettant d'effectuer des manipulations et des analyses courantes sur de tels ensembles de données. Elle a pour but d'être la couche fondamentale pour l'avenir du calcul statistique en Python. Elle sert de complément solide à la pile Python scientifique existante tout en mettant en œuvre et en améliorant les types d'outils de manipulation de données que l'on trouve dans d'autres langages de programmation statistique tels que R[70].



*Figure 5.5 Logo Pandas*

### Présentation du module Matplotlib :

Matplotlib est une bibliothèque de traçage disponible pour le langage de programmation Python en tant que composant de NumPy, une ressource de traitement numérique de données volumineuses. Matplotlib utilise une API orientée objet pour intégrer les tracés dans les applications Python.

Comme Python est largement utilisé dans l'apprentissage automatique, des ressources comme NumPy et matplotlib sont souvent utiles pour modéliser les technologies d'apprentissage automatique. L'idée est que les programmeurs accèdent à ces bibliothèques pour des tâches clés dans un environnement Python plus large, et intègrent les résultats avec tous les autres éléments et fonctionnalités d'un programme d'apprentissage automatique, d'un réseau neuronal ou d'une autre machine avancée. L'utilité de NumPy et de matplotlib est liée aux nombres - l'utilité de matplotlib est spécifiquement liée aux outils de traçage visuel. En un sens, ces ressources sont donc plus analytiques que génératives. Cependant, toute cette infrastructure fonctionne

ensemble[pour permettre aux programmes d'apprentissage automatique de produire des résultats qui sont utiles aux manipulateurs humains[70].



*Figure 5. 6Logo Matplotlib*

### Présentation du module Scikit\_mobility :

scikit-mobility, une bibliothèque python qui a l'ambition de fournir aux scientifiques et aux praticiens un environnement pour reproduire les recherches existantes et effectuer des analyses de données de mobilité, en particulier la bibliothèque permet à l'utilisateur de :

1. charger et représenter les données de mobilité, tant au niveau individuel que collectif, grâce à des structures de données faciles à utiliser (TrajDataFrame et FlowDataFrame) basées sur les bibliothèques python standard numpy, pandas et geopandas, ainsi que de visualiser des trajectoires et des flux sur des cartes interactives basées sur les bibliothèques python folium etmatplotlib.

2. nettoyer et prétraiter les données de mobilité à l'aide de techniques de pointe, telles que le regroupement de trajectoires, la compression, la segmentation et le filtrage, la bibliothèque fournit également à l'utilisateur un moyen de suivre toutes les opérations effectuées sur les données d'origine.

3. analyser les données de mobilité en utilisant les principales mesures caractérisant les modèles de mobilité tant au niveau individuel que collectif, telles que le calcul des distances de déplacement et caractéristiques, les entropies d'objet et de localisation, les fréquences de localisation, les temps d'attente, les matrices origine-destination, et Suite.

4. exécuter les modèles génératifs mécanistes les plus populaires pour simuler la mobilité individuelle, tels que le modèle d'exploration et de retour préférentiel (EPR) et ses variantes, et les flux de navette et de migration, tels que le modèle de gravité et le modèle de rayonnement.

5. estimer le risque de confidentialité associé à l'analyse d'un ensemble de données de mobilité donné grâce à la simulation du risque de ré-identification associé à un vaste répertoire d'attaques à la vie privée[71].



*Figure 5. 7 Logo scikitmobility*

### Présentation du module MobilityDB :

MobilityDB est implémenté comme une extension de PostgreSQL et PostGIS.

Il implémente des types de bases de données persistantes et des opérations de requête pour gérer les trajectoires géospatiales et leurs propriétés variables dans le temps.

MobilityDB fournit deux principaux types de trajectoires géospatiales : tgeompoint pour un point de géométrie temporelle et tgeogpoint pour un point de géographie temporelle.

Les deux peuvent représenter un point géospatial qui se déplace dans le temps, où les coordonnées sont respectivement exprimées en géométrie ou en géographie PostGIS.

Il fournit également un riche ensemble de fonctions pour gérer ces types ; comprenant : entrée/sortie, relations topologiques, vitesse, direction, distance, etc.

Notez que les valeurs de retour de plusieurs de ces fonctions sont temporelles. Par exemple, la vitesse d'une voiture est une valeur qui change avec le temps. Par conséquent, le système de types de MobilityDB comprend des types pour les flottants temporels, les booléens temporels, le texte temporel et les entiers temporels.

Le développement de base de MobilityDB est effectué par une équipe d'universitaires de l'Université libre de Bruxelles, en Belgique.

Les idées principales, y compris le système de types, les algorithmes et la conception de l'API, ont été validées par la communauté scientifique dans plusieurs publications évaluées par des pairs[72].



*Figure 5. 8Logo mobilityDB*

## Présentation du module SQLAlchemy

SQLAlchemy est une boîte à outils SQL open source et un mappeur relationnel objet (ORM) pour le langage de programmation Python, SQLAlchemy est une bibliothèque qui facilite la communication entre les programmes Python et les bases de données.

La plupart du temps, cette bibliothèque est utilisée comme outil ORM (Object Relational Mapper) qui traduit les classes Python en tables sur des bases de données relationnelles et convertit automatiquement les appels de fonction en instructions SQL.

SQLAlchemy fournit une interface standard qui permet aux développeurs de créer du code indépendant de la base de données pour communiquer avec une grande variété de moteurs de base de données[73].





### **5.3 Données utilisées pour la solution**

Cet ensemble de données de trajectoire GPS a été collecté dans le projet Geolife (Microsoft ResearchAsia) par l'utilisateur sur une période de 6 mois (d'octobre 2008 à mars 2009).

Une trajectoire GPS de cet ensemble de données est représentée par une séquence de points horodatés, dont chacun contient les informations de latitude, longitude.

Cet ensemble de données contient une trajectoire avec une distance totale de 1292 kilomètres et une durée totale de 3506 heures.

Ces trajectoires ont été enregistrées par différents enregistreurs GPS et téléphones GPS, et ont une variété de taux d'échantillonnage.

91,5% des trajectoires sont enregistrées dans une représentation dense, par ex. toutes les 1 à 5 secondes ou tous les 5 à 10 mètres par point.

Cet ensemble de données a recodé un large éventail de mouvements extérieurs des utilisateurs, y compris non seulement les routines de la vie 'comme rentrer à la maison et aller au travail', mais aussi certains divertissements et activités sportives, comme le shopping, les visites, les restaurants, la randonnée et le cyclisme.

Cet ensemble de données de trajectoire peut être utilisé dans de nombreux domaines de recherche, tel que l'exploration de modèles de mobilité, la reconnaissance de l'activité des utilisateurs, les réseaux sociaux basés sur la localisation, la confidentialité de la localisation et la recommandation de localisation.

## 5.4 Implémentation et fonctionnement

### 5.4.1 importation de jeu de donne

Avec l'utilisation de la Library Pandas on va ouvrir le fichier compressé Win rare qui contient notre dataset :

```
[ ] url = '/content/geolife_sample.txt.gz'
df = pd.read_csv(url, sep=',', compression='gzip',infer_datetime_format=True, parse_dates=['datetime'])
tdf = skmob.TrajDataFrame(df, latitude='lat', longitude='lon',datetime='datetime')
print(tdf.head())
print(tdf.tail())
```

	lat	lng	datetime	uid
0	39.984094	116.319236	2008-10-23 05:53:05	1
1	39.984198	116.319322	2008-10-23 05:53:06	1
2	39.984224	116.319402	2008-10-23 05:53:11	1
3	39.984211	116.319389	2008-10-23 05:53:16	1
4	39.984217	116.319422	2008-10-23 05:53:21	1
	lat	lng	datetime	uid
217648	39.999896	116.327290	2009-03-19 05:46:02	5
217649	39.999899	116.327352	2009-03-19 05:46:07	5
217650	39.999945	116.327394	2009-03-19 05:46:12	5
217651	40.000015	116.327433	2009-03-19 05:46:17	5
217652	39.999978	116.327460	2009-03-19 05:46:37	5

Figure 5.10 importation de jeu de donne

### 5.4.2 application du modèle proposé sur le jeu de donné

Pour appliquer le modèle de trajectoire proposé qui est basé sur les arrête dans la trajectoire, ont utilisé une méthode d'une bibliothèque **sci kit** mobilité comme présenté plus haut dans le chapitre, pour celle on doit tout d'abord installé la bibliothèque.

```
!apt-get install -qq curl g++ make
!curl -L http://download.osgeo.org/libspatialindex/spatialindex-src-1.8.5.tar.gz | tar xz
import os
os.chdir('spatialindex-src-1.8.5')
!./configure
!make
!make install
!pip install rtree
!ldconfig
!pip install scikit-mobility
```

Figure 5.11 installation de scikitmobility

```
import skmob
from skmob.preprocessing import detection

stdf = detection.stops(tdf, stop_radius_factor=0.5, minutes_for_a_stop=20.0, spatial_radius_km=0.2, leaving_time=False )
print(stdf.head())
```

Figure 5.12 méthode stop detection

### 5.4.3 application du modèle d'évaluation sur le jeu de donné

Le modèle d'évaluation proposé décrit la trajectoire jour par jour, son principe est de prendre la position de l'objet mobile chaque jour, pour cela on doit procéder par un resampling de minute par minute pour ne pas perdre de l'information, en suit le resampling par jour.

Contrairement au modèle qu'on a proposé, ce modèle d'évaluation décrit la trajectoire en se basant sur temps non pas sur le déplacement, le jeu de données des 2 modèles est presque équivalent en taille ce qui est important pour l'apprentissage pour permettre la comparaison.

```
df_2 = df_1.resample('T').median()
df_2 = df_2.bfill()

df_2.shape

(210335, 2)

print(df_2.isnull().sum())

lat      0
lng      0
dtype: int64
```

Figure 5. 13Resample par minute

```
daily_df = df_2.resample('D').median()

daily_df.head()
```

<b>datetime</b>	<b>lat</b>	<b>lng</b>
2008-10-24	39.998934	116.326776
2008-10-25	40.000104	116.325931
2008-10-26	39.996687	116.326655
2008-10-27	39.996687	116.322375
2008-10-28	40.010840	116.321843

Figure 5.14 Resample par jour

#### 5.4.4 enregistrement de la trajectoire dans la BD

Pour l'enregistrement de la trajectoire dans la base de données **mobilityDb** on procède tout d'abord par l'importation des modules nécessaires pour lier notre code python avec la base de données.

```
!pip install python-mobilitydb

!apt-get install -qq curl g++ make
!curl -L http://download.osgeo.org/libspatialindex/spatialindex-src-1.8.5.tar.gz | tar xz
import os
os.chdir('spatialindex-src-1.8.5')
!./configure
!make
!make install
!pip install rtree
!ldconfig
!pip install scikit-mobility

!pip install mobilitydb_sqlalchemy

!pip install geopandas
```

Figure 5.15 installation des modules

```
import geopandas
gdf = geopandas.GeoDataFrame(tdf, geometry=geopandas.points_from_xy(tdf.lng, tdf.lat))
```

Figure 5.16 adapté le dataframe au mobilityDB

```
print(gdf.head())
print(gdf.tail())

          geometry
datetime
2008-10-24 04:12:30 POINT (116.32134 40.00415)
2008-10-24 04:12:35 POINT (116.32146 40.00383)
2008-10-24 04:12:40 POINT (116.32143 40.00378)
2008-10-24 04:12:45 POINT (116.32143 40.00369)
2008-10-24 04:12:50 POINT (116.32143 40.00359)

          geometry
datetime
2009-03-19 05:46:02 POINT (116.32729 39.99990)
2009-03-19 05:46:07 POINT (116.32735 39.99990)
2009-03-19 05:46:12 POINT (116.32739 39.99995)
2009-03-19 05:46:17 POINT (116.32743 40.00002)
2009-03-19 05:46:37 POINT (116.32746 39.99998)

gdf.dtypes

geometry    geometry
dtype: object
```

Figure 5.17 le type du dataframe

```

[ ] import datetime
import pandas as pd
from mobilitydb_sqlalchemy import TGeomPoint
from shapely.geometry import Point
from sqlalchemy import Column, Integer, create_engine
from sqlalchemy.ext.declarative import declarative_base
from sqlalchemy.orm import sessionmaker

[ ] engine = create_engine('postgresql://postgres:123456789@localhost/trajectory', echo =True)

session = sessionmaker(bind=engine)()

[ ] Base = declarative_base()
class Trips(Base):
    __tablename__ = "trips_test_001"
    trip_id = Column(Integer, primary_key=True)
    trip = Column(TGeomPoint)

[ ] Base.metadata.create_all(engine)

[ ] trip = Trips(trip_id=1, trip=gdf,)
session.add(trip)
session.commit()

```

Figure 5. 18 construction de la table de la bd et enregistrement de la trajectoire

### 5.4.5 préparation de jeu de donnée pour l'apprentissage

Dans cette étape de l'implémentation on va commencer par importer les modules nécessaires au **machine learning** :

```
[ ] # import packages
import math
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.utils import Sequence
from datetime import timedelta
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import time
import os

[ ] import tensorflow as tf
from tensorflow.python.keras.models import Sequential
from tensorflow.python.keras.layers.recurrent import LSTM,RNN
from tensorflow.python.keras.layers.core import Dense, Activation, Dropout
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error
from sklearn.utils import shuffle
```

Figure 5. 19 importation des module d'apprentissage

Maintenant vient l'étape ou en dois divisé notre jeu de donnée en deux un dataframe pour l'entrainement et un autre pour le test dans les proportions de 75% entrainement, 25% test :

```
[ ] train_df,test_df = stdf[1:189], stdf[189:]
```

Figure 5. 20 division du jeu de donne

Après la division du jeu de données on procède par « **Scaling the values** »

```
#Scaling the values
train = train_df
scalers={}
for i in train_df.columns:
    scaler = MinMaxScaler(feature_range=(-1,1))
    s_s = scaler.fit_transform(train[i].values.reshape(-1,1))
    s_s=np.reshape(s_s,len(s_s))
    scalers['scaler_'+ i] = scaler
    train[i]=s_s
test = test_df
for i in train_df.columns:
    scaler = scalers['scaler_'+i]
    s_s = scaler.transform(test[i].values.reshape(-1,1))
    s_s=np.reshape(s_s,len(s_s))
    scalers['scaler_'+i] = scaler
    test[i]=s_s
```

Figure 5. 21Scaling

Dans cette partie du code on construit une méthode pour transformer le **train\_df** et le **test\_df** on **X\_train ,X\_test,Y\_train,Y\_train**.

```
def split_series(series, n_past, n_future):
    #
    # n_past ==> no of past observations
    #
    # n_future ==> no of future observations
    #
    X, y = list(), list()
    for window_start in range(len(series)):
        past_end = window_start + n_past
        future_end = past_end + n_future
        if future_end > len(series):
            break
        # slicing the past and future parts of the window
        past, future = series[window_start:past_end, :], series[past_end:future_end, :]
        X.append(past)
        y.append(future)
    return np.array(X), np.array(y)
```

Figure 5. 22methode split series



```

#For this case, let's assume that given the past 25 stops observation, we need to forecast the next 3 stops observations.
n_past = 25
n_future = 3
n_features = 2

#Now convert both the train and test data into samples using the split_series function.
X_train, y_train = split_series(train.values,n_past, n_future)

X_train = X_train.reshape((X_train.shape[0], X_train.shape[1],n_features))
y_train = y_train.reshape((y_train.shape[0], y_train.shape[1], n_features))
X_test, y_test = split_series(test.values,n_past, n_future)
X_test = X_test.reshape((X_test.shape[0], X_test.shape[1],n_features))
y_test = y_test.reshape((y_test.shape[0], y_test.shape[1], n_features))

```

Figure 5. 23 utilisation de la méthode split séries

### 5.4.6 construction du modèle basé sur le LSTM et GRU

L'architecture de notre modèle comporte une couche d'encoder, une couche une couche LSTM et enfin une couche de décodeur pour le modèle LSTM et le modèle GRU une couche GRU a la place de la couche LSTM.

```

# E1D1
# n_features ==> no of features at each timestep in the data.
#
encoder_inputs = tf.keras.layers.Input(shape=(n_past, n_features))
encoder_l1 = tf.keras.layers.LSTM(100, return_state=True)
encoder_outputs1 = encoder_l1(encoder_inputs)
encoder_states1 = encoder_outputs1[1:]
decoder_inputs = tf.keras.layers.RepeatVector(n_future)(encoder_outputs1[0])
decoder_l1 = tf.keras.layers.LSTM(100, return_sequences=True)(decoder_inputs,initial_state = encoder_states1)
decoder_outputs1 = tf.keras.layers.TimeDistributed(tf.keras.layers.Dense(n_features))(decoder_l1)
model_e1d1 = tf.keras.models.Model(encoder_inputs,decoder_outputs1)
model_e1d1.summary()

```

Figure 5. 24construction du modèle LSTM

```

# E1D1_gru
# n_features ==> no of features at each timestep in the data.
#
encoder_inputs_gru = tf.keras.layers.Input(shape=(n_past, n_features))
encoder_l1_gru = tf.keras.layers.GRU(100, return_state=True)
encoder_outputs1_gru = encoder_l1_gru(encoder_inputs_gru)
encoder_states1_gru = encoder_outputs1_gru[1:]
decoder_inputs = tf.keras.layers.RepeatVector(n_future)(encoder_outputs1_gru[0])
decoder_l1_gru = tf.keras.layers.GRU(100, return_sequences=True)(decoder_inputs,initial_state = encoder_states1_gru)
decoder_outputs1_gru = tf.keras.layers.TimeDistributed(tf.keras.layers.Dense(n_features))(decoder_l1_gru)
model_e1d1_gru= tf.keras.models.Model(encoder_inputs_gru,decoder_outputs1_gru)
model_e1d1_gru.summary()

```

Figure 5. 25construction du modèle GRU

### 5.4.7 entraînement du modèle basé sur le LSTM et GRU

Pour l'entraînement des deux modèles on a choisie comme optimisateur **adam** et la fonction de perte **huberloss**.

```
reduce_lr = tf.keras.callbacks.LearningRateScheduler(lambda x: 1e-3 * 0.90 ** x)
model_e1d1.compile(optimizer=tf.keras.optimizers.Adam(), loss=tf.keras.losses.Huber(), metrics=['accuracy'])
history_e1d1=model_e1d1.fit(X_train,y_train,epochs=25,validation_data=(X_test,y_test),batch_size=32,verbose=0,callbacks=[reduce_lr])
```

Figure 5. 26 entraînement du modèle LSTM

```
[ ] reduce_lr = tf.keras.callbacks.LearningRateScheduler(lambda x: 1e-3 * 0.90 ** x)
model_e1d1_gru.compile(optimizer=tf.keras.optimizers.Adam(), loss=tf.keras.losses.Huber(), metrics=['accuracy'])
history_e1d1_gru=model_e1d1_gru.fit(X_train,y_train,epochs=25,validation_data=(X_test,y_test),batch_size=32,verbose=0,callbacks=[reduce_lr])
```

Figure 5. 27 entraînement du modèle GRU

### 5.4.8 prédiction avec les deux modèles

après la construction et l'entraînement des deux modèles LSTM et GRU on va procéder à la prédiction avec l'utilisation de `X_test`.

```
# Make prediction
def prediction(model):
    prediction = model.predict(X_test)
    for index,i in enumerate(train_df.columns):
        scaler = scalers['scaler_'+i]
        prediction[:, :, index] = scaler.inverse_transform(prediction[:, :, index])

    return prediction

prediction_gru = prediction(model_e1d1_gru)
prediction_lstm = prediction(model_e1d1)
```

Figure 5. 28 méthode pour la prédiction

## 5.5 Evaluation des résultats

### 5.5.1 évaluation avec la fonction de perte

Évaluation des deux modèle LSTM/GRU sur le modèle proposé avec le modèle d'évaluation avec les fonctions MAE, MSE, RMSE :

Tableau 1 comparatif de LSTM et GRU sur le modèle proposé et le modèle d'évaluation avec le MAE :

MAE	Modèle proposé	Modèle d'évaluation
LSTM	0.2434	0.76710
GRU	0.2196	0.8017

Tableau 2 comparatif de LSTM et GRU sur le modèle proposé et le modèle d'évaluation avec le MSE :

MSE	Modèle proposé	Modèle d'évaluation
LSTM	0.1024	4.02570
GRU	0.0587	2.85680

Tableau 3 comparatif de LSTM et GRU sur le modèle proposé et le modèle d'évaluation avec le RMSE :

RMSE	Modèle proposé	Modèle d'évaluation
LSTM	0.3200	0.2424
GRU	0.2006	0.8017

Les tableau 1, Tableau 2 et Tableau 3 regroupe les résultat obtenu des fonctions de perte MAE, MSE, RMSE sur les deux modèle de prédiction, et le modèle de trajectoire proposé et le modèle d'évaluation, nous remarquons que la valeur de perte du modèle proposé est nettement inférieure au modèle d'évaluation, et le modèle de prédiction qui utilise des couche de GRU a le taux de perte le plus bas.

On conclut que le modèle GRU et performant quel le modèle LSTM sur le modèle de trajectoire proposé « modèle basé sur les arrête »

### 5.5.2 évaluation avec les graphes de perte et de précision

Dans cette partie on affiche les graphes obtenus des deux modèles LSTM,GRU sur le modèle proposé :

Le graphe de perte du modèle GRU :

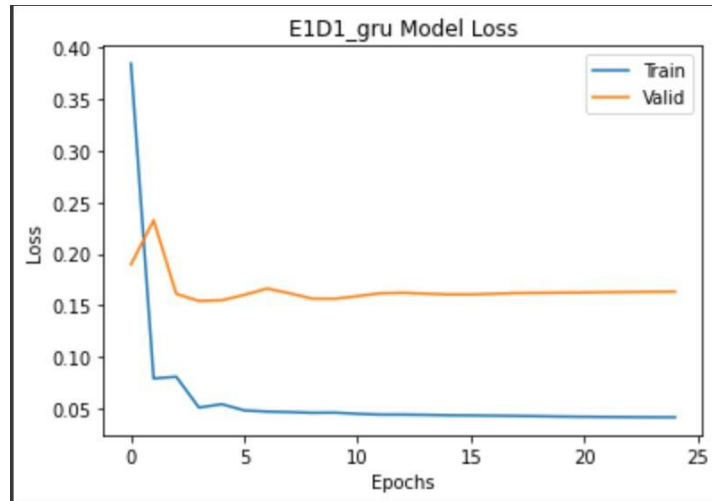


Figure 5. 29 graph de la fonction loss GRU du modèle proposé

Le graphe de validation accuracy /training accuracy :

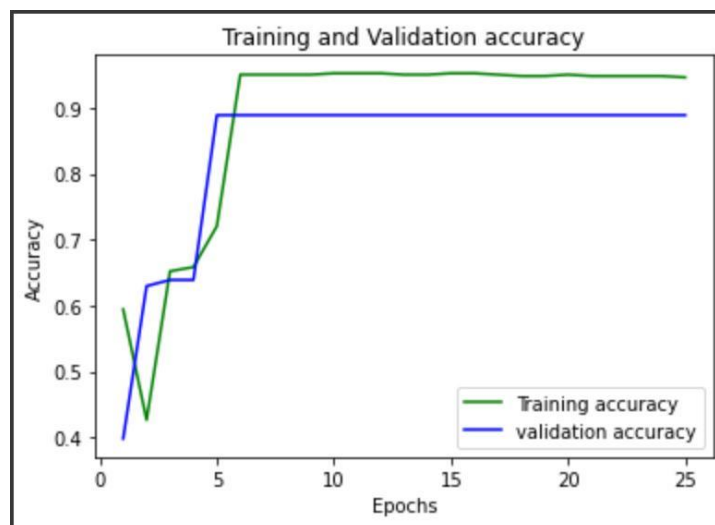


Figure 5. 30 graphe représente training accu et validation accu GRU du modèle proposé

Le graphe de perte du modèle LSTM:

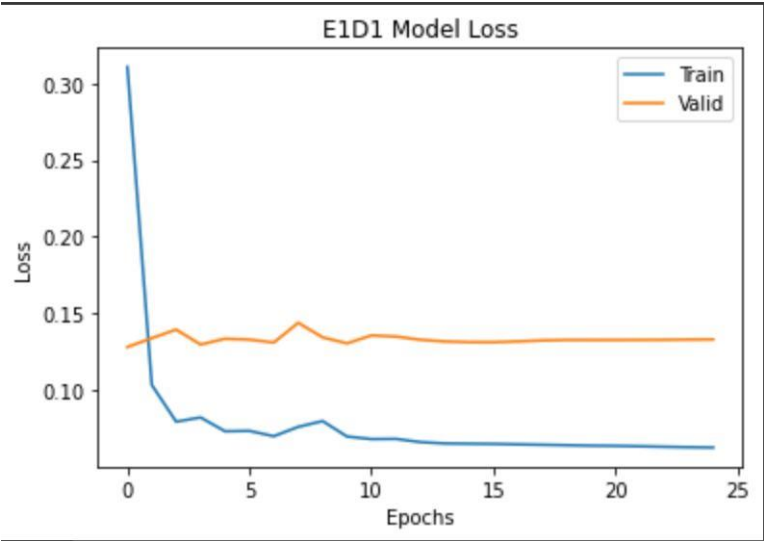


Figure 5. 31graph de la fonction loss LSTM du modèle proposé

Le graphe de validation accuracy /training accuracy :

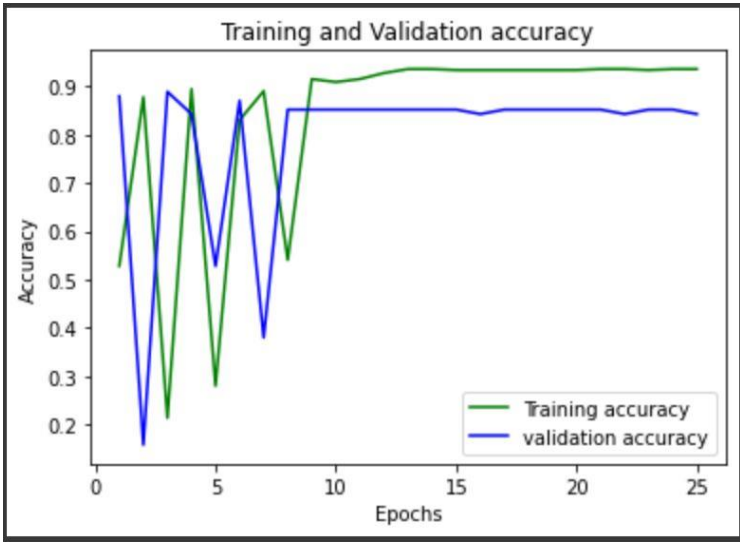


Figure 5. 32graphe représente training ccu et validation accu LSTM du modèle proposé

Dans les graphes suivant on affiche les résultats de perte et de validation accuracy du modèle d'évaluation :

Le graphe de perte du modèle GRU:

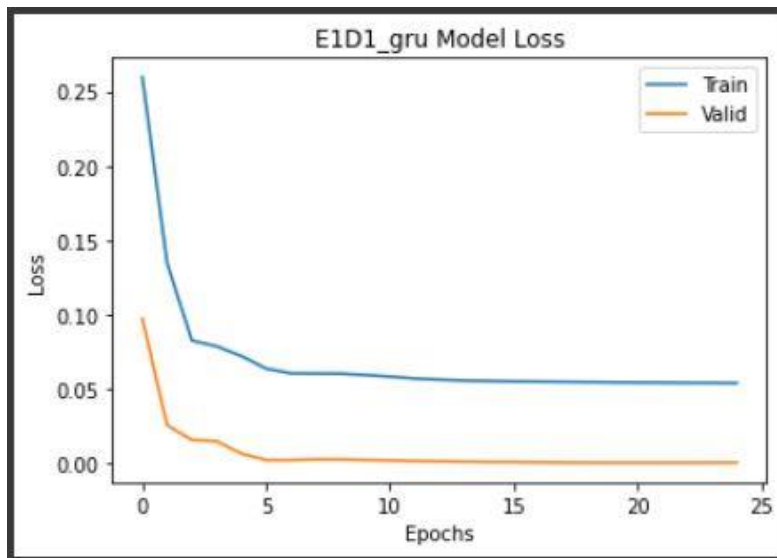


Figure 5. 33graph de la fonction loss GRU du modèle d'évaluation

Le graphe de validation accuracy /training accuracy :

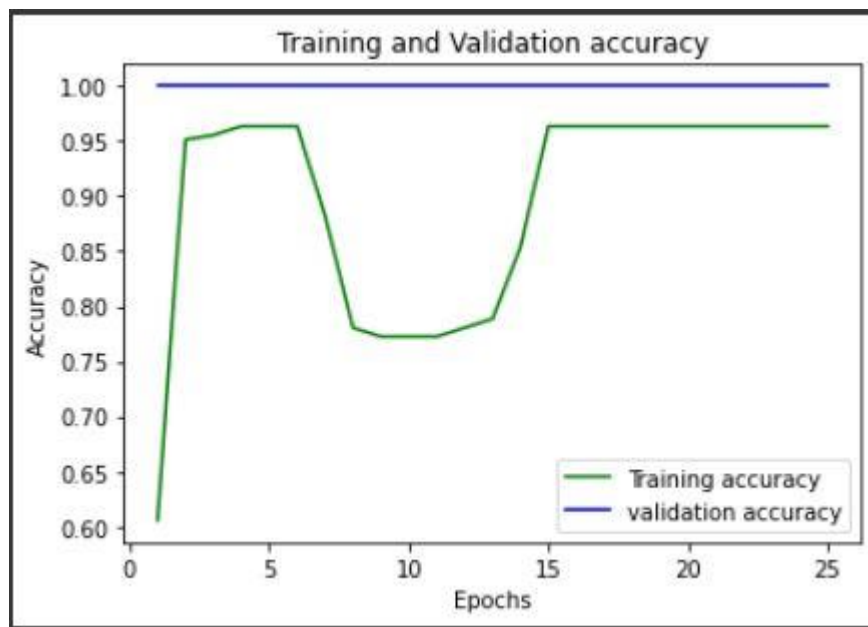


Figure 5. 34 graphe représente training accu et validation accu GRU du modèle d'évaluation

Le graphe de perte du modèle LSTM :

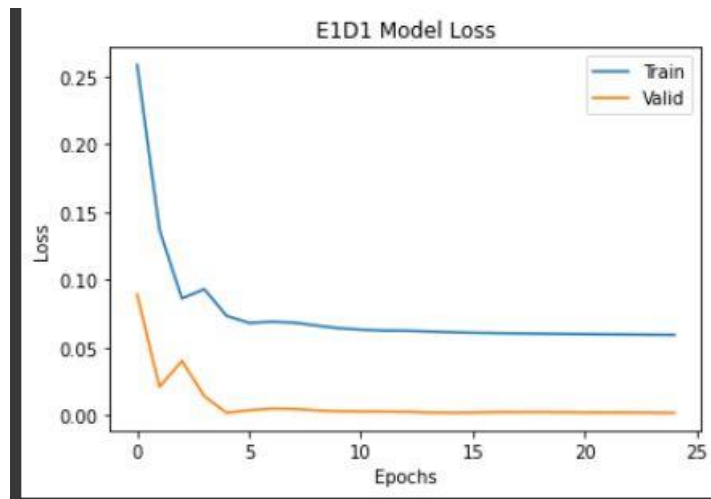


Figure 5. 35 graph de la fonction loss LSTM du modèle d'évaluation

Le graphe de validation accuracy /training accuracy :

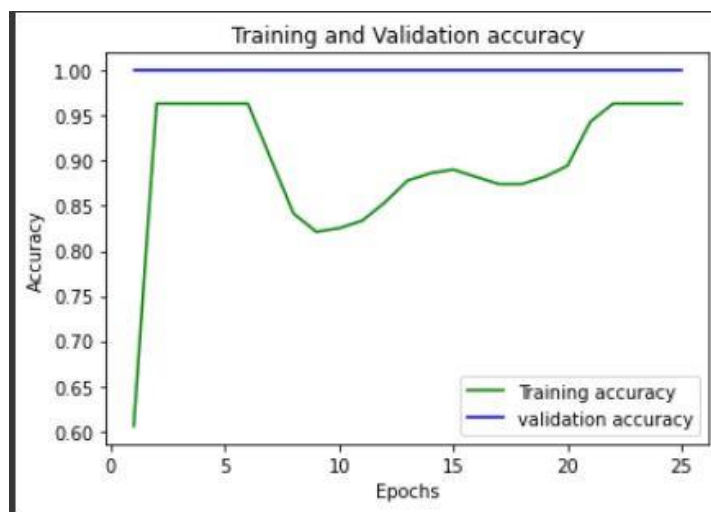


Figure 5. 36graphe représente training accu et validation accu LSTM du modèle d'évaluation

Dans les figures de **training loss et validation loss** nous remarquons que la différence entre la valeur de perte entre le **training loss** et le **validation loss** est entre 0.1 et 0.05.

Dans les figures qui représentent les performances des deux modèles LSTM et GRU sur le modèle d'évaluation, on constate qu'il y a un problème de « **Underfitting** », mauvaise performance sur les données d'entraînement et mauvaise généralisation à d'autres données.

Le graphe de **training accuracy validation accuracy** du modèle LSTM sur le modèle proposé souffre de fluctuation dans les premier epoch puis se stabilise.

Le graphe de **training accuracy validation accuracy** du modèle GRU sur le modèle proposé, ne souffre pas de problème de **underfitting** ni de **overfitting**.

On conclut que le modèle GRU sur le modèle de trajectoire basé sur les arrête a une bonne performance que celle du modèle LSTM, les modèle d'évaluation ne présente pas une bonne performance et une stabilité.

## **5.6 Conclusion**

A travers ce chapitre nous avons présenté les outils matériels et logiciels utilisés pour l'implémentation de notre modèle. Nous avons par la suite présenté en détail cette dernière et enfin nous avons exposé les résultats obtenus par notre modèle tout en les évaluant en les comparants avec les performances d'autres architectures.



## *Conclusion et perspective*

Parmi les données utilisées, celles liées à la mobilité des individus sont nécessaires pour des solutions visant l'amélioration de plusieurs domaines tels que la recommandation de service, le transport, les réseaux cellulaires, etc. Une des solutions qui puisse avoir un apport positif dans ces domaines et qui nécessite l'utilisation de ce type de données est la *prédiction* de la mobilité.

Nous sommes parvenus, par le biais de ce projet, à proposer une solution pour description de la trajectoire humaine, pour but de la prédire en se basant sur l'apprentissage profond, qui consiste à une couche d'autoencodeur et une couche de RNN (LSTM, GRU) compris, notre modèle a montré de bonnes performances sur le jeu de données, que nous avons créé à partir d'un jeu de données existant.

Cependant il reste à évaluer ce modèle, avec un ensemble de données bien plus volumineux et représentatif des données de mobilité dans une ville intelligente.

Ce travail fut une occasion pour nous de compléter de manière transversale nos compétences en informatique et d'élargir et d'approfondir nos connaissances en intelligence artificielle, aussi les connaissances dans le domaine de la mobilité humaine.

Cependant des perspectives d'améliorations de notre solution restent envisageables pour être enrichies par des fonctionnalités avancées telles que la possibilité de travailler avec plus de trajectoires de plusieurs individus.

## Bibliographie

- [1] Elodie Buard, Thomas Devogele, César Ducruet. « Trajectoires d'objets mobiles dans un espace support fixe ». Revue Internationale de Géomatique, Lavoisier, 2015, MODÉLISER LES DYNAMIQUES SPATIALES, 25 (3), pp.331-354. hal-01171521
- [2] Parent, C., Spaccapietra, S., Vangenot, C., Damiani, M. L., Machedo, M., & Porto, F. (2008). « Modélisation conceptuelle de trajectoires ».
- [3] Chardonnel, S., Du Mouza, C., Fauvet, M. C., Josselin, D., & Rigaux, P. (2004, June). « Patrons de mobilité : proposition de définition, de méthode de représentation et d'interrogation ». In Colloque Cassini'04-7ème conférence du GDR Sigma" Géomatique et Analyse Spatiale (pp. 19-23).
- [4] GÖNDÖR, Sebastian, UZUN, Abdalbaki, ROHRMANN, Till, et al. Predicting user mobility in mobile radio networks to proactively anticipate traffic hotspots. In : 2013 International Conference on MOBILE Wireless MiddleWARE, Operating Systems, and Applications. IEEE, 2013. p. 29-38.
- [5] ZHOU, Changqing, FRANKOWSKI, Dan, LUDFORD, Pamela, et al. Discovering personal gazetteers: an interactive clustering approach. In : Proceedings of the 12th annual ACM international workshop on Geographic information systems. 2004. p. 266-273.
- [6] Supervised vs Unsupervised Learning Understanding the differences between the two main types of machine learning methods ,Devin Soni, mars 22 , 2018
- [7] Understanding CNN (Convolutional Neural Network) Your first baby step to learn Deep Learning for Image Classification, Vincent Tatan, Dec 23, 2019
- [8] A Guide to RNN: Understanding Recurrent Neural Networks and LSTM Networks ,Niklas Donges, July 29, 2021
- [9] Illustrated Guide to LSTM's and GRU's: A step by step explanation, Michael Phi, Sep 24, 2018
- [10] Understanding GRU Networks, Simeon Kostadinov, Dec 16, 2017
- [11] Applied Deep Learning - Part 3: Autoencoders Arden Dertat Arden Dertat ; Oct 3, 2017

[12]Generative adversarial network: An overview of theory and applications Alankrita Aggarwal a , Mamta Mittal b , GopiBattinenic,a Department of Computer Science and Engineering, Panipat Institute of Engineering and Technology, Samalkha 132101, India.

[13]Liu, M.Y., &Tuzel, en O. (2016). “Coupled generative adversarial networks”,Loey, M., Smarandache, F., &Khalifa, en N. E. M. (2020). Within the lack of chest COVID19 X-ray dataset: A novel detection model based on GAN and deep transfer learning.Symmetry (Basel).. 10.3390/SYM12040651.

chapitre 2

[14] KOMNINOS, Nicos - Intelligent cities and globalisation of innovation network;Abingdon: Routledge, 2008, p. 267.

[15] CARAGLIU, Andrea & DEL BO, Chiara & NIJKAMP, Peter - "Smart cities in Europe", Serie Research Memoranda 0048; VU University Amsterdam, Faculty of Economics, Business Administration and Econometrics, 2009.

[16] Europea Commission - Europe 2020: a strategy for intelligent, sustainable and inclusive growth; Bruxelles, 2010.

[17] GIFFINGER, Rudolf; FERTNER, Christian; KRAMAR, Hans; KALASEK, Robert;PICHLER-MILANOVIC, Nataša; MEIJERS, Evert - Smart cities. Ranking of European medium-sized cities, Final report; Centre of Regional Science, Vienna UT, 2007.

[18] Joëlle Simard, Florence PaulhiacScherrer,LA VILLE INTELLIGENTE COMME VECTEUR POUR LE DÉVELOPPEMENT DURABLE : LE CAS DE LA VILLE DE MONTRÉAL,UNIVERSITÉ DE SHERBROOKE,Juillet 2015.

[19]Ana Isabel Torre-Bastida , Javier Del Ser,IbaiLaña, MaitenaIardia, MirenNekaneBilbao,Sergio Campos-Cordobést,Big Data for transportation and mobility:recent advances, trends and challenges,University of the Basque Country (UPV/EHU),july 2018.

[20](Golderzahi et Pao 2018, Sahni et al. 2018) .

[21]Hashem, I.A.T., et al., 2016. The role of big data in smart city. International Journal of Information Management, 36 (5), 748–758. doi:10.1016/j.ijinfomgt.2016.05.002.

- [22]Tsiropoulou, E.E., et al., 2017. RFID-based smart parking management system. *Cyber-Physical Systems*, 3 (1–4), 22–41. doi:10.1080/23335777.2017.1358765.
- [23]Rawat, P., et al., 2014. Wireless sensor networks: a survey on recent developments and potential synergies. *The Journal of Supercomputing*, 68 (1), 1–48.doi:10.1007/s11227-013-1021-9.
- [24]Sivanandam, R., et al. *Advanced Traveller Information System (ATIS) for Indian* ,2012.
- [25]Gonzalez, M.C., Hidalgo, C.A., and Barabasi, A.L., 2008. Understanding individual human mobility patterns. *Nature*, 453 (7196), 779. doi:10.1038/nature06958.
- [26]Deville, P., et al.,. Scaling identity connects human mobility and social interactions. *Proceedings of the National Academy of Sciences*, 113 (26), 7047–7052. doi:10.1073/pnas.15254,2016.
- [27]Bazzani, A., et al. 2011, May. Now casting of traffic state by GPS data. The metropolitan area of Rome. In: 2011 Proceedings of the 34th International Convention MIPRO. Opatija, Croatia: IEEE, 1615–1618.
- [28] Bin Guo, Zhu Wang, Zhiwen Yu, Yu Wang, Neil Y. Yen, Runhe Huang, Xingshe Zhou: *Mobile Crowd Sensing and Computing: The Review of an Emerging Human-Powered Sensing Paradigm*. *ACM Comput. Surv.* 48(1): 7:1-7:31 (2015).
- [30] C. Parent, S. Spaccapietra, E. Zimanyi, *Conceptual Modeling for Traditional and Spatio-Temporal Applications – The MADS Approach*, Springer, 2006.
- [31]I. Feathertone, N. Zhang “Amobility monitoring based advance reservation protocol” Q2Swin et 06 spin Oct 2006.
- [32] Vut Hong Nhan, K H Ryu “Future locatiT Liu, P Bahl “Mobility modeling location tracking and trajectory prediction in wireless ATM networks”; IEEE vol °16 n°06 Aug 98 on prediction of moving objects based on movements rules”; LNCIS 344 ICIC 2006.
- [33] T Liu, P Bahl “Mobility modeling location tracking and trajectory prediction in wireless ATM networks”; IEEE vol °16 n°06 Aug 98.
- [34] R.H. Gu“ting, M. Schneider, *Moving Objects Databases*. Diane D. Cerra, 2005.

- [35] V Bogorny et al., CONSTAnT – A Conceptual Data Model for Semantic Trajectories, p67-68-69.
- [36] Jinzhong Wang, Lijun Sun et Feng Xia, Urban Human Mobility: Data-Driven Modeling and Prediction in researchGate p1,2008.
- [37] V Bogorny et al., CONSTAnT – A Conceptual Data Model for Semantic Trajectories, p70.
- [38] Kathleen Stewart et Max J Egenhofer, Modeling Moving Objects Over Multiple Granularities in researchGate p1.
- [39] L. Song, D. Kotz, R. Jain, and X. He. Evaluating next-cell predictors with extensive wi-fi mobility data. *IEEE Transactions on Mobile Computing*, 5(12):1633–1649, 2006.
- [40] S. Qiao, D. Shen, X. Wang, N. Han, and W. Zhu. A self-adaptive parameter selection trajectory prediction approach via hidden markov models. *IEEE Transactions on Intelligent Transportation Systems*, 16(1):284–296, 2015
- [41] F. Terroso-Sáenz, J. Cuenca-Jara, A. González-Vidal, and A. F. Skarmeta. Human mobility prediction based on social media with complex event processing. *International Journal of Distributed Sensor Networks*, 12(9):5836392–5836392, 2016.
- [42] Y. Qiao, Z. Si, Y. Zhang, F. B. Abdesslem, X. Zhang, J. Yang, Y. Qiao, Z. Si, Y. Zhang, and F. B. Abdesslem. A hybrid markov-based model for human mobility prediction. *Neurocomputing*, 278:99–109, 2018.
- [43] M. Yin, M. Sheehan, S. Feygin, J. Paiement, and A. Pozdnoukhov. A generative model of urban activities from cellular data. *IEEE Transactions on Intelligent Transportation Systems*, 19(6):1682–1696, 2018.
- [44] J. Ziv and A. Lempel. Compression of individual sequences via variable-rate coding. *IEEE Press, New Jersey*, 1978.
- [45] A. Bhattacharya and S. K. Das. Lezi-update: an information-theoretic approach to track mobile users in pcs networks. pages 1–12, 1999.
- [46] M. R. Alam, M. B. I. Reaz, and M. A. M. Ali. Speed: An inhabitant activity prediction algorithm for smart homes. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 42(4):985–990, 2012.

- [47]K. Gopalratnam and D. J. Cook. Online sequential prediction via incremental parsing: The active lezi algorithm. *IEEE Intelligent Systems*, 22(1):52–58, 2007.
- [48]S. K. Pulliyakode and S. Kalyani. A modified ppm algorithm for online sequence prediction using short data records. *Communications Letters IEEE*, 19(3):423–426, 2015.
- [49]G. Walker. On periodicity in series of related terms. *Proceedings of the Royal Society of London*,131(818):518–532, 1931.
- [50]H. Tong, T. K. Kumar, and Y. Huang. *Stationary Time Series Models*. Wiley-Blackwell, New Jersey,2011.
- [51]F. N. David. Hypothesis-testing in time series analysis.*Biometrika*, 39(265):300–325, 1952.
- [52]G. Box and G. M. Jenkins. *Box and Jenkins: Time Series Analysis, Forecasting and Control*, pages 161–215. Palgrave Macmillan UK, Basingstoke, Hampshire, 2013.
- [53]Y. Yang, N. V. Chawla, P. Basu, B. Prabhala, and T. L. Porta. Link prediction in human mobility networks. In *Ieee/acm International Conference on Advances in Social Networks Analysis and Mining*, pages380–387, 2013.
- [54]M. De Domenico, A. Lima, and M. Musolesi. Interdependence and predictability of human mobility and social interactions. *Pervasive & Mobile Computing*,9(6):798–807, 2012.
- [55]S. Zeng, H. Wang, Y. Li, and D. Jin. Predictability and prediction of human mobility based on applicationcollected location data. In *IEEE International Conference on Mobile Ad Hoc and Sensor Systems*, pages28–36, 2017.
- [56]J. Joseph, F. Doshi-Velez, and N. Roy. A bayesian nonparametric approach to modeling mobility patterns. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, pages 1587–1593, Atlanta, 2010.
- [57]X. Song, H. Kanasugi, and R. Shibasaki. Deeptransport: Prediction and simulation of human mobility and transportation mode at a citywide level. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, pages 2618–2624, New York, 2016. International Joint Conferences on Artificial Intelligence Organization.

- [58]A. Monreale, F. Pinelli, R. Trasarti, and F. Giannotti. Wherenext: A location predictor on trajectory pattern mining. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 637–646, New York, 2009. ACM.
- [60]J. Feng, Y. Li, C. Zhang, F. Sun, F. Meng, A. Guo, and D. Jin. Deepmove: Predicting human mobility with attentional recurrent networks. In In Proceedings of the 2018 World Wide Web Conference on World Wide Web, pages 1459–1468, Lyon, 2018.
- [61]P. Baumann, C. Koehler, A. Dey, and S. Santini. Selecting individual and population models for predicting human mobility. IEEE Transactions on Mobile Computing, 2018.
- [62]K. Ouyang, R. Shokri, D. S. Rosenblum, and W. Yang. A non-parametric generative model for human trajectories. In Proceedings of the 27th International Joint Conference on Artificial Intelligence, pages 3812–3817, Stockholm, 2018.
- [64] J. Yuan, Y. Zheng, and X. Xie, “Discovering regions of different functions in a city using human mobility and pois,” in SIGKDD’12, Beijing, China, August 12-16, 2012.
- [65]Kuhlman, Dave. "A Python Book: Beginning Python, Advanced Python, and Python Exercises". Section 1.1. Archived from the original (PDF) on 23 June 2012.
- [66]Abadi, Martín; Barham, Paul; Chen, Jianmin; Chen, Zhifeng; Davis, Andy; Dean, Jeffrey; Devin, Matthieu; Ghemawat, Sanjay; Irving, Geoffrey; Isard, Michael; Kudlur, Manjunath; Levenberg, Josh; Monga, Rajat; Moore, Sherry; Murray, Derek G.; Steiner, Benoit; Tucker, Paul; Vasudevan, Vijay; Warden, Pete; Wicke, Martin; Yu, Yuan; Zheng, Xiaoqiang . "TensorFlow: A System for Large-Scale Machine Learning" (PDF),2016.
- [67]Pine, David (2014). "Python resources". Rutgers University. Retrieved 2017-04-07.
- [68]Chollet, François . "Xception: Deep Learning with Depthwise Separable Convolutions". 2016
- [69]Wes McKinney. "pandas: a Foundational Python Library for Data Analysis and Statistics" (PDF). Retrieved 2 August 2018.
- [70]"Matplotlib Lead Developer Explains Why He Can't Fix the Docs—But You Can – NumFOCUS". NumFOCUS. 2017-10-05.
- [71]Roberto Pellungrin, Gianni Barlacchi, Luca Pappalardo, scikit-mobility: a Python library for the analysis, October 2019.

[72]Esteban Zimanyi,MahmoudSakr, MobilityDB: A Mobility Database Based on PostgreSQL and PostGIS,December 2020

[73]Michael ,Bayer,SQLAlchemy,MITLicense,May 11, 2021

[74]Tiago CarneiroPessoa,RaulMedeiros,ThiagoNepomuceno,Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications,October 2018