

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Université de Béjaia  
Faculté des Sciences Exactes  
Département d'Informatique

## MEMOIRE DE MASTER RECHERCHE

Informatique

Option

*Systèmes d'Information Avancés*

Thème

Conception d'un système de stationnement  
intelligent en utilisant des solutions IdO et Big Data

Présenté par :

Mlle. Ferroudj Wissam

Mlle. Djellaoui Kenza

Soutenu le 25 Juin 2023 devant le jury composé de :

Président	Dr. M. Moktefi	MCB	Université de Béjaia
Promotrice	Pr. L. Bouallouche	Professeur	Université de Béjaia
Co-promotrice	Dr. S. Ouyahia	MCB	Université de Béjaia
Examineur	Dr. YA. Bennai	MCB	Université de Béjaia
Examineur	M. A. Goutal	Doctorant LMD	Université de Béjaia

Béjaia, Juin 2023.

## *\* Remerciements \**

Nous souhaitons exprimer notre profonde gratitude à notre Créateur pour nous avoir accordé la force, l'endurance et la persévérance nécessaires pour mener à bien ce travail. Sans sa grâce et son soutien, nous n'aurions pas pu atteindre cet objectif.

Nous tenons également à remercier chaleureusement notre encadreur, le Pr. Bouallouche, ainsi que nos co-encadreurs, Mme Yessad et Mme Lahlah, pour leur précieuse aide, leur expertise et leur soutien tout au long de notre mémoire. Leur disponibilité, leur patience et leurs commentaires constructifs ont grandement contribué à améliorer la qualité de notre travail, le rendant plus cohérent et précis.

Nous n'oublions pas non plus nos enseignants et nos familles, en particulier nos parents, nos frères et sœurs, pour leur amour, leur confiance, leurs conseils et leur soutien inconditionnel tout au long de nos études et de la réalisation de ce mémoire. Leur encouragement et leur soutien moral ont été une source d'inspiration et de motivation pour nous. Nous sommes conscients que sans leur appui, nous n'aurions pas pu accomplir ce travail avec autant de succès.

Nous tenons donc à adresser nos sincères remerciements à tous ceux qui ont contribué à la réussite de ce travail. Que Dieu vous bénisse et vous protège.

✱ *Dédicaces* ✱

Je dédie ce mémoire .....

À mes parents extraordinaires,

Quelle dédicace pourrait vous être supérieure, vous qui êtes au-delà de toutes les louanges et de tous les mots? Je profite cette occasion mémorable où je soutiens mon master, pour vous exprimer ma profonde gratitude pour tout ce que vous avez fait. Votre soutien inébranlable, votre amour inconditionnel et vos sacrifices ont été les clés de mon parcours. Ma réussite n'aurait pas été possible sans vous. Je vous souhaite tout le bonheur, Je prie pour que le bon Dieu m'aide à vous rendre fiers et à vous offrir le bonheur que vous méritez.

À ma sœur et mon frère bien-aimés,

Cette dédicace est un témoignage de l'amour et de l'appréciation que j'ai pour vous. Vous êtes bien plus que des frères et sœurs pour moi, vous êtes mes complices, mes amis et mes confidents. Votre présence dans ma vie a été une bénédiction inestimable.

À mon oncle Mohamed,

A mon deuxième père, je dédie cette dédicace avec tout mon amour et ma gratitude sincère. vous êtes unique et exemplaire dans votre rôle d'oncle aimant et attentionné.

À ma chère famille proche, oncles et tantes, cousins et cousines, à mes grands parents et amis, Enfin à la petite Émilie avec sa présence joyeuse et innocente.  
Et tous ceux qui m'aiment.

*M. FERROUDJ Wissam*

✧ *Dédicaces* ✧

Je dédie ce mémoire .....

À mes chers parents,

Aucun mot ne peut exprimer pleinement l'amour inconditionnel, les sacrifices et le soutien que vous m'avez offerts. Vous êtes une source inépuisable de bienveillance et de guidance dans ma vie. Je tiens à vous exprimer mon profond respect pour tout ce que vous avez fait et continuez de faire pour moi.

Que Dieu vous accorde une santé robuste, une longue vie remplie de bonheur et de sérénité. Je prie pour qu'Il me donne la capacité de vous apporter le bonheur que vous méritez, en retour de tout ce que vous avez donné.

À mes chers grands frères,

Il est impossible d'exprimer par de simples mots toute la gratitude et l'amour profond que je porte envers vous. Vous êtes bien plus que des frères pour moi, vous êtes ma source infinie de force, de joie et de réconfort.

À mes grands parents et mes deux tentes,

Je vous porte dans mon cœur avec une profonde gratitude et un amour infini.

À mes chers amis.

Katia, Dihya , Nabila , Wissam, Ouardia, Khalida et Shelly Je vous remercie du fond du cœur pour votre amitié sincère et votre soutien inconditionnel.

À l'hommage de mon grand père

Que Dieu le Tout-Puissant lui accorde Son infinie Miséricorde et l'accueille dans Son éternel Paradis.

À toute ma famille proche soit elle ou lointaine.

Et tous ceux qui m'aiment.

*M. DJELLAOUI kenza*

## RÉSUMÉ

Trouver rapidement une place de stationnement est désormais une nécessité quotidienne incontournable pour nous tous, dans toutes les zones urbaines. Cependant, la gestion inadéquate des ressources en matière de stationnement entraîne de nombreux problèmes. La recherche de places de stationnement est devenue un défi majeur pour les conducteurs, entraînant une augmentation de la congestion du trafic, de la pollution atmosphérique, des pertes de temps considérables et une grande frustration.

Face à cette problématique, les solutions basées sur l'IdO, le big data et l'apprentissage automatique offrent des alternatives prometteuses. C'est dans ce contexte que ce mémoire intervient avec deux contributions majeures. La première se concentre sur la proposition d'une architecture flexible pour la gestion des données big data générées par différents capteurs installés dans les parcs de stationnement. La deuxième contribution concerne un modèle d'apprentissage profond hybride Conv1D-BiGRU qui prédit avec précision l'occupation des places à court terme. Les résultats obtenus démontrent des performances élevées, ouvrant la voie à une meilleure gestion des ressources de stationnement, avec une MSE de 0.00017, un MAE de 0.0095, un RMSE de 0.0129 et une précision d'environ 99%.

**Mots clés :** Stationnement intelligent, Big data, IdO, apprentissage automatique, apprentissage profond, Architecture, modèle hybride Conv1D-BiGRU.

## ABSTRACT

Quickly finding a parking space has become an indispensable daily necessity for all of us, in all urban areas. However, inadequate parking resource management leads to numerous problems. The search for parking spaces has become a major challenge for drivers, resulting in increased traffic congestion, air pollution, significant time losses, and immense frustration.

In response to this issue, solutions based on IoT (Internet of Things), big data, and machine learning offer promising alternatives. It is within this context that this master's dissertation intervenes with two major contributions. The first focuses on proposing a flexible architecture for managing big data generated by various sensors installed in parking lots. The second contribution relates to a hybrid Conv1D-BiGRU deep learning model that accurately predicts short-term parking space occupancy. The obtained results demonstrate high performance, paving the way for improved parking resource management, with an MSE (Mean Squared Error) of 0.00017, an MAE (Mean Absolute Error) of 0.0095, an RMSE (Root Mean Squared Error) of 0.0129, and an accuracy of approximately 99%.

**Keywords :** Smart Parking, Big data, IoT, machine learning, deep learning, Architecture, hybrid Conv1D-BiGRU model.

# Table des matières

<b>Table des matières</b>	<b>i</b>
<b>Liste des figures</b>	<b>v</b>
<b>Liste des tableaux</b>	<b>vii</b>
<b>Liste des acronymes</b>	<b>ix</b>
<b>Liste de communication</b>	<b>xi</b>
<b>Introduction générale</b>	<b>1</b>
<b>1 Généralités sur le stationnement intelligent</b>	<b>3</b>
1.1 Introduction . . . . .	3
1.2 Les villes intelligentes . . . . .	3
1.2.1 Définition . . . . .	4
1.2.2 Domaines des villes intelligentes . . . . .	5
1.3 Stationnement intelligent . . . . .	7
1.3.1 Définition . . . . .	7
1.3.2 Fonctionnement de stationnement intelligent . . . . .	7
1.4 La technologie IdO . . . . .	8
1.4.1 Définition . . . . .	8
1.4.2 Utilisation . . . . .	9
1.5 Big data . . . . .	11
1.5.1 Définition . . . . .	11
1.5.2 Caractéristiques du Big data . . . . .	11
1.5.3 Principaux enjeux du Big data . . . . .	14
1.6 Machine learning . . . . .	15
1.6.1 Notions liées à l'apprentissage automatique . . . . .	15

---

1.6.2	Algorithmes d'apprentissage automatique appliqués dans le stationnement intelligent . . . . .	17
1.7	Avantages de la mise en place de stationnement intelligent . . . . .	20
1.8	Marché de stationnement intelligent . . . . .	21
1.9	Conclusion . . . . .	21
<b>2</b>	<b>État de l'art sur le stationnement intelligent</b>	<b>23</b>
2.1	Introduction . . . . .	23
2.2	Objectifs de recherche dans le domaine du stationnement intelligent . . . . .	23
2.3	Synthèse des travaux connexes . . . . .	24
2.3.1	Approches basées sur la gestion et le traitement des données massives	24
2.3.2	Approches basées sur la détection de l'occupation des places de stationnement . . . . .	27
2.3.3	Comparatif des travaux basés sur la détection de l'occupation des places de stationnement . . . . .	30
2.3.4	Approches basées sur la prévision de l'occupation des places de stationnement . . . . .	31
2.3.5	Comparatif des travaux basés sur la prévision de l'occupation des places de stationnement . . . . .	34
2.4	Progrès des systèmes de stationnement intelligents en Algérie . . . . .	35
2.5	Constat et discussion . . . . .	36
2.6	Conclusion . . . . .	37
<b>3</b>	<b>Nouvelle architecture d'analyse Big Data pour le stationnement intelligent</b>	<b>38</b>
3.1	Introduction . . . . .	38
3.2	Choix des outils/platformes BD . . . . .	38
3.2.1	Scalabilité . . . . .	39
3.2.2	Tolérance aux pannes . . . . .	39
3.2.3	Modes de traitement des méga données . . . . .	39
3.2.4	Taille des données supportées . . . . .	41
3.2.5	Sécurité . . . . .	41
3.3	Infrastructure de stockage . . . . .	41
3.3.1	HBase . . . . .	42
3.3.2	Mongo DB . . . . .	42
3.3.3	Cassandra . . . . .	42
3.3.4	Comparaison entre les bases de données NoSQL . . . . .	43

---

3.4	Plateformes de traitement et d'analyse de données . . . . .	45
3.4.1	Hadoop . . . . .	45
3.4.2	Spark : 3G de Big Data . . . . .	45
3.4.3	Flink : 4G de Big Data . . . . .	45
3.4.4	Kafka . . . . .	46
3.4.5	Comparaison entre Spark, Hadoop et Flink . . . . .	46
3.5	Nouvelle Architecture d'analyse de Big Data . . . . .	48
3.5.1	Présentation générale de l'architecture . . . . .	48
3.5.2	Les différents niveaux de notre architecture . . . . .	49
3.6	Conclusion . . . . .	52
<b>4</b>	<b>Prévision de l'occupation des espaces de stationnement, proposition, implémentation et validation</b>	<b>53</b>
4.1	Introduction . . . . .	53
4.2	Motivation . . . . .	53
4.3	Gated Recurrent Unit . . . . .	54
4.3.1	Architecture GRU . . . . .	54
4.3.2	Avantages de GRU . . . . .	55
4.4	Nouveau modèle hybride Conv1D-BiGRU pour la Prévision de l'occupation des parkings . . . . .	56
4.5	Outils et bibliothèques utilisés . . . . .	57
4.5.1	Python . . . . .	57
4.5.2	Google Collaboratory . . . . .	57
4.5.3	Tensorflow et Keras . . . . .	58
4.6	Présentation de l'ensemble de données . . . . .	58
4.7	Tendances et analyse périodique des données . . . . .	60
4.7.1	Analyse des tendances d'occupation des parkings : . . . . .	60
4.7.2	Analyse de corrélation entre les variables . . . . .	63
4.8	Pré-traitement / Préparation des données . . . . .	64
4.8.1	Nettoyage . . . . .	64
4.8.2	Ajout des variables endogènes et exogènes . . . . .	65
4.8.3	Normalisation des données . . . . .	66
4.8.4	Création des séquences . . . . .	66
4.8.5	Fractionnement des données . . . . .	67
4.8.6	Préparation des tenseurs . . . . .	68
4.9	Mise en œuvre et évaluation . . . . .	69



4.9.1 Métriques de performances . . . . .	69
4.9.2 Architecture du modèle et les paramètres associés . . . . .	70
4.9.3 Présentation des résultats . . . . .	72
4.10 Conclusion . . . . .	77
<b>Conclusion et perspectives</b>	<b>78</b>
<b>Bibliographie</b>	<b>80</b>

# Liste des figures

1.1	Illustration de différents objets du quotidien connectés à Internet. . . . .	9
1.2	Le nombre total d'appareils connectés à l'Internet des Objets (IdO) installés dans le monde entier, de 2018 à 2025, en milliards. . . . .	10
1.3	Différence entre perception classique et un neurone récurant. . . . .	19
1.4	Le marché des systèmes de gestion de stationnement intelligent. . . . .	21
2.1	Architecture en couches proposée par Yang et Xiong. . . . .	25
2.2	Le système de stationnement intelligent proposé par Cokro et al. . . . .	26
2.3	L'architecture à 4 niveaux pour l'analyse des données de l'IdO proposée par Cokro et al. . . . .	26
2.4	L'architecture du système proposé par Nguyen et al. . . . .	27
2.5	Architecture du système proposé par Jung et al. . . . .	28
2.6	Diagramme de flux du système proposé par Jung et al. . . . .	28
3.1	Architecture d'analyse de Big Data basée sur l'apprentissage profond pour le stationnement intelligent. . . . .	48
4.1	Architecture d'une cellule GRU . . . . .	54
4.2	Occupation horaire de deux parkings dans une période d'un mois . . . . .	60
4.3	Occupation par heure de 4 parkings . . . . .	61
4.4	Courbe d'occupation moyenne pendant une journée . . . . .	62
4.5	Occupation moyenne par jour de la semaine . . . . .	63
4.6	Matrice de corrélation entre les variables . . . . .	64
4.7	Nettoyage des données. . . . .	65
4.8	Ajout des données endogènes. . . . .	65
4.9	Ajout des données exogènes. . . . .	66
4.10	Normalisation des données. . . . .	66
4.11	Création des séquences . . . . .	67
4.12	Fractionnement des données pour les séries temporelles . . . . .	67

---

4.13 Fractionnement des données. . . . .	68
4.14 Redimensionnement des données. . . . .	69
4.15 Architecture du modèle conv1D-BiGRU. . . . .	71
4.16 Perte MSE du modèle conv1D-BiGRU . . . . .	73
4.17 Erreur absolue moyenne du modèle conv1D-BiGRU. . . . .	73
4.18 Comparaison des trois modèles en termes de métriques . . . . .	74
4.19 Graphe des prévisions du modèle GRU. . . . .	75
4.20 Graphe des prévisions du modèle BiGRU. . . . .	76
4.21 Graphe des prévisions du modèle conv1D-BiGRU. . . . .	77

# Liste des tableaux

2.1	Tableau comparatif des travaux basés sur la détection de l'occupation des places de stationnement . . . . .	31
2.2	Tableau comparatif des travaux basés sur la prévision de l'occupation des places de stationnement . . . . .	35
3.1	Tableau de comparaison entre les bases de données NoSQL . . . . .	44
3.2	Tableaux de comparaison entre Spark, Hadoop et Flink . . . . .	47
4.1	Les paramètres du modèle conv1D-BiGRU. . . . .	72
4.2	Comparaison des résultats . . . . .	73



# Liste des acronymes

<i>API</i>	Application Programming Interface
<i>ACC</i>	Accuracy
<i>BD</i>	Big Data
<i>BDD</i>	Base De Données
<i>BP</i>	Bach Propagation
<i>CAGR</i>	Compound Annual Growth Rate
<i>CEO</i>	Chief Executive Officer
<i>CNN</i>	Convolutional Neural Network
<i>CQL</i>	Cassandra Query Language
<i>DAG</i>	Directed Acyclic Graph
<i>DL</i>	Deep Learning
<i>DNN</i>	Deep Neural Network
<i>ETL</i>	Extraction, Transformation, Chargement
<i>FMR</i>	Fixed Model Reuse
<i>FPS</i>	Frames Per Second
<i>GCNN</i>	Graph Convolutional Neural Network
<i>GRU</i>	Gated Recurrent Unit
<i>GPU</i>	Graphics Processing Unit
<i>HDFS</i>	Hadoop Distributed File System
<i>HPC</i>	High Performance Computing
<i>HTTP</i>	HyperText Transfer Protocol
<i>IA</i>	Intelligence Artificielle
<i>IBM</i>	International Business Machines Corporation
<i>ICT</i>	Information and Communication Technologies
<i>IEEE</i>	Institute of Electrical and Electronics Engineers.
<i>IdO</i>	Internet Des Objets
<i>IoT</i>	Internet of Things
<i>JSON</i>	JavaScript Object Notation
<i>LSTM</i>	Long Short Term Memory
<i>LTE</i>	Long Term Evolution
<i>MAE</i>	Mean Absolute Error
<i>MAPE</i>	Mean Absolute Percentage Error
<i>ML</i>	Machine Learning
<i>MQTT</i>	Message Queuing Telemetry Transport

---

<i>MSE</i>	Mean Squared Error
<i>M2M</i>	Machine to Machine
<i>NaN</i>	Not a Number
<i>NFC</i>	Near-Field Communication
<i>NoSQL</i>	No Structured Query Language
<i>OCDE</i>	L'Organisation de Coopération et de Développement Economiques
<i>PHP</i>	Hypertext Preprocessor
<i>RAM</i>	Random Access Memory
<i>RCNN</i>	Regional Convolutional Neural Network
<i>RDD</i>	Resilient Distributed Datasets
<i>REST</i>	REpresentational State Transfer
<i>RFID</i>	Radio Frequency Identification
<i>RMSE</i>	Root Mean Square Error
<i>SQL</i>	Structured Query Language
<i>STI</i>	Systèmes de transport intelligent
<i>SOA</i>	Service-Oriented Architecture
<i>SVR</i>	Support Vector Regression
<i>TIC</i>	Technologies de l'Information et de la Communication
<i>UN</i>	Nations Unies
<i>VGG</i>	Visual Geometry Group
<i>WiFi</i>	Wireless Fidelity
<i>XML</i>	Extensible Markup Language

# Liste de communication

Samira YESSAD, Souaad BOUSSOUFA-LAHLAH, Wissam FERROUDJ, Kenza DJEL-LAOUI et Louiza BOUALLOUCHE-MEDJKOUNE. "Architecture d'Analyse de Big Data Basée sur l'Apprentissage Profond pour le Stationnement Intelligent : Étude et Comparaison des Outils", Colloque international des objets et des systèmes connectés (COC'2023), Tunisie, Juin 2023. <https://coc23.sciencesconf.org/>



# Introduction générale

Les villes intelligentes ont émergé comme une solution prometteuse aux différents problèmes urbains dans le monde, offrant un environnement confortable et pratique pour les résidents attirant ainsi une population croissante. La planète compte actuellement 7,7 milliards d'habitants et les Nations Unies (U.N.) prévoient que la population humaine atteindra 9 milliards d'ici 2050 [24]. 54% de cette population vit dans des villes, cette proportion devrait atteindre 68% d'ici 2050 [102]. Cependant, avec cette augmentation rapide de la population, le nombre de véhicules dans ces villes ne cesse de croître, créant un besoin de stationnement accru. En conséquence, la recherche de places de stationnement est devenue un problème majeur pour les conducteurs, entraînant une augmentation de la congestion du trafic, de la pollution atmosphérique, de la perte de temps et de la frustration pour les conducteurs.

L'une des principales causes des embouteillages est la mauvaise gestion des ressources en matière de stationnement. Aux heures de pointe dans la plupart des grandes villes, le trafic généré par les voitures à la recherche d'une place de parking représente jusqu'à 40% du trafic total [53]. Les conducteurs parcourent en moyenne une distance de seulement 0,5 mile (environ 800 mètres) à une vitesse de 10 miles/h (environ 16 km/h) lorsqu'ils cherchent une place de stationnement, ce qui leur prend en moyenne 15 minutes [89].

Face à cette situation, un système de stationnement intelligent utilisant les technologies de l'Internet des Objets (IdO) et du Big Data (BD) offre une solution innovante à ce problème en proposant une architecture technique solide pour la gestion des données de stationnement. Ce système dirige les conducteurs vers des places de stationnement disponibles prédites dans des délais réduits grâce à des algorithmes basés sur l'apprentissage profond. C'est dans cette direction que s'insère le présent travail qui a comme objectif la conception d'un système de stationnement intelligent en utilisant les technologies IoT et Big Data avec deux contributions principales.

La première consiste à proposer une architecture générale pour gérer les données massives

provenant des infrastructures IdO installées au sein des parcs de stationnement. La deuxième contribution concerne l'implémentation d'un modèle d'apprentissage profond pour prédire l'occupation des places de stationnement.

Pour cela, ce document est composé de quatre chapitres. Le premier a pour but d'introduire des généralités sur le fonctionnement et les différentes technologies des systèmes de stationnement intelligents dans le contexte des villes intelligentes. Le deuxième chapitre est un état de l'art des travaux récents portant sur le stationnement intelligent. Le troisième chapitre, quant à lui, vise à expliquer la nouvelle architecture proposée pour la gestion des données de stationnement tout en commençant par une étude des différentes technologies et outils de stockage et de traitement des données.

Le dernier chapitre présente les détails de la mise en œuvre du modèle d'apprentissage profond proposé, ainsi que les résultats obtenus.

# Généralités sur le stationnement intelligent

## 1.1 Introduction

Notre projet porte sur le stationnement intelligent, qui est un domaine d'application clés des villes intelligentes. Pour bien comprendre le contexte de notre projet, ce premier chapitre établira les fondements nécessaires pour une compréhension approfondie des concepts essentiels liés aux villes intelligentes en général, puis plus spécifiquement ceux liés au stationnement intelligent. Par la suite, nous examinerons en détail la définition et le fonctionnement de ce dernier, en mettant l'accent sur les technologies fondamentales qui le soutiennent, notamment l'Internet des Objets (IdO) et le Big Data (BD). A la fin, nous soulignerons les avantages de la mise en place des systèmes de stationnement intelligents dans le contexte du développement urbain, ainsi que son rôle sur le marché mondial.

## 1.2 Les villes intelligentes

Le concept des villes intelligentes, également connu sous le terme anglais "smart city" d'origine américain [82], a émergé selon certaines études [91] dans les années 1990 dans le cadre du mouvement de croissance intelligente aux États-Unis [39]. Ce mouvement vise à promouvoir un développement urbain durable et équilibré, en mettant l'accent sur l'utilisation efficace des ressources. Avec l'avènement du numérique, les avancées technologiques telles que l'internet et les Technologies de l'Information et de la Communication (TIC) ont offert de nouvelles possibilités pour améliorer la qualité de vie dans les zones urbaines.

Bien que des travaux antérieurs aient posé les bases du concept de villes intelligentes, le premier ouvrage influant sur ce thème n'a été publié qu'en 1994 par le professeur William J.

Mitchell, intitulé "City of Bits : Space, Place, and the Infobahn" [68], dans lequel il décrit une nouvelle forme de ville qui intègre des espaces virtuels inter-connectés par les autoroutes de l'information dans le contexte de la miniaturisation et de la révolution des télécommunications numériques continues.

Les travaux sur les villes intelligentes ont pris de l'ampleur dans les années 2000, avec l'intérêt grandissant des entreprises telles que l'entreprise International Business Machines Corporation (IBM) et Cisco, ainsi que de la Commission européenne [93]. En 2004, la ville de Songdo en Corée du Sud est devenue la première ville considérée comme véritablement intelligente, marquant le début d'un mouvement mondial vers le développement de villes intelligentes [110]. Depuis lors, de nombreuses autres villes ont lancé des projets et des initiatives pour devenir plus intelligentes, et le développement des villes intelligentes est un phénomène en constante évolution à l'échelle mondiale.

### 1.2.1 Définition

Il n'y a pas eu de définition unique ou universelle de ce qu'est une ville intelligente, car cela dépend des priorités et des besoins spécifiques de chaque ville. De ce fait, on trouve plusieurs définitions dans la littérature, parmi lesquelles nous citons :

"La ville intelligente est une ville instrumentée, interconnectée et intelligente qui utilise les TIC pour détecter, analyser et intégrer des informations critiques sur les systèmes centraux des villes " [46].

"Les villes intelligentes sont des territoires dotés d'une grande capacité d'apprentissage et d'innovation, qui repose sur la créativité de leur population, leurs institutions de création de connaissances et leur infrastructure numérique de communication et de gestion des connaissances" [56].

"Une ville intelligente est une entité locale - un district, une ville, une région ou un petit pays - qui adopte une approche holistique de l'utilisation des technologies de l'information avec une analyse en temps réel qui encourage le développement économique durable" [94].

Une ville est intelligente lorsque les investissements dans l'infrastructure traditionnelle, le développement social et l'infrastructure moderne de communication (TIC) modernes alimentent une croissance durable et une qualité de vie élevée, avec une gestion judicieuse

des ressources naturelles [109].

En somme, toutes les définitions précédentes s'accordent sur le fait que les villes intelligentes sont apparues en réponse aux défis posés par l'urbanisation [97], tels que la surpopulation, l'épuisement des ressources et le réchauffement climatique. Une ville intelligente, ou smart city, est un environnement connecté qui vise à optimiser l'utilisation des ressources publiques, réduire les coûts de fonctionnement des administrations publiques et améliorer la qualité de vie de ses citoyens [46]. Cela est rendu possible grâce à l'intégration des TIC [?] [63] dans différents secteurs de la ville. Le but ultime est d'offrir aux citoyens un environnement de vie confortable et agréable [94].

## 1.2.2 Domaines des villes intelligentes

Ces villes se distinguent par six caractéristiques principales, à savoir la gouvernance intelligente, l'économie intelligente, la mobilité intelligente, l'environnement intelligent, le citoyen intelligent et l'habitat intelligent [77] [93] [55]. Ces caractéristiques sont inter-connectées et complémentaires, et elles doivent être prises en compte pour créer une ville intelligente et promouvoir une interaction harmonieuse entre les citoyens et leur environnement [55].

### 1.2.2.1 Gouvernance intelligente

La gouvernance intelligente implique la prise en compte de la participation politique, des installations civiles et de l'ouverture administrative [93], qui vise à utiliser des technologies numériques pour améliorer la prise de décision, la transparence et la participation citoyenne d'un côté et permet également de connaître les droits et les barrières de l'autre côté [71].

### 1.2.2.2 Économie intelligente

L'économie intelligente facilite le développement du commerce électronique et de nouvelles opportunités pour les entrepreneurs [83]. Elle englobe les différents facteurs liés à la compétitivité économique d'une ville, notamment son importance économique sur les marchés nationaux et internationaux, la promotion de l'entrepreneuriat, la flexibilité dans les méthodes de travail et de production, l'innovation et l'esprit d'entrepreneuriat [93] [99].

### 1.2.2.3 Environnement intelligent

Cette notion recouvre plusieurs aspects de l'environnement urbain, notamment la pollution, les situations habituelles et climatiques, la contamination, l'organisation des magasins et la protection de l'environnement (tels que la qualité de l'air, la qualité de l'eau, la gestion des déchets, la consommation d'énergie et les émissions de gaz à effet de serre) qui sont de plus en plus importants [22]. Les zones urbaines s'engagent à minimiser l'empreinte de l'écosystème et à utiliser des énergies propres (solaire, éolienne, etc.) et une gestion durable des ressources [93].

### 1.2.2.4 Citoyen intelligent

Le concept de "citoyen intelligent" est le résultat de plusieurs attributs liés à l'expansion des investissements sociaux [42]. Parmi ceux-ci, on peut citer la créativité, la flexibilité, le niveau de qualification, l'ouverture d'esprit, la capacité à apprendre tout au long de la vie, la diversité culturelle et éthique, ainsi que la participation active à la vie publique [91].

### 1.2.2.5 Habitat intelligent

Ce concept se concentre sur l'amélioration de la qualité de vie au sein de la ville intelligente [57]. Pour y parvenir, sept éléments sont pris en compte, à savoir les lieux culturels, les conditions de santé, la sécurité individuelle, la qualité du logement, les institutions éducatives, l'attractivité touristique et la cohésion sociale [55] [57]. Ces éléments sont considérés comme des indicateurs clés pour évaluer le niveau de développement et de progrès de la ville intelligente en terme de qualité de vie.

### 1.2.2.6 Mobilité intelligente

Elle inclut des solutions pour le transport en commun, le covoiturage, les véhicules autonomes et les modes de transport doux tels que le vélo et la marche. Elle permet notamment la gestion intelligente du trafic et du stationnement de manière efficace et intelligente [31]. Elle intègre les TIC pour soutenir l'optimisation des flux de circulation des véhicules, mais aussi pour recueillir les opinions des citoyens sur l'habitabilité des villes ou la qualité des services de transport public locaux [115].

Les six caractéristiques précédentes représentent les caractéristiques clés et principales pour une ville intelligente auxquelles s'accordent toutes les recherches et les organisations,

cependant, il y a d'autres qui considèrent que la cybersécurité, l'interconnexion et la sécurité sont aussi des points fondamentaux à ne pas négliger [77].

## 1.3 Stationnement intelligent

### 1.3.1 Définition

Le stationnement intelligent, dit "smart parking" en anglais, est une des solutions de ville intelligente les plus adoptées à travers le monde. Cette technologie moderne permet d'automatiser et de réduire le temps de recherche d'une place de stationnement, de faciliter la circulation et d'offrir une sécurité routière. Certaines solutions complètent cette technologie avec d'autres services tels que la rapidité de paiement et de réservation en ligne.

Plusieurs entités telles que les aéroports, les universités, les centres commerciaux et les garages municipaux ont commencé à réaliser les avantages significatifs de la technologie de stationnement automatisé [85].

Le principe consiste à équiper le parking avec divers types de capteurs intelligents capables de détecter les places vides en temps réel. Les informations recueillies à travers ces capteurs et à l'aide de l'IdO seront traitées et analysées selon le type de données collectées [95].

### 1.3.2 Fonctionnement de stationnement intelligent

Un système de gestion de stationnement vise à surveiller, contrôler et optimiser l'utilisation des places de stationnement. Voici un aperçu général de son fonctionnement

#### 1.3.2.1 Détection et Collecte des données

Des capteurs IdO sont utilisés pour détecter l'occupation des places de stationnement. Ces capteurs peuvent prendre la forme de capteurs de présence intégrés au sol, de caméras de surveillance ou de capteurs de mouvement. Les données collectées par les capteurs sont transmises à un système central de gestion de stationnement. Ces données comprennent des informations sur l'emplacement, l'heure d'occupation et d'autres détails pertinents.

#### 1.3.2.2 Traitement et analyse des données

Les données collectées sont traitées à l'aide des techniques de BD. Les données traitées sont utilisées pour analyser les modèles de stationnement, les comportements des utilisateurs

et les tendances de demande. Les algorithmes d'apprentissage automatique et d'Intelligence Artificielle (IA) sont appliqués pour comprendre et prédire la disponibilité future des places de stationnement, ainsi que pour anticiper les heures de pointe et les périodes de forte demande.

### 1.3.2.3 Affichage des informations / Guidage vers les places disponibles

Les informations sur la disponibilité des places sont généralement affichées aux conducteurs à l'entrée des parkings ou via des panneaux d'affichage dynamiques installés dans les rues ou les zones de stationnement. Certains systèmes de gestion de stationnement intelligent en plus de l'affichage d'informations, proposent des systèmes de guidage qui dirigent les conducteurs vers les places de stationnement disponibles. Cela peut être fait à l'aide de panneaux de signalisation, d'applications mobiles ou de systèmes de navigation embarqués.

### 1.3.2.4 Gestion de la tarification

Dans les systèmes de stationnement payant, le système de gestion de stationnement facilite également le processus de paiement. Cela peut inclure des bornes de paiement automatiques, des applications mobiles ou des systèmes de paiement sans contact. De plus, le système de gestion de stationnement peut contrôler l'accès aux parkings à l'aide de barrières ou de portails automatiques.

## 1.4 La technologie IdO

### 1.4.1 Définition

L'Internet des Objets (IdO), également connu sous le nom d'Internet des choses ou "Internet of Things (IoT)" en anglais, est un système d'objets physiques interconnectés et qui peuvent être connectés à Internet et communiquer entre eux à l'aide de capteurs et d'autres technologies de communication. Le terme a été inventé par Kevin Ashton en 1999 pour décrire l'idée d'une interconnexion entre des objets de monde physique, qui étaient auparavant déconnectés du monde numérique, ainsi qu'avec les ordinateurs et les réseaux informatiques [85]. Avec l'IdO, les objets peuvent collecter des données, échanger des informations et même prendre des décisions autonomes, sans intervention humaine directe [95]. Cette technologie a le potentiel de transformer notre façon de vivre et de travailler, en permettant aux entreprises et aux individus de mieux comprendre et de contrôler tout leur environnement physique.

L'IdO peut englober une large gamme d'objets physiques, allant des appareils électroniques





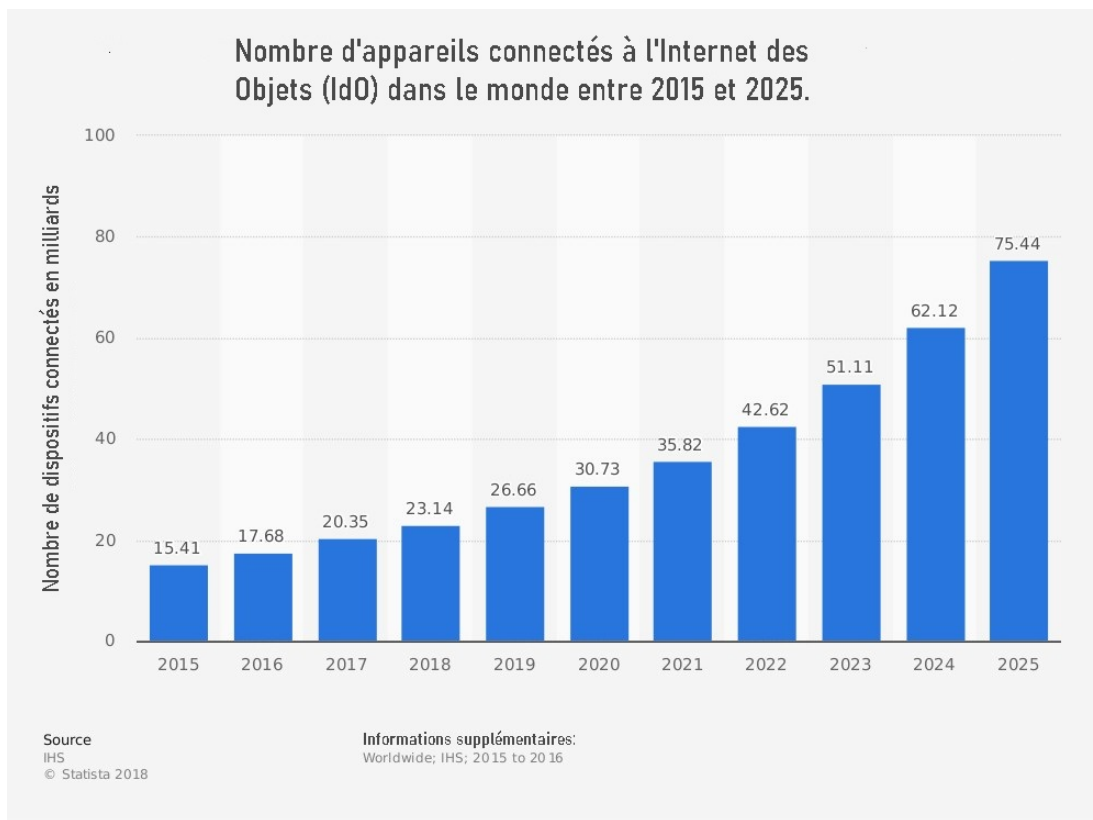


FIGURE 1.2 – Le nombre total d'appareils connectés à l'Internet des Objets (IdO) installés dans le monde entier, de 2015 à 2025, en milliards.

Cependant, la croissance de taux d'utilisation d'IdO au fil des années pose également un défi majeur pour la gestion des données collectées par les capteurs et leur utilisation efficace. Les données volumineuses, communément appelées "Big Data" (BD), nécessitent des outils et des techniques spécifiques pour leur traitement et leur analyse. En effet, sans une approche adéquate pour extraire des informations significatives à partir de ces données massives, leur utilité est limitée. Ainsi, la gestion des données volumineuses collectées par la ville intelligente est un enjeu crucial pour la réussite de son utilisation dans l'urbanisation des villes, et le BD est la clé pour y parvenir.

## 1.5 Big data

### 1.5.1 Définition

Big Data (BD) est le terme qui décrit un ensemble de données non structurées et complexes provenant de sources hétérogènes (tels que la prolifération des appareils connectés, la popularité des réseaux sociaux, les transactions en ligne, les données de localisation, le web (données de navigation, publicitaires, les sites d'e-commerce) sous une diversité de format (texte, image, vidéo, audio, ...) et nécessitent des technologies puissantes pour stocker, gérer, analyser et visualiser l'information. En d'autre terme le BD représente les ensembles d'informations caractérisés par un volume, une vitesse et une variété si élevés qu'ils nécessitent une technologie et des méthodes analytiques spécifiques pour leur transformation en valeur.

### 1.5.2 Caractéristiques du Big data

Les caractéristiques du BD sont générées par une multitude de facteurs et de sources. La notion des 3Vs est une référence aux grandes caractéristiques qui ont marqué les données sur lesquelles repose le phénomène du BD depuis son apparition, à savoir le volume, la variété et la vitesse. Cependant, avec les progrès réalisés dans les solutions de stockage et de traitement, ces trois défis ne représentent plus les seuls piliers du BD. D'autres aspects, traduits par d'autres "Vs", ont émergé avec l'application du BD dans de nombreux domaines, faisant passer le phénomène de 3Vs [51] [47] à 5Vs[44], ensuite, à 7Vs [59] [60], voire même 10Vs [54]. Une liste des Vs les plus souvent employés est donnée dans ce qui suit [72].

#### 1.5.2.1 Volume

De grands volumes de données numériques sont générés en continu à partir des millions d'application et d'appareils connectés.

#### 1.5.2.2 Variété

La variété fait référence à la diversité des types de données générées dans laquelle on peut distinguer la variété des sources et la variété du contenu.

**Variété des sources :** Les données sont de différentes sources qui peuvent être des applications de gestion internes ou externes à une entreprise ou également des données de

différents capteurs, ou encore des données comportementales client comme les réseaux sociaux et la géolocalisation.

**Variété du contenu :** Comme dit précédemment les données sont générées de différentes sources disparates ce qui explique la différence entre les formats de ces dernières, elles peuvent être des données structurées (BDD, XML), semi structurées (pages web, emails, commentaires), non structurées (image, vidéo, audio).

### 1.5.2.3 Vitesse

Également appelé vitesse, désigne la rapidité avec laquelle les données sont arrivées, et doivent être stockées et traitées en temps réel [54]. Pour illustrer cette rapidité, il est estimé qu'en une seule minute 500 heures de contenus sont mises en ligne sur YouTube, 69 millions de messages sont envoyés sur Whatsapp et Facebook Messenger, et plus d'un 1,3 million d'euros sont dépensés en ligne [10].

### 1.5.2.4 Vérité

La vérité fait référence à la qualité et à l'exactitude des données collectées [54], qui peuvent être impactées par leur origine diverse. Les données peuvent contenir des éléments manquants, des ambiguïtés, des erreurs ou même des informations non fiables. La vérité est essentielle pour garantir un niveau élevé de confiance dans les données utilisées pour prendre des décisions. Il est donc crucial d'éliminer toute donnée incomplète ou invalide, car une grande quantité de telles informations peut entraîner des confusions. Par exemple, si les informations sur les médicaments qu'un patient prend sont modifiées ou incomplètes, cela peut mettre sa vie en danger .

### 1.5.2.5 Valeur

L'avènement du monde numérique a rendu l'accès aux données aussi rapide et direct que la réflexion sur la fiabilité et la valeur de l'information à disposition [26]. La valeur est considérée comme le plus important des "Vs" en matière de BD [54], car elle implique la nécessité de déterminer la véritable utilité des bonnes données et de les adapter aux besoins spécifiques de l'entreprise [52].

### 1.5.2.6 Validité

Bien que la validité soit souvent confondue à la véracité des données car elles semblent similaires en matière d'exactitude, ces deux notions sont en réalité distinctes. En effet, les données sont valides lorsqu'elles passent du stade exploratoire à celui d'action. Cette caractéristique du BD est vitale pour détecter des relations cachées entre les éléments au sein de sources massives de données [54].

### 1.5.2.7 Volatilité

La volatilité des BD se réfère à la période de validité des données et à la durée pendant laquelle elles doivent être stockées [54]. Cette caractéristique du BD est importante car certaines données peuvent devenir obsolètes rapidement, tandis que d'autres peuvent être utiles pendant une période plus longue.

### 1.5.2.8 Variabilité

La variabilité fait référence aux incohérences du flux de données, qui peuvent être difficiles à gérer, en particulier avec l'augmentation de l'utilisation des médias sociaux [49].

### 1.5.2.9 Viabilité

La viabilité du BD fait référence à la capacité des données de maintenir leur qualité, leur intégrité, leur accessibilité et leur disponibilité sur une période prolongée, tout en étant capables de se développer et de produire davantage de données en cas de besoin [54].

### 1.5.2.10 Viscosité

La viscosité dans le contexte du BD correspond à la difficulté de modifier ou de supprimer des données, car elles sont souvent intégrées à des systèmes complexes ou sont importantes pour les processus métiers. Plus précisément, la viscosité est liée au degré de corrélation et d'interdépendance entre les différentes structures de données. Étant donné que les données sont souvent collectées à partir de multiples sources qui peuvent être hétérogènes, cela peut rendre la gestion des données plus difficile [54].

### 1.5.3 Principaux enjeux du Big data

Avec l'avènement du BD dans divers domaines, de nombreux défis émergent chaque jour. Ces défis concernent les différentes étapes d'acquisition, de nettoyage, d'analyse, et d'interprétation des données [70] [74].

#### 1.5.3.1 Acquisition des données

Il s'agit de la collecte des données à partir de différentes sources. L'acquisition des données peut poser des problèmes tels que la qualité des données, la variété des sources, la confidentialité des données, etc [21] [87].

#### 1.5.3.2 Nettoyage des données

Il est important de s'assurer que les données collectées sont fiables et précises. Le nettoyage des données consiste à éliminer les données inutiles ou erronées, à corriger les erreurs et à normaliser les données pour faciliter leur traitement [21].

#### 1.5.3.3 Intégration et agrégation des données

Les données collectées proviennent souvent de différentes sources et sont stockées dans des formats différents. L'intégration et l'agrégation des données consistent à les rassembler et à les structurer de manière à faciliter leur analyse [87].

#### 1.5.3.4 Analyse des données

L'analyse des données est un processus clé pour extraire des informations utiles à partir des données. Pour y parvenir, il est essentiel de sélectionner les bonnes techniques telles que la fouille de données, l'apprentissage automatique ou la visualisation de données, mais également les modèles adéquats en fonction des données et de l'objectif d'analyse. En effet, le choix de ces méthodes et modèles est crucial pour obtenir des résultats pertinents et exploitables pour la prise de décision [70] [87].

#### 1.5.3.5 Interprétation des résultats

Une fois les données analysées, il est important de les interpréter correctement, de faire des hypothèses pour en tirer des conclusions significatives exploitables [25] [70]. Cela nécessite une compréhension approfondie des données.

## 1.6 Machine learning

Le Machine Learning (ML) ou apprentissage automatique en français est une branche de l'intelligence artificielle qui permet aux ordinateurs d'apprendre à partir de données, sans qu'ils aient besoin d'être programmés explicitement. Les algorithmes de ML utilisent des modèles mathématiques pour analyser les données, détecter des tendances et des modèles, et prendre des décisions en fonction de ces données. Dans le contexte du stationnement intelligent, le ML est utilisé pour améliorer la détection des places de stationnement disponibles et prédire les niveaux futurs d'occupation du parking. Les algorithmes de ML peuvent également être utilisés pour optimiser la tarification des places de stationnement en fonction des heures de pointe et de la demande.

### 1.6.1 Notions liées à l'apprentissage automatique

#### 1.6.1.1 Préparation des données

Est une étape cruciale dans tout projet d'apprentissage automatique. Elle comprend généralement deux étapes principales : le nettoyage des données et la normalisation des données.

**Le nettoyage des données** implique la suppression de colonnes inutiles ou redondantes, ainsi que des valeurs aberrantes ou manquantes (NaN signifie Not A Number). Les colonnes inutiles peuvent être des attributs qui ne sont pas pertinents pour la tâche d'apprentissage automatique que l'on souhaite réaliser. Les valeurs aberrantes sont des observations qui sont très différentes des autres observations et peuvent fausser les résultats de notre modèle.

**La normalisation des données** est une étape importante pour s'assurer que toutes les caractéristiques (ou variables) ont la même échelle. Cela évite que certaines caractéristiques ne dominent les autres en raison de leurs valeurs plus grandes. Deux méthodes couramment utilisées pour la normalisation sont la méthode Min-Max et la standardisation Z-score. La méthode Min-Max redimensionne les valeurs de sorte qu'elles se situent toutes entre un certain intervalle (par exemple, entre 0 et 1). La méthode Z-score calcule la moyenne et l'écart type des valeurs, puis les ajuste pour obtenir une distribution centrée sur zéro avec une variance unitaire.

### 1.6.1.2 Sous-ajustement / sur-ajustement

Le sous-ajustement (underfitting) et le sur-ajustement (overfitting) sont deux phénomènes courants dans l'apprentissage automatique.

Le sous-ajustement se produit lorsque le modèle ne parvient pas à capturer les modèles sous-jacents des données d'entraînement. Cela se traduit par une faible performance du modèle tant sur les données d'entraînement que sur les données de test.

D'autre part, le sur-ajustement se produit lorsque le modèle s'adapte trop étroitement aux données d'entraînement. Il mémorise essentiellement les exemples d'entraînement spécifiques au lieu d'apprendre les schémas généraux. Cela se traduit par une performance élevée sur les données d'entraînement, mais une performance médiocre sur les données de test ou de nouvelles données. Le modèle sur-ajusté peut être trop complexe et trop spécialisé pour les données d'entraînement, ce qui entraîne une perte de capacité de généralisation [105].

### 1.6.1.3 Fractionnement des données

Une fois que les données sont préparées, il est courant de diviser l'ensemble de données en trois ensembles distincts : l'ensemble d'entraînement (train set), l'ensemble de test (test set) et l'ensemble de validation (validation set).

**L'ensemble d'entraînement** est utilisé pour former notre modèle. C'est là que notre algorithme apprend les schémas et les relations entre les caractéristiques et les résultats attendus.

**L'ensemble de test** est utilisé pour évaluer les performances du modèle. Il est essentiel de séparer cet ensemble de l'ensemble d'entraînement, car il permet de vérifier comment le modèle se comporte sur des données qu'il n'a jamais vues auparavant. Cela aide à détecter les problèmes de sur-ajustement (overfitting), où le modèle mémorise les données d'entraînement mais ne généralise pas bien sur de nouvelles données.

**L'ensemble de validation** est utilisé pour ajuster les hyperparamètres du modèle. Les hyperparamètres sont des paramètres qui ne sont pas appris par le modèle lui-même, mais qui influencent ses performances. En utilisant l'ensemble de validation, on peut essayer différentes combinaisons d'hyperparamètres et choisir celle qui donne les meilleures performances.

Une répartition courante est de diviser les données en environ 70-80% pour l'ensemble



d'entraînement, 10-15% pour l'ensemble de test et 10-15% pour l'ensemble de validation. Cette répartition équilibrée permet d'avoir suffisamment de données pour l'entraînement, l'évaluation et l'ajustement des hyperparamètres.

#### 1.6.1.4 Entraînement du modèle

L'étape d'entraînement consiste à ajuster les paramètres du modèle en utilisant l'ensemble d'entraînement. Le modèle apprend à partir des exemples d'entraînement en minimisant une fonction de perte (loss function) qui mesure l'écart entre les prévisions du modèle et les valeurs réelles des données d'entraînement. L'algorithme d'optimisation est utilisé pour ajuster les poids et les biais du modèle afin de réduire cette perte.

#### 1.6.1.5 Évaluation du modèle

Une fois que le modèle est entraîné, il est évalué sur l'ensemble de test pour estimer ses performances sur de nouvelles données. Des métriques d'évaluation appropriées sont utilisées pour mesurer l'exactitude, la précision, le rappel, l'erreur quadratique ou toute autre mesure pertinente selon le problème. Cela permet de comprendre comment le modèle se comporte en généralisation sur des données qu'il n'a pas vu auparavant.

### 1.6.2 Algorithmes d'apprentissage automatique appliqués dans le stationnement intelligent

Cette section abordera quelques-uns des algorithmes les plus couramment utilisés dans le domaine du stationnement intelligent [19] [112]. Cependant, pour la suite de ce mémoire et une compréhension approfondie il est nécessaire d'accorder une attention particulière aux algorithmes basés sur les réseaux de neurones. Ces algorithmes d'apprentissage automatique sont utilisés en fonction des caractéristiques des données (comme l'occupation des places de stationnement, Localisation et date et heure) et des objectifs spécifiques du système de stationnement intelligent (Prévision de la disponibilité future....). Ils permettent d'optimiser la gestion des places de stationnement, de prédire la disponibilité future et d'améliorer l'expérience des conducteurs.

### 1.6.2.1 Machines à vecteurs de support

Machines à vecteurs de support ou bien Support Vector Machine (SVM) en anglais sont des algorithmes de classification utilisés pour séparer les données d'occupation des places de stationnement en différentes catégories en fonction de critères spécifiques.

Son architecture est basée sur la recherche d'un hyperplan optimal qui sépare de manière optimale les différentes classes de données. Les SVM cherchent à trouver l'hyper plan qui maximise la marge entre les points de données les plus proches de chaque classe, permettant ainsi une meilleure généralisation aux nouvelles instances.

Les SVM utilisent également une approche dite à marge d'erreur, qui consiste à tolérer certaines erreurs de classification afin d'obtenir un modèle plus robuste et moins sensible aux données aberrantes.

### 1.6.2.2 Régression par vecteurs de support

Régression par Vecteurs de Vupport (SVR) est un algorithme utilisé pour modéliser et prédire la demande de places de stationnement en fonction de variables telles que l'heure de la journée, le jour de la semaine, etc.

L'architecture des SVR cherche à minimiser l'écart entre les valeurs prédites et les valeurs réelles, tout en tolérant une marge d'erreur définie par un paramètre de régularisation.

Les SVR utilisent une approche basée sur les vecteurs de support pour identifier les exemples les plus pertinents et construire un modèle de régression robuste. Ils permettent également de gérer des données aberrantes en introduisant des paramètres de régularisation appropriés.

### 1.6.2.3 Réseaux de neurones convolutifs (CNN)

Les CNN pour Convolutional Neural Network sont une catégorie des Réseaux de neurones artificiels (Artificial Neural Network ANN) spécialement conçus pour le traitement des données structurées en grille, comme les images, les vidéos ou les données audio.

La particularité des CNN réside dans l'utilisation des couches de convolution et de pooling qui permettent de capturer des caractéristiques locales dans les données d'entrée. Ces caractéristiques sont ensuite utilisées par des couches entièrement connectées pour effectuer des classifications ou des prévisions. Plusieurs variantes des CNN existe AlexNet, Visual Geometry Group (VGG), ...etc.

Les CNN sont particulièrement efficaces dans les tâches de vision par ordinateur, où la détection de motifs spatiaux. Ils sont couramment utilisés pour l'analyse d'images dans le contexte du stationnement intelligent [66]. Ils permettent la détection et la reconnaissance des véhicules, ainsi que l'analyse des images de surveillance.

#### 1.6.2.4 Réseaux de neurones récurrents (RNN)

Les réseaux de neurones récurrents (RNN) sont une catégorie spécifique de ANN qui sont conçus pour traiter des données séquentielles ou temporelles.

Contrairement aux réseaux de neurones classiques où l'information se propage de manière linéaire de la couche d'entrée à la couche de sortie, les RNN ont des connexions récurrentes qui leur permettent de conserver une mémoire pour mémoriser des informations sur les séquences précédentes et d'influencer les sorties futures. La figure 1.3 illustre bien la différence entre un neurone artificiel classique et un neurone récurant.

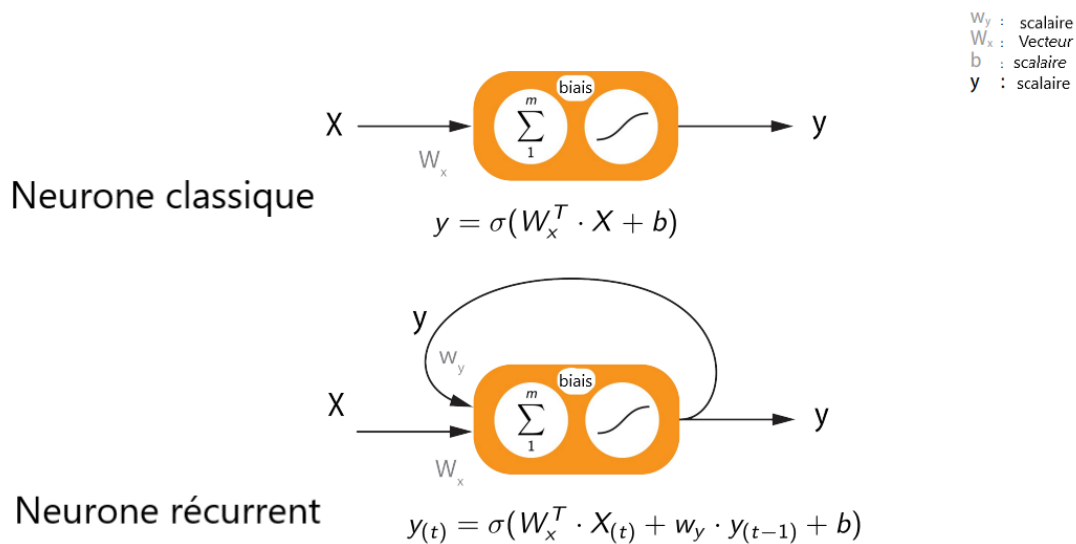


FIGURE 1.3 – Différence entre perception classique et un neurone récurant.

Ils sont utilisés dans des applications telles que la reconnaissance vocale, la traduction automatique et la prévision de séquences temporelles.

Cependant, les RNN peuvent également rencontrer des problèmes tels que le "vanishing gradient" (gradient qui tend à disparaître lors de la rétro-propagation) et le "exploding gradient" (gradient qui devient très grand). Pour surmonter ces problèmes, des variantes de RNN telles que les Long Short-Term Memory (LSTM) et les Gated Recurrent Unit (GRU)

ont été développées. Ces variantes utilisent des mécanismes de porte (gates) pour mieux contrôler le flux d'informations dans le réseau et résoudre les problèmes de gradient.

## 1.7 Avantages de la mise en place de stationnement intelligent

La mise en place d'un système de stationnement intelligent offre de nombreux avantages pour les conducteurs, les usagers de la route et la ville elle-même. Voici quelques-uns de ces avantages :

- Obtention d'informations précises en temps réel sur les places de stationnement occupées ou disponibles dans chaque parking.
- Réduction du temps de recherche d'un espace de stationnement libre.
- Amélioration de la sécurité du trafic pour les conducteurs et les usagers de la route.
- Réduction de la pollution et des émissions de gaz toxiques grâce à une meilleure gestion de l'espace de stationnement.
- Communication d'informations aux usagers avant, pendant et après le stationnement pour une expérience utilisateur améliorée.
- Facilité et commodité d'utilisation des places de stationnement.
- Augmentation de l'activité et de la mobilité dans la ville grâce aux technologies modernes.
- Génération de revenus significative grâce à une surveillance et une gestion améliorées et en temps réel de l'espace de stationnement disponible.
- Simplification de l'expérience de stationnement et ajout de valeur pour les intervenants du stationnement, tels que les conducteurs et les commerçants.
- Utilisation de données pour des décisions intelligentes, notamment à travers des applications d'état en temps réel et des rapports d'analyse historique.
- Réduire le nombre de véhicules stationnés illégalement sur le bord de la route, ce qui réduit les embouteillages et la circulation dans les zones où il n'y a pas assez de places de stationnement [33].
- Guidage et information des usagers et des touristes sur les places de stationnement disponibles et non occupées.

## 1.8 Marché de stationnement intelligent

Selon l'estimation des analystes de marché américain des Systèmes de Transport Intelligent (STI) [16], le marché mondial des systèmes de stationnement intelligents devrait connaître une croissance significative. On prévoit un taux de croissance annuel composé de rendement d'investissements (CAGR : Compound Annual Growth Rate) de 21,2 % entre 2022 et 2030. Et la valeur estimée devrait atteindre 30,16 milliards de dollars d'ici 2030, comme le montre la figure 1.4.

(Le CAGR de 21,2 % représente le taux de croissance moyen annuel sur la période de 2022 à 2030.)



FIGURE 1.4 – Le marché des systèmes de gestion de stationnement intelligent.

## 1.9 Conclusion

En conclusion, nous pouvons dire que les concepts présentés dans ce chapitre ont une importance considérable pour améliorer la qualité de vie dans les villes en permettant la collecte et l'analyse de données massives en temps réel. Nous avons particulièrement abordé le concept de stationnement intelligent dans ce chapitre, est l'un des domaines qui mérite

une attention particulière, et un exemple concret de l'utilisation des technologies IdO et BD dans les villes intelligentes. Dans le chapitre suivant, nous allons approfondir dans ce concept et examiner les travaux réalisés en utilisant les différentes technologies précédemment citées pour améliorer la gestion du stationnement.

# État de l'art sur le stationnement intelligent

## 2.1 Introduction

Nombreux sont les travaux qui cherchent à remédier aux problèmes de stationnement intelligent dans les villes modernes, et à augmenter l'efficacité de leurs systèmes. Dans ce contexte, plusieurs technologies comme l'IdO, le Big Data et les algorithmes d'apprentissage automatique sont de plus en plus utilisées pour améliorer la performance du stationnement intelligent.

Dans ce chapitre, nous allons nous intéresser aux travaux les plus récents dans ce domaine. Une synthèse de ces travaux sera mise en avant selon une classification définie, ensuite, nous allons examiner les différentes approches proposées dans un tableau comparatif en mettant l'accent sur les techniques d'apprentissage automatique dans ce domaine. Enfin, nous terminons ce chapitre par une discussion.

## 2.2 Objectifs de recherche dans le domaine du stationnement intelligent

Dans la littérature, plusieurs recherches ont été menées avec des propositions variées, mais elles visent généralement toutes les mêmes objectifs, comme par exemple, assurer une gestion efficace des emplacements de stationnement pour fournir aux conducteurs des informations précises d'une part et augmenter les bénéfices et les revenus d'autre part. Les différentes propositions peuvent être résumées comme suit :

- Concevoir un système de stationnement intelligent capable d'améliorer l'efficacité de l'utilisation des espaces de stationnement.

- Analyse des corrélations spatio-temporelles entre les parkings pour améliorer leur gestion.
- Détection de l'occupation des places de stationnement pour gérer le trafic routier.
- Détection des places de stationnement disponibles.
- Détection des véhicules présents dans les parkings.
- Détection d'objets pertinents pour le stationnement.
- Utilisation d'indicateurs pour évaluer l'occupation des places de stationnement.
- Regroupement temporel et détection des anomalies dans les parkings urbains.
- Prévion de la disponibilité des places de stationnement à court et long terme.

## 2.3 Synthèse des travaux connexes

Dans la littérature, divers travaux sur le stationnement sont proposés. Les recherches dans ce domaine sont abondantes et couvrent un large éventail de sujets. Dans cette section, nous allons présenter une synthèse de certains travaux récents que nous classifions tout d'abord en trois classes. La première classe concerne les approches basées sur le traitement des données massives, la deuxième c'est les approches basées sur la détection l'occupation des places de stationnement et finalement les approches basées sur la prévision de l'occupation .

### 2.3.1 Approches basées sur la gestion et le traitement des données massives

La gestion de données massives est un domaine de recherche en pleine expansion dans le développement de systèmes de stationnement intelligents. A cet égard, plusieurs architectures basées sur l'IdO et BD ont été proposées afin de mieux gérer les grandes quantités de données générées par les appareils de l'IdO utilisés dans le stationnement intelligents en utilisant des plateformes de traitement des mégas données et de stockage telles que Hadoop et Spark.

#### **Travail de Yang et Xiong [106]**

Dans l'article de Yang et Xiong, on a proposé l'utilisation de technologies de BD pour remplacer les méthodes traditionnelles de gestion des données pour résoudre les problèmes de stationnement.

Les auteurs proposent une conception et construction d'une plateforme à 4 couches. La couche de base pour le matériel, la couche de données pour l'acquisition, le pré-traitement et



le stockage des données, la couche système pour la gestion des différents modules, et enfin, la couche applicative qui comprend principalement des petits programmes WeChat et une application de terminal mobile. Afin de fournir aux gestionnaires du gouvernement un support de données efficace, ces données sont générées à partir de plusieurs logiciels et d'équipement de stationnement, cette plateforme permet d'avoir une supervision globale sur la gestion des parcs de stationnement, de demander des données, d'observer des fluctuations et les tendances des parkings.

Le système vise à améliorer considérablement l'efficacité de la gestion du stationnement et réduire le temps de recherche de places de stationnement. L'architecture en couches proposée est présentée dans la Figure 2.1.

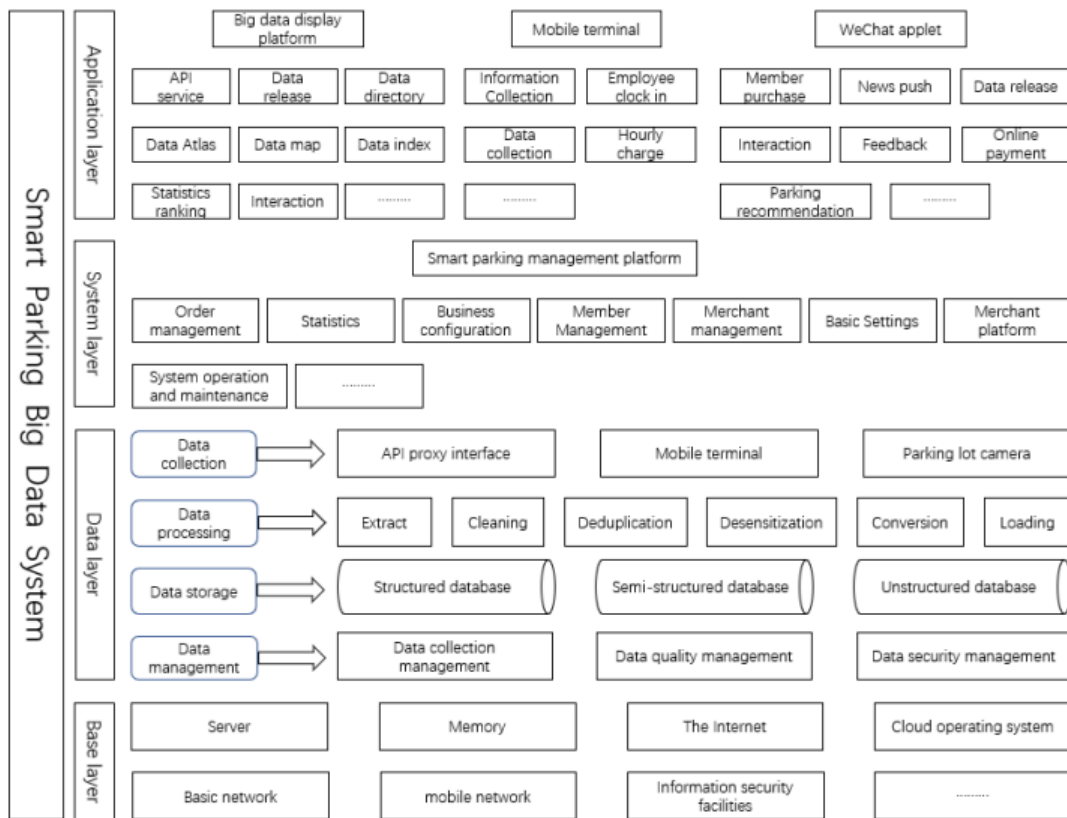


FIGURE 2.1 – Architecture en couches proposée par Yang et Xiong.

**Travail de Cokro et al. [38]**

Dans [38], les auteurs ont proposé la conception d'un système comme une solution pour le

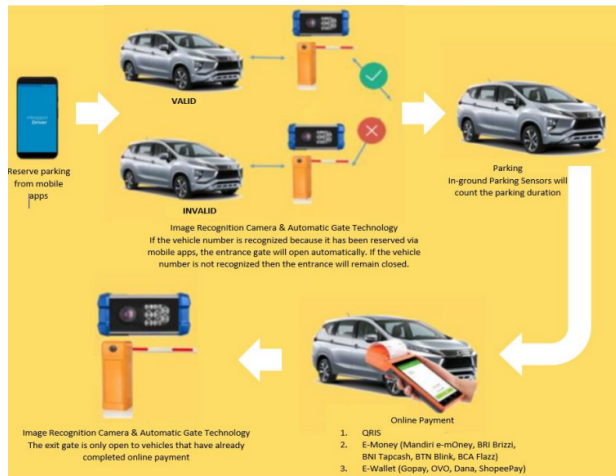


FIGURE 2.2 – Le système de stationnement intelligent proposé par Cokro et al.

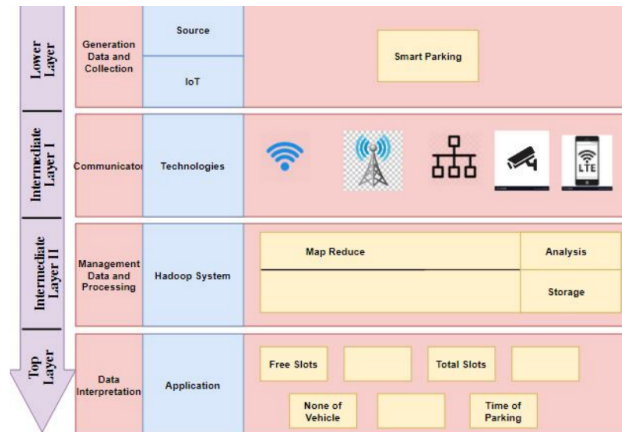


FIGURE 2.3 – L'architecture à 4 niveaux pour l'analyse des données de l'IdO proposée par Cokro et al.

gouvernement régional de DKI Jakarta, en particulier le département des transports de DKI Jakarta, pour atteindre l'objectif de redevance de de stationnement et maximiser les recettes qui ont été perdues au fil des ans. La Figure 2.2 montre le fonctionnement du système proposé qui utilise la technologie de reconnaissance d'image et les portails de parking automatiques.

La Figure 2.3 présente l'architecture du système proposé qui est composée de quatre couches. La plus basse est la couche de génération et de collecte de données, elle s'occupe de la collecte et de la combinaison des données perçues par les appareils de l'IdO ainsi que de leur filtrage préliminaire. La deuxième couche, qui est la première couche intermédiaire, est responsable de la communication des données. La troisième couche est la deuxième couche intermédiaire qui est la couche centrale de l'ensemble du système analytique et est responsable du traitement des données en utilisant des outils comme Hadoop Spark, Storm, VoltDb, HIVE, HBASE et SQL. La couche supérieure est celle d'interprétation des données qui représente l'application d'exploitation des données par l'utilisateur final.

### Travail de Nguyen et al. [76]

La solution proposée dans [76] est un système de surveillance de l'occupation des parkings autour d'une ville. Le système est basé sur le paradigme Fog/IoT qui effectue des analyses de données au niveau du brouillard, sans diffuser et stocker toutes les données de détection dans le cloud.

Un traitement et une analyse Big Data des données de la plateforme proposée sont effectués en utilisant Hadoop MapReduce pour fournir des informations précises aux conducteurs. Le

résultat du traitement des données permet de connaître rapidement les places vacantes dans le parking. La solution utilise le cloud pour stocker les résultats des données reçues de tous les nœuds Fog. Plus de détails sur l'architecture proposée sont illustrés dans la Figure 2.4.

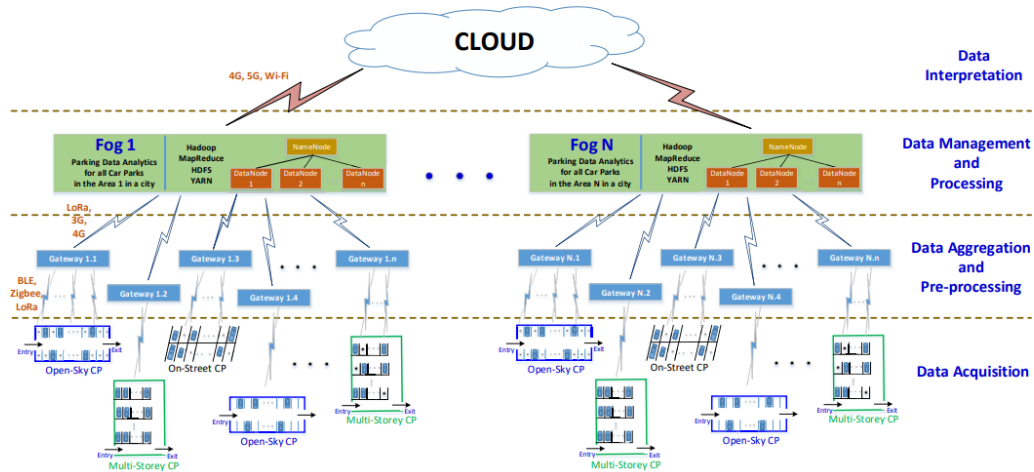


FIGURE 2.4 – L'architecture du système proposé par Nguyen et al.

### Travail de Jung et al. [48]

Dans [48], à des fins des statistiques, l'auteur a développé un système de gestion de stationnement intelligent basé sur des capteurs de stationnement à ultrasons et des balises Bluetooth. Les capteurs IdO à ultrasons sont contrôlés par une carte Arduino pour collecter les données de stationnement, qui sont ensuite transmises via le protocole MQTT (Message Queuing Telemetry Transport) au serveur en temps réel qui stocke les informations dans une base de données et fournit ainsi l'état de stationnement. Lors d'une réservation d'une place la balise Bluetooth à l'entrée du parking permet de reconnaître automatiquement la place de stationnement réservée et communiquer toutes les informations au serveur. Les Figures 2.5 et 2.6 montrent respectivement l'architecture et le diagramme de flux du système proposé dans ce travail.

## 2.3.2 Approches basées sur la détection de l'occupation des places de stationnement

### Travail de Radiuk et al. [84]

L'approche de Radiuk et al [84] est basée sur la technologie de vision par ordinateur et a été développée pour détecter les emplacements de stationnement vides et occupés dans les



FIGURE 2.5 – Architecture du système proposé par Jung et al.

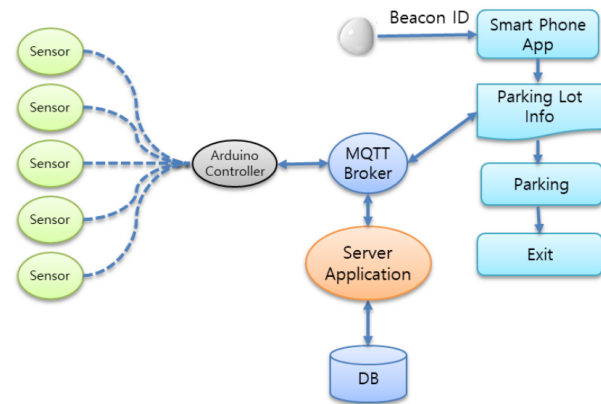


FIGURE 2.6 – Diagramme de flux du système proposé par Jung et al.

images de parking. Cette approche utilise un modèle de réseau neuronal convolucional qui a été entraîné sur un ensemble de données d'images de parking appelé KhNUParking à l'aide d'une descente de gradient stochastique avec un momentum de 0,8.

Les résultats obtenus ont été comparés avec d'autres architectures de réseaux neuronaux bien connues (AlexNet (), VGG-16 (), MobileNetV2() et GCV API ()) et ont montré une amélioration de la précision de la détection des emplacements de stationnement. Pour développer un système de stationnement intelligent, Google Cloud Vision a été utilisé comme détecteur de places de parking. Les performances de ce système ont été évaluées en utilisant un seul GPU et ont montré une précision de 85,4% pour la détection de 66 places de parking en environ 0,14 seconde, ce qui est considéré comme une performance décente et pratique.

### Travail de Mettupally et Menon [66]

Dans [66], Mettupally et Menon ont proposé une solution pour résoudre le problème des places de stationnement en utilisant la vision par ordinateur, l'analyse big data et le réseau de neurones. Leur solution s'appuie sur le développement d'un réseau de neurone convolutif régional et ceci en combinant deux types de CNN qui sont le Faster RCNN et le Mask RCNN pour obtenir un CNN hybride qu'ils ont appelé FMR CNN (Faster Mask RCNN), ils l'ont ensuite appliqué aux jeux de données PKlot et CoCo, et pour réduire le temps nécessaire à l'entraînement du modèle ils ont opté à l'utilisation de poids pré-entraînés chargés par le dataset COCO sur le modèle proposé FMR-CNN. Enfin, ils ont comparé les résultats de ce dernier avec ceux obtenus en appliquant un simple CNN à 4 couches et ont constaté que la précision moyenne du nouveau modèle FMR-CNN est de 92%, ce qui surpasse la précision moyenne de

détection de voiture du modèle CNN traditionnel qui a 88% de précision avec un écart de 4%.

### **Travail de Amarasooriya et al. [23]**

L'étude effectuée dans l'article intitulé " Implementation of Smart Parking System Using Image Processing" [23] consiste à traiter des séquences vidéo provenant d'une caméra à l'aide de l'algorithme de détection d'objets YOLO v3.

L'algorithme de détection d'objets YOLO v3 (You Only Look Once) v3 est une architecture de réseau de neurones convolutifs utilisée pour la détection d'objets en temps réel. La version trois (v3) s'agit d'une amélioration de versions précédentes de YOLO, qui se distingue par sa capacité à détecter efficacement les objets dans une image en une seule passe (d'où son nom "You Only Look Once"). Cette architecture permet de prédire les coordonnées de boîtes englobantes (bounding boxes) et les classes des objets présents dans une image. Contrairement à d'autres méthodes de détection d'objets qui utilisent des régions d'intérêt (Region Proposal Network), YOLO v3 divise l'image en une grille et prédit directement les boîtes englobantes et les classes pour chaque cellule de la grille.

L'objectif d'utiliser la variante YOLO dans cette étude est de détecter les coordonnées des parkings et des véhicules garés. Pour évaluer la performance du modèle, les chercheurs ont utilisé la base de données PKLot et ont effectué des tests dans différentes conditions météorologiques. Les résultats ont montré que le modèle proposé atteint une performance moyenne de 88,01%. Ils ont obtenu une meilleure performance par temps ensoleillé qui est de 91.25% et la performance la plus faible par temps pluvieux qui est de 84.28%.

### **Travail de Thakur et al. [100]**

Les chercheurs Thakur et al. [100] ont développé et évalué un classificateur pour la détection et la classification des places de parking en utilisant le dataset d'images PKLot dans différentes conditions météorologiques. Ils ont proposé deux modèles dans leur approche. Le premier modèle utilisait ResNet-50 en combinaison avec l'algorithme SVM (Support Vector Machine) et a atteint une précision de 98,9 %. Le deuxième modèle utilisait VGG16 en combinaison avec les fonctionnalités d'OpenCV, une bibliothèque open-source pour le traitement d'images et la vision par ordinateur, et a obtenu une précision de 93,4 %.

ResNet-50 est une architecture de réseau de neurones profonds qui résout le problème de la disparition du gradient en permettant aux informations de contourner certaines couches du réseau. Les caractéristiques extraites ont ensuite été utilisées par le classificateur SVM pour distinguer les emplacements vides des emplacements occupés.

VGG16 est une autre architecture de réseau de neurones convolutifs utilisée pour extraire des caractéristiques à partir d'images. Dans ce cas, il a été combiné avec les fonctionnalités d'OpenCV pour améliorer les performances de détection et de classification.

#### **Travail de Awaisi et al. [27]**

Les chercheurs dans [27] ont choisi d'étudier la vision par ordinateur dans la gestion des parkings dans un cadre basé sur le renforcement par apprentissage profond (DRL) pour les systèmes de stationnement intelligents compatibles avec l'Internet industriel des objets (IIo).

Il utilise le Q-learning pour identifier les places de stationnement vides en moins de temps et avec une plus grande précision. Pour cela, l'utilisation des caméras est requise. Un entraînement préalable des données recueillies à partir des caméras est effectué en construisant le DNN hors ligne, ensuite la phase d'apprentissage Q profond en ligne, où le DRL est utilisé pour détecter précisément les véhicules et les places de stationnement disponibles, et attribuer les places aux véhicules.

Les résultats de simulations montrent que l'utilisation de Deep Neural Network (DNN) offre une grande précision dans la détection des véhicules et des places de stationnement en les comparant aux modèles basés sur les CNN.

### **2.3.3 Comparatif des travaux basés sur la détection de l'occupation des places de stationnement**

Le Tableau 2.1 a pour objectif de synthétiser et analyser les articles étudiés dans la section précédente. Il mettra en évidence les méthodologies et résultats présentés par les auteurs, permettant ainsi une comparaison approfondie des travaux de détection.

Auteurs	Année	DataSet	Modèle	Résultat (ACC)
Mettupally et Menon [66]	2019	PKlot, CoCo	FMR CNN (Faster Mask RCNN)	92%
Radiuk et al. [84]	2022	dataset d'images de parking KhNUParking	CNN	85,4%
Awaisi et al. [27]	2023	Des données recueillies à partir de caméras	Renforcement par apprentissage profond (DRL) + Q-learning	élevée
Amarasooriya et al. [23]	2023	Dataset sur les places de parking -Image (PKlot)	YOLO v3	88.01%
Thakur et al. [100]	2023	Dataset sur les places de parking -Image (PKlot)	Resnet50 + SVM VGG16 + OpenCV	98,9% et 93.4 %

TABLEAU 2.1 – Tableau comparatif des travaux basés sur la détection de l'occupation des places de stationnement

### 2.3.4 Approches basées sur la prévision de l'occupation des places de stationnement

Il y a des travaux dans le domaine de stationnement intelligent qui se concentrent sur la prévision des espaces de stationnement vacants. Étant donné que cette tâche ne peut pas être réalisée efficacement en utilisant des CNN et leurs variantes, plusieurs études se sont tournées vers l'utilisation des RNN (Réseaux de Neurones Récurrents) et de leurs architectures dérivées telles que LSTM et GRU. Dans les sections suivantes, nous présentons quelques-uns de ces travaux.

#### Travail de Zheng et al. [114]

Dans l'étude publiée par Zheng et al. [114], trois méthodes d'apprentissage automatique pour prédire le taux d'occupation des parkings ont été comparé à savoir l'arbre de régression, le réseau neuronal et la régression de vecteur de support (SVR), sur deux ensembles de données provenant de Melbourne et de San Francisco (SFpark).

En se concentrant sur la combinaison de caractéristiques comprenant le nombre de places de stationnement occupées, les chercheurs ont constaté que la méthode de l'arbre de régression était la plus performante pour prédire le taux d'occupation des parkings, indépendamment du niveau d'occupation. Plus précisément, la valeur de l'erreur quadratique

moyenne (MSE) tendait vers 0.002 et un MAE = 0.032 lors de l'entraînement du modèle sur le dataset de Melbourne en utilisant la troisième combinaison de caractéristiques.

### **Travail de Xiangdong et al. [104]**

Les auteurs de l'article [104] proposent une prévision à court terme de l'espace de stationnement disponible en milieu urbain en utilisant un réseau neuronal LSTM. Ce choix est basé sur les caractéristiques des séries chronologiques des données de stationnement. Pour optimiser les paramètres du modèle et économiser du temps lors de l'entraînement, les auteurs introduisent un algorithme génétique dans leur approche.

Les données utilisées pour cette étude ont été collectées dans un parc de stationnement en Chine sur une période de 14 jours en 2018, avec des intervalles de mesures de 3 minutes. Pour comparer l'efficacité du modèle proposé, les auteurs ont également utilisé un réseau neuronal BP (Back Propagation) pour prédire l'espace de stationnement disponible en utilisant les mêmes paramètres mais sur des intervalles différents.

Les résultats montrent que les deux modèles donnent de bons résultats de prévision, avec une précision supérieure à 90%. Cependant, à mesure que le nombre d'intervalles augmente, la précision de prévision du réseau neuronal BP diminue considérablement. En revanche, le LSTM optimisé par algorithme génétique maintient une bonne précision de prévision, avec une valeur de MAPE de 6,82% pour un pas de temps.

### **Travail de Canli et Toklu [35]**

Une autre étude [35] propose une approche novatrice pour prédire le taux d'occupations à court (15–30 Min.) et moyen (1–2 Heures) terme. Les chercheurs ont utilisé des données météorologiques d'Istanbul ainsi que des données provenant de 607 parkings composés de 19 paramètres.

Ils ont développé un modèle basé sur les GRU, une variante récente des RNNs, et ont également évalué les performances des modèles LSTM et RNN. En appliquant ces trois techniques à différentes combinaisons de données d'apprentissage, les chercheurs ont obtenu 27 modèles. Ils ont comparé les résultats obtenus en utilisant des mesures telles MAE, MSE, RMSE, et la précision pour évaluer la capacité prédictive des modèles.

Le modèle GRU, avec 10 pas de temps et 70% de données d'apprentissage minimales, a montré les meilleures performances avec une précision de 0,91%, MAE=0.9, MSE=2.35 et RMSE=1.53. Cependant, les modèles basés sur les RNN ont présenté des temps d'appren-



tissage plus rapides que ceux basés sur GRU et LSTM.

#### **Travail de Zeng et al. [37]**

Zeng et al. [37] ont combiné la transformée en ondelettes (WT) avec une LSTM bidirectionnelle (Bi-LSTM) pour améliorer encore la précision de la prévision en utilisant une réduction du bruit de seuil. Ils ont comparé les résultats obtenus à partir de différents modèles Conv, LSTM, Conv+LSTM, Bi-LSTM, Conv+Bi-LSTM, DWT-Conv, DWT-LSTM, et DWT-Bi-LSTM. Les meilleures valeurs de RMSE = 7.42746, MSE = 55.16721 et MAE = 4.64591 ont été obtenues avec le modèle DWT-Bi-LSTM.

#### **Travail de Feng et al. [40]**

Dans [40], une comparaison était faite entre deux modèles LSTM et ConvLSTM-DN sur les données de 9 parkings publics de Santa Monica, California, USA. Et ceci est fait sur la base des métriques RMSE, MAE et MAPE avec des pas du temps de 5, 15, 30, 45 et 60 min.

Les résultats montrent que bien que le modèle de ConvLSTM-DCN présente une légère baisse de précision par rapport au modèle LSTM dans la prévision à 5 minutes, il surpasse ce dernier dans toutes les autres configurations de prévision, à savoir celles à 15, 30, 45 et 60 minutes. Plus précisément, l'approche dConvLSTM-DCN offre des avantages respectifs de 0,81%, 1,84%, 2,68 % et 4,81 % dans les configurations de prévision à 15, 30, 45 et 60 minutes.

#### **Travail de Yang et al. [108]**

Un modèle d'apprentissage profond Graph-CNN été adopté dans [108] pour prédire l'occupation au niveau des blocs de stationnement 30 minutes à l'avance. Le modèle prend des données de circulation routière multi-sources comme entrée, par exemple, le stationnement, la circulation et les conditions météorologiques. La performance du modèle est évaluée au moyen d'une étude de cas dans le centre-ville de Pittsburgh.

Le modèle basé sur GCNN surpasse les autres méthodes de base, y compris le LSTM multicouche et le LASSO avec un MAPE moyen de test de 10,6% lors de la prévision des occupations de stationnement au niveau des blocs 30 minutes à l'avance. Il a été remarqué également que L'intégration de la vitesse du trafic et des données météorologiques peut améliorer considérablement les performances de prévision.

#### **Travail de Sadia K, Reza R Alam A, et al. [86]**

La référence [86] présente une étude comparative entre un modèle LSTM et un modèle de régression de forêt aléatoire (RF) pour la prévision de la disponibilité des places de stationnement à Kuala Lumpur, en Malaisie. Les données utilisées est une collecte d'enregistrement toutes les 15 minutes entre 2016 et 2017, totalisant 34 933 enregistrements. Les résultats de l'analyse indiquent que le modèle de RF présente une meilleure performance en terme de MAE par rapport au modèle LSTM. Le MAE de ce dernier sur l'ensemble d'entraînement est de 0,067 et de 0,06 sur l'ensemble de test, tandis que pour le modèle de RF, le MAE est de 0,025 sur l'ensemble d'entraînement et de 0,026 sur l'ensemble de test. Cependant, ces résultats ne peuvent pas être généralisés à tous les cas, car cette étude a été réalisée pour une région spécifique. Pour des ensembles de données plus importants, le modèle LSTM, en tant qu'approche de réseau neuronal, peut être mieux formé et donner de meilleurs résultats.

### **Travail de Zeng et al. [113]**

Dans cette étude, un modèle GRU-LSTM superposé est proposé pour prédire l'occupation des parkings. Ce modèle combine les avantages du GRU en terme d'efficacité de prévision et ceux du LSTM en termes de précision. Deux expérience d'entrainement de modèles été effectué , la première est l'entraînement sur des séries uni-variées, par contre la deuxième prend en compte plusieurs facteurs tels que l'occupation, les conditions météorologiques et les vacances. L'étude utilise des données réelles du parking de Chongqing Wenxin Road et compare 13 modèles de prévision, entre autres le RNN,GRU, Bi-GRU, le LSTM-RNN empilé et le LSTM-Bi-LSTM empilé.

Les résultats comparatifs démontrent la supériorité du modèle proposé en termes de précision et d'efficacité dans la prévision à court terme de l'occupation des parkings. De plus, l'incorporation d'informations sur les facteurs externes qui influencent potentiellement le comportement des utilisateurs, améliore significativement la performance de la prévision en ayant ainsi comme résultats  $MSE=2.2225$ ,  $RMSE= 4.9395$  et une valeur de MAE de 1.5721.

### **2.3.5 Comparatif des travaux basés sur la prévision de l'occupation des places de stationnement**

Le Tableau 2.2 a pour objectif de synthétiser et analyser les articles étudiés dans la section précédente. Il mettra en évidence les méthodologies et résultats présentés par les auteurs, permettant ainsi une comparaison approfondie des approches de prévision.

Auteurs	Année	DataSet	Modèle	Résultat (ACC, MAE, MSE, MAPE, RMSE)
Zheng et al. [114]	2015	Données de Melbourne et de San Francisco (SFpark)	SVR	MSE= 0.002 MAE= 0.032
Yang et al. [108]	2019	Données de centre-ville de Pittsburgh	Graph-CNN	MAPE=10,6%
Xiangdong et al. [104]	2019	Les données d'un Park en Chine	LSTM	MAPE= 90%
Sadia K, Reza R Alam A, et al. [86]	2021	Des données réelles à Kuala Lumpur, en Malaisie	RF	MAE= 0.025 %
Zeng et al. [113]	2022	Des données réelles du parking de Chongqing Wenxin Road	GRU-LSTM	MSE=2.2225, RMSE= 4.9395, MAE= 1.5721.
Canlı et al. [35]	2022	Les données de 607 parkings à Istanbul	GRU	ACC= 91% MAE=0.9 MSE=2.35 RMSE=1.53%
Zeng et al. [37]	2022	-	DWT-Bi-LSTM	RMSE= 7.42746 MSE= 55.16721 MAE= 4.6
Feng et al. [40]	2022	Les données de 9 parkings publics de Santa Monicaa, California et US	dConvLSTM-DCN	-

TABLEAU 2.2 – Tableau comparatif des travaux basés sur la prévision de l'occupation des places de stationnement

## 2.4 Progrès des systèmes de stationnement intelligents en Algérie

En se concentrant sur la situation spécifique des systèmes de stationnement intelligents en Algérie, il est important de noter que le pays a également fait des progrès dans ce domaine. Le premier parking intelligent, connu sous le nom de Parking intelligent 1er Mai, a été implanté dans la commune de Sidi M'hamed à Alger. Opérationnel depuis novembre 2014, ce projet novateur a été réalisé grâce à la coordination entre l'APC de Sidi M'hamed et l'EGCTU (Etablissement de gestion de la circulation et de transport urbain). Ce parking intelligent

utilise une structure en hauteur capable d'accueillir jusqu'à 16 véhicules sur une plateforme rotative.

Cette initiative pionnière en Algérie s'inspire des expériences réussies de pays comme le Japon et la Russie, où les systèmes de stationnement intelligents ont démontré leur efficacité. Forts de cette première expérience pilote réussie, les responsables envisagent de généraliser progressivement ce concept dans d'autres communes d'Alger qui sont confrontées quotidiennement à un flux important de véhicules et font face à de réels problèmes de stationnement.

## 2.5 Constat et discussion

Après avoir étudié l'état de l'art, nous avons identifié trois approches clés dans le domaine du stationnement intelligent : la **gestion** des données massives, la **détection** en temps réel des emplacements disponibles et la **prévision** du nombre d'emplacements futurs.

L'approche basée sur la gestion et le traitement des données massives offre un potentiel considérable pour améliorer la gestion du stationnement intelligent. Elle permet une exploitation optimale des données générées par les appareils de l'IdO et facilite la mise en place de systèmes de gestion efficaces et innovants. Cependant, des défis tels que la confidentialité des données et la sélection appropriée des technologies doivent être pris en compte lors de la conception et de l'implémentation de ces systèmes.

D'autre part, la majorité des travaux de la deuxième approche qui est la détection en temps réel ont donné lieu à des études intéressantes et à des résultats satisfaisants. Cependant ils reposent sur l'identification des véhicules et la détection des emplacements des véhicules en utilisant des caméras de surveillance ce qui peut poser des problèmes de confidentialité pour les propriétaires de véhicules ce qui exige une prise en considération de la sécurité des informations.

En revanche, les systèmes de prévision du nombre d'emplacements de stationnement disponibles utilisent souvent des données historiques collectées par différents capteurs ( des capteurs à ultrason, des capteurs magnétiques ou des capteurs de pression) qui peuvent détecter la présence ou l'absence de véhicules dans un emplacement de stationnement pour modéliser les schémas de stationnement et prévoir les niveaux de demande futurs en étudiant les tendances et les saisonnalités à court et long terme grâce aux **RNN**, ce qui est un point crucial à prendre en considération.

Par conséquent, il est justifié que cette dernière approche vaut la peine d'être explorée en détail et poursuivre des recherches dans ce domaine pour explorer pleinement les avantages

potentiels de l'utilisation des RNN dans la prévision des emplacements de stationnement disponibles.

## 2.6 Conclusion

Dans ce chapitre, nous avons examiné plusieurs travaux avancés dans le domaine de stationnement. Nous avons constaté que le traitement des données massives offre des perspectives intéressantes pour améliorer la gestion du stationnement intelligent. Les techniques de collecte, d'analyse et de traitement des données massives permettent d'obtenir des informations précieuses pour optimiser la disponibilité des places de stationnement et réduire les problèmes de congestion. D'autre part, les algorithmes d'apprentissage automatique ont démontré leur efficacité dans la prévision de la disponibilité des places de stationnement.

Dans l'ensemble, l'étude de l'état de l'art nous a permis de contribuer à deux niveaux pour améliorer la gestion du stationnement intelligent, et qui seront détaillés dans les chapitres suivants.

# Nouvelle architecture d'analyse Big Data pour le stationnement intelligent

## 3.1 Introduction

Après avoir étudié la littérature sur le domaine du stationnement intelligent, nous avons constaté deux approches principales dans lesquelles nous pouvons contribuer. La première consiste à proposer une architecture générale pour gérer les données massives provenant des infrastructures IdO installées au sein des parcs de stationnement. La deuxième contribution concerne l'implémentation d'un modèle d'apprentissage profond basé pour prédire l'occupation des places de stationnement .

Ce chapitre vise essentiellement à présenter notre nouvelle architecture. Cependant, pour ce faire nous avons été confrontés à de nombreuses technologies, sur lesquelles nous avons mené une étude approfondie pour pouvoir enfin proposer une nouvelle architecture d'analyse de BD pour un système de stationnement intelligent.

## 3.2 Choix des outils/platformes BD

Les applications des villes intelligentes génèrent des données de manière continue à partir de différentes sources. Ainsi, les données ont diverses caractéristiques résumées dans les 10V du BD du chapitre précédent. Le taux de génération de données varie selon les capteurs utilisés et ceci en fonction des exigences de l'application (applications de soins et de santé, de gestion de l'énergie, de marketing, des services clients, des transports, de développement durable et de l'éducation, etc). Prenons comme exemples, les capteurs GPS qui sont capables de générer

des données dans des intervalles du temps très réduits d'environ quelques secondes, tandis que les capteurs de températures peuvent générer des données dans des intervalles distants d'environ une heure [79].

Cette hétérogénéité et ce grand volume de données ont fait qu'aucun outil traditionnel de gestion des bases de données ne peut vraiment gérer le BD, d'où l'émergence de nouvelles technologies permettant le stockage (MongoDB [12], Redis [15], Casandra [2], Hbase [2], Neo4j [14], etc.), le traitement et l'exploitation efficace de ce type de données (Hadoop, Spark, Flink, Storm, Kafka, etc [2].).

Ainsi lors du choix d'une plateforme pour le traitement des données, il est essentiel d'examiner les besoins de l'application concernée. Pour ce faire, l'utilisateur doit se poser quelques questions fondamentales, sur par exemple, les délais de traitement et la taille des données à traiter avant de prendre une décision éclairée [92]. Dans ce cadre, plusieurs métriques et critères entrent en jeu.

### 3.2.1 Scalabilité

La scalabilité fait référence à la capacité d'un système à fonctionner efficacement lorsqu'il est soumis à une augmentation de la charge de travail. En d'autres termes, un système doit être en mesure de gérer une augmentation du nombre d'utilisateurs, de transactions ou de données sans que cela n'affecte négativement ses performances. Pour évaluer la scalabilité d'un outil, il est important de considérer des facteurs tels que la capacité de traitement, la capacité de stockage, la bande passante et la disponibilité des ressources.

### 3.2.2 Tolérance aux pannes

La tolérance aux pannes fait référence à la capacité d'un système à continuer de fonctionner même en présence de pannes matérielles ou logicielles. Pour évaluer la tolérance aux pannes d'un outil ou d'une plateforme, il est important de considérer des facteurs tels que la redondance, les mécanismes de récupération, la détection des pannes et les mécanismes de réparation automatique.

### 3.2.3 Modes de traitement des méga données

Selon les besoins en matière de données, les infrastructures informatiques peuvent traiter les données en mode batch (en lot), en temps quasi réel ou en temps réel [43]. Par exemple,

Hadoop est couramment utilisé pour le traitement par lots, tandis que Spark est utilisé pour le traitement en temps réel [73].

### 3.2.3.1 Mode batch ou traitement par lot

C'est un mode de traitement des mégadonnées dans lequel les données sont collectées, stockées et traitées en blocs. Les données sont collectées sur une période donnée, puis traitées en une seule opération. Les tâches de traitement en mode batch sont planifiées à l'avance et exécutées à des intervalles réguliers, souvent pendant les heures creuses (soit au repos, en fin de journée ou pendant la nuit) [43] afin de ne pas affecter les performances des applications en temps réel.

Les traitements en batch sont souvent utilisés pour les opérations de traitement lourdes qui prennent du temps et des ressources, tels que les rapports, les analyses ou la génération de statistiques.

### 3.2.3.2 Traitement en temps quasi réel/en continu (stream processing en anglais)

C'est le traitement ou l'analyse de flux continu d'événements. Les données sont traitées à la volée, sans avoir besoin d'attendre que toutes les données soient collectées avant de les traiter. Ce mode de traitement est souvent utilisé pour traiter de grands volumes de données en temps proche au réel.

### 3.2.3.3 Traitement en temps réel

C'est un mode de traitement qui permet de traiter les données au moment où elles sont créées ou reçues, sans délai perceptible. Ce qui signifie que le temps de réponse pour le traitement est très court. Il est souvent utilisé dans des applications en temps réel de surveillance, de contrôle de processus, de détection de fraude, de transactions financières, etc. À la différence d'un stream processing qui s'occupe de la vélocité de données, le traitement temps réel s'occupe de la latence des données [98]. c'est-à-dire le temps qu'il faut pour traiter les données et obtenir une réponse en temps réel. Il met l'accent sur la capacité à traiter les données instantanément, sans délai perceptible, pour prendre des décisions ou agir en conséquence.



Bien que les termes de traitement en temps réel et de traitement en flux soient souvent utilisés de manière interchangeable, il existe une différence subtile entre les deux ;

La différence entre les deux modes de traitement est que le traitement en temps réel nécessite une réponse immédiate tandis que le traitement en flux peut prendre un certain temps pour traiter les données et générer des résultats. Le traitement en temps réel est plus couramment utilisé dans des applications critiques qui nécessitent une réponse immédiate, tandis que le traitement en flux est plus adapté aux applications qui nécessitent un traitement continu et qui peuvent tolérer un léger délai de traitement.

### 3.2.4 Taille des données supportées

La taille des données supportées est la capacité d'une plateforme à gérer des fichiers de données de grande taille, souvent supérieure à des gigaoctets. Les plateformes doivent être capables de stocker et de traiter de grandes quantités de données, en évitant les goulets d'étranglement liés aux limitations de taille de fichier ou de stockage. Les plateformes qui offrent une haute capacité de stockage et des mécanismes de partitionnement de données sont souvent préférées.

### 3.2.5 Sécurité

La sécurité est essentielle lors du traitement de données sensibles. Il est important de choisir une solution qui offre des fonctionnalités de sécurité robustes pour protéger les données.

## 3.3 Infrastructure de stockage

En plus des systèmes de stockage traditionnels basés sur une représentation structurée des données sous forme de relations et utilisant Structured Query Language (SQL), les villes intelligentes ont également besoin de structures de stockage plus avancées capables de stocker des données semi-structurées et non structurées [90] connus sous le nom NoSQL (pour Not Only SQL). De ce fait, il existe plusieurs types de bases de données dédiées aux Big Data, chacune ayant ses propres caractéristiques et avantages [78] [69]. Voici une comparaison de trois bases de données Big Data les plus populaires [6] [41].

### 3.3.1 HBase

C'est une base de données distribuée open source développée par les fondations logicielles Apache [2], le fruit d'une amélioration de Google Bigtable. Cette plateforme principalement écrite en java, peut stocker des quantités massives de données allant de téraoctets à pétaoctets sous forme de tableaux [2]. Sous Hadoop, HBase est utilisée pour stocker les données massives au niveau de Hadoop Distributed File System (HDFS). Un des avantages de cette plateforme est de fournir un accès aléatoire et rapide à une grande quantité de données structurées [96]. HBase est utilisée pour l'analyse des journaux en ligne, les distributions Hadoop, les applications à forte écriture (applications lourdes) et MapReduce.

### 3.3.2 Mongo DB

MongoDB [12] est une base de données multiplateforme orientée document. Elle offre des performances élevées et est de nature dynamique où nous n'avons pas besoin de prédéfinir un schéma pour les données. MongoDB stocke les données au format JSON, ce qui facilite l'envoi des données sous la forme désirée, et elle offre une vitesse, une disponibilité et une évolutivité élevées. Elle est considérée comme la meilleure base de données NoSQL pour gérer l'analyse de données en temps réel [41]. Elle est souvent utilisée dans les tâches d'analytique et IA, les applications mobiles et jeux et les applications IdO et systèmes de gestion de contenu [4]. Plusieurs grandes entreprises françaises adoptent MongoDB, comme Criteo, Axa Banque, Le Figaro et plus récemment Bouygues Telecom [12].

### 3.3.3 Cassandra

Cassandra [2] est un système de gestion de base de données open source, caractérisé par une architecture distribuée capable de gérer une grande quantité de données sur différents serveurs de base, et offre une haute disponibilité sans aucune sorte de panne. Il est écrit en Java et développé par Apache Software Foundation. Il peut être utilisé pour stocker les données des capteurs, des systèmes de messagerie, des sites Web de commerce électronique et des applications toujours actives [4]. Il dispose d'un système distribué peer-to-peer sur ses nœuds, ainsi les données sont réparties entre tous les nœuds du cluster [18].

### 3.3.4 Comparaison entre les bases de données NoSQL

Nous avons mené une riche recherche bibliographique pour comprendre les caractéristiques de ces trois bases de données et présenter une étude comparative entre elles selon quelques critères que nous présentons dans le Tableau 3.1.

	Hbase	MongoDB	Cassandra
Modèle de base de données	Orienté colonnes	Orienté document	Orienté colonnes
Réplication [78] [69]	Maître-esclave.	Maître-esclave.	Maître-esclave
Langage de programmation code/ pris en charge	Java/ C, C#, C++, Groovy, Java, PHP, Python, Scala	C++/ C, C#, C++, Erlang, Haskell, Java, JavaScript, Perl, PHP, Python, Ruby, Scala	Java/ C#, C++, Clojure, Erlang, Go, Haskell, Java, Node.js, Perl, PHP, Python, Ruby, Scala
Les systèmes d'exploitation du serveurs	Linux, Unix, Windows	Solaris, Linux, OS X, Windows	BSD, Linux, OS X, Windows
Latence [41] [20] [65]	Temps d'exécution élevé pour les lectures et réduit pour les mises à jour	Temps d'exécution moyen pour les lectures et les mises à jour	Temps d'exécution moyen pour les lectures et réduit pour les mises à jour
Langage de requêtes [69]	API calls, REST, XML, Thrift	Système de fichiers à mémoire volatile	CQL, API calls
Sécurité [111]	Authentification :moyen, Contrôle d'accès :moyen, Chiffrement :faible, Vérification :moyen	Authentification :moyen, Contrôle d'accès :élevé, Chiffrement :moyen, Vérification :faible	Authentification :faible, Contrôle d'accès :faible, Chiffrement :moyen, Vérification :faible
Tolérance aux pannes [64]	Point de défaillance unique dans l'approche maître-esclave (la panne dans le nœud maître). Peut être surmonté grâce à un regroupement de basculement.	Plusieurs fonctionnalités et mécanismes pour garantir la disponibilité continue des données avec la réplication et l'approche de sharding car en cas de panne d'une instance les autres instances continueront de fonctionner.	Cassandra peut continuer à fonctionner même en cas de défaillance d'un ou plusieurs nœuds grâce à son architecture pair à pair.

TABLEAU 3.1 – Tableau de comparaison entre les bases de données NoSQL

## 3.4 Plateformes de traitement et d'analyse de données

Il s'agit de l'ensemble des plateformes analytiques de BD qui servent d'interface pour collecter les données, effectuer les analyses et les traitements requis en utilisant des modes de traitement de données appropriés. Les trois plus grandes technologies les plus utilisées dans le domaine de stationnement sont Hadoop, Spark et Flink [101]. Nous présentons ainsi une comparaison de ces trois technologies considérant quelques critères tel que présenté dans le Tableau 3.2.

### 3.4.1 Hadoop

Hadoop est une plateforme open source qui fait partie de la fondation Apache basée sur Java. Elle permet de gérer et traiter de grands ensembles de données, structurées et non structurées, qui résident sous la forme de clusters, dans le cadre d'un système distribué. Cette plateforme informatique a été conçue afin de résoudre les problèmes liés à la volumétrie et à la variété des données. Elle offre un espace de stockage massif pour tous les types de données, une immense puissance de traitement ainsi que la possibilité de prendre en charge une quantité de tâches virtuellement illimitées [2]. Parmi les sociétés géantes qui utilisent Hadoop on trouve Twitter, LinkedIn, ou encore eBay et Amazon.

### 3.4.2 Spark : 3G de Big Data

Spark est un moteur open-source conçu en 2009 afin d'accélérer le traitement des systèmes Hadoop. Il permet une analyse et un traitement ultra-rapide de Big Data en cluster. Son utilisation est principalement dédiée aux applications du Machine Learning et des pipelines de données [81]. Il est idéal pour des informations immédiates en temps réel et un accès aux données existantes. Parmi les grandes sociétés du monde de l'internet qui ont développé Spark à très grande échelle, on trouve Netflix, Yahoo et ebay.

### 3.4.3 Flink : 4G de Big Data

Flink est un moteur de traitement distribué et une plateforme open-source d'analyse de données évolutif. Il est conçu pour répondre aux problématiques de traitement de données par lots grâce à la combinaison du traitement en streaming (en mémoire) et du traitement par lot (sur disque) [36] avec une gestion automatique de la mémoire. Il a été développé par les deux créateurs Fabian Hueske et Aljoscha Krettek en janvier 2015 auteur de l'idée

de « streaming d'abord, avec le batch comme cas particulier de streaming ». Dès que les données arrivent, elles sont immédiatement "transférées" via un programme de streaming. Il est également capable de gérer les données tardives dans les flux en utilisant des filigranes. De plus, Flink fournit un mode de compatibilité très puissant qui permet d'utiliser le code Storm, MapReduce, ... existant sur le moteur d'exécution Flink.

Contrairement à Spark, Flink est indépendant de Hadoop mais il peut rejoindre d'autres infrastructures de l'écosystème Hadoop, telles que Spark, Storm et Samza [2], dans la course à la gestion en flux du Big Data ainsi qu'à l'utilisation de HDFS (Hadoop Distributed File System) pour lire, écrire, stocker et traiter les données [75].

#### 3.4.4 Kafka

Kafka est un système de messagerie distribué qui est souvent utilisé pour la transmission de données en temps réel entre différents composants d'une architecture de traitement de données [103]. Contrairement à Hadoop, Spark et Flink, Kafka ne traite pas les données, mais fournit plutôt un système de transport de données haute performance, bien que certaines fonctionnalités de traitement de flux de données soient disponibles dans Kafka à travers des API telles que Kafka Streams [45]. En utilisant Kafka avec Hadoop, Spark ou Flink, les données peuvent être ingérées en temps réel à partir de différentes sources, stockées dans des clusters Hadoop ou dans des bases de données NoSQL, puis traitées en temps réel à l'aide de Spark ou Flink. Kafka peut également être utilisé pour envoyer les résultats des analyses de données en temps réel à d'autres systèmes ou applications.

#### 3.4.5 Comparaison entre Spark, Hadoop et Flink

Le tableau 3.2 illustre une comparaison des outils de traitement de Big Data ; Hadoop, Spark et Flink [75].

Fonctionnalité	Spark	Hadoop	Flink
Scalabilité	Très scalable en utilisant la structure de données RDD (Resilient Distributed Dataset) distribuée sur plusieurs nœuds, l'optimisation des opérations en mémoire et la minimisation des E/S disque.	Une grande scalabilité grâce à HDFS qui permet le parallélisme et à son modèle de calcul MapReduce	Une grande scalabilité, parallélise les opérations sur les flux en les distribuant sur plusieurs nœuds pour s'adapter à des volumes de données importants et à des charges de travail croissantes.
Tolérance aux pannes	Obtenue en stockant la chaîne de transformations	Haute et obtenue en répliquant des blocs de données	Basée sur les instantanés distribués de Chandy-Lamport pour un débit élevé.
Mode de traitement	Traitement par lots et en streaming	Traitement par lots	Traitement en streaming et par lots
Taille des données supportées	plusieurs dizaines ou centaines de Go	Pétaoctets à exaoctets	Gigaoctets à pétaoctets et au-delà
Coût	Coûteux en mémoire et en nombre de clusters	Moins coûteux	Coûteux en mémoire et en nombre de clusters
Gestion de la mémoire	Gestion configurable	Gestion configurable	Gestion automatique
Langage pris en charge	Java, R, Scala, Python	Java, c++, Ruby, Groovy, Perl, Python.	Java, Scala, Python et R
Les bibliothèques ML	MLlib	Mahoot	Flink ML
Sécurité	Sécurisé grâce à l'intégration avec Hadoop	Extrêmement sécurisé via l'authentification, l'autorisation, le chiffrement, l'audit des accès et l'intégration avec par exemple Kerberos	Support d'authentification des utilisateurs via l'authentification, l'autorisation et le chiffrement des données.
Rapidité de traitement et performances	Traitement rapide grâce au stockage des données intermédiaires en mémoire.	Traitement ralenti à cause de la lecture et écriture à partir d'un disque	Le plus rapide grâce aux opérateurs d'itération en boucle fermée natifs qui accélèrent l'apprentissage automatique et le traitement des graphes.
Simplicité d'utilisation	Facile grâce à ses opérateurs de haut niveau	Difficile à cause de codage de chaque opération à la main	Facile grâce à ses opérateurs de haut niveau

TABLEAU 3.2 – Tableaux de comparaison entre Spark, Hadoop et Flink

### 3.5 Nouvelle Architecture d'analyse de Big Data

Compte tenu de l'étude bibliographique et de l'étude comparative des différents outils que nous avons effectuée, nous proposons une nouvelle architecture d'analyse de Big Data basée sur l'apprentissage profond pour le stationnement intelligent comme présentée dans la Fig. 3.1.

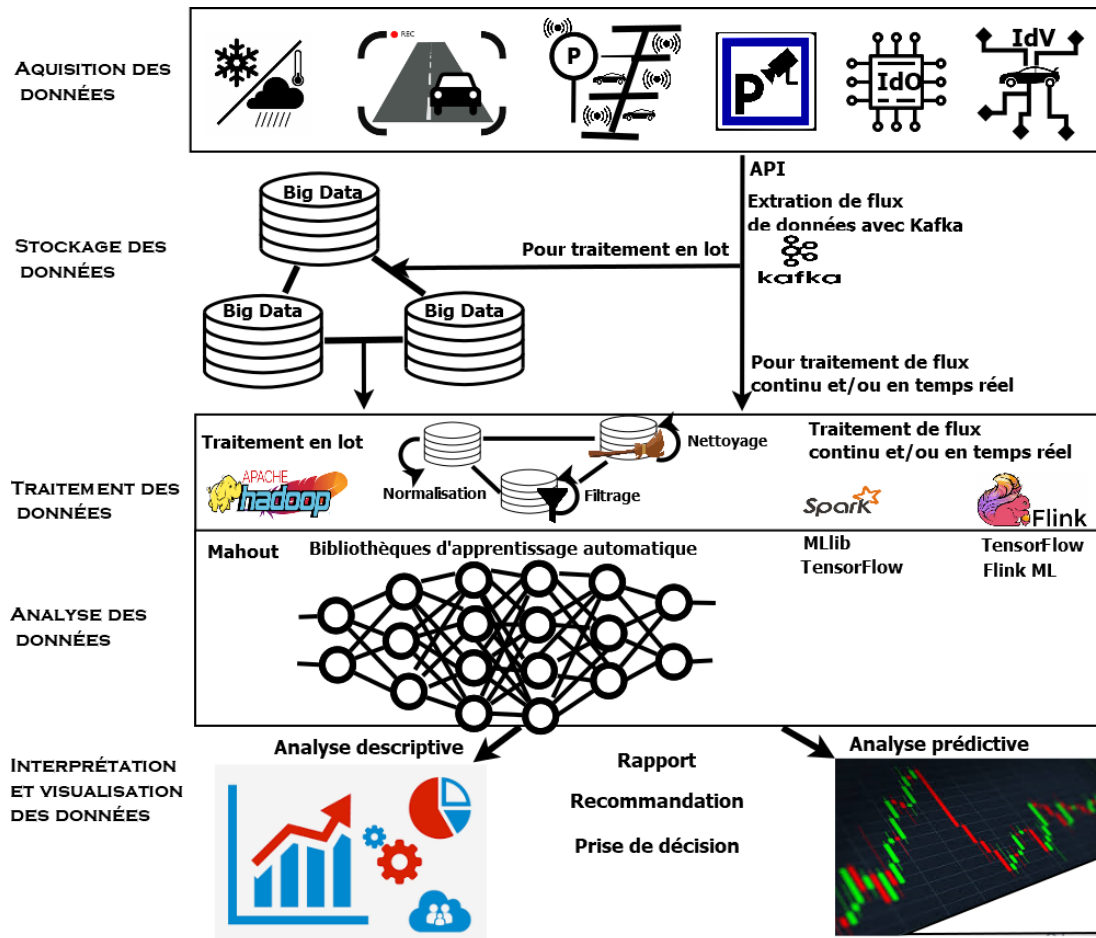


FIGURE 3.1 – Architecture d'analyse de Big Data basée sur l'apprentissage profond pour le stationnement intelligent.

#### 3.5.1 Présentation générale de l'architecture

Notre architecture composée de 5 niveaux (Acquisition, stockage, traitement et analyse enfin la couche d'interprétation et de visualisation des données) intègre à la fois le traitement



des données par lot, en flux continu et en temps réel pour pouvoir effectuer une analyse descriptive et prédictive.

La première peut donner l'occupation des places de stationnement à différentes heures de pointe de la journée pour aider les autorités de la ville à élargir des parkings ou d'en ouvrir d'autres à proximité. Elle peut aussi aider à constater la demande de stationnement dans différentes zones lors d'événements spéciaux et guider les conducteurs pour qu'ils trouvent un parking approprié rapidement et réduire les embouteillages dans la ville. En plus, l'analyse des valeurs moyennes d'occupation de chaque parking à chaque heure d'un jour donné peut fournir aux automobilistes plus d'informations sur le stationnement à l'échelle horaire, ce qui pourrait les aider à prendre une décision de stationnement en fonction de l'heure à laquelle ils veulent entrer dans la ville.

D'un autre côté, une analyse prédictive peut estimer le temps d'arrivée du conducteur en fonction duquel on peut prédire si une place de stationnement dans un parking reste vacante à son arrivée.

En plus, on peut utiliser la détection de l'occupation pour fournir des informations en temps réel sur la disponibilité des places de stationnement et procéder à des ajustements de prix et les limites de temps en fonction de la demande pour une utilisation efficace des places de stationnement dans une zone encombrée, etc [76] [58].

Il est essentiel de souligner que notre architecture est flexible et peut être adaptée en fonction des besoins spécifiques du projet. Nous accordons une attention particulière à la sécurité des données, et aux performances du système pour garantir des opérations fluides et fiables et qui peuvent être assurées grâce aux trois plateformes Hadoop, Spark et Flink qui intègrent des mécanismes de sécurité fiables tels que l'authentification, l'autorisation et le chiffrement.

## 3.5.2 Les différents niveaux de notre architecture

Notre architecture se compose de cinq niveaux qui sont essentiels pour assurer une analyse efficace des données de stationnement intelligent.

### 3.5.2.1 Premier niveau, l'acquisition

Il est chargé de collecter les données provenant de différentes sources. Principalement, on peut utiliser les capteurs de stationnement, les caméras et les appareils de l'IdO qui

permettent une acquisition en temps réel des informations sur l'occupation des places de stationnement. Cependant, d'autres données pertinentes peuvent être aussi récupérées, comme les données météorologiques, les données du trafic routier et d'autres informations concernant les véhicules connectés. A ce niveau, il est crucial d'avoir des mécanismes de collecte fiables et efficaces pour garantir la qualité des données.

Dans un système de stationnement intelligent, plusieurs types de capteurs peuvent être installés pour collecter des données sur l'occupation des places de stationnement. Voici quelques-uns des capteurs couramment utilisés :

- **Capteurs de détection de présence** : Ces capteurs détectent la présence d'un véhicule dans une place de stationnement. Ils peuvent être basés sur différentes technologies, notamment :
  - **Capteurs magnétiques** : Ils utilisent des champs magnétiques pour détecter la présence d'un véhicule.
  - **Capteurs de pression** : Ils mesurent les changements de pression induits par le poids d'un véhicule.
  - **Capteurs infrarouges** : Ils détectent la présence d'un véhicule en mesurant les changements dans les rayonnements infrarouges.
- **Capteurs de comptage** : Ces capteurs comptent le nombre de véhicules entrant et sortant d'une zone de stationnement. Ils sont souvent utilisés pour suivre l'occupation globale d'un parking ou pour contrôler l'accès à un parking.
- **Caméras de surveillance** : Les caméras sont utilisées pour capturer des images ou des vidéos des places de stationnement. Elles peuvent être utilisées pour détecter la présence de véhicules, lire les plaques d'immatriculation, effectuer une reconnaissance des caractéristiques des véhicules, etc.
- **Capteurs ultrasoniques** : Ces capteurs utilisent des ondes sonores pour mesurer la distance entre le capteur et un objet, généralement un véhicule. Ils peuvent être utilisés pour détecter la présence de véhicules dans une place de stationnement.
- **Capteurs de mouvement** : Ces capteurs détectent les mouvements à l'intérieur d'une place de stationnement pour déterminer si elle est occupée ou non. Ils peuvent être basés sur des technologies telles que les capteurs infrarouges passifs (PIR) ou les

capteurs à micro-ondes.

- **Capteurs de guidage** : Ces capteurs aident les conducteurs à trouver rapidement des places de stationnement disponibles. Ils peuvent être intégrés dans le sol, les panneaux de signalisation ou les éclairages et utilisent des indicateurs visuels pour guider les conducteurs vers les places libres.

### 3.5.2.2 Deuxième niveau, le Stockage

IL est responsable du stockage et de la gestion évolutive des données collectées. A ce niveau, des APIs peuvent être utilisées pour récupérer les données collectées dans le premier niveau pour les envoyer soit en flux continu pour le traitement en streaming ou en temps réel en utilisant par exemple Kafka ou bien pour le stockage pour un éventuel traitement en lot en utilisant les différentes technologies de stockage distribué comme Hadoop Distributed File System (HDFS) ou Apache Cassandra. cette couche prend en compte les besoins en termes de volume, de vélocité et de variété des données pour assurer un stockage et un envoi des données efficace.

### 3.5.2.3 Troisième niveau, le Traitement

Il s'occupe du traitement des données en temps réel et par lots. En utilisant Apache Kafka comme plateforme de streaming, les flux de données peuvent être intégrés, traités et distribués en temps réel. Pour le traitement par lots, il est possible d'exploiter la puissance d'Hadoop avec Spark pour effectuer des analyses complexes et des calculs sur de grands volumes de données et pour le traitement en temps réel avec une faible latence, nous pouvons utiliser Apache Flink.

### 3.5.2.4 Quatrième niveau, l'Analyse

Il est essentiel pour extraire des informations significatives à partir des données collectées. Des outils tels que SQL, Python et des bibliothèques d'apprentissage automatique (Mahout, MLib, et Flink ML pour les outils Hadoop, Spark et Flink respectivement) peuvent être utilisés pour effectuer une analyse descriptive et prédictive. Cela permet de comprendre l'occupation des places de stationnement à différentes heures de la journée, les demandes spéciales lors d'événements et les valeurs moyennes d'occupation de chaque parking à chaque heure,

comme on peut aussi prédire l'occupation des espaces de stationnement. Ces informations sont précieuses pour les autorités de la ville, les gestionnaires de parkings et les conducteurs.

### **3.5.2.5 Enfin, le niveau de l'Interprétation et la Visualisation**

il permet de représenter visuellement les résultats de l'analyse. Nous utilisons des tableaux de bord interactifs, des graphiques, des cartes ou des applications mobiles pour présenter les données de manière compréhensible aux utilisateurs finaux. Cela facilite la prise de décision et améliore l'expérience des conducteurs.

## **3.6 Conclusion**

Il est primordial de bâtir une bonne architecture technique appropriée sur laquelle les données seront collectées, stockées, traitées et analysées en temps réel. Cette architecture doit être à la fois résistante aux pannes, puissante et surtout évolutive. Car, c'est le choix de la bonne architecture qui aide à la bonne exploitation des grands flux de données qui voient le jour à chaque instant, et qui à son tour mène automatiquement à la réalisation d'importantes valeurs ajoutées à long terme sur le plan économique des entreprises.

# Prévision de l'occupation des espaces de stationnement, proposition, implémentation et validation

## 4.1 Introduction

Dans ce chapitre, nous allons commencer par présenter notre deuxième contribution qui consiste en la proposition d'un nouveau modèle basé sur les RNNs pour la prévision de l'occupation des parcs de stationnement. Ce modèle est une partie du niveau analyse des données (niveau 4) de notre architecture proposée dans le chapitre précédent. Ensuite, nous allons présenter les étapes de l'implémentation et la validation du modèle de prévision proposé pour le stationnement intelligent. Pour mettre en œuvre le modèle de prévision des places de stationnement disponibles, nous avons utilisé un ensemble d'outils et bibliothèques que nous allons aussi présenter dans ce chapitre. De plus, pour valider le modèle nous avons utilisé un dataset de parking que nous présenterons suivi d'une étape de pré-traitement et de préparation des données. Enfin, nous expliquerons notre méthode d'évaluation du modèle, ainsi que les résultats obtenus.

## 4.2 Motivation

Au cours de notre recherche, nous avons identifié différentes approches utilisées pour la prévision des espaces de stationnement disponibles. Parmi ces approches, les RNN ont été largement étudiés en raison de leur capacité à modéliser des séquences temporelles mieux que les algorithmes classiques comme ARIMA, SARIMA, encore mieux que les algorithmes ML tels les CNN, les SVM, etc.

Nous avons également constaté, lors de l'étude des travaux précédents, que les GRU sont particulièrement performants pour les problèmes de stationnement par rapport aux autres variantes des RNN y compris les cellules LSTM.

C'est donc dans cette perspective de performance et d'adaptabilité aux séquences temporelles du stationnement que nous avons choisi d'utiliser les GRUs comme base pour notre modèle de prévision. Et pour mieux comprendre cette variante, la section suivante va détailler son architecture.

## 4.3 Gated Recurrent Unit

Un réseau Gated Recurrent Unit (GRU), en français réseau récurrent à portes, est une variante améliorée des LSTM introduite en 2014. Une unité GRU requiert moins de paramètres à apprendre et est appelée "cellule". La cellule est l'unité de base de chaque architecture d'un RNN, elle est équivalente à un neurone artificielle dans l'architecture des ANN.

### 4.3.1 Architecture GRU

Une cellule GRU est composée de plusieurs composants clés que nous allons détailler dans le sous titre suivant. Notamment une porte de réinitialisation (reset gate), une porte de mise à jour (update gate) et une une sortie GRU.

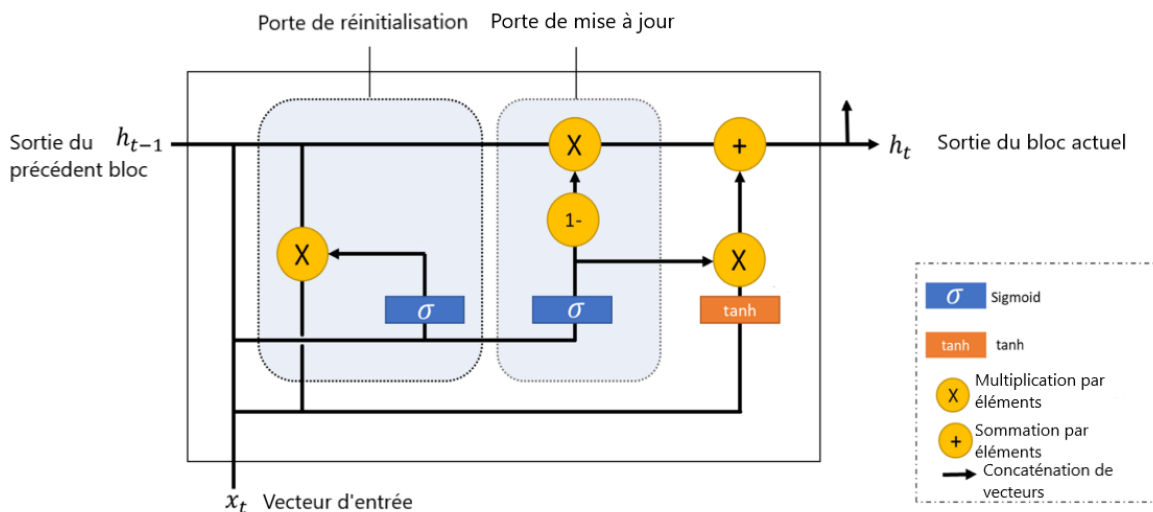


FIGURE 4.1 – Architecture d'une cellule GRU

#### 4.3.1.1 La porte de réinitialisation

Joue un rôle essentiel dans le contrôle de la quantité d'informations passées que le réseau doit utiliser pour calculer la prochaine sortie. Pour ce faire, l'état caché précédent est combiné avec les données d'entrée, puis passe par une fonction sigmoïde. Cette fonction sigmoïde permet de déterminer quelles coordonnées sont pertinentes et doivent être conservées. Ensuite, le résultat est multiplié par l'ancien état caché, ce qui permet de préserver les coordonnées importantes de l'état précédent. Ainsi, une partie de l'état précédent est oubliée à travers cette porte, ne laissant que les informations significatives pour la suite du traitement.

#### 4.3.1.2 Porte de mise à jour

Détermine la quantité d'informations actuelles qui sera oubliée lors du calcul de la prochaine sortie. Elle permet au modèle de choisir quelles informations sont pertinentes pour le contexte actuel.

#### 4.3.1.3 La sortie final

Elle est obtenue en combinant l'état caché précédent (partiellement effacé par la porte de réinitialisation) avec l'entrée actuelle du réseau. Cette combinaison est ensuite normalisée à l'aide de la fonction tangente hyperbolique ( $\tanh$ ) pour obtenir des valeurs comprises entre -1 et 1.

Ensuite, toutes les coordonnées de l'état caché sont filtrées en fonction de leur pertinence pour les prévisions, en utilisant la sortie de la porte de mise à jour. Les coordonnées jugées "inutiles" sont annulées, tandis que les coordonnées jugées pertinentes sont conservées.

La sortie du réseau GRU est obtenue en combinant l'information passée et l'information actuelle, tout en filtrant les coordonnées inutiles et en préservant les coordonnées pertinentes pour les prévisions

### 4.3.2 Avantages de GRU

**Efficacité de calcul** : GRU est une version améliorée des LSTM, avec une structure plus simple, ce qui les rend plus rapides à entraîner et à exécuter.

**Réduction du problème de disparition du gradient** : Les RNN souffrent souvent du problème de disparition du gradient à cause de leurs liens récurrents (les gradients se

propagent de moins en moins profondément dans le réseau, ce qui rend l'apprentissage des dépendances à long terme difficile). Les GRU ont été conçus pour résoudre partiellement ce problème en introduisant des mécanismes de portes adaptatives qui permettent de contrôler le flux d'informations dans le réseau. Cela permet aux GRU de mieux capturer les dépendances à long terme dans les séquences.

**Utilisation efficace de la mémoire :** Au lieu d'avoir une cellule de mémoire séparée (le cas des LSTM), les GRU utilisent une seule variable pour représenter l'état caché et la mémoire. Cela permet une utilisation plus efficace de la mémoire dans le réseau.

**Adaptabilité aux données :** Les GRU sont capables d'apprendre à retenir et à oublier des informations en fonction des dépendances temporelles des données. Les portes de réinitialisation et de mise à jour permettent de réguler l'importance relative des informations antérieures et nouvelles lors du processus de prévision. Cette capacité d'ajustement dynamique permet aux GRU de s'adapter de manière flexible aux différentes caractéristiques des séquences temporelles.

## 4.4 Nouveau modèle hybride Conv1D-BiGRU pour la Prévision de l'occupation des parkings

Notre décision d'utiliser les BiGRU comme base pour notre modèle repose sur les différents avantages qu'il offre par rapport aux autres variantes des RNN, tels que les LSTM et ceci grâce à son architecture représentée dans la figure 4.1 précédente.

Le BiGRU est une extension des GRU qui permet d'exploiter à la fois les informations passées et futures lors de la prédiction. Contrairement aux GRU classiques qui ne considèrent que les dépendances séquentielles passées, le BiGRU utilise deux ensembles de cellules GRU : une séquence est traitée dans l'ordre normal (de gauche à droite), tandis qu'une autre séquence est traitée dans l'ordre inverse (de droite à gauche). Cela permet au modèle d'exploiter les informations des deux directions et de capturer les dépendances séquentielles à la fois passées et futures.

En complément des BiGRU, nous avons choisi d'utiliser la couche Conv1D (Convolution à une dimension) pour extraire des caractéristiques temporelles pertinentes à partir des données brutes telles que les changements brusques, les tendances ou les motifs récurrents tel que les motifs saisonniers, c'est-à-dire les variations régulières qui se répètent chaque période.



La combinaison d'une couche Conv1D avec les BiGRU nous permet de bénéficier des avantages de chaque approche. La couche Conv1D peut identifier les motifs temporels significatifs dans les données, tandis que les GRU bidirectionnels peuvent capturer les dépendances séquentielles à court terme tels que les variations rapides dans les données.

## 4.5 Outils et bibliothèques utilisés

Dans cette partie, nous définissons les outils et bibliothèques que nous avons utilisés lors de la réalisation de notre modèle.

### 4.5.1 Python

Python est un langage de programmation qui permet d'écrire des instructions et de les exécuter de manière interactive. Il est polyvalent et prend en charge différents styles de programmation tels que la programmation orientée objet, la programmation procédurale et la programmation fonctionnelle [8] [29].

Python est le choix privilégié pour le développement d'applications d'apprentissage automatique en raison de sa simplicité, de sa puissance, de sa richesse en bibliothèques et de sa portabilité [29]. Il offre un environnement convivial et efficace pour concevoir, implémenter et déployer des modèles d'apprentissage automatique performants.

### 4.5.2 Google Collaboratory

Google Colab est une plateforme de développement et d'exécution de projets de ML basée sur le cloud. Elle offre un environnement de programmation interactif qui permet aux utilisateurs de créer, partager et exécuter des blocs de code python, appelés "notebooks", de manière collaborative [29], sans avoir besoin de télécharger, d'installer ni d'exécuter quoi que ce soit [7].

En plus d'être une solution gratuite qui donne accès à des ressources informatiques puissantes comme GPU (Graphics Processing Unit) et TPU (Tensor Processing Units), Google colab offre une sélection de bibliothèques logicielles préinstallées [29]. Ce qui facilite l'entraînement et l'exécution d'algorithmes de machine ML sur des jeux de données volumineux.

### 4.5.3 Tensorflow et Keras

TensorFlow est une bibliothèque de calcul numérique développée par Google, pour créer et entraîner des modèles d'apprentissage automatique et d'apprentissage profond, en particulier des réseaux de neurones profonds [29]. Elle fournit une infrastructure flexible pour la construction de graphiques de calcul et l'exécution de calculs distribués sur diverses plateformes matérielles, telles que des CPU, des GPU et des TPU.

Keras est une interface haut niveau pour la construction de modèles d'apprentissage automatique, spécifiquement de réseaux de neurones, qui s'exécute sur TensorFlow [29], mais elle est également disponible sur d'autres bibliothèques de calcul. Elle est conçue pour faciliter et accélérer le processus de développement des modèles, en fournissant des abstractions conviviales et une syntaxe simple. Keras permet aux développeurs de créer rapidement des architectures de réseau de neurones en empilant des couches, de configurer facilement les hyperparamètres et de s'entraîner sur les données. Il offre également des fonctionnalités avancées telles que le transfert d'apprentissage, l'augmentation des données et la validation croisée.

À partir de TensorFlow 2.0, Keras est intégrée en tant que couche d'abstraction haut niveau dans TensorFlow [30]. Ce qui donne la possibilité d'utiliser l'API de Keras pour créer et entraîner des modèles d'apprentissage automatique en utilisant les fonctionnalités de TensorFlow.

## 4.6 Présentation de l'ensemble de données

Le dataset SFpark fourni des enregistrements de la disponibilité mesurée des parkings dans les rues de San Francisco. Ces données ont été obtenues à partir de l'API publique du projet SFpark [32].

L'API SFpark renvoie le nombre de places de parking libres et le nombre total de places de parking disponibles par segment de route pour un total de 5 314 places de parking réparties sur 579 segments de route dans la zone pilote. Les données sont fournies sous forme de 6 fichiers CSV distincts [9].

Les observations de disponibilité des parkings ont été collectées pendant une période allant du 13 juin 2013 au 24 juillet 2013, avec une fréquence d'interrogation de l'API d'environ 5 minutes. Au total, le dataset contient environ 7 millions d'observations d'occupation des parkings pour les segments de route.

Le fichier spécifique "sfpark\_filtered\_136\_247\_100taxis" utilisé dans notre projet se concentre sur 420 segments de parkings et comprend un total de 12 096 intervalles de temps pour lesquels des données de disponibilité des parkings sont disponibles. Les colonnes du fichier sont les suivantes :

**Timestamp :** Indique le moment où l'API SFpark a été interrogée, arrondi à la minute la plus proche.

**Segmentid :** Identifiant du segment de stationnement défini par SFpark.

**Capacity :** Nombre total de places de stationnement dans le segment de stationnement. Veuillez noter que la capacité peut varier dans le temps en raison des restrictions de stationnement.

**Occupied :** Nombre actuel de places de stationnement occupées dans le segment de stationnement, rapporté par les capteurs de SFpark.

**ObservedK :** A une valeur de 1 si, pendant la période indiquée par l'horodatage, au moins un taxi est passé sur ce segment de route, et 0 sinon. K correspond à la  $K^{eme}$  répétition de l'échantillonnage aléatoire des taxis et de leur choix de voie. Si observedK a la valeur 1, cela signifie qu'un véhicule sonde aurait rapporté la même occupation que les capteurs de SFpark.

**DiffK :** Différence entre la dernière mesure d'occupation des taxis et le nombre actuel de places de stationnement occupées. K correspond à la  $K^{eme}$  répétition mentionnée ci-dessus.

Il y a un bloc d'informations simulées sur la disponibilité des places de stationnement, basées sur les données de véhicules sondes pour la détection des foules dans les parkings. Les informations sont fournies pour 10 répétitions afin de réduire les biais causés par le sous-échantillonnage aléatoire. Les variables dans ce bloc sont observedK, DiffK.

Ces informations permettent de comprendre la disponibilité simulée des places de stationnement basée sur les données de véhicules sondes.

## 4.7 Tendances et analyse périodique des données

Avant tout, une analyse des données de stationnement est effectuée pour une meilleure compréhension des données ; l'analyse préliminaire permet de mieux comprendre les tendances des données de stationnement, et peut aider à identifier les caractéristiques (variables) les plus pertinentes pour la prévision de l'occupation des parkings. En examinant les relations entre les différentes variables et la variable cible "occupeid", on peut déterminer quelles caractéristiques sont les plus prédictives et lesquelles peuvent être éliminées ou transformées.

### 4.7.1 Analyse des tendances d'occupation des parkings :

Il y a une forte corrélation d'occupation entre les différents parkings, comme le montrent les figures 4.2 et 4.3. Ces graphiques représentent les tendances horaires sur une période d'un mois et de deux mois respectivement. Cette corrélation suggère que l'occupation des parkings est influencée par des facteurs communs, tels que les heures de pointe de la journée ou les événements spéciaux.

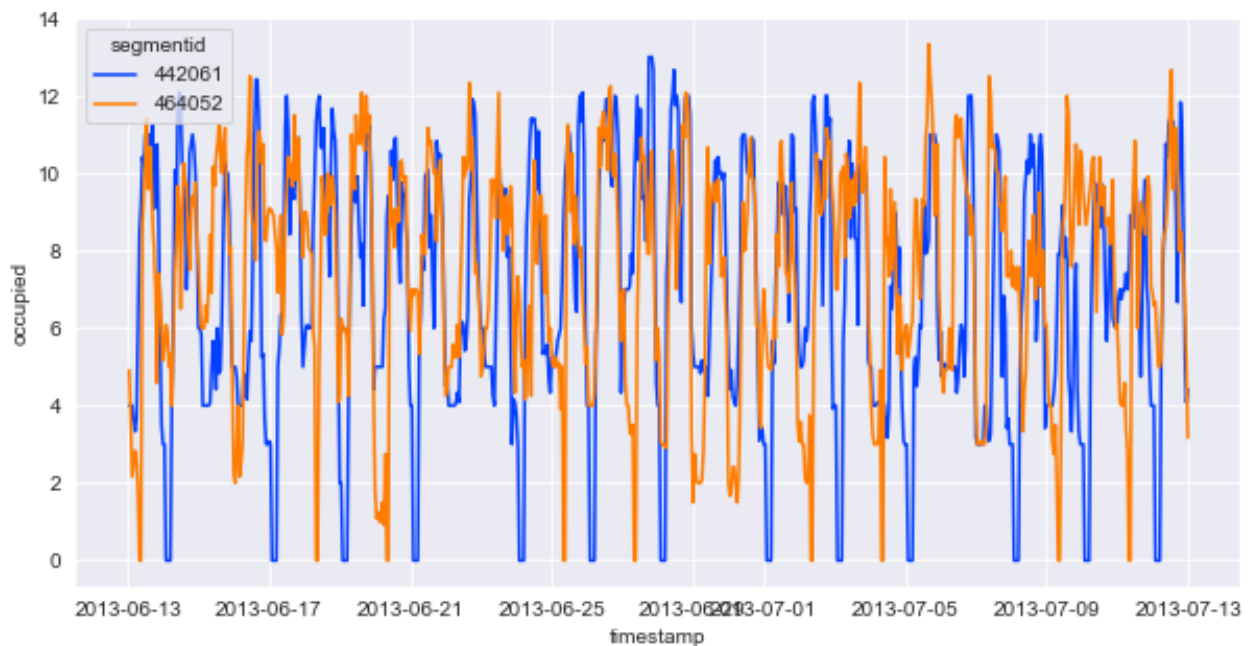


FIGURE 4.2 – Occupation horaire de deux parkings dans une période d'un mois

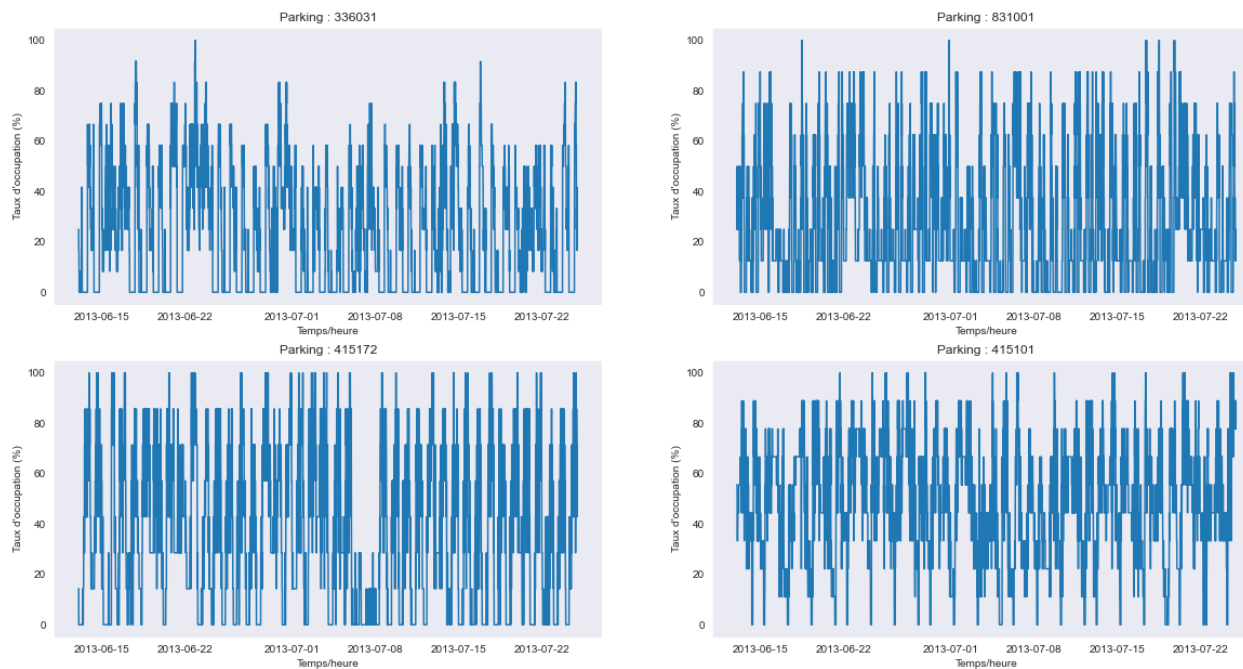


FIGURE 4.3 – Occupation par heure de 4 parkings

D'après le graphique de la figure 4.4 qui montre l'occupation moyenne des parkings pour la journée de 19-06-2013 on observe que :

Le taux d'occupation des parkings augmente après 5h00 du matin et atteint environ 55% aux alentours de midi. Après midi, le taux d'occupation diminue légèrement. On observe ensuite une nouvelle augmentation après 16h00, atteignant son maximum aux alentours de 18h00 pour une occupation de 70%. Ces tendances peuvent s'expliquer par les heures de début et de fin d'activité dans les bureaux administratifs, les entreprises de services et les écoles de San Francisco. Les heures de début d'activité se situent généralement entre 8h00 et 9h00 du matin, tandis que les heures de fin d'activité sont généralement entre 16h00 et 18h00 en soirée. Ainsi, l'augmentation de l'occupation des parkings le matin peut être attribuée aux personnes se rendant au travail ou à l'école, tandis que la diminution légère de l'occupation l'après-midi peut être due à des horaires de travail flexibles ou à des activités après l'école. La nouvelle augmentation en fin d'après-midi peut être causée par les personnes quittant leur travail ou leur école pour rentrer chez elles.

Ces observations montrent une corrélation entre les heures d'activité des bureaux administratifs, des entreprises de services et des écoles, et les fluctuations d'occupation des parkings tout au long de la journée.

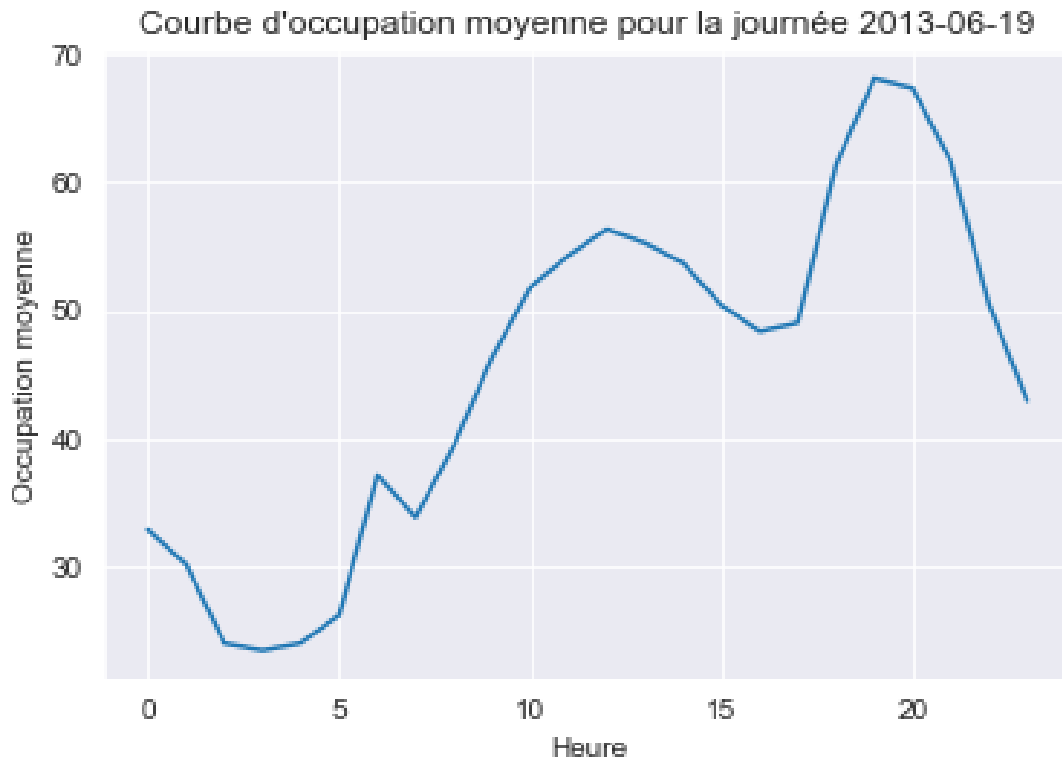


FIGURE 4.4 – Courbe d'occupation moyenne pendant une journée

L'histogramme de la figure 4.5 pour représenter l'occupation moyenne par jour de la semaine. Chaque barre représente la moyenne d'occupation pour un jour de la semaine.

Nous remarquons que es parkings sont plus utilisés le week-end surtout le premier jour du weekend (Samedi) que les autre jours. Cette situation est confirmé par : les véhicules à usage individuel sont plus nombreux le week-end pour raisons telles que les achats, etc..

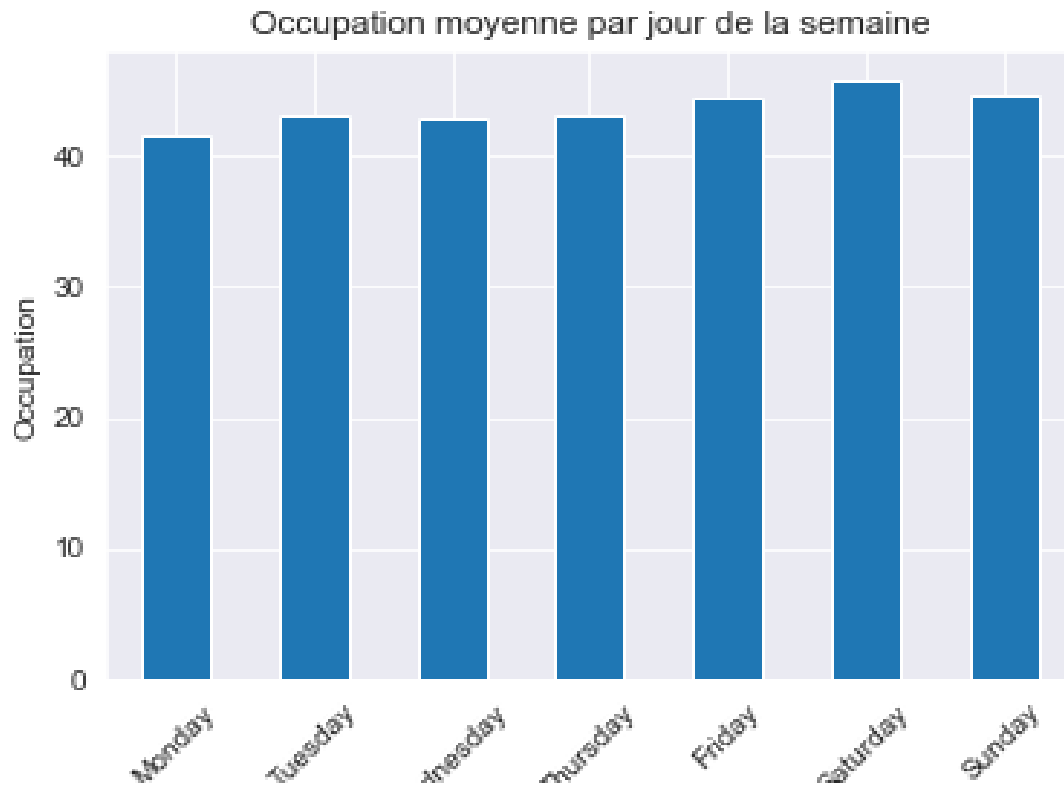


FIGURE 4.5 – Occupation moyenne par jour de la semaine

#### 4.7.2 Analyse de corrélation entre les variables

La figure 4.6 la matrice de corrélation entre les variables du dataset. Une corrélation élevée est observée entre la variable cible "occupied" et la variable "capacity". Cela indique qu'il existe une relation significative entre l'occupation des parkings et la capacité totale du parking. En revanche, les variables "diffk" présentent une faible corrélation avec la variable cible. Cela suggère que ces variables supplémentaires ne contribuent pas de manière significative à la prévision de l'occupation des parkings. Par conséquent, il est possible de conclure que les variables les plus importantes pour la prévision de l'occupation des parkings sont "timestamp" "segmentid" et "capacity", tandis que les variables supplémentaires peuvent être négligées ou exclues du modèle.

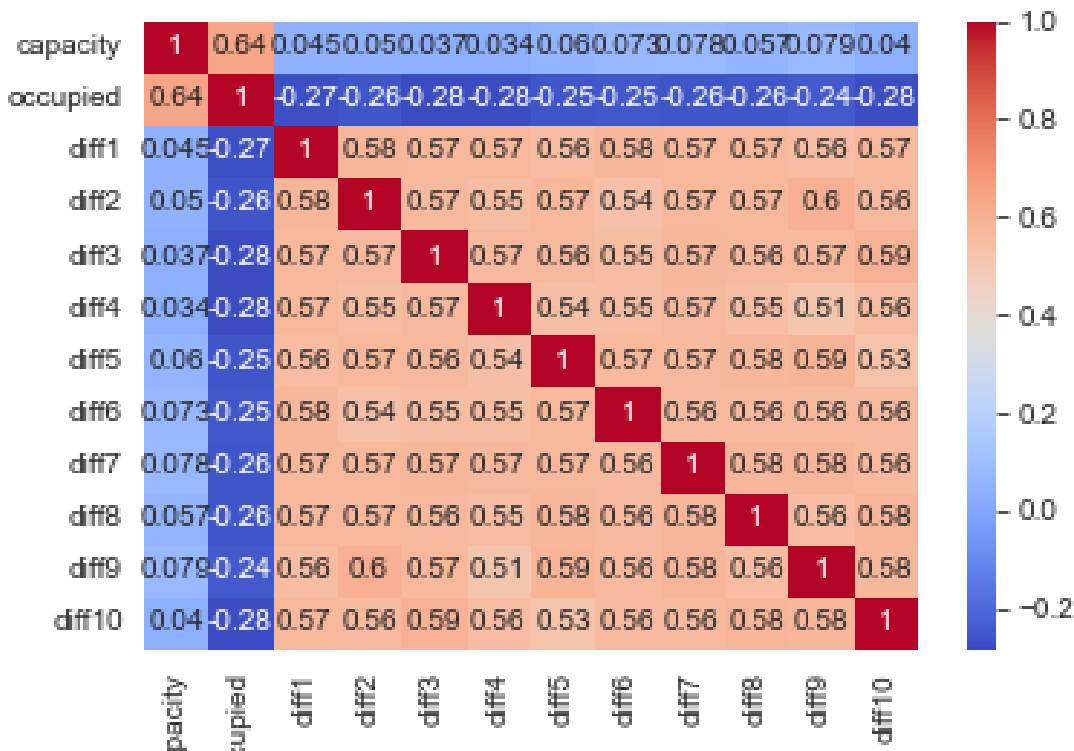


FIGURE 4.6 – Matrice de corrélation entre les variables

## 4.8 Pré-traitement / Préparation des données

### 4.8.1 Nettoyage

Nous avons choisi de ne prendre en compte que les quatre premières colonnes du dataset sur un total de 24 variables, comme illustré dans la figure 4.7, car les variables supplémentaires `observedK` et `diffK` ne semblent pas être directement pertinentes pour notre tâche de prévision des espaces de stationnement disponibles. En effet, les variables `diffk` présentent une faible corrélation avec notre variable cible "occupied", tandis que les valeurs de `observedK` sont toutes définies comme "NAN" (non disponibles). En se concentrant uniquement sur les variables "timestamp", "segmentid", "capacity" et "occupied", qui fournissent les informations essentielles sur le moment de la collecte des données, l'identifiant du segment de stationnement, la capacité totale et le nombre de places occupées, nous simplifions le modèle et nous nous focalisons sur les caractéristiques clés nécessaires à notre objectif de prévision.



```
[6] # Sélection des colonnes qui sont pertinentes pour notre analyse ( 4 variables sur un total de 24)
df = df.loc[:, ["timestamp", "segmentid", "occupied", "capacity"]]

# Supprimer les lignes contenant des valeurs NaN
df = df.dropna()
```

FIGURE 4.7 – Nettoyage des données.

## 4.8.2 Ajout des variables endogènes et exogènes

L'ajout de variables **endogènes** telles que la température, les précipitations et l'humidité peut aider à améliorer les performances de prévision des réseaux GRU en apportant des informations supplémentaires qui peuvent avoir une corrélation avec l'occupation des parkings. Par exemple, si les précipitations sont élevées, cela peut avoir un impact sur le nombre de voitures qui cherchent à se garer dans un parking couvert.

La température peut également avoir une influence sur l'occupation des parkings. En effet, en période de chaleur, les gens préfèrent se déplacer en voiture plutôt qu'à pied ou en vélo, ce qui peut augmenter la demande de places de stationnement.

Dans notre cas, nous avons téléchargé les données météorologiques horaires à partir du site web de la NASA (National Aeronautics and Space Administration) [13]. Ensuite, nous avons effectué une interpolation linéaire pour lisser les données à des intervalles de temps de 5 minutes afin de les faire correspondre à notre jeu de données comme le montre la figure 4.8.

```
[38] # interpoler la temperature sur des intervalles de 5 min
temperature_interpole= np.interp(df_agg['timestamp'], data_weather['datetime'], data_weather['T2M'])

# interpoler les precipitations sur des intervalles de 5 min
precipitation_interpole= np.interp(df_agg['timestamp'], data_weather['datetime'], data_weather['PRECTOTCORR'])

# Ajouter les deux colonnes temperature et precipitation à notre dataframe
df_agg['temperature']=temperature_interpole
df_agg['precipitation']=precipitation_interpole
```

FIGURE 4.8 – Ajout des données endogènes.

Quant aux variables **exogènes** telles que l'heure, le jour de la semaine ou bien le mois, elles peuvent également avoir un impact sur l'occupation des parkings. Par exemple, il est possible que les parkings soient plus occupés en fin de semaine ou pendant les heures de pointe de la journée. Comme le montre la figure 4.9, nous avons ajouter des variables :

```
[39] import holidays

sf_holidays = holidays.US()

df_agg['weekday'] = df_agg['timestamp'].dt.weekday
df_agg['month'] = df_agg['timestamp'].dt.month
df_agg['hour'] = df_agg['timestamp'].dt.hour
df_agg['holiday'] = df_agg['timestamp'].apply(lambda x: x.date() in sf_holidays)
df_agg['holiday'] = df_agg['holiday'].astype(int)
```

FIGURE 4.9 – Ajout des données exogènes.

Ces variables fournissent des informations supplémentaires permettant au modèle à mieux capturer les tendances et les saisons de l'occupation du parking pour améliorer sa capacité de prévision.

### 4.8.3 Normalisation des données

Pour la normalisation de nos données, nous avons utilisé la standardisation MinMax détaillée dans le chapitre 1. Pour cela, nous avons utilisé la fonction `MinMaxScaler` de la bibliothèque `sklearn`, et qui est montré dans la figure 4.10.

```
[3] from sklearn.preprocessing import MinMaxScaler

values = df.values
scaler = MinMaxScaler(feature_range=(0, 1))
scaled = scaler.fit_transform(values)
```

FIGURE 4.10 – Normalisation des données.

### 4.8.4 Création des séquences

Une séquence, fait référence au nombre d'observations consécutives utilisées comme entrée pour le modèle GRU. C'est la longueur de la fenêtre glissante utilisé pour créer des paires d'entrée-sortie dans l'apprentissage du modèle.

Dans notre cas la séquence est de longueur 4 avec des intervalles de temps de 5min, ce qui signifie que le modèle va capturer les informations sur les 20 dernières minutes (4 observations

x 5 minutes =20 min) les plus récentes comme entrée pour prédire la prochaine observation. Une illustration est représentée dans la figure 4.11.

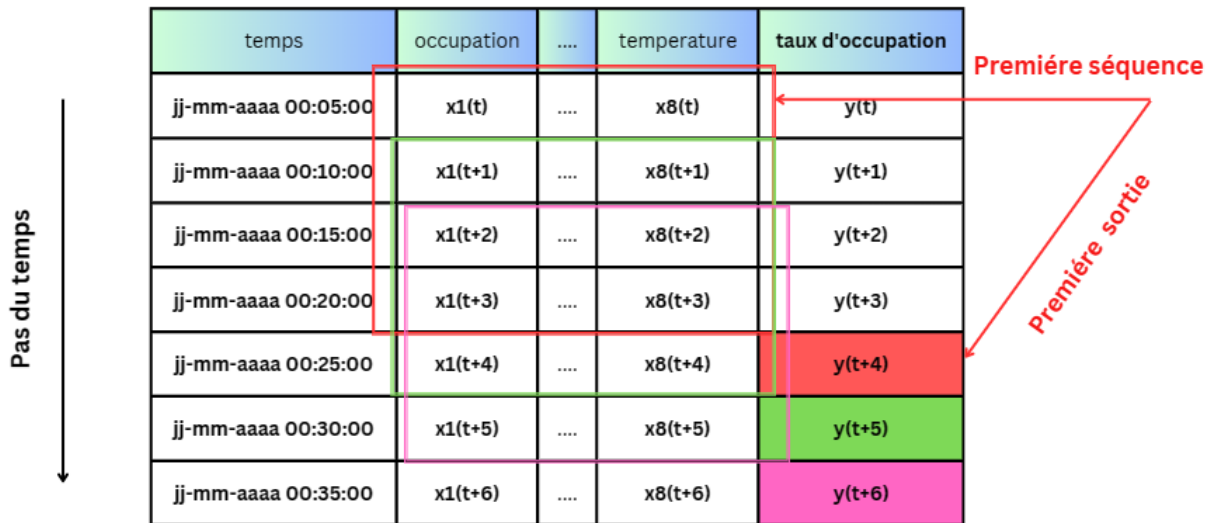


FIGURE 4.11 – Création des séquences

### 4.8.5 Fractionnement des données

Le fractionnement des données pour les RNN n'est pas aléatoire comme pour le reste des modèles. Dans notre cas, nous devons prendre en considération l'ordre temporel des données. Il est important de s'assurer que les séquences de données dans l'ensemble de test se trouvent après les séquences d'entraînement dans le temps. Cela permet d'éviter toute fuite d'informations du futur vers le passé lors de l'entraînement du modèle. Une présentation schématique est donnée dans la figure 4.12.

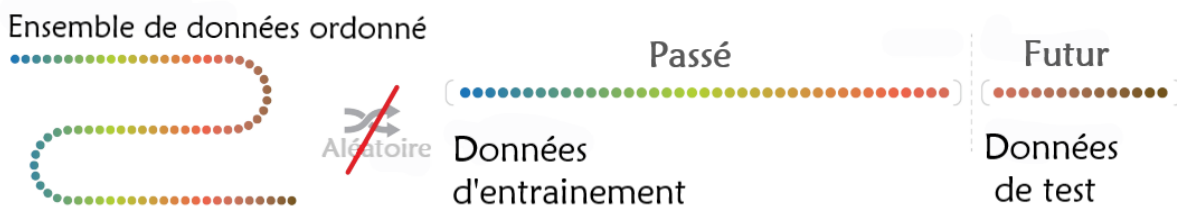


FIGURE 4.12 – Fractionnement des données pour les séries temporelles

En ce qui concerne les tailles de données dans chaque ensemble, celles-ci ont été divisées en deux sous-ensembles : trois quart pour la phase d'entraînement l'équivalent de 80%, et

le reste pour le test 20% et ceci en prenant en considération l'ordre chronologique comme le montre la figure 4.13.

```
[8] train_size= int(0.8 * len(df))

train = values[:train_size, :]
test = values[train_size:, :]

train_x, train_y = train[:, :target_column_index], train[:, target_column_index]
test_x, test_y = test[:, :target_column_index], test[:, target_column_index]
```

FIGURE 4.13 – Fractionnement des données.

### 4.8.6 Préparation des tenseurs

Les réseaux de neurones récurrents et particulièrement les GRU s'attendent à ce que les entrées de données soient effectuées en 3D.

Lorsque nous préparons les données pour l'entraînement d'un modèle, nous devons convertir les séquences de données en tenseurs, qui sont des tableaux multidimensionnels, afin de pouvoir les utiliser comme entrée pour le modèle GRU. Elles doivent donc être présentées sous la forme suivante : [nombre de séquences, pas de temps (taille de la séquence), nombre de variables]. Les dimensions de ce format sont définies comme suit :

**Le nombre de séquences :** Il s'agit du nombre de séquences temporelles distinctes qui existent dans nos données.

**La longueur de la séquence :** Comme déjà défini cela correspond au nombre d'instant de temps que nous utilisons pour prédire la valeur suivante. Par exemple, si nous utilisons une séquence de longueur 4, nous utilisons les 4 instants de temps précédents pour prédire la valeur suivante ( 5 ème ).

**Le nombre de variables :** Il s'agit du nombre de variables ou de caractéristiques que nous utilisons pour représenter chaque instant de temps dans notre séquence.

La figure 4.14 montre comment nous avons effectuer le redimensionnement en 3D aux différents ensembles de données.

```
[21] time_steps = 4
      num_features = train_x.shape[1]
      num_targets = 1

      train_x = train_x.reshape((-1, time_steps, num_features))
      train_y = train_y.reshape((-1, time_steps, num_targets))
      test_x = test_x.reshape((-1, time_steps, num_features))
      test_y = test_y.reshape((-1, time_steps, num_targets))
```

FIGURE 4.14 – Redimensionnement des données.

## 4.9 Mise en œuvre et évaluation

Cette section va présenter les détails de mise en œuvre du modèle proposé, ainsi que les résultats obtenus.

### 4.9.1 Métriques de performances

Les métriques de performances sont des mesures essentiels dans l'évaluation des modèles d'apprentissage automatique. Elles permettent de quantifier l'efficacité des modèles et de fournir des mesures objectives de leur précision [29].

- Erreur quadratique moyenne (MSE) :

L'erreur quadratique moyenne mesure la moyenne des erreurs quadratiques entre les valeurs prédites et les valeurs réelles. Elle est souvent utilisée dans les problèmes de régression pour quantifier la qualité des prévisions. Une MSE plus faible indique une meilleure performance du modèle [29].

Formule mathématique :

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2 \quad (4.1)$$

où  $y_i$  représente les valeurs réelles,  $\bar{y}$  est la valeur prédite correspondante, et  $n$  est le nombre total d'échantillons.

- Erreur absolue moyenne (MAE) :

L'erreur absolue moyenne mesure la moyenne des erreurs absolues entre les valeurs

prédites et les valeurs réelles. Comme la MSE, elle est couramment utilisée dans les problèmes de régression. Elle est moins sensible aux valeurs aberrantes que la MSE [29].

Formule mathématique :

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \bar{y}| \quad (4.2)$$

où  $y_i$  représente les valeurs réelles,  $\bar{y}$  est la valeur prédite correspondante, et  $n$  est le nombre total d'échantillons.

— Racine de l'erreur quadratique moyenne (RMSE) :

La racine carrée de l'erreur quadratique moyenne est une version de la MSE où la racine carrée est prise pour rendre la métrique plus interprétable dans les mêmes unités que les valeurs d'origine. Elle est également utilisée dans les problèmes de régression et fonctionne de la même manière que la MSE [29].

Formule mathématique :

$$\text{RMSE} = \sqrt{\text{MSE}} \quad (4.3)$$

— Exactitude (Accuracy) :

L'exactitude mesure la proportion d'échantillons correctement classés par rapport au nombre total d'échantillons. Cette métrique est couramment utilisée dans les problèmes de classification, où le modèle prédit des étiquettes de classe [29].

Formule mathématique :

$$\text{Exactitude} = \frac{\text{Nombre d'échantillons correctement classés}}{\text{Nombre total d'échantillons}} \quad (4.4)$$

## 4.9.2 Architecture du modèle et les paramètres associés

### 4.9.2.1 L'architecture du modèle

L'architecture de notre modèle comporte quatre types de couches comme le montre la figure 4.15 suivante :

```
[18] # Architecture du modèle
      model.summary()

Model: "sequential"
-----
Layer (type)                Output Shape              Param #
-----
conv1d (Conv1D)              (None, 1, 32)            544
conv1d_1 (Conv1D)            (None, 1, 64)            4160
bidirectional (Bidirectiona (None, 1, 256)            148992
1)
bidirectional_1 (Bidirectio (None, 128)              123648
nal)
dense (Dense)                (None, 1)                 129
-----
Total params: 277,473
Trainable params: 277,473
Non-trainable params: 0
-----
```

FIGURE 4.15 – Architecture du modèle conv1D-BiGRU.

Voici une explication détaillée de chaque couche :

**1. Deux Couches Conv1D** : utilisées pour extraire des motifs locaux ( Ces motifs peuvent être des variations brusques, des tendances, des répétitions...etc) à partir des données séquentielles. Elles filtrent les entrées en appliquant des opérations de convolution sur des fenêtres glissantes de taille fixe.

**2. Deux couches BiGRU** : L'utilisation de couches bidirectionnelles permet d'exploiter l'information contextuelle dans les deux directions temporelles, en améliorant ainsi la compréhension des séquences.

**3. Une couche dense** : C'est une couche entièrement connectée qui prend les sorties des couches précédentes et effectue des calculs de transformation linéaire, suivis d'une fonction d'activation. Cette couche dense est généralement utilisée pour combiner les informations apprises à partir des couches précédentes et générer des sorties finales.

### 4.9.2.2 Les paramètres associés

Lors de la création d'un modèle d'apprentissage profond, plusieurs paramètres de base sont pris en considération. Les principaux paramètres et valeurs du modèle mis en place sont indiqués dans le tableau 4.1.

Paramètre	Valeur
Fonctions d'activation	relu
Fonction de perte	MSE
Descente de gradient (optimiseur)	Adam
Métriques	MSE, MAE, RMSE, Accuracy
Nombre d'epochs	15
Batch_size	128

TABLEAU 4.1 – Les paramètres du modèle conv1D-BiGRU.

### 4.9.3 Présentation des résultats

Pour valider notre approche, nous avons mesuré les métriques présentées précédemment sur notre modèle. ensuite effectué une comparaisons des résultats obtenues à celles de deux modèles, GRU et BiGRU, à deux couches entraîné sur le même dataset et avec les mêmes paramètres précédents.

#### 4.9.3.1 Mesure des métriques

Nous avons enregistré l'évolution des MAE et MSE du modèle au fil des époques, comme illustré dans les figures 4.16 et 4.17. Ces figures démontrent une diminution progressive des erreurs au cours des époques d'entraînement sur l'ensemble de test.



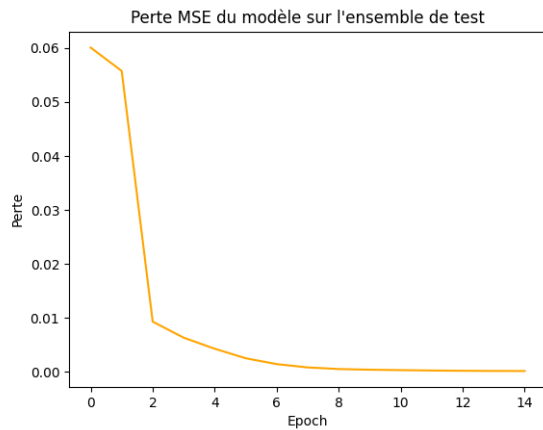


FIGURE 4.16 – Perte MSE du modèle conv1D-BiGRU .

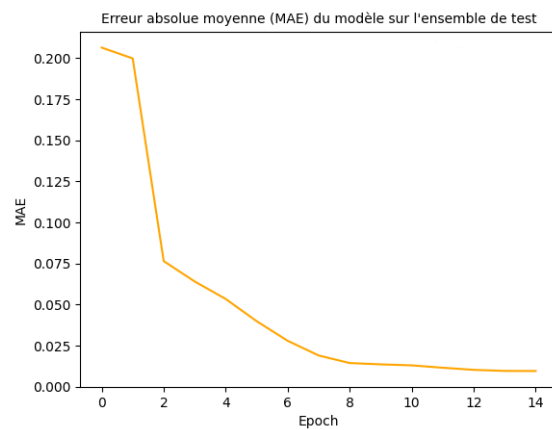


FIGURE 4.17 – Erreur absolue moyenne du modèle conv1D-BiGRU.

La diminution des erreurs MAE et MSE indique que le modèle s'améliore progressivement dans sa capacité à prédire avec précision les valeurs cibles. Plus précisément, une diminution de l'erreur MAE signifie que les prévisions du modèle s'approchent davantage des valeurs réelles en termes d'erreur absolue moyenne. De même, une diminution de l'erreur MSE indique que les prévisions du modèle s'ajustent progressivement aux valeurs réelles en termes de la différence quadratique moyenne.

#### 4.9.3.2 Comparaison des résultats

La comparaison a été effectuée au moyen de l'erreur absolue moyenne, de l'erreur quadratique moyenne, de la racine carrée de l'erreur quadratique moyenne et de la précision. Le tableau 4.2 ainsi que l'histogramme de la figure 4.18 montrent l'évaluation des erreurs et de la précision des trois modèles.

Modèle	MSE	MAE	RMSE	Accuracy
GRU	0.01	0.0291	0.0316	0.983
BiGRU	0.0066	0.0219	0.0257	0.988
Conv1D-BiGRU	0.00017	0.0095	0.0129	0.997

TABLEAU 4.2 – Comparaison des résultats

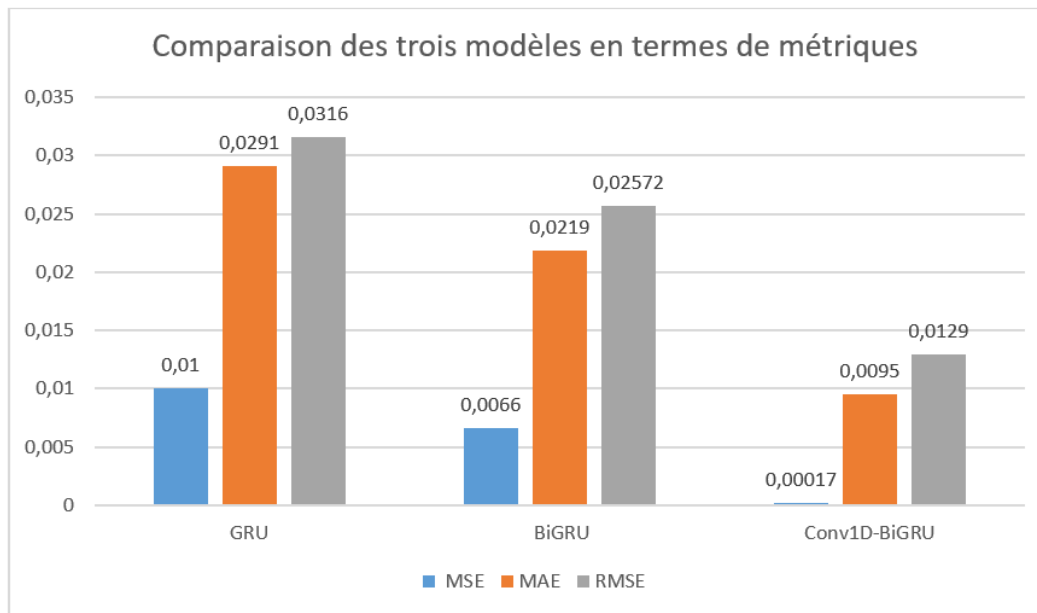


FIGURE 4.18 – Comparaison des trois modèles en termes de métriques

Pour bien illustrer la différences entre les trois modèles nous avons enregistré les graphes des prévision sur l'ensemble de test.

La figure 4.19 illustre la relation entre les valeurs réelles et prédites du taux d'occupation sur l'ensemble de données de test dans le modèle GRU. Les séries temporelles en bleu représentent les valeurs réelles, tandis que les prévisions pour l'ensemble de données de test sont représentées en orange. On observe une corrélation évidente entre les deux graphiques. Cependant, il est clair que les prévisions présentent des divergences significatives par rapport aux valeurs réelles, indiquant que les prévisions ne sont pas assez précises pour capturer avec précision les variations dans les données.

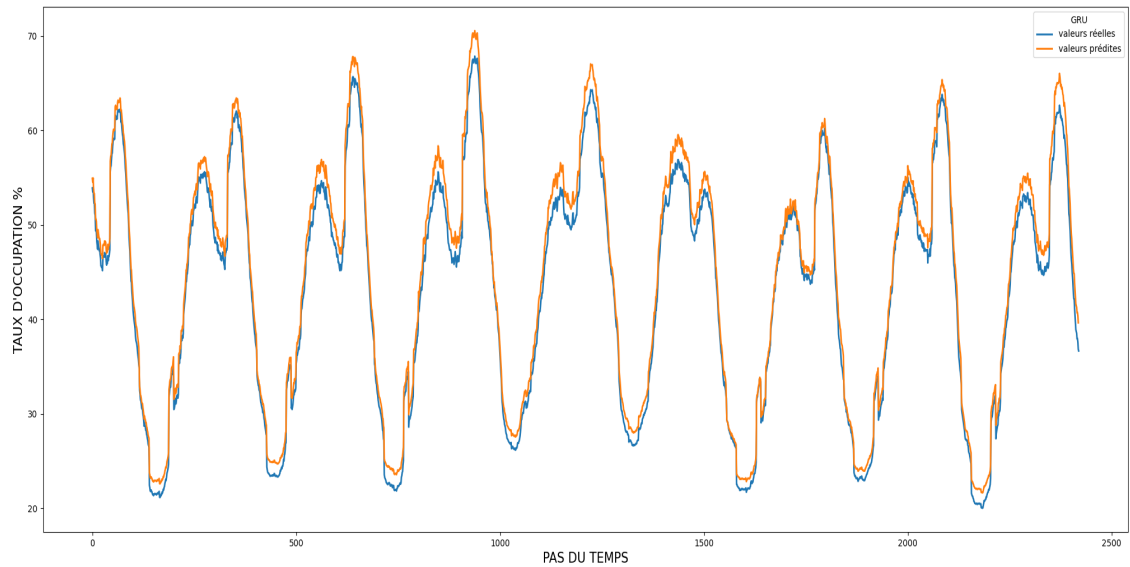


FIGURE 4.19 – Graphe des prévisions du modèle GRU.

La figure 4.20 présente la relation entre les valeurs réelles et prédites du taux d'occupation sur l'ensemble de données de test dans le modèle **BiGRU**. On observe une correspondance moyenne entre les deux graphiques. Globalement, le modèle BiGRU présente de meilleures performances que le modèle GRU en termes de prévision des séries temporelles.

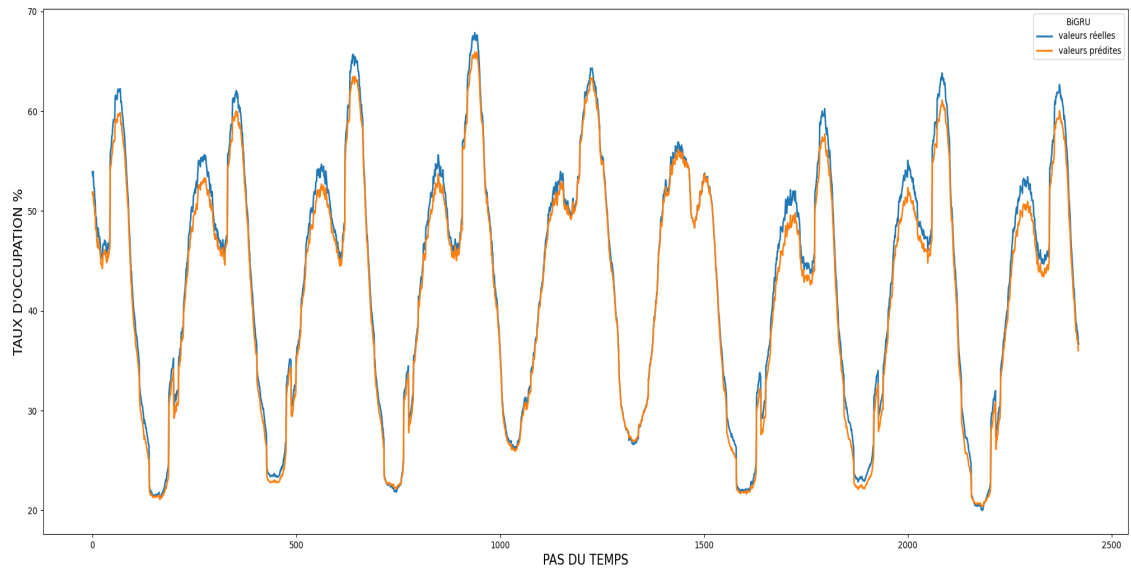


FIGURE 4.20 – Graphe des prévisions du modèle BiGRU.

La figure 4.21 présente la relation entre les valeurs réelles et prédites du taux d'occupation sur l'ensemble de données de test dans le modèle **Conv1D-BiGRU**. On observe une correspondance significative entre les deux graphiques, avec quelques légères erreurs. Globalement, le modèle Conv1D-BiGRU présente de meilleures performances que les modèles GRU et BiGRU en termes de prévision des séries temporelles.

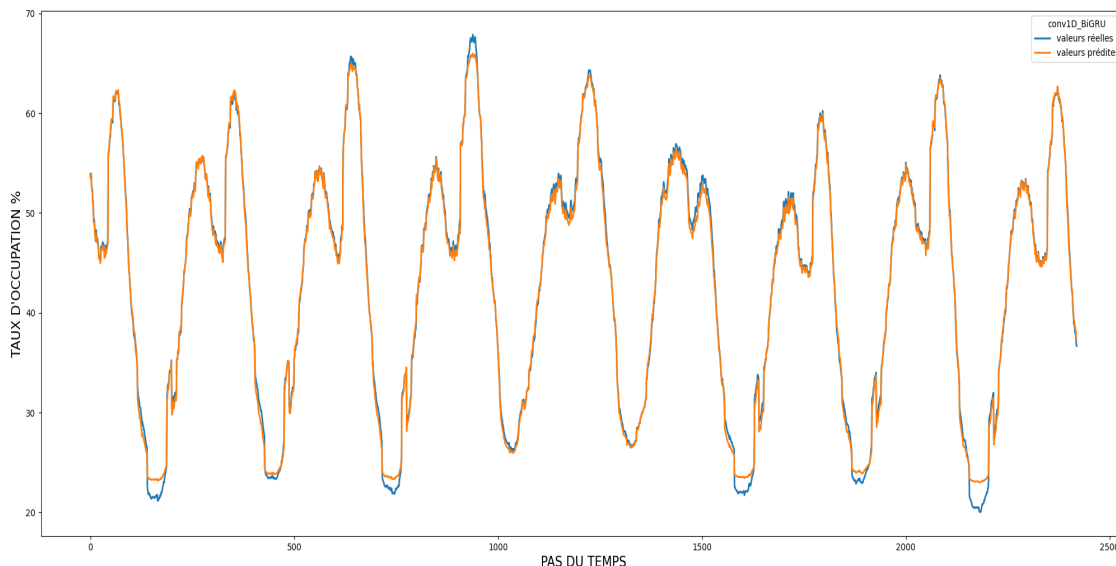


FIGURE 4.21 – Graphe des prévisions du modèle conv1D-BiGRU.

Ces résultats solides confirment que notre modèle Conv1D-BiGRU est capable de capturer avec précision les variations dans les séries temporelles et de fournir des prévisions plus fiables en le comparant aux deux modèles GRU et BiGRU. Ainsi, notre approche combinant la couche Conv1D avec le BiGRU s'avère être une solution prometteuse pour la prévision des séquences temporelles dans le domaine de stationnement intelligent.

## 4.10 Conclusion

À travers ce chapitre, nous avons présenté notre approche et décrit les outils et bibliothèques utilisés pour l'implémentation de notre modèle. Nous avons également expliqué les différentes étapes de préparation des données, suivi de la présentation et de la discussion des résultats obtenus. En conclusion, notre approche démontre des performances supérieures que celles des deux autres modèles testés GRU et BiGRU.

## Conclusion et perspectives

En conclusion, cette étude a comme objectif de concevoir un système de stationnement intelligent en utilisant des solutions IdO et Big Data. Tout au long de ce travail, nous avons exploré les concepts essentiels des villes intelligentes et nous nous sommes concentrés sur le stationnement intelligent en tant que domaine clé.

Dans le premier chapitre, nous avons établi les bases nécessaires pour comprendre les principes fondamentaux du sujet. Nous avons mis en évidence les technologies essentielles telles que l'Internet des Objets, le Big Data et le ML qui soutiennent ce domaine. De plus, nous avons souligné les avantages et les enjeux de la mise en place de système de stationnement intelligent dans le contexte du développement urbain.

Dans le deuxième chapitre, nous avons effectué une revue de littérature des travaux de recherche existants dans le domaine du stationnement intelligent. Nous avons identifié les approches utilisées et quelques limites. Cette analyse approfondie nous a permis de situer notre travail par rapport aux recherches existantes.

Dans le troisième chapitre, nous avons proposé une nouvelle architecture d'analyse des mégadonnées adaptée à un système de stationnement intelligent. Cette architecture constitue notre première contribution significative. Nous avons souligné l'importance de choisir une architecture technique solide pour la collecte, le stockage, le traitement et l'analyse des données massives générées par les infrastructures IdO des systèmes de stationnement.

Cette contribution a été acceptée et présentée lors du Colloque sur les Objets et Systèmes Connectés 2023 en Tunisie sous le titre "Architecture d'analyse de Big Data basée sur l'apprentissage profond pour le stationnement intelligent : Étude et Comparaison des Outils". Cette présentation témoigne de la qualité de notre travail et de son impact sur la communauté scientifique.

Dans le quatrième chapitre, nous avons mis en œuvre un modèle de prévision de l'occupation des parkings basé sur l'apprentissage automatique. Cette deuxième contribution majeure

nous a permis d'obtenir des résultats prometteurs, dépassant les performances des modèles GRU simple et BiGRU.

Enfin, notre mémoire a permis de répondre à son objectif initial en proposant une architecture d'analyse des mégadonnées et un modèle de prévision performant pour un système de stationnement intelligent. Ces contributions ouvrent la voie à des améliorations concrètes dans la gestion du stationnement.

En guise de perspectives, nous envisageons d'utiliser les plateformes Big Data à savoir Hadoop, Spark et Flink pour distribuer le modèle de prévision proposé sur différents clusters, afin d'accélérer les calculs et d'obtenir les prédictions le plus rapidement possible.

Cependant le temps, la rapidité de traitement et les performances de ces outils sont des paramètres importants à prendre en considération dans notre contexte.

En effet, l'étude comparative effectuée dans le chapitre trois s'avère insuffisante pour choisir entre les technologies Big Data. Il est donc nécessaire de mener une étude expérimentale pour comparer les bibliothèques de Machine Learning disponibles dans Hadoop, Spark et Flink, en tenant compte des besoins spécifiques de notre sujet. Cette étude nous permettra de choisir l'outil le plus approprié pour notre projet.

# Bibliographie

- [1] <https://www.geeksforgeeks.org/>. Dernier accès le 2023-02-20.
- [2] The Apache Software Foundation. <https://www.apache.org/>. Dernier accès le 22-05-2023.
- [3] "colloque sur les objets et systèmes connectés 2023". <https://coc23.sciencesconf.org/>. "Dernier accès le 2023-06-21".
- [4] Comparaison entre nosqldatabase. <https://logz.io/blog/nosql-database-comparison/>. Dernier accès le 2023-01-15.
- [5] connected devices. <https://www.dreamstime.com/stock-illustration-cloud-computing-security-concept-smart-phones-vector-illustration-background-connected-devices-image62189291>. Dernier accès le 13-06-2023.
- [6] DB-Engines Ranking. <https://db-engines.com/en/ranking>. Dernier accès le 01-04-2023.
- [7] Documentation google colab. <https://research.google.com/colaboratory/faq.html>. Dernier accès le 08-06-2023.
- [8] General python. <https://docs.python.org/>. Dernier accès le 08-06-2023.
- [9] Harvard. <https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/YLWCSU>. Dernier accès le 08-06-2023.
- [10] Internet. <https://www.bercynumerique.finances.gouv.fr/une-minute-sur-internet-en-2021>. Dernier accès le 04-05-2023.
- [11] Iot. <https://yoomweb.com/blogue/hi-tec/statistiques-internet-objets-iot.html>. Dernier accès le 04-06-2023.



- 
- [12] MongoDB. <https://www.mongodb.com/>. Dernier accès le 30-04-2023.
- [13] National aeronautics and space administratio. <https://power.larc.nasa.gov/data-accessviewer/>. Dernier accès le 08-06-2023.
- [14] Neo4j. <https://neo4j.com/fr/>. Dernier accès le 01-04-2023.
- [15] Redis. <https://redis.io/>. Dernier accès le 01-04-2023.
- [16] researchandmarkets. <https://www.researchandmarkets.com/reports/5440538/smart-parking-systems-market-size-share-and>. Dernier accès le 2023-05-03.
- [17] Statistiques cisco. <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html>. Dernier accès le 2023-05-03.
- [18] Apache cassandra. *Website. Available online at <http://planetcassandra.org/what-is-apache-cassandra>*, 13, 2014.
- [19] Muhammad Zainal Abidi and Reza Pulungan. A systematic review of machine-vision-based smart parking systems. *Sci. J. Informatics*, 7(2) :213–227, 2020.
- [20] Veronika Abramova, Jorge Bernardino, and Pedro Furtado. Experimental evaluation of NoSQL databases. *International Journal of Database Management Systems*, 6(3) :1, 2014.
- [21] Eiman Al Nuaimi, Hind Al Neyadi, Nader Mohamed, and Jameela Al-Jaroodi. Applications of big data to smart cities. *Journal of Internet Services and Applications*, 6(1) :1–15, 2015.
- [22] Antonio Marcos Alberti, Mateus AS Santos, Ricardo Souza, Hirley Dayan Lourenço Da Silva, Jorge Roberto Carneiro, Vitor Alexandre Campos Figueiredo, and Joel JPC Rodrigues. Platforms for smart environments and future internet design : A survey. *IEEE Access*, 7 :165748–165778, 2019.
- [23] PMD Amarasooriya and MPPL Peiris. Implementation of smart parking system using image processing. *MPPL, Implementation of Smart Parking System Using Image Processing (June 10, 2023)*, 2023.
- [24] Deborah Anderson. Population and the environment—time for another contraception revolution. *N Engl J Med*, 381(5) :397–399, 2019.

- [25] Gennady Andrienko, Natalia Andrienko, Steven Drucker, Jean-Daniel Fekete, Danyel Fisher, Stavros Idreos, Tim Kraska, Guoliang Li, Kwan-Liu Ma, Jock Mackinlay, et al. Big data visualization and analytics : Future research challenges and emerging applications. In *BigVis 2020-3rd International Workshop on Big Data Visual Exploration and Analytics*, 2020.
- [26] Karim. Aoulad Abdelouarit. *Élaboration d'un outil basé sur le Big Data pour améliorer la recherche dans un environnement e-Learning*. PhD thesis, 11 2022.
- [27] Kamran Sattar Awaisi, Assad Abbas, Hasan Ali Khattak, Arsalan Ahmad, Mazhar Ali, and Abbas Khalid. Deep reinforcement learning approach towards a smart parking architecture. *Cluster Computing*, 26(1) :255–266, 2023.
- [28] Houria Benmehania and Latifa Bensakta. *L'intégration des Big Data dans le processus ETL (Extrat, Transform, Laod)*. PhD thesis, Université laarbi tebessi tebessa, 2016.
- [29] Ekaba Bisong. Google colaboratory. building machine learning and deep learning models on google cloud platform. *Apress, Berkeley, CA*, pages 59–64, 2019.
- [30] Ekaba Bisong. Tensorflow 2.0 and keras. *Building Machine Learning and Deep Learning Models on Google Cloud Platform : A Comprehensive Guide for Beginners*, pages 347–399, 2019.
- [31] Can Biyik, Ahmad Abareshi, Alexander Paz, Rosa Arce Ruiz, Rosaria Battarra, Christopher DF Rogers, and Carmen Lizarraga. Smart mobility adoption : A review of the literature. *Journal of Open Innovation : Technology, Market, and Complexity*, 7(2) :146, 2021.
- [32] Fabian Bock and Sergio Di Martino. On-street parking availaibility data in san francisco, from stationary sensors and high-mileage probe vehicles. *Data in brief*, 25 :104039, 2019.
- [33] Ahlem Boumergued and Sarah Nasri. *Conception et réalisation d'une application mobile pour un parking intelligent*. PhD thesis, Université Mohamed el-Bachir el-Ibrahimi Bordj Bou Arréridj, 2021.
- [34] Florent Brissaud, Dominique Charpentier, Anne Barros, and Christophe Bérenguer. Capteurs intelligents : Nouvelles technologies et nouvelles problématiques pour la sûreté de fonctionnement. In *Maîtrise des Risques et de Sûreté de Fonctionnement, Lambda-Mu 16*, pages 3A–2, 2008.

- [35] Hikmet Canlı and Si Toklu. Design and implementation of a prediction approach using big data and deep learning techniques for parking occupancy. *Arabian Journal for Science and Engineering*, pages 1–16, 2022.
- [36] Paris Carbone, Asterios Katsifodimos, Stephan Ewen, Volker Markl, Seif Haridi, and Kostas Tzoumas. Apache flink : Stream and batch processing in a single engine. *The Bulletin of the Technical Committee on Data Engineering*, 38(4), 2015.
- [37] Zeng Chao, Ma Changxi, Wang Ke, and Cui Zihao. Predicting vacant parking space availability : A dwt-bi-lstm model. *Physica A : Statistical Mechanics and its Applications*, 599 :127498, 2022.
- [38] Raden SB Cokro, Edi Yusuf Wirawan, Yopi Putra, Ayu Puspitarini, Gunawan Wang, and Emil Robert Kaburuan. Designing smart parking system through the use of iot and big data. *International Journal of Advanced Trends in Computer Science and Engineering*, 10(5), 2021.
- [39] Bollier David. *How Smart Growth Can Stop Sprawl : A Fledgling Citizen Movement Expands*. Essential Books, 1998.
- [40] Yajing Feng, Yingying Xu, Qian Hu, Sujatha Krishnamoorthy, and Zhenzhou Tang. Predicting vacant parking space availability zone-wisely : A hybrid deep learning approach. *Complex & Intelligent Systems*, 8(5) :4145–4161, 2022.
- [41] Andrea Gandini, Marco Gribaudo, William J Knottenbelt, Rasha Osman, and Pietro Piazzolla. Performance evaluation of NoSQL databases. In *Computer Performance Engineering : 11th European Workshop, EPEW 2014, Florence, Italy, September 11-12, 2014. Proceedings 11*, pages 16–29. Springer, 2014.
- [42] Xiaohu Ge, Song Tu, Guoqiang Mao, Cheng-Xiang Wang, and Tao Han. 5g ultra-dense cellular networks. *IEEE Wireless Communications*, 23(1) :72–79, 2016.
- [43] Maziar Goudarzi. Heterogeneous architectures for big data batch processing in MapReduce paradigm. *IEEE Transactions on Big Data*, 5(1) :18–33, 2019.
- [44] Reihaneh Hariri, Erik Fredericks, and Kate Bowers. Uncertainty in big data analytics : survey, opportunities, and challenges. *Journal of Big Data*, 6(1) :1–16, 2019.
- [45] Bhole Rahul Hiranman, Chapte Viresh, and Karve Abhijeet. A study of apache kafka in big data stream processing. In *2018 International Conference on Information , Communication, Engineering and Technology (ICICET)*, pages 1–3, 2018.

- [46] IBM. A smarter planet : The next leadership agenda. Council on Foreign Relations, 2008.
- [47] Prajakta Joshi. Evaluating the data mining technology in big data in the retail industry.
- [48] Jung and Inhwon. An iot-based smart parking management system. *International Journal of Computational Vision and Robotics*, 10(2) :122–132, 2020.
- [49] Avita Katal, Mohammad Wazid, and Rayan Goudar. Big data : issues, challenges, tools and good practices. In *2013 Sixth international conference on contemporary computing (IC3)*, pages 404–409. IEEE, 2013.
- [50] Karambir Kaur and Monika Sachdeva. Performance evaluation of newsql databases. In *2017 International Conference on Inventive Systems and Control (ICISC)*, pages 1–5. IEEE, 2017.
- [51] Navroop Kaur, Sandeep K Sood, and Prabal Verma. Cloud resource management using 3vs of internet of big data streams. *Computing*, 102 :1463–1485, 2020.
- [52] Basel Kayyali, David Knott, and Kuiken Steve Van. The big-data revolution in us health care : Accelerating value and innovation. *Mc Kinsey & Company*, 2(8) :1–13, 2013.
- [53] Sanam Kazi, Shirgaonkar Nuzhat, Ansari Nashrah, and Qureshi Rameeza. Smart parking system to reduce traffic congestion. In *2018 International Conference on Smart City and Emerging Technology (ICSCET)*, pages 1–4. IEEE, 2018.
- [54] Nawsher Khan, Mohammed Alsaqer, Habib Shah, Gran Badsha, Aftab Ahmad Abbasi, and Soulmaz Salehian. The 10 vs, issues and challenges of big data. In *Proceedings of the 2018 international conference on big data and education*, pages 52–56, 2018.
- [55] Ayca Kirimtat, Ondrej Krejcar, Attila Kertesz, and M Fatih Tasgetiren. Future trends and current state of smart city concepts : A survey. *IEEE access*, 8 :86448–86467, 2020.
- [56] Nicos Komninos. *Intelligent Cities : Innovation, Knowledge Systems, and Digital Spaces*. Routledge, 2002.
- [57] TV Kumar. Smart living for smart cities. *Smart Living for Smart Cities*, pages 3–70, 2020.

- [58] Rakesh Kumar Lenka, Rabindra Barik, Nihal Kumar Das, Kriti Agarwal, Debesh Mohanty, and Swati Vipsita. PSPS : An IoT based predictive smart parking system. In *2017 4th IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics (UPCON)*, pages 311–317. IEEE, 2017.
- [59] Sabina Leonelli. *La recherche scientifique à l'ère des Big Data : Cinq façons dont les Big Data nuisent à la science et comment la sauver*. Editions Mimésis, 2019.
- [60] Carson Leung. Big data visualization of association rules and frequent patterns. In *Encyclopedia of Data Science and Machine Learning*, pages 1284–1298. IGI Global, 2023.
- [61] Kevin YK Leung and Hiu Yan Lee. Implementing the smart city : Who has a say ? some insights from hong kong. *International Journal of Urban Sciences*, 27(sup1) :124–148, 2023.
- [62] Mariusz Luterek. Smart cities and citizen orientation : The growing importance of “smart people” in developing modern cities. In *Information Systems : 16th European, Mediterranean, and Middle Eastern Conference, EMCIS 2019, Dubai, United Arab Emirates, December 9–10, 2019, Proceedings 16*, pages 209–222, 2020.
- [63] Gina Paola Maestre-Góngora. Data visualization guide for smart city technologies. In *Electronic Governance with Emerging Technologies : First International Conference, EGETC 2022, Tampico, Mexico, September 12–14, 2022, Revised Selected Papers*, page 176. Springer Nature, 2023.
- [64] V Manoj. Comparative study of nosql document, column store databases and evaluation of cassandra. *International Journal of Database Management Systems*, 6(4) :11, 2014.
- [65] Houcine Matallah, Ghalem Belalem, and Karim Bouamrane. Evaluation of NoSQL databases : MongoDB, Cassandra, HBase, Redis, Couchbase, OrientDB. *International Journal of Software Science and Computational Intelligence (IJSSCI)*, 12(4) :71–91, 2020.
- [66] Sai Nikhil Reddy Mettupally and Vineetha Menon. A smart eco-system for parking detection using deep learning and big data analytics. In *2019 SoutheastCon*, pages 1–4. IEEE, 2019.
- [67] Umar Farooq Minhas, Rui Liu, Ashraf Aboulnaga, Kenneth Salem, Jonathan Ng, and Sean Robertson. Elastic scale-out for partition-based database systems. In *2012 IEEE*

- 28th International Conference on Data Engineering Workshops*, pages 281–288. IEEE, 2012.
- [68] William J Mitchell. *City of bits : space, place, and the infobahn*. MIT press, 1996.
- [69] ABM Moniruzzaman and Syed Akhter Hossain. NoSQL database : New era of databases for big data analytics-classification, characteristics and comparison. *arXiv preprint arXiv :1307.0191*, 2013.
- [70] Sadek Kahil Moustafa. *Visualisation interactive des données pour l'analyse et la recherche exploratoire dans un contexte big data*. PhD thesis, Université Echahid Cheikh Larbi-Tebessi-Tébessa, 2023.
- [71] Dewi Mutiara, Siti Yuniarti, and Bambang Pratama. Smart governance for smart city. In *IOP Conference Series : Earth and Environmental Science*, volume 126, page 012073. IOP Publishing, 2018.
- [72] Tenali Nagamani and Babu Gatram Rama Mohan. A systematic literature review and future perspectives for handling big data analytics in covid-19 diagnosis. *New Generation Computing*, pages 1–38, 2023.
- [73] Hamid Nasiri, Saeed Nasehi, and Maziar Goudarzi. Evaluation of distributed stream processing frameworks for IoT applications in Smart Cities. *Journal of Big Data*, 6 :1–24, 2019.
- [74] T Nasser and RS Tariq. Big data challenges. *J Comput Eng Inf Technol* 4 : 3. doi : [http://dx.doi.org/10.4172/2324.9307\(2\)](http://dx.doi.org/10.4172/2324.9307(2)), 9307(2), 2015.
- [75] Elham Nazari, Mohammad Hasan Shahriari, and Hamed Tabesh. Bigdata analysis in healthcare : apache hadoop, apache spark and apache flink. *Frontiers in Health Informatics*, 8(1) :14, 2019.
- [76] Sang Nguyen, Zoran Salcic, and Xuyun Zhang. Big data processing in fog-smart parking case study. In *2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom)*, pages 127–134. IEEE, 2018.
- [77] Florine Nicaise, Gauthier Corbeau, and Benoît Gailly. *Analyse rigoureuse d'un concept d'innovation sur base de la littérature scientifique et identification des principales impli-*

- cations managériales : le cas Smart City*. PhD thesis, Louvain School of Management, Université catholique de Louvain, 2021.
- [78] Ahmed Oussous, Fatima-Zahra Benjelloun, Ayoub Ait Lahcen, and Samir Belfkih. Comparison and classification of NoSQL databases for big data. *International Journal of Database Theory and Application*, 6(4.2013), 2013.
- [79] Debajyoti Pal, Tuul Triyason, and Praisan Padungweang. Big data in smart-cities : Current research and challenges. *Indonesian Journal of Electrical Engineering and Informatics (IJEI)*, 6(4) :351–360, 2018.
- [80] Debajyoti Pal, Tuul Triyason, and Praisan Padungweang. Big data in smart-cities : Current research and challenges. *Indonesian Journal of Electrical Engineering and Informatics (IJEI)*, 6(4) :351–360, 2018.
- [81] Hudan Pan, Ruijin Guo, Jie Zhu, Qi Wang, Yanmei Ju, Ying Xie, Yanfang Zheng, Zhifeng Wang, Ting Li, Zhongqiu Liu, et al. A gene catalogue of the sprague-dawley rat gut metagenome. *Gigascience*, 7(5) :giy055, 2018.
- [82] Antoine Picon. L'avènement de la ville intelligente. *Sociétés*, (2) :9–24, 2016.
- [83] Yelena Popova and Sergejs Popovs. Impact of smart economy on smart areas and mediation effect of national economy. *Sustainability*, 14(5), 2022.
- [84] Pavlo Radiuk, Olga Pavlova, Houda El Bouhissi, Volodymyr Avsiyevych, and Volodymyr Kovalenko. Convolutional neural network for parking slots detection. *CEUR-WS*, 2022.
- [85] Karen Rose, Scott Eldridge, and Lyman Chapin. The internet of things : An overview. *The internet society (ISOC)*, 80 :1–50, 2015.
- [86] Kishwara Sadia, Rehnuma Reza, Albina Alam, and Muhammad Arifur Rahman. Car parking availability prediction : A comparative study of lstm and random forest regression approaches. *International Journal of Automation, Artificial Intelligence and Machine Learning*, 2(1) :16–29, 2021.
- [87] Laurie A Schintler and Connie L McNeely. *Encyclopedia of big data*. Springer, 2022.
- [88] Abderrahim Sebri. Cours framework big data. 2022.

- [89] Sheela Shanthakumari, P Selvarani, J Senthil Murugan, WT Chembian, M Mithun Kumar, M Karthikeyan, and M Govindaraj. A study on smart parking space allocator and parking management using opencv. *Mathematical Statistician and Engineering Applications*, 72(1) :98–104, 2023.
- [90] Sugam Sharma. Expanded cloud plumes hiding big data ecosystem. *Future Generation Computer Systems*, 59 :63–92, 2016.
- [91] Dilpreet Singh and Chandan Reddy. A survey on platforms for big data analytics. *Journal of Big Data*, 2, 2014.
- [92] Dilpreet Singh and Chandan Reddy. A survey on platforms for big data analytics. *Journal of big data*, 2(1) :1–20, 2015.
- [93] Tarana Singh, Arun Solanki, Sanjay Kumar Sharma, Anand Nayyar, and Anand Paul. A decade review on smart cities : Paradigms, challenges and opportunities. *IEEE Access*, 2022.
- [94] Smart Nation Programme Office. Smart nation singapore, 2016.
- [95] Ferhane Souhayla. *l'internet des objets révolutionne notre vie quotidienne : application pour une maison intelligent*. PhD thesis, Universite laarbi tebessi tebessa, 2021.
- [96] Christof Strauch and Walter Kriha. Nosql databases. *Lecture Notes, Stuttgart Media University*, 20(24) :79, 2011.
- [97] Bühnová Barbora Stuchlíková Kristýna Michaela, Faldík Oldřich and Trenz Oldřich. Evaluation of smart city concept characteristics. In *European Scientific Conference of Doctoral Students*, page 193, 2020.
- [98] Nicoleta Tantalaki, Stavros Souravlas, and Manos Roumeliotis. A review on big data real-time stream processing and its scheduling techniques. *International Journal of Parallel, Emergent and Distributed Systems*, 35(5) :571–601, 2020.
- [99] Pillai J. Tepaniuk J. and Bak-Jensen B. Battery energy storage management for smart residential buildings. In *Proceedings of the 53rd International Universities Power Engineering Conference (UPEC)*, pages 1–6. IEEE, 2018.
- [100] Narina Thakur, Eshanika Bhattacharjee, Rachna Jain, Biswaranjan Acharya, and Yu-Chen Hu. Deep learning-based parking occupancy detection framework using resnet and vgg-16. *Multimedia Tools and Applications*, pages 1–24, 2023.



- 
- [101] Faheem Ullah, Shagun Dhingra, Xiaoyu Xia, and Ali Babar. Evaluation of distributed data processing frameworks in hybrid clouds. *arXiv preprint arXiv :2201.01948*, 2022.
- [102] Nations United. World population prospects 2019 : Highlights, 2019.
- [103] Han Wu, Zhihao Shang, and Katinka Wolter. Learning to reliably deliver streaming data with apache kafka. In *2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 564–571, 2020.
- [104] Li Xiangdong, Cen Yuefeng, Cen Gang, and Xu Zengwei. Prédiction des places de stationnement disponibles à court terme à l’aide du modèle lstm. In *2019 14th International Conference on Computer Science & Education (ICCSE)*, pages 631–635. IEEE.
- [105] Ying Xue. An overview of overfitting and its solutions. *Journal of Physics : Conference Series*, 1168(2) :022022, 2019.
- [106] Cheng Yang and Xiong ShiYONG. Implanting big data technology into the construction of “smart parking”. In *2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, pages 310–313. IEEE, 2020.
- [107] Shuguan Yang, Wei Ma, Xidong Pi, and Sean Qian. Une approche d’apprentissage profond pour la prédiction en temps réel de l’occupation du stationnement dans les réseaux de transport intégrant de multiples sources de données spatio-temporelles. *Transportation Research Part C : Emerging Technologies*, 107 :248–265.
- [108] Shuguan Yang, Wei Ma, Xidong Pi, and Sean Qian. A deep learning approach to real-time parking occupancy prediction in transportation networks incorporating multiple spatio-temporal data sources. *Transportation Research Part C : Emerging Technologies*, 107 :248–265, 2019.
- [109] Tan Yigitcanlar, Md Kamruzzaman, Marcus Foth, Jamile Sabatini-Marques, Eduardo Da Costa, and Giuseppe Ioppolo. Can cities become smart without being sustainable? a systematic review of the literature. *Sustainable cities and society*, 45 :348–365, 2019.
- [110] Seungho Yoo. Songdo : The hype and decline of world’s first smart city. *Sustainable cities in Asia*, pages 146–160, 2017.

- 
- [111] Anam Zahid, Rahat Masood, and Muhammad Awais Shibli. Security of sharded NoSQL databases : A comparative analysis. In *2014 conference on information assurance and cyber security (CIACS)*, pages 1–8. IEEE, 2014.
- [112] Fotios Zantalis, Grigorios Koulouras, Sotiris Karabetsos, and Dionisis Kandris. A review of machine learning and iot in smart transportation. *Future Internet*, 11(4) :94, 2019.
- [113] Chao Zeng, Changxi Ma, Ke Wang, and Zihao Cui. Parking occupancy prediction method based on multi factors and stacked gru-lstm. *IEEE Access*, 10 :47361–47370, 2022.
- [114] Yanxu Zheng, Sutharshan Rajasegarar, and Christopher Leckie. Prédiction de disponibilité des parkings équipés de capteurs dans les villes intelligentes. In *2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, pages 1–6. IEEE.
- [115] Ivana Šemanjski, Sadko Mandžuka, and Sidharta Gautama. Smart mobility. In *2018 International Symposium ELMAR*, pages 63–66, 2018.