



HAL
open science

Prédiction des besoins pour la gestion de serveurs mobiles en périphérie

Nina Santi

► **To cite this version:**

Nina Santi. Prédiction des besoins pour la gestion de serveurs mobiles en périphérie. Apprentissage [cs.LG]. Université de Lille, 2023. Français. NNT : 2023ULILB050 . tel-04548817v2

HAL Id: tel-04548817

<https://theses.hal.science/tel-04548817v2>

Submitted on 22 May 2024

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

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

UNIVERSITÉ DE LILLE

ÉCOLE DOCTORALE MADIS 631

UNITÉ DE RECHERCHE INRIA LILLE - NORD EUROPE

Thèse préparée par **Nina Santi**
et a été soutenue publiquement le **19 décembre 2023**
en vue de l'obtention du grade de **Docteur en Informatique**
Discipline **Informatique et applications**

Prédiction des besoins pour la gestion de serveurs mobiles en périphérie

Devant le jury composé de :

Saadi Boudjit

Maître de conférences, Université Sorbonne
Paris Nord (USPN)

Rapporteur

Hervé Rivano

Professeur des Universités, Institut national
des sciences appliquées (INSA), Lyon

Rapporteur

Christelle Caillouet

Maîtresse de conférences, Inria Côte d'Azur

Examinatrice

Ahmed Rahmani

Professeur des Universités, École Centrale
de Lille

Examineur et Président du Jury

Nathalie Mitton

Directrice de recherche, Inria Lille - Nord
Europe

Directrice de thèse

Résumé

L'informatique en périphérie est un paradigme émergent au sein de l'Internet des Objets (IoT) et complémentaire à l'informatique en nuage. Ce paradigme propose l'implémentation de serveurs de calcul situés à proximité des utilisateurs, réduisant ainsi la pression et les coûts de l'infrastructure réseau locale. La proximité avec les utilisateurs suscite de nouveaux cas d'utilisation, tels que le déploiement de serveurs mobiles embarqués sur des drones ou des robots, offrant une alternative moins coûteuse, plus éco-énergétique et flexible par rapport aux infrastructures fixes lors d'événements ponctuels ou exceptionnels. Cependant, cette approche soulève également de nouveaux enjeux pour le déploiement et l'allocation de ressources en temps et en espace, souvent dépendants de la batterie.

Dans le cadre de cette thèse, nous proposons des outils et des algorithmes de prédiction pour la prise de décision concernant l'allocation de ressources fixes et mobiles, à la fois en termes de temps et d'espace, au sein d'environnements dynamiques. Nous mettons à disposition des jeux de données riches et reproductibles qui reflètent l'hétérogénéité inhérente aux applications de l'Internet des Objets (IoT), tout en présentant un taux de contention et d'interférence élevé. Pour cela, nous utilisons le FIT-IoT Lab, un banc d'essai ouvert dédié à l'IoT, et nous mettons l'ensemble du code à disposition de manière ouverte. De plus, nous avons développé un outil permettant de générer de manière automatisée et reproductible des traces de l'IoT. Nous exploitons ces jeux de données pour entraîner des algorithmes d'apprentissage automatique basés sur des techniques de régression afin de les évaluer dans leur capacité à prédire le débit des applications de l'IoT. Dans une démarche similaire, nous avons également entraîné et analysé un réseau neuronal de type transformateur temporel pour prédire plusieurs métriques de la Qualité de Service (QoS). Afin de tenir compte de la mobilité des ressources, nous générons des traces de l'IoT intégrant des points d'accès mobiles embarqués sur des robots TurtleBot. Ces traces, qui intègrent la mobilité, sont utilisées pour valider et tester un framework d'apprentissage fédéré reposant sur des transformateurs temporels parcimonieux. Enfin, nous proposons un algorithme décentralisé de prédiction de la densité de la population humaine par régions, basé sur l'utilisation d'un filtre à particules. Nous testons et validons cet algorithme à l'aide du simulateur Webots dans un contexte de serveurs embarqués sur des robots, et du simulateur ns-3 pour la partie réseaux.

Abstract

Multi-access Edge computing is an emerging paradigm within the Internet of Things (IoT) that complements Cloud computing. This paradigm proposes the implementation of computing servers located close to users, reducing the pressure and costs of local network infrastructure. This proximity to users is giving rise to new use cases, such as the deployment of mobile servers mounted on drones or robots, offering a cheaper, more energy-efficient and flexible alternative to fixed infrastructures for one-off or exceptional events. However, this approach also raises new challenges for the deployment and allocation of resources in terms of time and space, which are often battery-dependent. In this thesis, we propose predictive tools and algorithms for making decisions about the allocation of fixed and mobile resources, in terms of both time and space, within dynamic environments. We provide rich and reproducible datasets that reflect the heterogeneity inherent in Internet of Things (IoT) applications, while exhibiting a high rate of contention and interference. To achieve this, we are using the FIT-IoT Lab, an open testbed dedicated to the IoT, and we are making all the code available in an open manner. In addition, we have developed a tool for generating IoT traces in an automated and reproducible way. We use these datasets to train machine learning algorithms based on regression techniques to evaluate their ability to predict the throughput of IoT applications. In a similar approach, we have also trained and analysed a neural network of the temporal transformer type to predict several Quality of Service (QoS) metrics. In order to take into account the mobility of resources, we are generating IoT traces integrating mobile access points embedded in TurtleBot robots. These traces, which incorporate mobility, are used to validate and test a federated learning framework based on parsimonious temporal transformers. Finally, we propose a decentralised algorithm for predicting human population density by region, based on the use of a particle filter. We test and validate this algorithm using the Webots simulator in the context of servers embedded in robots, and the ns-3 simulator for the network part.

Table des matières

Table des matières	iv
1 Introduction et contexte	1
1.1 Des réseaux sans fil à l'Internet des Objets	1
1.1.1 Réseaux sans-fil	1
1.1.2 Internet des Objets (IoT)	1
1.2 Informatique en périphérie multi-accès	3
1.3 MEC: de nouveaux défis	4
1.3.1 Ressources fixes	4
1.3.2 Ressources mobiles	5
1.4 Projet européen : DRUID-NET	6
1.4.1 Présentation du projet et des équipes	6
1.4.2 Objectifs du projet	6
1.5 Problématiques	7
1.6 Contributions et organisation de cette thèse	8
1.6.1 Contributions de cette thèse	8
1.6.2 Organisation et contenu du document	9
1.7 Liste des publications	10
2 Génération de traces	11
2.1 Jeux de données de l'IoT pour la prédiction	11
2.1.1 Données IoT: une demande croissante	11
2.1.2 Nos besoins	11
2.1.3 Pourquoi générer nos propres données ?	12
2.1.4 Science ouverte et reproductibilité	13
2.2 Génération de traces dynamiques et hétérogènes de l'IoT	14
2.2.1 Caractérisation des applications	14
2.2.2 Dispositif expérimental	15
2.2.3 Résultats	16
2.2.4 Conclusion et discussion	19
2.3 Génération automatisée et reproductible de données	20
2.3.1 Sisyphe: Outil pour la génération de données automatisée et reproductible	20
2.3.2 Données d'applications IoT enrichies	24
2.3.3 Résultats	26
2.3.4 Analyse statistique des résultats	26
2.3.5 Conclusion	29
2.4 Génération de traces intégrant des ressources mobiles	30
2.4.1 Caractérisation des applications Edge	30
2.4.2 Dispositif expérimental	31

2.4.3	Paramètres des expériences	32
2.4.4	Résultats	33
2.4.5	Conclusion	33
2.5	Conclusion	34
3	Prédiction de la Qualité de Service	35
3.1	Pourquoi prédire la Qualité de Service ?	35
3.2	Prédiction du débit dans des environnements dynamiques	35
3.2.1	Pourquoi et comment prédire le débit ?	35
3.2.2	Travaux déjà existants sur la prédiction de débit	36
3.2.3	Données	37
3.2.4	Apprentissage automatique pour la prédiction du débit	37
3.2.5	Évaluation des performances des modèles	39
3.2.6	Et maintenant, les résultats !	41
3.2.7	Conclusion	43
3.3	Analyse de modèles d'apprentissage automatique pour la prédiction multi-métriques	43
3.3.1	Une caractéristique c'est bien, plusieurs c'est mieux !	43
3.3.2	Travaux existants sur la prédiction de la QoS avec des réseaux de neurons profonds	43
3.3.3	Un court détour par les fenêtres glissantes	44
3.3.4	Transformateurs temporels	44
3.3.5	Configuration de l'expérimentation	45
3.3.6	Résultats : mode univarié	46
3.3.7	Résultats : mode multivarié	49
3.3.8	Mode univarié ou multivarié ?	50
3.3.9	Conclusion	50
3.4	Conclusion	50
4	Prédiction de la densité humaine	53
4.1	Prédiction de la densité humaine	53
4.1.1	Pourquoi prédire la mobilité humaine ?	53
4.1.2	Mais est-ce que la mobilité humaine est prédictible ?	53
4.1.3	Prédiction de la mobilité pour le MEC	54
4.2	État de l'art	54
4.2.1	Prédiction de la mobilité à l'échelle de la ville	54
4.2.2	Prédiction de la mobilité dans les réseaux mobiles	55
4.2.3	Le meilleur des deux mondes	56
4.3	Contexte technique	56
4.3.1	Les filtres à particules	56
4.3.2	Filtre à particules pour la prédiction de la densité humaine	57
4.4	Contribution	59
4.4.1	Simulateur de Mobilité Humaine	59
4.4.2	Collecte de données d'observation	59

4.4.3	Diffusion des données	60
4.5	Configuration des expériences	62
4.5.1	Expérience sur la communication	62
4.5.2	Expérience sur la partie calcul du filtre	62
4.6	Résultats	65
4.6.1	Protocole de comméragé	65
4.6.2	Prévisions du filtre à particule	71
4.7	Conclusion	73
5	Étude sur la gestion des ressources	75
5.1	Gestion des ressources MEC : allocation et déploiement	75
5.2	Allocation de ressources	75
5.2.1	Modélisations des tâches	76
5.2.2	Quelles ressources peut-on allouer ?	76
5.2.3	Allouer des ressources, mais pour quels objectifs ?	78
5.2.4	Méthodes d'allocation des ressources	80
5.3	Déploiement de ressources mobiles	84
5.3.1	Comment modéliser un système avec des ressources mobiles	85
5.3.2	Méthodes pour le déploiement	85
5.4	Conclusion	87
6	Conclusion générale	89
6.1	Récapitulatif des travaux	89
6.2	Discussion des résultats	90
6.3	Perspectives et travaux futurs	92
6.3.1	Travaux futurs	92
6.3.2	Les perspectives de recherche dans les réseaux MEC avec unités mobiles	93
	Bibliographie	95

Table des figures

1.1	Le modèle OSI [3]. Nos travaux se situent au niveau de la couche 3 des réseaux.	1
1.2	Informatique en périphérie multi-accès.	3
1.3	Informatique en périphérie multi-accès avec ressources mobiles.	5
2.1	Cartes M3 du FIT IoT-LAB [69].	16
2.2	Résultats bruts de l'expérimentation.	16
2.3	PDR et latence pour l'application d'urgence.	17
2.4	PDR et latence pour l'application d'éclairage.	18
2.5	Vue d'ensemble de l'architecture de Sisyphé. Les éléments en gras sont utilisés dans ce chapitre.	23
2.6	Architecture de l'expérimentation.	24
2.7	Exemple simple de manipulation des données produites.	26
2.8	Graphiques de distribution de probabilité des données de qualité de service pour toutes les applications IoT.	27
2.10	Architecture de l'expérimentation avec traces mobiles.	31
2.11	Topologie de l'expérimentation pour la génération des données mobiles.	32
2.9	Turtlebot 2 [81].	32
2.12	Exemples de résultats pour les données avec points d'accès mobiles.	33
3.1	Représentation graphique de la MAE.	39
3.2	Représentation graphique de la MSE.	39
3.3	Comparaison des résultats MAE pour tous les algorithmes.	41
3.4	Comparaison des résultats MSE pour tous les algorithmes.	41
3.5	Comparaison du temps de prédiction pour tous les algorithmes.	41
3.6	Comparaison du temps d'entraînement pour tous les algorithmes.	41
3.7	Illustration des fenêtres glissantes.	44
3.8	Architecture d'un transformateur temporel.	45
3.9	Illustration des têtes d'attention.	45
3.10	MAE de la prédiction univariée du débit pour tous les ensembles de données.	46
3.11	MAE de la prédiction multivariée du débit pour tous les ensembles de données.	46
3.12	MAE de la prédiction PDR multivariée pour tous les ensembles de données.	46
3.13	MSE de la prédiction univariée du débit pour tous les ensembles de données.	48
4.1	Modèle espace-état.	56
4.2	Algorithme du filtre à particules.	57
4.3	Expérimentation pour les performances de prédiction.	62
4.4	Comparaison du nombre de gossips envoyés par intervalles de mise à jour pour le mode <i>push</i> .	66
4.5	Comparaison du nombre de gossips envoyés par intervalles de requêtes pour le mode <i>pull</i> .	67

Liste des tableaux

2.1	Paramètres des simulations	14
2.2	Paramètres de l'expérimentation.	24
2.3	Caractéristiques brutes des données produites.	26
2.4	Caractéristiques statistiques des ensembles de données sur le débit et la latence pour toutes les applications IoT	27
2.5	Caractéristiques statistiques des ensembles de données sur le PDR et le PLR pour toutes les applications IoT	27
2.6	Paramètres de l'expérimentation pour la génération de données mobiles. . .	30
2.7	Configurations de réseaux avec différentes complexités	31
3.1	Mesures de performance de tous les algorithmes pour les applications IoT. .	42
3.2	Hyperparamètres utilisés dans toutes les méthodes pour la prédiction univariée du débit dans tous les ensembles de données.	45
3.3	Résultats des prévisions univariées pour le débit, les meilleurs résultats sont mis en évidence en gras.	46
3.4	Résultats des prévisions univariées pour le PDR, les meilleurs résultats sont mis en évidence en gras.	47
3.5	Résultats des prévisions univariées pour le PLR, les meilleurs résultats sont mis en évidence en gras.	47
3.6	Résultats des prévisions univariées pour la latence, les meilleurs résultats sont en gras.	47
3.7	Résultats des prévisions multivariées pour le débit, les meilleurs résultats sont mis en évidence en gras.	47
3.8	Résultats des prévisions multivariées pour le PDR, les meilleurs résultats sont mis en évidence en gras.	47
3.9	Résultats des prévisions multivariées pour le DPP, les meilleurs résultats sont mis en évidence en gras.	47
3.10	Résultats des prévisions multivariées pour la latence, les meilleurs résultats sont mis en évidence en gras.	48
4.1	Paramètres du filtre à particules.	64
4.2	Performances par le nombre de gossip envoyés et interval de mise à jour pour le mode <i>push</i>	65
4.3	Performances par le nombre de gossip envoyés et interval de requête pour le mode <i>pull</i>	68
4.4	Données de performance par mode.	69
4.5	Données de performance par nombre de noeuds en mode hybride.	70
4.6	Données de performance par nombre de noeuds pour le <i>flooding</i>	70
4.7	Performances selon l'intervalle d'observation. Les résultats pour la moyenne de la RMSE sont en gras.	72
4.8	Performances des prévisions des modèles.	72

1.1 Des réseaux sans fil à l'Internet des Objets

1.1.1 Réseaux sans-fil

Les réseaux sans fil ont révolutionné notre monde et font désormais partie intégrante de notre quotidien. En effet, on utilise ce type de réseaux, par exemple, lorsque nous nous connectons en Wi-Fi depuis notre ordinateur ou que nous utilisons un navigateur depuis notre smartphone. Il existe également des usages tout aussi fréquents, mais moins visibles, tels que dans le domaine de la santé, ainsi que des usages qui feront partie du monde de demain, comme les voitures connectées. On appelle ces réseaux des réseaux informatiques, c'est-à-dire un ensemble d'équipements informatiques qui s'échangent des informations entre eux sans être connectés par des câbles physiques. Les appareils utilisent principalement des ondes radio, en utilisant l'amplitude, la fréquence ou la phase pour transmettre les informations. Les propriétés de ces ondes sont utilisées de manière similaire à la communication en morse avec de la lumière en allumant et éteignant la lampe rapidement. Cependant, les ondes radio ne sont pas le seul *support physique* disponible pour échanger de l'information, on peut également utiliser l'infrarouge [1] ou encore la lumière visible [2]. Une fois que l'on connaît le moyen physique de communiquer entre les appareils, il est nécessaire que les appareils se comprennent entre eux et donc parlent le même langage. Comme pour le morse, il faut un alphabet commun. C'est pourquoi le modèle OSI [3] a été créé, il divise la communication de bout en bout des appareils en sept couches distinctes et indépendantes. Ce modèle a tout d'abord été créé pour les réseaux filaires, mais il est également utilisé pour les réseaux sans fil. Cela permet de simplifier le développement, la maintenance et l'interopérabilité des réseaux en concentrant les efforts sur des fonctions spécifiques. Dans cette thèse, nos travaux se situent au niveau de la couche 3 des réseaux.

1.1.2 Internet des Objets (IoT)

Parmi les réseaux sans fil, un type de réseau bien particulier ne cesse de se développer depuis ces dernières années : l'Internet des Objets (IoT). On pourrait donner comme définition générale que l'IoT est un ensemble d'objets interconnectés mondialement et configurables avec des protocoles de communications standardisés [4]. C'est donc un *Internet* composé d'*objets*. Cependant, l'IoT a connu de nombreux changements et a intégré différentes technologies au fil de son évolution, et encore aujourd'hui, il connaît une rapide évolution. À ses débuts, l'IoT était défini comme des objets connectés interopérables et identifiables de manière unique grâce à la technologie d'identification par radiofréquence (RFID) [5]. Désormais, l'Internet des objets (IoT) englobe les réseaux de capteurs sans fil (WSNs) [6] et les réseaux de capteurs sans fil rechar-

1.1	Des réseaux sans fil à l'Internet des Objets . . .	1
1.1.1	Réseaux sans-fil	1
1.1.2	Internet des Objets (IoT)	1
1.2	Informatique en périphérie multiaccès	3
1.3	MEC: de nouveaux défis .	4
1.3.1	Ressources fixes	4
1.3.2	Ressources mobiles	5
1.4	Projet européen : DRUID-NET	6
1.4.1	Présentation du projet et des équipes	6
1.4.2	Objectifs du projet	6
1.5	Problématiques	7
1.6	Contributions et organisation de cette thèse	8
1.6.1	Contributions de cette thèse	8
1.6.2	Organisation et contenu du document	9
1.7	Liste des publications . .	10



FIGURE 1.1 – Le modèle OSI [3]. Nos travaux se situent au niveau de la couche 3 des réseaux.

[3] : International Telecommunication UNION. *ITU-T Recommendation X.200 : Information Technology - Open Systems Interconnection - Basic Reference Model : The Basic Model*. Recommendation X.200. ITU-T, 1994

[4]: Li Da XU, Wu He et Shancang Li. « Internet of Things in Industries : A Survey ». In : *IEEE Transactions on Industrial Informatics* 10.4 (2014)

[5]: Kevin ASHTON. « Internet of Things ». In : *RFID Journal* (juin 2009)

[6]: I.F. AKYILDIZ, W. SU, Y. SANKARASUBRAMANIAM et E. CAYIRCI. « Wireless sensor networks : a survey ». In : *Computer Networks* 38.4 (2002)

[7]: Shibo HE, Jiming CHEN, Fachang JIANG, David K.Y. YAU, Guoliang XING et Youxian SUN. « Energy provisioning in wireless rechargeable sensor networks ». In : *2011 Proceedings IEEE INFOCOM*. 2011

[8]: Khan MUHAMMAD, Rafik HAMZA, Jamil AHMAD, Jaime LLORET, Haoxiang WANG et Sung Wook BAIK. « Secure Surveillance Framework for IoT Systems Using Probabilistic Image Encryption ». In : *IEEE Transactions on Industrial Informatics* 14 (2018)

[9]: Muhammad AYAZ, Mohammad AMMAD-UDDIN, Zubair SHARIF, Ali MANSOUR et El-Hadi M. AGGOUNE. « Internet-of-Things (IoT)-Based Smart Agriculture : Toward Making the Fields Talk ». In : *IEEE Access* 7 (2019)

[10]: Hugh BOYES, Bil HALLAQ, Joe CUNNINGHAM et Tim WATSON. « The industrial internet of things (IIoT) : An analysis framework ». In : *Computers in Industry* 101 (2018)

[11]: Ammar GHARAIBEH, Mohammad A. SALAHUDDIN, Sayed Jahed HUSSINI, Abdallah KHREISHAH, Issa KHALIL, Mohsen GUIZANI et Ala AL-FUQAHA. « Smart Cities : A Survey on Data Management, Security, and Enabling Technologies ». In : *IEEE Communications Surveys Tutorials* 19.4 (2017)

[16]: Pradipta GHOSH, Andrea GASPARRI, Jiong JIN et Bhaskar KRISHNAMACHARI. « Robotic Wireless Sensor Networks ». In : *Mission-Oriented Sensor Networks and Systems*. 2017

[17]: L. ROMEO, A. PETITTI, R. COLELLA, G. VALECCE, P. BOCCADORO, A. MILELLA et L.A. GRIECO. « Automated Deployment of IoT Networks in Outdoor Scenarios using an Unmanned Ground Vehicle ». In : *2020 IEEE International Conference on Industrial Technology (ICIT)*. 2020

[18]: Rajesh M, Anu GEORGE et Sudarshan T.S.B. « Energy efficient deployment of Wireless Sensor Network by multiple mobile robots ». In : *2015 International Conference on Computing and Network Communications (CoCoNet)*. 2015

geables (WRSNs) [7] utilisés pour mesurer et contrôler l'environnement, ainsi qu'une variété de capteurs et d'appareils hétérogènes, tels que les caméras de surveillance [8]. De plus, de nombreuses applications ont vu le jour, notamment dans l'agriculture [9], l'industrie (IIoT) [10] et les villes intelligentes [11]. Ces nouvelles applications doivent relever les défis inhérents à l'IoT. Notamment, les appareils IoT peuvent être soumis à des contraintes importantes en termes d'énergie, de puissance de calcul et de communication. En effet, certains de ces objets fonctionnent sur batterie, ce qui les oblige à économiser leur énergie lors des communications. De même, ils doivent optimiser leurs calculs pour préserver leur autonomie et assurer leur mission. De plus, les communications sans fil peuvent être peu fiables et sujettes à des pertes d'informations. Pour compliquer encore davantage les choses, les objets IoT présentent une grande hétérogénéité, ce qui signifie que ces contraintes varient d'un objet à l'autre. De nombreuses technologies, prenant en compte ces aspects, ont fait l'objet de recherches intensives et sont désormais intégrées à l'IoT [12-15]. L'utilisation de robots au sein des réseaux de capteurs sans fil (WSNs) est intéressante pour apporter de la mobilité et de la flexibilité aux capteurs sans fil. Cette convergence entre la robotique et les WSNs est désignée sous le terme de "Réseau Robotique de Capteurs Sans Fil" (RWSN ou WRSN) [16]. Ces deux domaines présentent un intérêt car ils permettent de combler les lacunes de l'un en exploitant les avantages de l'autre. En effet, la mobilité des robots peut être mise à profit pour repositionner les capteurs [17, 18], collecter des données, où les robots sont appelés des *mobile sinks* [19, 20], voire même pour recharger les batteries des capteurs [21]. De leur côté, les réseaux de capteurs peuvent contribuer en aidant les robots à se localiser ou à cartographier leur environnement, par exemple [22]. La mise en place de RWSNs exige des compétences et une expertise à la fois en réseaux et en robotique, soulevant des défis stimulants.

L'informatique en nuage

Le NIST [23] (Institut national des normes et de la technologie des États-Unis) définit l'informatique en nuage (*Cloud Computing* en anglais) comme un modèle qui met à disposition un *pool* partagé de ressources informatiques de manière omniprésente, pratique et à la demande, qui sont configurables et rapidement approvisionnées, puis libérées avec un minimum d'effort de gestion ou d'interaction avec le fournisseur de services. En termes plus simples, le Cloud offre un accès à des serveurs distants, souvent accessibles via Internet. Lorsqu'il est associé à l'IoT, le Cloud permet de créer des applications IoT puissantes. En effet, les objets IoT sont généralement contraints en termes de ressources de calcul et d'énergie, c'est pourquoi ils peuvent recourir au Cloud pour effectuer des calculs complexes et gourmands en puissance de calcul, ainsi que pour stocker massivement des données en vue de proposer des services de plus en plus intelligents [24, 25]. Le Cloud présente de nombreux avantages, tels que la réduction des coûts et la scalabilité, ce qui en fait une technologie de choix pour l'IoT [26].

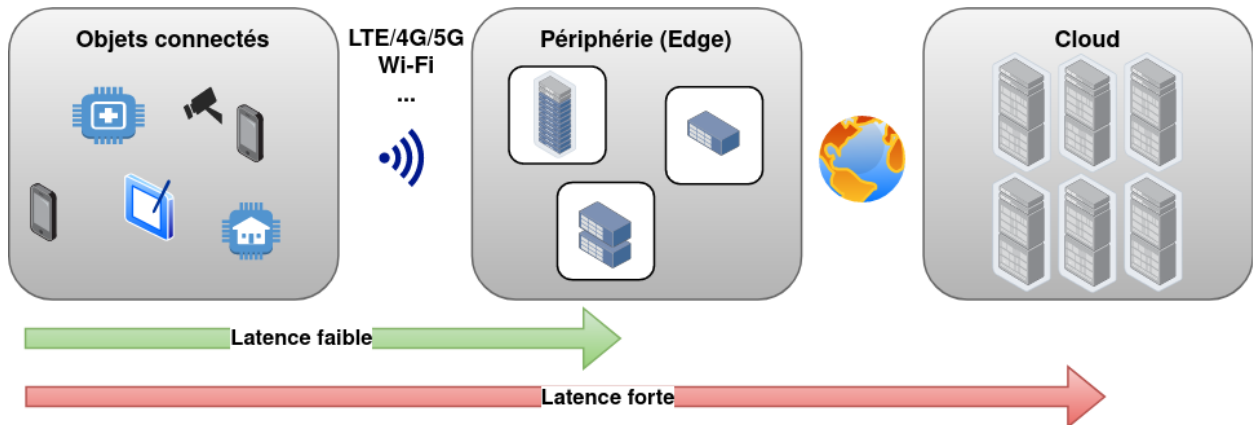


FIGURE 1.2 – Informatique en périphérie multi-accès.

1.2 Informatique en périphérie multi-accès

Des ressources distantes aux ressources en périphérie

Cependant, l'informatique en nuage est limitée par son emplacement distant par rapport aux utilisateurs finaux et aux appareils, ce qui entraîne une latence élevée et des retards. Ce problème est exacerbé par la lourde génération de données à partir des appareils IoT, qui surchargent le réseau et créent éventuellement des goulets d'étranglement lorsque les tâches ne sont pas traitées suffisamment rapidement [27]. Effectivement, les fameux chiffres du rapport annuel de Cisco sur l'Internet montrent qu'au niveau mondial, le nombre moyen d'appareils et de connexions par habitant passera de 2,4 en 2018 à 3,6 d'ici 2023. De plus, les connexions machine-à-machine (M2M) seront la catégorie d'appareils et de connexions connaissant la croissance la plus rapide, avec une multiplication par près de 2,4 [28]. Cela pose problème pour les applications mobiles ou IoT qui ont des contraintes de temps strictes et des exigences élevées en matière de qualité de service (QoS) et de qualité d'expérience (QoE), telles que la réalité augmentée ou virtuelle, la reconnaissance d'images, la surveillance, etc. Ces limitations du Cloud ont incité l'industrie et la recherche à envisager des solutions pour déplacer les capacités de l'informatique en nuage vers le bord du réseau. Un nouveau paradigme a donc vu le jour, appelé l'informatique en périphérie (*Edge Computing*), où les dispositifs connectés envoient leurs tâches à des nœuds de calcul situés à la périphérie du réseau, c'est-à-dire à proximité des utilisateurs ou des objets qui produisent des données [27]. Cette proximité présente des avantages par rapport au cloud, notamment : (i) la réduction de la latence [27, 29, 30], (ii) l'économie d'énergie [27, 29], (iii) une meilleure protection de la vie privée [29, 30] et (iv) la prise en compte de l'emplacement et du contexte [27, 29]. Ces avantages sont essentiels pour répondre aux exigences strictes des applications en temps réel, en particulier dans le cas des applications critiques pour la mission et des applications soumises à des contraintes temporelles, où le temps est une ressource importante [30].

[19]: Xuxun LIU, Tian WANG, Weijia JIA, Anfeng LIU et Kaikai CHI. « Quick Convex Hull-Based Rendezvous Planning for Delay-Harsh Mobile Data Gathering in Disjoint Sensor Networks ». In : *IEEE Transactions on Systems, Man, and Cybernetics : Systems* 51.6 (2021)

[20]: Bala Gangadhara GUTAM, Praveen Kumar DONTA, Chandra Sekhara Rao ANNAVARAPU et Yu-Chen HU. « Optimal rendezvous points selection and mobile sink trajectory construction for data collection in WSNs ». In : *Journal of Ambient Intelligence and Humanized Computing* 14 (2021)

[21]: Xiaoguo YE et Weifa LIANG. « Charging utility maximization in wireless rechargeable sensor networks ». In : *Wireless Networks* 23.7 (oct. 2017)

[22]: Dionisis KANDRIS, Christos Theodoros NAKAS, Dimitrios VOMVAS et Grigorios E. KOULOURAS. « Applications of Wireless Sensor Networks : An Up-to-Date Survey ». In : *Applied System Innovation* (2020)

[23]: Peter MELL et Timothy GRANCE. *The NIST Definition of Cloud Computing*. Rapp. tech. National Institute of Standards et Technology, 2011

[24]: Motahareh NAZARI JAHANTIGH, Amir MASOUD RAHMANI, Nima JAFARI NAVIMIROUR et Ali REZAEI. « Integration of Internet of Things and cloud computing : a systematic survey ». In : *IET Communications* 14.2 (2020). eprint : <https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/iet-com.2019.0537>

[25]: Christos STERGIU, Kostas E. PSANNIS, Byung-Gyu KIM et Brij GUPTA. « Secure integration of IoT and Cloud Computing ». In : *Future Generation Computer Systems* 78 (2018)

[26]: Alessio BOTTA, Walter de DONATO, Valerio PERSICO et Antonio PESCAPÉ. « Integration of Cloud computing and Internet of Things : A survey ». In : *Future Generation Computer Systems* 56 (2016)

[27]: Weisong SHI, Jie CAO, Quan ZHANG, Youhui LI et Lanyu XU. « Edge Computing : Vision and Challenges ». In : *IEEE Internet of Things Journal* 3.5 (2016)

[28]: CISCO. *Cisco Annual Internet Report (2018–2023)*. White Paper. Cisco Systems, 2020

[29]: Yuyi MAO, Changsheng YOU, Jun ZHANG, Kaibin HUANG et Khaled B. LETAIEF. « A Survey on Mobile Edge Computing : The Communication Perspective ». In : *IEEE Communications Surveys Tutorials* 19.4 (2017)

[30]: Mahadev SATYANARAYANAN. « The Emergence of Edge Computing ». In : *Computer* 50.1 (2017)

[30]: Mahadev SATYANARAYANAN. « The Emergence of Edge Computing ». In : *Computer* 50.1 (2017)

[31]: Ashkan YOUSEFPOUR, Caleb FUNG, Tam NGUYEN, Krishna P. KADIYALA, Fatemeh JALALI, Amirreza NIAKANLAHJI, Jian KONG et Jason P. JUE. « All One Needs to Know about Fog Computing and Related Edge Computing Paradigms : A Complete Survey ». In : *ArXiv abs/1808.05283* (2018)

[34]: Y. C. HU, M. PATEL, D. SABELLA, N. SPRECHER et V. YOUNG. *Mobile Edge Computing - A Key Technology Towards 5G*. Rapp. tech. White Paper No. 11. ETSI (European Telecommunications Standards Institute), 2015

"Edge computing", "Cloudlets", "Fog Computing"...

On retrouve de nombreux termes pour désigner différents paradigmes de l'informatique en périphérie, chacun étant plus ou moins spécifique, ce qui peut entraîner de la confusion. L'informatique en brouillard (Fog Computing) et l'informatique en périphérie apportent tous deux des services du Cloud à la périphérie du réseau [31]. Cependant, l'informatique en périphérie se concentre davantage sur les objets, tandis que l'informatique en brouillard se concentre sur l'ensemble de l'infrastructure, de la périphérie au cloud [27, 31]. Les cloudlets sont des "centres de données en boîte" situés à proximité des utilisateurs et accessibles via WiFi. Ils ressemblent à des points d'accès WiFi, mais disposent de capacités de calcul pour fournir des services de cloud à proximité des utilisateurs avec peu d'entretien et une faible consommation d'énergie [32]. Cependant, ils ont été écartés en raison de leur accès WiFi, qui implique une couverture limitée, une prise en charge difficile de la mobilité entre les cloudlets et des préoccupations en matière de sécurité. Similaires aux cloudlets, les micro-centres de données (micro-data-centers) sont composés de 10 serveurs ou moins et sont placés à proximité des utilisateurs [33]. Enfin, le Mobile Edge Computing (MEC) a été défini par l'ETSI en 2015 comme une informatique en périphérie intégrée dans les réseaux d'accès radio (RAN) pour servir les utilisateurs mobiles [34]. Le terme mobile edge computing a évolué vers *multiaccess computing* [35] (en français informatique en périphérie multiaccès), permettant l'intégration de technologies d'accès radio hétérogènes (RATs), telles que la 5G, la LTE, le WiFi, etc., dans le paradigme et ajoutant ainsi des cas d'utilisation supplémentaires n'impliquant pas uniquement des appareils mobiles, comme illustré dans la Figure 2.1. Dans cette thèse, nous utilisons donc le terme MEC (multiaccess edge computing) comme un terme générique qui englobe également le paradigme de l'informatique en périphérie mobile. Le MEC est le candidat le plus prometteur pour les applications critiques pour la mission et les applications soumises à des contraintes temporelles fortes, en raison de sa proximité, de son bon support de mobilité pour les utilisateurs mobiles et de son intégration de plusieurs technologies d'accès.

1.3 MEC : de nouveaux défis

De par sa nature différente du cloud, le MEC présente de nouveaux défis tout en ouvrant également la voie à de nouvelles applications, telles que le déploiement de serveurs MEC intégrés à des robots ou des drones.

1.3.1 Ressources fixes

Dans les réseaux MEC, les serveurs locaux sont limités en ressources, et en tant que paradigme récent, ils relèvent des défis ouverts pour gérer ces ressources de capacité limitée [27, 36]. Par conséquent, il est essentiel de gérer correctement les ressources MEC pour répondre efficacement aux demandes des utilisateurs. La gestion des ressources en MEC se divise en trois aspects : (i) la décision de déchargement des données (*offloading*), (ii) l'allocation des ressources et (iii) la gestion de

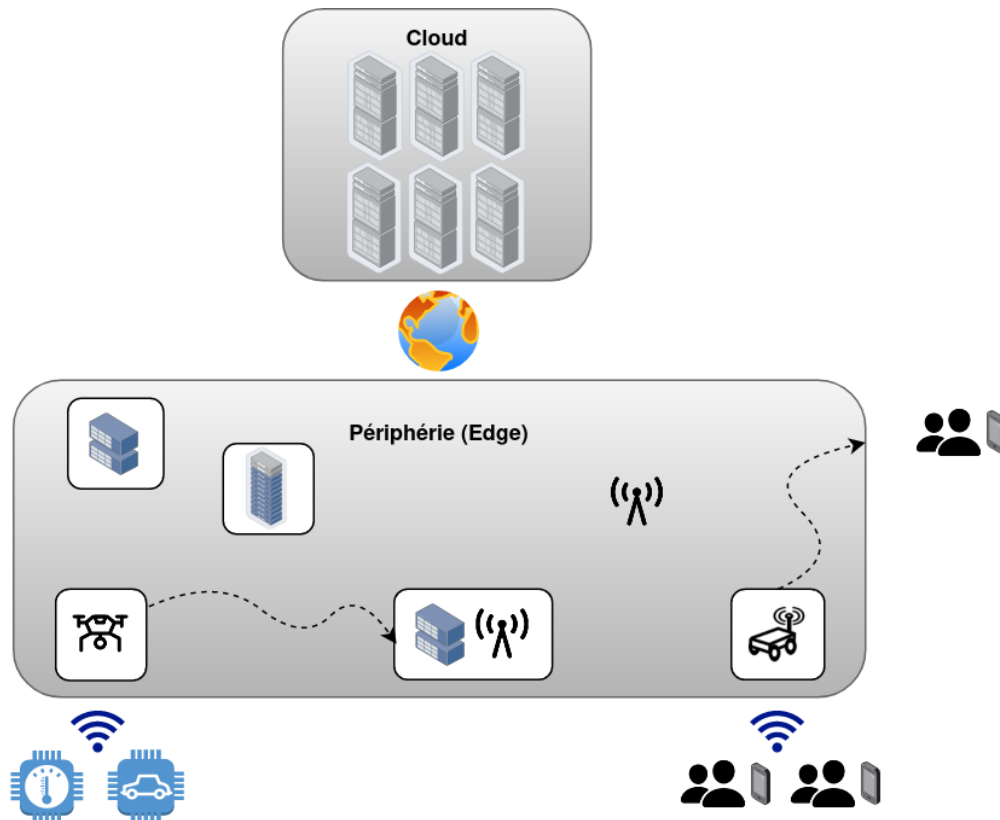


FIGURE 1.3 – Informatique en périphérie multi-accès avec ressources mobiles.

la mobilité des utilisateurs, c'est-à-dire la migration de services. Dans cette thèse, nous nous concentrons sur les outils pouvant aider à prendre des décisions en matière d'allocation de ressources. Effectivement, les ressources matérielles du MEC sont limitées en capacité, contrairement à celles du Cloud, et elles peuvent même fonctionner sur batterie. Le schéma d'allocation des ressources est donc essentiel pour gérer efficacement ces ressources limitées et répondre aux demandes en termes de contraintes de latence, de QoS et de QoE des applications.

1.3.2 Ressources mobiles

Lorsque les serveurs MEC sont montés sur des UAV (véhicules aériens sans pilote) ou des robots terrestres, ils sont adaptés pour répondre aux besoins des utilisateurs mobiles ou des appareils IoT lors d'événements temporaires ou de situations d'urgence. En effet, des ressources fixes pourraient s'avérer trop coûteuses, trop rigides à déployer ou simplement nécessaires pour une durée limitée. Ainsi, les drones et les robots possèdent la mobilité nécessaire pour être rapidement déployés dans les zones de forte demande ou d'urgence, de manière temporaire, et sont suffisamment flexibles pour se déplacer afin de suivre la dynamique des demandes [37-39]. La gestion et l'allocation des ressources est ainsi élargie avec un quatrième aspect : (iv) la gestion des ressources mobiles, qui comprend le déploiement des ressources, c'est-à-dire leur nombre et leur emplacement, leur planification de trajet et de nouveaux coûts tels que les retards de déploiement. La Figure 1.3 illustre le MEC avec ressources mobiles.

[35]: F. GIUST, G. VERIN, K. ANTEVSKI, J. CHOU, Y. FANG, W. FEATHERSTONE, F. FONTES, D. FRYDMAN, A. LI, A. MANZALINI, D. PURKAYASTHA, D. SABELLA, C. WEHNER, K-W WEN et Z. ZHOU. *MEC Deployments in 4G and Evolution Towards 5G*. Rapp. tech. White Paper No. 24. ETSI (European Telecommunications Standards Institute), 2018

[27]: Weisong SHI, Jie CAO, Quan ZHANG, Youhuizi LI et Lanyu XU. « Edge Computing : Vision and Challenges ». In : *IEEE Internet of Things Journal* 3.5 (2016)

[36]: Dimitrios DECHOUNIOTIS, Nikolaos ATHANASOPOULOS, Aris LEIVADEAS, Nathalie MITTON, Raphael JUNGERS et Symeon PAPAVASSILIOU. « Edge Computing Resource Allocation for Dynamic Networks : The DRUID-NET Vision and Perspective ». In : *Sensors* 20.8 (2020)

[37]: M. ULEMA, D. ZUCKERMAN, P. CHATZIMISIOS, F. GRANELLI, N. MANGRA, E. MARKAKIS, K. NAMUDURI, Y. NIKOLOUDAKIS, P. RAWAT, M. Z. SHAKIR et T. ZHANG. *Public Safety Technology Gaps and Opportunities*. White Paper. 2021

[38]: Tarik TALEB, Konstantinos SAMDANIS, Badr MADA, Hannu FLINCK, Sunny DUTTA et Dario SABELLA. « On Multi-Access Edge Computing : A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration ». In : *IEEE Communications Surveys Tutorials* 19.3 (2017)

[39]: Muhammad ABRAR, Ushna AJMAL, Ziyad M. ALMOHAIMEED, Xiang GUI, Rizwan AKRAM et Roha MASROOR. « Energy Efficient UAV-Enabled Mobile Edge Computing for IoT Devices : A Review ». In : *IEEE Access* 9 (2021)

[40]: CHIST-ERA consortium. <https://www.chistera.eu/projects/druid-net>

[36]: Dimitrios DECHOUNIOTIS, Nikolaos ATHANASOPOULOS, Aris LEIVADEAS, Nathalie MITTON, Raphael JUNGERS et Symeon PAPAVALASSILOU. « Edge Computing Resource Allocation for Dynamic Networks : The DRUID-NET Vision and Perspective ». In : *Sensors* 20.8 (2020)

1.4 Projet européen : DRUID-NET

L'informatique en périphérie est un paradigme qui englobe plusieurs domaines de recherche, et ses applications sont très variées, pouvant être étroitement liées à de nombreux domaines. Les défis associés à l'informatique en périphérie sont nombreux et complexes, et nécessitent une approche holistique qui combine différentes expertises. C'est précisément dans cette optique que le projet européen CHIST-ERA DRUID-NET a été lancé avec plusieurs partenaires scientifiques internationaux, dont cette thèse fait partie.

1.4.1 Présentation du projet et des équipes

Cette thèse s'inscrit dans le cadre d'un projet européen de trois ans, de 2020 à 2023, financé par CHIST-ERA [40], appelé DRUID-NET. Au sein du domaine de l'informatique en périphérie, DRUID-NET, dont l'acronyme signifie *eDge computing ResoUrce allocatIon for Dynamic NETworks*, vise à résoudre les défis liés à la gestion des ressources MEC, comme cela a été exposé dans la section précédente, en distribuant de manière dynamique les ressources en fonction des variations rapides de la demande [36]. Ce projet adopte une approche holistique en réunissant plusieurs équipes de recherche internationales possédant diverses expertises, venant des institutions ci-dessous :

- Université technique nationale d'Athènes - Grèce
- Université catholique de Louvain - Belgique
- École de Technologie Supérieure - Canada
- INRIA - Lille - France
- Université Queen's de Belfast - Royaume-Uni

1.4.2 Objectifs du projet

En prenant en considération à la fois la mobilité de l'utilisateur final et les paramètres de l'environnement de connexion sans fil, DRUID-NET a pour objectif de développer des modèles de performance dynamiques pour les profils réseau des applications IoT en utilisant les outils de prédiction et de modélisation appropriés. À cet effet, le projet s'est concentré sur les objectifs clés suivants :

Estimation du profil de charge de travail DRUID-NET utilise des modèles de données et mathématiques pour classifier et catégoriser les besoins des différentes applications IoT en se basant sur des caractéristiques de trafic telles que les schémas de transmission, la corrélation spatiale et temporelle du trafic, la taille des trames, et la régularité du trafic. Cette approche crée des mécanismes de prédiction pour gérer la dynamique et l'incertitude des profils de trafic des applications IoT, en ciblant des catégories spécifiques ayant des exigences similaires en termes de ressources.

Modélisation des performances Le projet vise à développer des modèles de trafic et de ressources dynamiques pour émuler le trafic IoT. Le projet DRUID-NET utilise des modèles hybrides dynamiques pour prendre en compte différentes métriques de performance et ressources. Ce cadre

aide à équilibrer précision, complexité et faisabilité en temps réel de l'allocation des ressources, tout en capturant les changements structurels tels que la mobilité des utilisateurs et les changements de protocoles.

Allocation de ressources Le mécanisme d'allocation de ressources a pour but d'allouer conjointement la communication, le calcul et le stockage. Il utilise deux approches : la première résout rapidement les problèmes d'optimisation multi-objectif pour obtenir un point de fonctionnement optimal, tandis que la seconde se concentre sur le comportement dynamique de l'allocation des ressources. Des méthodes d'analyse théorique et des contrôleurs pour la stabilité sont développés pour assurer la robustesse aux changements rapides de charge de travail, de disponibilité des ressources et de problèmes de réseau.

Conception conjointe de contrôleurs et d'algorithmes d'allocation de ressources Le cadre DRUID-NET permet d'encapsuler, de comparer et de modifier l'impact des perturbations dans les futures applications de contrôle où les ressources doivent être utilisées avec parcimonie. Cette recherche vise à concevoir des contrôleurs qui traitent un mélange de phénomènes indésirables en modifiant l'approvisionnement des ressources à l'algorithme de contrôle, si cela est jugé nécessaire.

Dans ces différents objectifs clés, nous avons collaboré dans cette thèse à divers travaux sur l'estimation du profil de charge de travail et la modélisation des performances. Nous expliquons plus en détail les problématiques spécifiques à cette thèse.

1.5 Problématiques

Nous avons vu que la gestion des ressources dans le contexte du MEC, qu'il s'agisse de ressources fixes ou mobiles, présente des défis complexes en raison de plusieurs facteurs. De même, nous avons déjà examiné les approches adoptées par le projet DRUID-NET pour relever ces défis. Maintenant, nous allons nous pencher plus spécifiquement sur les problématiques auxquelles nous avons été confrontés durant cette thèse au sein de DRUID-NET. Tout d'abord, les ressources MEC sont limitées en termes de puissance de traitement et de capacité de stockage. De plus, les réseaux sans fil sont connus pour leur manque de fiabilité et leur propension aux pertes d'informations. Par ailleurs, les applications MEC peuvent avoir des exigences strictes en termes de Qualité de Service (QoS) et de Qualité de l'Expérience (QoE). Finalement, les environnements d'application du MEC évoluent rapidement. Pour faire face à ces défis de la gestion des ressources à disposition, une approche consiste à *anticiper* les demandes et les besoins, lesquels peuvent être modélisés à l'aide des métriques de la Qualité de Service du réseau. En effet, les fluctuations de la charge de trafic et les variations des conditions des canaux peuvent entraîner des perturbations dans le réseau, provoquant des engorgements et ralentissant considérablement les opérations. En prévoyant ces variations de trafic et les évolutions des conditions du réseau, il devient possible d'allouer les ressources à l'avance ou d'éviter une allocation excessive. Ainsi, la **prédiction de la QoS du réseau** permet de prévenir à la fois la surutilisation et la sous-utilisation des ressources. La mobilité des utilisateurs est un autre élément crucial dans la gestion des ressources,

[41]: Jan PLACHY, Zdenek BECVAR et Emilio Calvanese STRINATI. « Dynamic resource allocation exploiting mobility prediction in mobile edge computing ». In : 2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC). 2016

non seulement pour l'allocation des ressources fixes, mais également pour le déploiement des ressources mobiles. Il est donc essentiel de positionner correctement les ressources, en tenant compte des délais de déplacement pour les ressources mobiles, ainsi que du temps nécessaire à la configuration et au démarrage des machines virtuelles [41] qui fourniront les services pour les ressources fixes. Ceci garantit leur disponibilité au moment nécessaire. C'est pourquoi la **prédiction de la mobilité des foules** joue un rôle clé dans l'anticipation de ces déplacements. Divers mécanismes de prédiction, tels que l'apprentissage automatique, peuvent contribuer à anticiper ces aspects.

C'est à ces divers défis pour le déploiement de ressources MEC mobiles que nous nous sommes intéressées durant cette thèse, en mettant l'accent sur les outils de prédiction pour la QoS et la mobilité des utilisateurs. De plus, nous avons examiné les travaux déjà réalisés dans le domaine de la gestion des ressources du MEC afin de contribuer à la création de nouveaux schémas exploitant des outils de prédiction.

1.6 Contributions et organisation de cette thèse

1.6.1 Contributions de cette thèse

Cette thèse regroupe les divers travaux réalisés au cours de ces trois années de recherche visant à faciliter la gestion des ressources MEC. Voici les contributions apportées :

De l'importance des données : Nous avons vu précédemment que la prédiction peut jouer un rôle crucial dans la gestion des ressources, notamment en ce qui concerne la puissance et la mémoire, dans les réseaux dynamiques. Cependant, il est important de noter que de nombreux mécanismes de prédiction se basent sur des données historiques. C'est pourquoi, nous présentons des ensembles de données ainsi que des outils permettant leur génération dans une approche ouverte. Nous détaillons la méthodologie utilisée pour créer un premier ensemble de données contenant des traces applicatives de l'IoT. De plus, nous mettons à disposition un outil qui permet de générer automatiquement et de manière reproductible ce type de données, ainsi qu'un nouvel ensemble de données enrichi. Toutes ces ressources en données sont analysées en détail et mises à la disposition de la communauté de recherche. Enfin, nous proposons un ensemble de données contenant des traces applicatives associées à des points d'accès MEC mobiles.

Prédiction de la Qualité de Service : En utilisant les données générées, nous proposons ensuite des modèles de prédiction de la Qualité de Service (QoS). Dans un premier temps, nous comparons différents modèles d'apprentissage automatique utilisant la régression pour prédire le débit. Ces modèles sont entraînés sur le premier ensemble de données fourni dans le Chapitre précédent et comparés les uns aux autres. Ensuite, nous prédisons plusieurs métriques de la qualité de service, à savoir le débit, le taux de paquets délivrés, le taux de paquets perdus et la latence, à l'aide d'un modèle de transformateurs temporels dans le but d'anticiper les besoins en ressources *Edge*. Ce modèle de transformateur est comparé à d'autres méthodes de prédiction, et les résultats sont analysés en détail.

Prédiction de la mobilité humaine : Nous présentons un algorithme de prédiction de la densité humaine dans l'espace dans le but d'anticiper les lieux de déploiement des ressources. Contrairement aux modèles de prédiction précédents, cette approche explore une voie qui ne repose pas sur des données historiques pour prédire la mobilité, qu'il s'agisse de jours "réguliers" ou de jours "irréguliers" où la mobilité humaine est perturbée par des événements imprévus ou exceptionnels, tels qu'une catastrophe naturelle ou un événement sportif. En effet, la mobilité humaine régulière est ainsi perturbée, et nous ne disposons pas de données historiques pour permettre à un modèle d'apprentissage d'apprendre ces nouveaux schémas. Pour résoudre ce défi, nous utilisons un filtre à particules couplé à un simulateur de mobilité humaine ainsi qu'un algorithme dit de "gossip".

Étude sur la gestion des ressources MEC : Finalement, nous présentons un état de l'art sur la gestion des ressources MEC. Nous mettons à disposition cette étude en tant qu'outil pour la conception d'un nouveau schéma de gestion des ressources exploitant les prédictions de la QoS et de la mobilité humaine. Tout d'abord, nous effectuons une revue des travaux portant sur l'allocation des ressources MEC, qu'elles soient fixes ou mobiles. Nous définissons la modélisation des tâches, les types de ressources pouvant être alloués, ainsi que les objectifs de réduction ou de maximisation de l'allocation. Ensuite, nous passons en revue les différentes méthodes disponibles, en exposant leurs avantages et inconvénients. Enfin, nous faisons de même pour le déploiement dans des espaces de ressources mobiles, en mettant particulièrement l'accent sur les UAVs. Nous présentons les différentes modélisations de systèmes existantes ainsi que les méthodes associées.

1.6.2 Organisation et contenu du document

Nous organisons cette thèse en six chapitres. Tout d'abord, dans le Chapitre 1, nous introduisons le contexte de cette thèse en établissant des liens entre plusieurs domaines clés, notamment les réseaux sans fil, l'Internet des objets (IoT) et l'informatique en périphérie multi-accès. Nous clarifions les termes utilisés dans ce contexte pour mieux situer nos travaux, tout en exposant les problématiques de recherche et les motivations qui les sous-tendent. De plus, nous présentons le projet européen qui soutient ces travaux. Le Chapitre 2 se consacre à la génération de données, tandis que le Chapitre 3 se focalise sur les algorithmes de prédiction de la QoS en utilisant les données présentées au Chapitre 2. Le Chapitre 4 présente l'algorithme de prédiction de la densité humaine. Enfin, dans le Chapitre 5, nous recensons l'état de l'art en matière de gestion des ressources MEC. Dans le Chapitre 6, nous concluons en récapitulant en détail les diverses contributions apportées au cours de cette thèse. Nous formulons une conclusion générale qui englobe l'ensemble de ces travaux, et nous discutons des perspectives pour de futures recherches.

1.7 Liste des publications

Journaux

Nina SANTI et Nathalie MITTON. « A resource management survey for mission critical and time critical applications in multi access edge computing ». In : *ITU Journal on Future and Evolving Technologies*. Volume 2 (2021), Issue 2. 2.2 (nov. 2021)

Firdose SAEIK, Marios AVGERIS, Dimitrios SPATHARAKIS, Nina SANTI, Dimitrios DECHOUNIOTIS, John VIOLOS, Aris LEIVADEAS, Nikolaos ATHANASOPOULOS, Nathalie MITTON et Symeon PAPAVALASSIOU. « Task Offloading in Edge and Cloud Computing : A Survey on Mathematical, Artificial Intelligence and Control Theory Solutions ». In : *Computer Networks* 195 (août 2021)

Conférences internationales

Nina SANTI, Rémy GRÜNBLATT, Brandon FOUBERT, AROOSA HAMEED, John VIOLOS, Aris LEIVADEAS et Nathalie MITTON. « Automated and Reproducible Application Traces Generation for IoT Applications ». In : *Q2SWinet 2021 - 17th ACM Symposium on QoS and Security for Wireless and Mobile Networks*. Alicante, Spain : ACM, nov. 2021

AROOSA HAMEED, John VIOLOS, Nina SANTI, Aris LEIVADEAS et Nathalie MITTON. « A Machine Learning Regression approach for Throughput Estimation in an IoT Environment ». In : *iThings-2021 : The 14th IEEE International Conference on Internet of Things*. Melbourne, Australia, déc. 2021

Ateliers internationaux

Kevin JIOKENG, Nathalie MITTON et Nina SANTI. *Generating Mobility-Aware Traces for IoT Applications*. In : *IEEE Global Communications Conference 2022, Demo sessions*. <https://hal.science/hal-03851704>. 2022

Conférences nationales

Nina SANTI, Brandon FOUBERT et Nathalie MITTON. « Comment générer des traces applicatives avec FIT IoT-LAB pour la science ouverte ». In : *CORES 2021 – 6ème Rencontres Francophones sur la Conception de Protocoles, l'Évaluation de Performance et l'Expérimentation des Réseaux de Communication*. La Rochelle, France, sept. 2021

2.1 Jeux de données de l’IoT pour la prédiction

Afin d’anticiper les demandes et les besoins, nous avons constaté dans la Section 1.5 qu’il était possible de **prédire la Qualité de Service (QoS)** du réseau. Pour ce faire, il est nécessaire de créer des modèles de prédiction. Dans ce chapitre, nous abordons donc la première étape vers la prédiction de la QoS : la génération des données.

2.1.1 Données IoT : une demande croissante

Les données sont désormais au cœur de nombreuses recherches. Avec la récente croissance de l’apprentissage automatique et de l’apprentissage profond, il est aujourd’hui essentiel de trouver des ensembles de données complets et de qualité pour produire des modèles pertinents et réalistes. C’est d’autant plus le cas en IoT où les données de réseaux sont essentielles pour comprendre les caractéristiques du réseau, leurs modèles et les modéliser. Dans le cadre de cette thèse, ces données nous servent à concevoir des modèles de prédiction de la qualité de service du réseau afin d’anticiper une demande en ressources supplémentaires par le *Edge* et d’aider à la prise de décision en termes d’allocation et de gestion. Mais de telles données peuvent également être utiles pour détecter des activités anormales ou des comportements caractéristiques d’attaques déjà connues en formant des modèles d’apprentissage automatique [48, 49] ou d’apprentissage profond [50]. De même, les chercheurs peuvent utiliser ce type de données pour former des modèles afin d’identifier et de classer les différents appareils IoT présents dans le réseau [51]. C’est pourquoi ces travaux sont considérés à part entière, et nous souhaitons les valoriser dans ce chapitre, car ils ont non seulement été nécessaires pour nos travaux, mais peuvent également l’être pour d’autres recherches.

2.1.2 Nos besoins

Nous considérons des environnements très dynamiques propres au MEC, de plus, les communications sans fil sont de nature très incertaine, peu fiables et peuvent entraîner des pertes de données. C’est d’autant plus le cas dans les environnements que nous considérons. Nous savons qu’en IoT, il existe de nombreuses applications très hétérogènes en raison des appareils utilisés, mais aussi en raison du trafic généré par ceux-ci. Les appareils échangent et envoient des données à des fréquences différentes et dans des contextes différents. Il nous faut donc des données qui reflètent ces différentes couches de complexité pour permettre de reproduire la réalité et mettre en avant l’efficacité des techniques d’apprentissage automatique utilisées. Nous avons donc besoin de données de qualité, et pour cela, elles doivent remplir quelques critères principaux, c’est-à-dire :

2.1 Jeux de données de l’IoT pour la prédiction	11
2.1.1 Données IoT: une demande croissante	11
2.1.2 Nos besoins	11
2.1.3 Pourquoi générer nos propres données ?	12
2.1.4 Science ouverte et reproductibilité	13
2.2 Génération de traces dynamiques et hétérogènes de l’IoT	14
2.2.1 Caractérisation des applications	14
2.2.2 Dispositif expérimental	15
2.2.3 Résultats	16
2.2.4 Conclusion et discussion	19
2.3 Génération automatisée et reproductible de données	20
2.3.1 Sisyphe: Outil pour la génération de données automatisée et reproductible	20
2.3.2 Données d’applications IoT enrichies	24
2.3.3 Résultats	26
2.3.4 Analyse statistique des résultats	26
2.3.5 Conclusion	29
2.4 Génération de traces intégrant des ressources mobiles	30
2.4.1 Caractérisation des applications Edge	30
2.4.2 Dispositif expérimental	31
2.4.3 Paramètres des expériences	32
2.4.4 Résultats	33
2.4.5 Conclusion	33
2.5 Conclusion	34

[52]: Tamas PFLANZNER, Zoltan FEHER et Attila KERTESZ. « A Crawling Approach to Facilitate Open IoT Data Archiving and Reuse ». In : *2019 Sixth International Conference on Internet of Things : Systems, Management and Security (IOTSMS)*. 2019

[53]: M. KUZMAN, X. d. TORO GARCÍA, S. ESCOLAR, A. CARUSO, S. CHESSA et J. C. LÓPEZ. « A Testbed and an Experimental Public Dataset for Energy-Harvested IoT Solutions ». In : *IEEE Int. Conference on Industrial Informatics (INDIN)*. 2019

[54]: Z. XIAO, F. LI, R. WU, H. JIANG, Y. HU, J. REN, C. CAI et A. IYENGAR. « TrajData : On Vehicle Trajectory Collection With Commodity Plug-and-Play OBU Devices ». In : *IEEE Internet of Things Journal* 7 (2020)

[55]: R. JACOB, R. Da FORNO, R. TRUB, A. BIRI et L. THIELE. « Dataset : Wireless Link Quality Estimation on FlockLab - and Beyond ». In : *Proceedings of the 2nd Workshop on Data Acquisition To Analysis*. 2019

[56]: Nickolaos KORONIOTIS, Nour MOUSTAFA, Elena SITNIKOVA et Benjamin TURNBULL. « Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics : Bot-IoT dataset ». In : *Future Generation Computer Systems* 100 (2019)

[57]: Abdullah ALSAEDI, Nour MOUSTAFA, Zahir TARI, Abdun MAHMOOD et Adnan ANWAR. « TON_IoT Telemetry Dataset : A New Generation Dataset of IoT and IIoT for Data-Driven Intrusion Detection Systems ». In : *IEEE Access* 8 (2020)

[58]: Ivan VACCARI, Giovanni CHIOLA, Maurizio AIELLO, Maurizio MONGELLI et Enrico CAMBIASO. « MQTTset, a New Dataset for Machine Learning Techniques on MQTT ». In : *Sensors* 20.22 (2020)

[59]: Mohamed Amine FERRAG, Othmane FRIHA, Djallel HAMOUDA, Leandros MAGLARAS et Helge JANICKE. « Edge-IIoTset : A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications for Centralized and Federated Learning ». In : *IEEE Access* 10 (2022)

[60]: Sajjad DADKHAH, Hassan MAHDIKHANI, Priscilla Kyei DANSO, Alireza ZOHOURIAN, Kevin Anh TRUONG et Ali A. GHORBANI. « Towards the Development of a Realistic Multidimensional IoT Profiling Dataset ». In : *2022 19th Annual International Conference on Privacy, Security Trust (PST)*. 2022

- **Complètes** : elles doivent contenir toutes les informations requises.
- **Valides** : elles doivent provenir de sources fiables ou être générées depuis un environnement maîtrisé.
- **Précises** : elles doivent représenter fidèlement la réalité et le cas d'utilisation visé.
- **Actualisées** : elles doivent être à jour, ce qui est d'autant plus important dans le domaine de l'IoT (Internet des objets), où le matériel est en constante évolution, de même que les logiciels et les paradigmes.

2.1.3 Pourquoi générer nos propres données ?

Nous avons d'abord cherché dans la littérature les données déjà existantes :

Pflanzner et al. [52] proposent un service d'indexation pour récupérer des traces ouvertes d'applications disponibles sur Internet. [53] présente un ensemble de données provenant d'un banc d'essai de capteurs météorologiques alimentés par l'énergie solaire. Les données contiennent la consommation et la production d'énergie de l'application pour aider la recherche dans les applications IoT auto-alimentées. Xiao et al. [54] introduisent une solution prête à l'emploi pour collecter des données de trajectoire de véhicule. Leur solution comprend un algorithme d'apprentissage profond pour combler les lacunes dans la collecte de données, telles que les défaillances du GPS. [55] fournit un ensemble de données provenant de la plateforme FlockLab. La collecte de données s'effectue sur une longue période afin de capturer des motifs périodiques à long terme qui affectent la qualité des liaisons sans fil. Koroniotis et al. [56] présentent un ensemble de données Bot-IoT qui inclut des trafics provenant d'attaques couramment utilisées par les botnets. L'ensemble comprend plusieurs caractéristiques du trafic réseau et utilise comme application IoT une station météo, un réfrigérateur intelligent, une lumière activée par le mouvement, une porte de garage activée à distance et un thermostat intelligent. Le jeu de données est complet, mais les applications IoT sont simulées à l'ordinateur et peuvent donc potentiellement moins bien refléter les conditions réelles des traffics IoT. TON_IoT est un jeu de données proposé dans [57] pour aider à la détection d'attaque d'intrusion basée sur les données. Les données sont étiquetées pour indiquer si elles représentent une attaque ou non, ainsi qu'avec le sous-type d'attaque. L'architecture de l'expérimentation est composée d'une couche *Cloud*, *Edge* et des capteurs IoT. Vaccari et al. [58] présentent MQTTset, un jeu de données axé sur le protocole MQTT entre 8 capteurs et qui est validé par un système de détection. Ferrag et al. [59] fournissent l'ensemble Edge-IIoTset pour la cybersécurité centrée sur les applications IoT et de l'IoT pour l'industrie (IIoT). 8 capteurs sont déployés avec un broker MQTT. Les auteurs testent leurs données avec des modèles de détection d'intrusion centralisée et fédérée basée sur l'apprentissage profond. Dadkha et al. [60] produisent un ensemble de données IoT avec 60 appareils équipés de WiFi, ZigBee et Z-Wave. Ils y ajoutent deux jeux de données d'attaques. L'expérimentation est menée localement dans le laboratoire de l'Institut canadien de cybersécurité (CIC). NetFlow, proposé dans [61], regroupe les ensembles de caractéristiques de quatre jeux de données pour la détection d'intrusion, à savoir UNSW-NB15, BoT-IoT, ToN-IoT et

CSE-CIC-IDS2018, afin d'avoir une base de comparaison pour ces quatre jeux de données.

Nous avons noté que ces jeux de données sont souvent spécialisés dans un domaine (particulièrement la cybersécurité) et donc pas assez *complets* ni *précis* pour nos cas d'utilisation. De plus, les environnements expérimentaux et les codes de la construction à la récupération des résultats sont souvent fermés et non accessibles publiquement, on ne peut donc pas vérifier la *fiabilité* des données ni savoir si elles sont *actualisées*. C'est donc dans cet esprit que nous voulons générer nos propres données en ouvrant le plus possible notre démarche.

2.1.4 Science ouverte et reproductibilité

Pour que ces travaux soient profitables à d'autres recherches, nous avons ouvert publiquement les résultats dans le cadre d'une démarche de science ouverte. En effet, dans un contexte de demande croissante de données, le mouvement de la science ouverte apporte une réponse en permettant une grande disponibilité des données ouvertes, ce qui contribue directement à faire progresser la recherche. En plus de mettre à disposition des ressources libres, les données ouvertes représentent un atout majeur en ce qui concerne la reproductibilité des résultats, qui revêt une importance capitale dans de nombreux domaines scientifiques. De plus en plus de conférences et de revues proposent de soumettre, en parallèle d'un article, une courte extension présentant les données variées utilisées pour les expériences et/ou les simulations, appelées *artefacts*, ce qui témoigne de l'intérêt croissant pour cette démarche. De plus, il est reconnu que plusieurs types de biais peuvent être présents dans les données [62], et de tels biais peuvent entraîner des comportements indésirables dans les modèles. Il est donc important de prendre des précautions lors de la conception et de la création de jeux de données. La reproductibilité, telle que définie par ACM [63], apparaît comme le meilleur moyen d'accroître la confiance dans les résultats basés sur la génération de données. Des étapes intermédiaires, telles que la répétabilité et la reproductibilité, peuvent également promouvoir et confirmer l'intégrité de ces résultats. La répétabilité revêt une importance particulière dans le cadre de la création d'ensembles de données pour l'IoT et les réseaux sans fil, car le spectre radio, les technologies de communication et les systèmes d'exploitation qui alimentent ces réseaux subissent constamment des changements. Cela rend les jeux de données pour l'IoT facilement obsolètes ou, pire encore, dommageables. C'est pourquoi les données présentées dans le cadre de ce chapitre sont entièrement répétables, car nous fournissons le code et les outils nécessaires à leur création et à leur re-génération.

Dans ce Chapitre, nous allons examiner les différents jeux de données d'applications de l'IoT et du *Edge* que nous avons produits, ainsi que les différents outils destinés à aider la communauté de la recherche à réutiliser ces données, tout en facilitant la création de nouvelles.

[61]: Mohanad SARHAN, Siamak LAYEGHY, Nour MOUSTAFA et Marius PORTMANN. « NetFlow Datasets for Machine Learning-Based Network Intrusion Detection Systems ». In : (2021)

[62]: Ninareh MEHRABI, Fred MORSTATTER, Nripsuta SAXENA, Kristina LERMAN et Aram GALSTYAN. « A Survey on Bias and Fairness in Machine Learning ». In : 54.6 (2021)

[63]: *Artifact Review and Badging Version 1.1*. Août 2020

TABLE 2.1 – Paramètres des simulations

Application	Nombre de nœuds	Fréquence d’envoi de message	Taille de charge utile (octets)	Durée (minutes)
Systèmes CVC	100	1 paquet par 4 minutes	60	60
Éclairages intelligents	100	1 paquet par 8 minutes	30	90
Systèmes d’urgence	40	1 paquet par 30 secondes	127	10
Surveillance	30	99 paquets par seconde	127	10
Réalité virtuelle ou augmentée	10	197 paquets par seconde	127	10
VoIP	10	16 paquets par seconde	127	10

2.2 Génération de traces dynamiques et hétérogènes de l’IoT

Nous allons maintenant présenter ici les premières traces d’applications IoT que nous avons générées. Comme déjà motivé précédemment, nous cherchons à obtenir des traces complètes qui reflètent la réalité complexe et hétérogène des réseaux IoT. Dans le cadre de cette thèse, ces traces sont utilisées pour créer des modèles de prédiction de la qualité de service (QoS) différentes en fonction des applications, que l’on retrouve dans le Chapitre 3.2, afin que les ressources Edge soit allouées de manière optimale. Pour ce faire, nous avons développé une méthodologie spécifique. Tout d’abord, nous caractérisons les différents types de trafic hétérogènes à travers différents paramètres d’expérimentation tels que leur fréquence de communication et la taille des paquets échangés. Ensuite, nous simulons ces applications sur la plateforme FIT IoT-LAB [64] afin de générer les traces. Les paramètres des simulations, tels que le nombre de nœuds employés, sont choisis en fonction des caractéristiques de l’application émulée. Les traces ainsi générées sont enrichies de plusieurs données permettant de déduire des métriques utiles, telles que le taux de livraison des paquets et le délai de bout en bout. Nous partageons ces traces obtenues en accès ouvert, ainsi qu’une base de code pour générer, manipuler et analyser les données obtenues [65].

2.2.1 Caractérisation des applications

Nous avons décidé, dans un premier temps, de produire des traces applicatives IoT en caractérisant et définissant plusieurs applications IoT, chacune d’entre elles fournissant des données spécifiques à une application particulière. En effet, chaque application IoT possède ses propres caractéristiques, et disposer de données spécifiques à chaque application permet de créer un jeu de données complet et précis. Nous allons maintenant expliquer comment nous caractérisons ces applications IoT. Les caractéristiques des applications IoT sont définies en termes de la fréquence des messages échangés, du nombre de nœuds utilisés, de leur densité et du type de message [66-68]. Par exemple, les systèmes CVC et les éclairages intelligents présentent une fréquence d’échange faible, mais avec un nombre important de nœuds. Ces deux applications échangent des fichiers simples. En revanche, les systèmes d’urgence utilisent un nombre modéré de nœuds avec une fréquence d’envoi de message également modérée. Les messages échangés comprennent des images et

[64]: Cedric ADJIH, Emmanuel BACCELLI, Eric FLEURY, Gaetan HARTE, Nathalie MITTON, Thomas NOEL, Roger PISSARD-GIBOLLET, Frédéric SAINT-MARCEL, Guillaume SCHREINER, Julien VANDAELE et Thomas WATTEYNE. « FIT IoT-LAB : A Large Scale Open Experimental IoT Testbed ». In : Milan, Italy, déc. 2015

[65]: *Applications traces*. <https://gitlab.inria.fr/druident/application-traces>

[66]: Tanweer ALAM. « Cloud-Based IoT Applications and Their Roles in Smart Cities ». In : *Smart Cities* 4.3 (2021)

[67]: Philipp SCHULZ, Maximilian MATTHE, Henrik KLESSIG, Meryem SIMSEK, Gerhard FETTWEIS, Junaid ANSARI, Shehzad Ali ASHRAF, Bjoern ALMERTH, Jens VOIGT, Ines RIEDEL, Andre PUSCHMANN, Andreas MITSCHLE-THIEL, Michael MULLER, Thomas ELSTE et Marcus WINDISCH. « Latency Critical IoT Applications in 5G : Perspective on the Design of Radio Interface and Network Architecture ». In : *IEEE Communications Magazine* 55.2 (2017)

[68]: Amit Kumar SIKDER, Abbas ACAR, Hidayet AKSU, A. Selcuk ULUGAC, Kemal AKKAYA et Mauro CONTI. « IoT-enabled smart lighting systems for smart cities ». In : *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*. 2018

des SMS. Les systèmes de surveillance avec caméras sont caractérisés par un nombre modéré de nœuds, mais une fréquence de messages élevée, les messages étant des vidéos prises par les caméras. Quant aux applications de réalité augmentée et virtuelle, elles utilisent un petit nombre de nœuds, une fréquence de messages élevée et échangent des vidéos. En ce qui concerne la Voix sur IP (VoIP) dans l'Internet des Objets, elle fonctionne avec un faible nombre de nœuds et une fréquence de messages modérée, les messages véhiculant de la voix. Nous avons simulé l'échange d'images par des paquets de plus grande taille, comparativement aux applications qui échangent des fichiers simples. Une fois la caractérisation effectuée, nous déterminons les paramètres des simulations pour chaque type d'application. Ces paramètres sont résumés dans la Table 2.1. De plus, la durée des expérimentations est ajustée en fonction de la fréquence d'échange de paquets.

2.2.2 Dispositif expérimental

Nous allons maintenant présenter le matériel avec lequel ces données ont été générées, ainsi que les logiciels et protocoles utilisés.

Couche matérielle

Pour l'environnement d'expérimentation, nous avons choisi le FIT IoT-LAB, qui est un banc d'essai dédié à l'IoT [64]. Cette plateforme met à disposition plusieurs modèles de cartes équipées de capteurs et de puces radio, connectées dans une topologie logique maillée. Les cartes sont entièrement ouvertes, ce qui signifie que l'on peut flasher le firmware de notre choix à partir des systèmes d'exploitation pris en charge par le modèle de carte. Cette philosophie d'un banc d'essai totalement ouvert est essentielle si l'on souhaite rendre disponibles les données produites ainsi que la méthodologie de leur création. En effet, lors de la génération de traces, l'accès au matériel peut entraver la reproductibilité, en particulier lorsque du matériel exotique est utilisé. Fournir un accès distant à des machines x86-64 standard pourrait ne pas être utile pour la reproductibilité de la recherche sur les systèmes, car il est facile d'accéder à de telles machines, mais les réseaux sans fil, et en particulier les réseaux IoT, posent un défi différent. Leur évolution rapide signifie que certains matériels peuvent devenir obsolètes en quelques années seulement. De plus, leur diversité signifie que la reproduction de certaines expériences peut s'avérer très coûteuse, que ce soit en termes de matériel ou de temps. Par conséquent, tout chercheur ayant accès au banc d'essai peut reproduire les résultats à moindre coût. De plus, ces bancs d'essai favorisent un meilleur accès à la science en mettant davantage de ressources à la disposition d'un plus grand nombre de personnes, en particulier dans les réseaux IoT qui nécessitent de vastes déploiements. Cela renforce d'autant plus notre choix en faveur de la science ouverte. La plateforme met à notre disposition une API qui permet de réserver à l'avance les cartes pour nos expérimentations, de flasher notre programme, de récupérer les résultats et en général d'interagir avec les expérimentations et les cartes. Plusieurs villes, dont Grenoble et Lille par exemple, présentent des déploiements de cartes selon différentes topologies physiques. En ce qui concerne le matériel, nous avons choisi

[64]: Cedric ADJIH, Emmanuel BACCELLI, Eric FLEURY, Gaetan HARTE, Nathalie MITTON, Thomas NOEL, Roger PISSARD-GIBOLLET, Frederic SAINT-MARCEL, Guillaume SCHREINER, Julien VANDAELE et Thomas WATTEYNE. « FIT IoT-LAB : A Large Scale Open Experimental IoT Testbed ». In : Milan, Italy, déc. 2015

[69]: IoT-LAB M3. <https://www.iot-lab.info/docs/boards/iot-lab-m3/>

[70]: E. CALLAWAY, P. GORDAY, L. HESTER, J.A. GUTIERREZ, M. NAEVE, B. HEILE et V. BAHL. « Home networking with IEEE 802.15.4 : a developing standard for low-rate wireless personal area networks ». In : *IEEE Communications Magazine* 40.8 (2002)

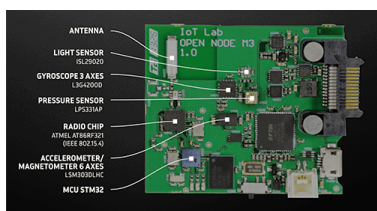


FIGURE 2.1 – Cartes M3 du FIT IoT-LAB [69].

[69] IoT-LAB M3. <https://www.iot-lab.info/docs/boards/iot-lab-m3/>

[71]: A. DUNKELS, B. GRONVALL et T. VOIGT. « Contiki - a lightweight and flexible operating system for tiny networked sensors ». In : *29th Annual IEEE International Conference on Local Computer Networks*. 2004

d'utiliser les cartes ouvertes M3 [69] pour toutes les simulations. Ces cartes sont basées sur les microcontrôleurs STM32 avec un processeur ARM Cortex-M3, car elles répondent aux besoins d'un large éventail d'applications. Elles sont équipées de plusieurs capteurs, dont un capteur de lumière, un capteur de température et de pression, ainsi qu'une puce radio compatible avec la norme IEEE 802.15.4 [70].

Couche logicielle

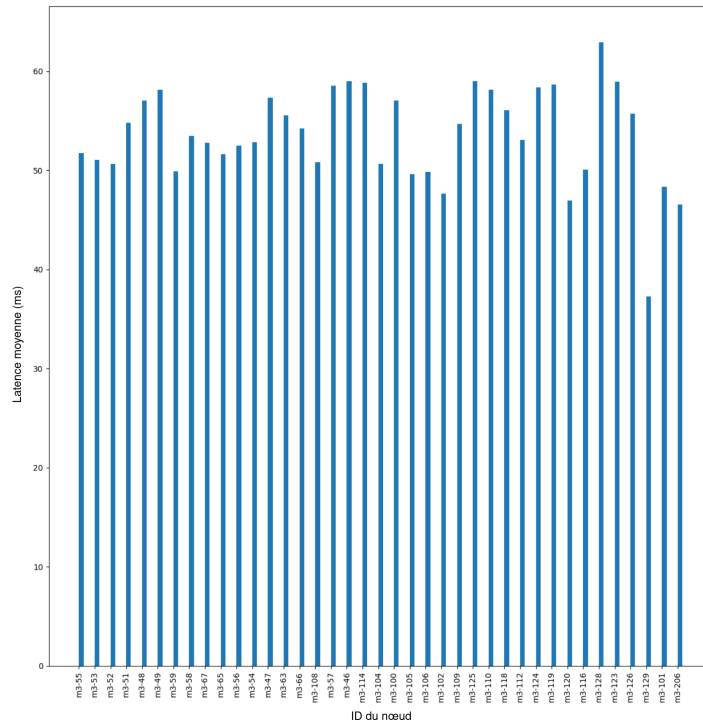
Pour la programmation des nœuds, nous nous sommes basés sur une base de code Contiki [71] que nous avons adaptée à nos besoins. Contiki est un système d'exploitation open-source conçu pour l'IoT et les systèmes embarqués qui sont limités en ressources. Les nœuds communiquent en mode broadcast via le protocole IEEE 802.15.4, ce qui signifie que les paquets envoyés par les nœuds contenant les messages sont adressés à tous les autres nœuds du réseau. Pour le protocole de routage, nous avons opté pour RPL, qui est un protocole de routage standard dans les réseaux de capteurs sans fil et l'Internet des objets. En effet, RPL est conçu pour prendre en compte les ressources limitées en termes de puissance de calcul, de mémoire et d'énergie des objets dans les réseaux IoT. Quant au protocole de communication, nous avons utilisé l'UDP, qui signifie *User Datagram Protocol* en français, Protocole de Datagramme d'Utilisateur. Nous avons ajusté la fréquence d'envoi des messages des nœuds ainsi que la taille de la charge utile des paquets envoyés en fonction des paramètres choisis, en ajoutant également un identifiant de paquet. De plus, nous avons demandé aux nœuds d'écrire en sortie les paquets qu'ils envoient ou reçoivent. La plateforme d'expérimentation nous a fourni les outils nécessaires pour flasher ce code sur les cartes ouvertes M3. Nous avons utilisé l'interface client Python pour programmer les nœuds avec notre code et lancer les simulations. Ensuite, nous avons récupéré les résultats des simulations, c'est-à-dire la sortie des nœuds, à l'aide d'un agrégateur de liens en série. En effet, dans la plateforme, chaque nœud possède une liaison série accessible via une socket TCP, qui permet d'interagir avec le nœud. L'agrégateur est un script Python fourni qui permet de récupérer les résultats de plusieurs nœuds en même temps, permettant ainsi d'agréger les liens en série. Lorsqu'un nœud écrit sa sortie, la plateforme ajoute un horodatage et le nom du nœud, ce qui est utile pour retrouver les informations, et ces données sont disponibles dans l'agrégateur. Tous les scripts utilisés sont disponibles avec les données.

2.2.3 Résultats

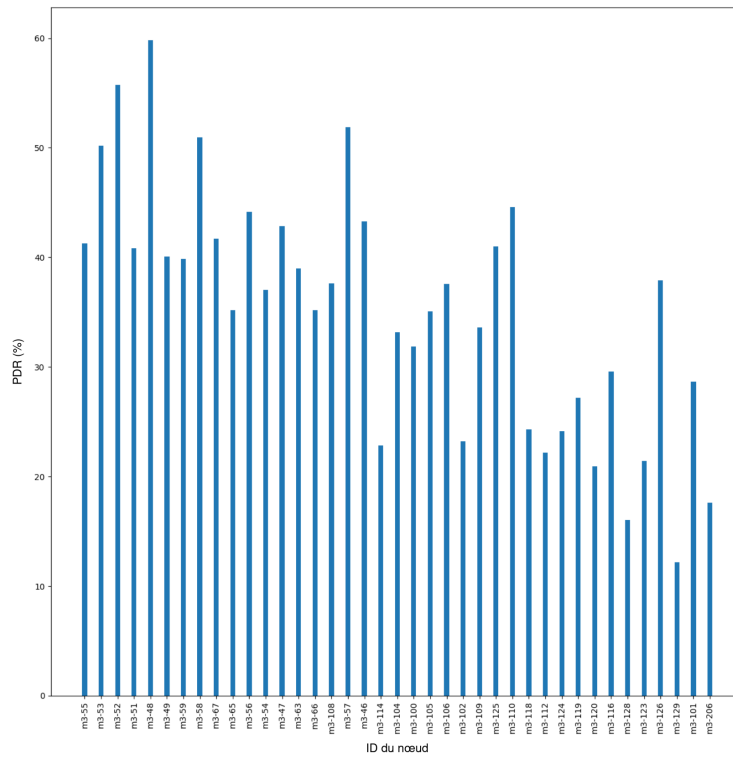
En tant que résultat brute, nous obtenons le trafic généré, c'est-à-dire les échanges de paquets entre les nœuds, consignés dans un fichier où chaque ligne représente une sortie d'un nœud, correspondant à un paquet envoyé mais également reçu pour pouvoir mesurer les paquets perdus. Chaque ligne contient l'horodatage de la transmission ou de la réception du paquet, le nom du nœud récepteur ou émetteur, le contenu du paquet indiquant s'il s'agit d'un message reçu ou transmis, ainsi qu'un identifiant de message, comme illustré dans la Figure 2.2. Pour une meilleure analyse, nous avons divisé ce fichier de données brutes en deux autres fichiers distincts : l'un contenant les messages reçus et l'autre

```
1610708144.171619;m3-48;Sending broadcast;3178
1610708144.209941;m3-114;Sending broadcast;3187
1610708144.216687;m3-59;Received from: fe80::956;3178
1610708144.217259;m3-53;Received from: fe80::956;3178
1610708144.217470;m3-51;Received from: fe80::956;3178
```

FIGURE 2.2 – Résultats bruts de l'expérimentation.

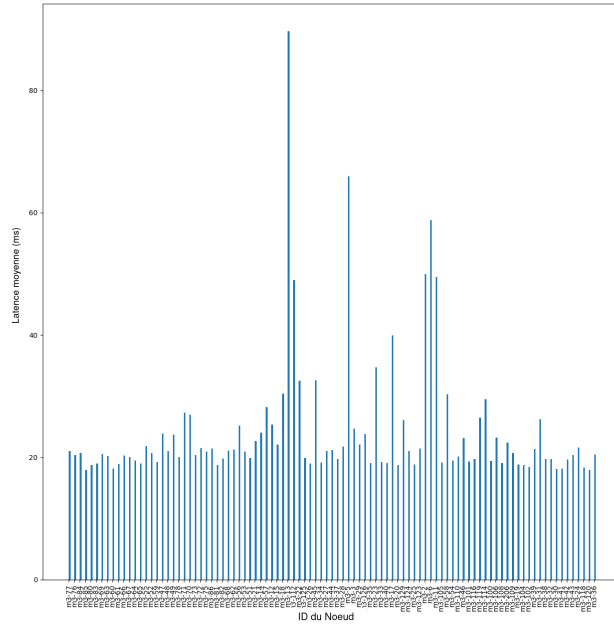


(a) Taux de latence moyenne par nœuds.

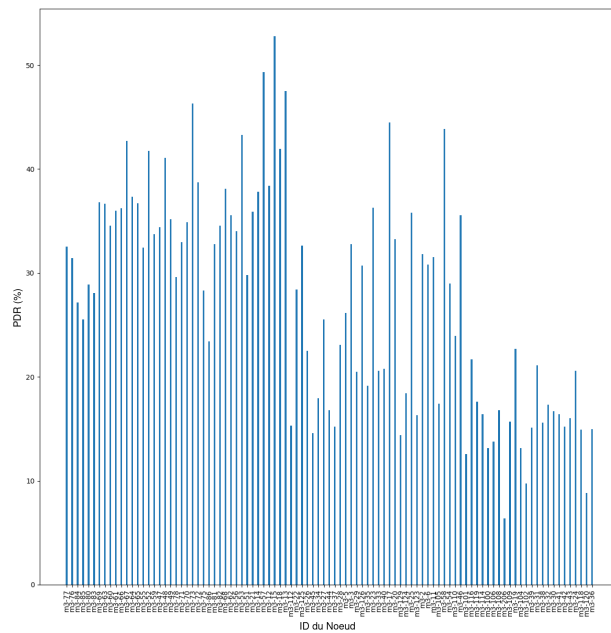


(b) PDR par nœuds.

FIGURE 2.3 – PDR et latence pour l'application d'urgence.



(a) Taux de latence moyenne par noeuds.



(b) PDR par noeuds.

FIGURE 2.4 – PDR et latence pour l'application d'éclairage.

les messages transmis. Ces deux fichiers conservent les informations des données brutes, mais dans le fichier des messages transmis, nous avons également ajouté le nombre de nœuds recevant le message. Dans le fichier des messages reçus, nous avons inclus le délai de réception en millisecondes. Dans notre cas, ces données servent à prédire initialement le délai en fonction de l'état du réseau et de l'application. C'est pourquoi nous avons extrait différentes métriques, telles que le taux de paquets délivrés et le délai moyen de transmission des paquets. La Figure 2.3 illustre la latence moyenne de transmission et le taux de livraison de paquets par nœud pour l'application d'urgence. La Figure 2.4 illustre les mêmes métriques pour l'application des éclairages intelligents. Si l'on se focalise sur le taux de latence moyen par nœud pour les deux applications, on peut constater que certains nœuds de l'application d'éclairage affichent une latence moyenne significativement plus élevée que les autres nœuds de cette application, ainsi que par rapport aux nœuds de l'application d'urgence. En revanche, les autres nœuds de l'application d'éclairage, qui ne présentent pas de latence "extrême", ont une latence inférieure à celle des nœuds de l'application d'urgence. Cela s'explique par le nombre plus élevé de nœuds dans l'application d'éclairage, ainsi que par sa topologie, qui conduit certains nœuds à se trouver en périphérie du réseau, entraînant ainsi une latence plus élevée. En ce qui concerne le PDR, l'application d'éclairage affiche un PDR légèrement plus faible que le système d'urgence. Cette différence s'explique également par la densité plus élevée des nœuds dans l'application d'éclairage. Cela met en évidence la nécessité d'obtenir des données provenant de plusieurs types de trafics IoT hétérogènes, chacun ayant ses propres caractéristiques distinctes. Il est intéressant de noter que d'autres métriques peuvent également être extraites et mises en relation. La séparation des données, les métriques et les graphiques présentés dans la Figure 2.3 ont été générés à l'aide d'un code Python, disponible en libre accès sur GitLab. Ce code a été conçu pour être réutilisable afin d'extraire d'autres métriques, telles que la fenêtre de réception des messages, ou pour travailler sur d'autres données présentant le même format.

2.2.4 Conclusion et discussion

Nous avons développé une méthodologie pour générer des traces applicatives de l'IoT à l'aide de la plateforme FIT IoT-LAB. Ces applications couvrent différents cas d'utilisation et types de trafic. Pour commencer, nous avons caractérisé le trafic de ces applications afin de définir les paramètres de simulation appropriés. Les traces ainsi générées ont été enrichies d'informations telles qu'un identifiant de message et un horodatage. Cela nous a permis d'extraire des métriques pertinentes, telles que le délai de réception, le taux de paquets perdus et la latence. Nous utilisons ces données pour tenter de prédire le débit dans un réseau IoT dans le Chapitre 3.3 afin de distribuer au mieux et en conséquence les ressources *Edge*. Il convient de noter plusieurs limitations de ces données. Elles pourraient être enrichies en ajoutant des informations supplémentaires, comme l'utilisation du CPU ou la structure de l'arbre RPL. De plus, certaines métadonnées nécessaires à la reproduction des expérimentations ne sont pas automatiquement fournies dans les résultats, par exemple, le site d'expérimentation. Les utilisateurs doivent rechercher ces métadonnées par eux-mêmes à la fin de leur expérience.

Enfin, les données sont actuellement stockées sous forme de fichiers texte simple. Une représentation plus avancée des résultats, telle que la mise à disposition d'une base de données, pourrait simplifier l'exploitation des résultats. Nous reprenons ces travaux par la suite afin de les enrichir et de les améliorer.

2.3 Génération automatisée et reproductible de données

Dans la Section 2.3, nous avons proposé des données IoT générées de manière ouverte pour, dans le cadre de cette thèse, pour la prédiction du débit. Cependant, nous avons également mis à disposition ces traces applicatives ainsi que les outils permettant leur génération, dans le but de favoriser la science ouverte et la reproductibilité de la recherche. Cette première étape a ouvert la voie à plusieurs développements futurs :

(i) En ce qui concerne les *données*, nous constatons qu'il est nécessaire d'introduire des couches de complexité supplémentaires pour mieux refléter les réseaux sans fil et Edge, en particulier pour permettre une prédiction multimétrique de la qualité de service.

(ii) En ce qui concerne la *génération de données*, notre objectif est de proposer un outil complet à la disposition de la communauté de la recherche, favorisant ainsi une génération d'expérimentations automatisée et reproductible.

En effet, lors de notre travail sur le point (i), nous avons constaté qu'il est particulièrement difficile de reproduire ou, encore plus complexe, de répéter la recherche et les expériences dans le domaine des réseaux sans fil, et plus généralement dans le domaine des systèmes informatiques. C'est pourquoi nous allons d'abord présenter un outil de génération automatisée et reproductible de traces que nous appelons *Sisyphé*, suivi d'un nouvel ensemble de données d'application IoT enrichies, produites grâce à *Sisyphé*.

2.3.1 Sisyphé : Outil pour la génération de données automatisée et reproductible

Pourquoi proposer un nouvel outil ?

Dans [72], les auteurs ont mesuré les deux exigences les plus "fondamentales" pour la reproduction, à savoir la possibilité d'accéder au code source et la possibilité de construire ce code source, pour un ensemble de 601 articles provenant de diverses conférences et revues de l'ACM. Sur les 402 articles qui n'ont pas été exclus pour des raisons de disponibilité du matériel, ou qui ont été exclus parce que leurs résultats n'étaient pas étayés par un code, seuls 54,0% remplissent les deux conditions de base, à savoir la disponibilité du code source et la réussite de la construction. Même si l'on espère que la situation est bien moins désastreuse de nos jours qu'en 2015, la première étape pour garantir la reproductibilité et la répétabilité est d'assurer la disponibilité du code source impliqué dans les étapes de traitement, d'analyse, de présentation et de création des

[72]: Christian COLLBERG et Todd A. PROEBSTING. « Repeatability in Computer Systems Research ». In : *Commun. ACM* 59.3 (fév. 2016)

travaux scientifiques. Cela inclut également le code expérimental, qui doit être accessible à tout moment. Nous nous sommes donc intéressés à l'existence d'outils de génération automatique d'artefacts d'expériences. Software Heritage est au cœur de la partie archivage du code source de Sisyphe [73], mais ne vise pas à archiver les résultats d'expériences ou les artefacts binaires, tels que les firmwares, ni à fournir l'automatisation des expériences, de la construction à la récupération des résultats, contrairement à Sisyphe. De même, les travaux [52-55], présentés dans la Section 2.1, ne visent pas à fournir une génération automatique et reproductible de données, indépendante d'un banc d'essai. À notre connaissance, il n'existe pas d'outil tel que celui que nous proposons ici.

Principes de conception

L'objectif principal de Sisyphe est de fournir un moyen de générer des artefacts d'expériences de manière autonome, répétable et reproductible. Pour atteindre cet objectif, Sisyphe repose sur plusieurs techniques et principes de conception que nous présentons ici.

Archivage du code source : Pour résoudre le problème de la disponibilité du code source, nous avons pris plusieurs mesures. Tout d'abord, le code source de la plateforme Sisyphe ainsi que le code source des expériences réalisées sur la plateforme sont distribués sous licence libre, ce qui permet à quiconque d'accéder au code source et de l'utiliser. Cependant, simplement distribuer le code source en le hébergeant sur une page web institutionnelle ou personnelle n'est pas suffisant. Il existe des risques que le code source disparaisse en cas de refonte du site web ou de changement d'affiliation. Pour garantir la disponibilité à long terme du code source, nous avons choisi de nous appuyer sur l'archivage des logiciels, rendu possible par Software Heritage [74]. Software Heritage est une organisation à but non lucratif soutenue par l'Inria et l'UNESCO, dont la mission est d'archiver à long terme les logiciels et les codes sources. Pour assurer la disponibilité du code source de Sisyphe, nous attribuons des identifiants uniques et stables appelés SWHID à différents composants logiciels de la plateforme. Ces SWHID permettent d'accéder facilement au code source de chaque composant de Sisyphe. Par exemple, l'identifiant du code source de Sisyphe, disponible sur <https://github.com/sisyphe-re/infrastructure-ng>, est `swh :1 :dir :6b23bcefe73bbaea308b6ea99d7e75cd37ded16fe`. Ces identifiants SWHID permettent aux utilisateurs d'explorer le code source d'un projet de manière similaire à l'interface web de Github. En plus de préserver le code source de la plateforme, nous utilisons également Software Heritage pour archiver le code source des expériences à chaque fois qu'elles sont exécutées de manière automatisée. Avant d'exécuter une expérience, nous nous assurons que le code source associé est présent dans les archives de Software Heritage. Cette approche garantit qu'il n'y a pas de décalage entre l'expérience en cours et ce qui est archivé à long terme, assurant ainsi la disponibilité et la reproductibilité du code source des expériences.

Builds reproductibles : Tout chercheur qui a déjà essayé d'exécuter un vieux code qu'il a écrit pour une contribution scientifique sait à quel point cela peut être difficile. L'accès au code source n'est qu'une partie

[73]: Roberto Di COSMO, Morane GRUENPETER et Stefano ZACCHIROLI. « Referencing Source Code Artifacts : a Separate Concern in Software Citation ». In : *Computing in Science and Engineering* (déc. 2019)

[52]: Tamas PFLANZNER, Zoltan FEHER et Attila KERTESZ. « A Crawling Approach to Facilitate Open IoT Data Archiving and Reuse ». In : *2019 Sixth International Conference on Internet of Things : Systems, Management and Security (IOTSMS)*. 2019

[53]: M. KUZMAN, X. d. TORO GARCÍA, S. ESCOLAR, A. CARUSO, S. CHESSA et J. C. LÓPEZ. « A Testbed and an Experimental Public Dataset for Energy-Harvested IoT Solutions ». In : *IEEE Int. Conference on Industrial Informatics (INDIN)*. 2019

[54]: Z. XIAO, F. LI, R. WU, H. JIANG, Y. HU, J. REN, C. CAI et A. IYENGAR. « TrajData : On Vehicle Trajectory Collection With Commodity Plug-and-Play OBU Devices ». In : *IEEE Internet of Things Journal* 7 (2020)

[55]: R. JACOB, R. Da FORNO, R. TRUB, A. BIRI et L. THIELE. « Dataset : Wireless Link Quality Estimation on FlockLab - and Beyond ». In : *Proceedings of the 2nd Workshop on Data Acquisition To Analysis*. 2019

[74]: Roberto Di COSMO et Stefano ZACCHIROLI. « Software Heritage : Why and How to Preserve Software Source Code ». In : *iPRES 2017 - 14th International Conference on Digital Preservation*. Kyoto, Japan, sept. 2017

[75]: J. M. PERKEL. « Challenge to Scientists : Does Your Ten-Year-Old Code Still Run ? » In : *Nature* 584.7822 (2020)

[76]: E. DOLSTRA. « The Purely Functional Software Deployment Model ». Thèse de doct. Utrecht University, 2006

[77]: Pierre BRUNISHOLZ, Etienne DUBLÉ, Franck ROUSSEAU et Andrzej DUDA. « WalT : A reproducible testbed for reproducible network experiments ». In : *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 2016

du "défi de la reproductibilité", comme le soulignent les résultats du "Ten Years Reproducibility Challenge" présentés dans [75] : le manque de documentation ou l'accès à des environnements informatiques obsolètes peuvent également représenter des obstacles difficiles à surmonter. Dans le cadre de Sisyphe, nous avons pris des mesures pour garantir la reproductibilité non seulement du code source, mais aussi de l'environnement de développement et de calcul. Pour cela, nous utilisons le gestionnaire de paquets Nix [76] et la distribution Linux NixOS. Nix adopte une approche fonctionnelle de la gestion des paquets en reliant les logiciels et leurs dépendances à l'aide de hachages cryptographiques. Cela nous permet de fournir des environnements de développement et de calcul reproductibles. Les tiers peuvent réutiliser notre code sans avoir d'impact sur le reste de leur système. Par exemple, ils peuvent construire les firmwares basés sur RIOT-OS (de la Section 2.3.2) en utilisant une seule commande, 'nix build', ce qui élimine les problèmes liés à la gestion des dépendances. De plus, l'architecture globale de Sisyphe est elle-même écrite en Nix et repose sur le système d'exploitation NixOS. Cette approche suit la philosophie "infrastructure en tant que code", ce qui signifie que l'ensemble de l'infrastructure est décrite de manière reproductible dans le code. Nix et NixOS s'efforcent d'être reproductibles jusqu'au niveau du bit, ce qui signifie qu'ils peuvent rapidement détecter les projets existants présentant des schémas non déterministes dans leurs constructions. Nous avons même découvert un exemple de tel comportement non déterministe dans le système d'exploitation RIOT pendant le développement de Sisyphe, ce qui souligne l'importance de la reproductibilité à ce niveau. En résumé, Sisyphe utilise Nix et NixOS pour garantir la reproductibilité non seulement du code source, mais aussi de l'environnement de développement et de calcul, ce qui facilite grandement la réutilisation et la reproduction des expériences.

Bancs d'essai : Comme motivé dans la Section 2.1, nous réutilisons le FIT IoT-Lab comme banc d'essai pour produire les données. Les bancs d'essai entrent directement dans la philosophie de reproductibilité en rendant rapidement accessibles les environnements d'expérimentation pour le plus grand nombre dans un environnement contrôlé. Cependant, l'un des inconvénients de ces bancs d'essai accessibles à distance est qu'ils offrent des conditions de fonctionnement rigides et peuvent, dans certains cas, ne pas fournir de résultats significatifs en raison de leurs déploiements particuliers, qui peuvent être considérés comme artificiels [77]. À cette fin, Sisyphe est agnostique en matière de bancs d'essai et n'essaie pas d'imposer l'utilisation d'un banc d'essai spécifique, tant que les expériences peuvent être menées de manière automatisée, ce qui nous amène au prochain et dernier paradigme de conception principal de Sisyphe.

Intégration continue et automatisation : L'environnement radio évolue rapidement avec l'évolution de l'attribution du spectre radio, les changements dans les habitudes des utilisateurs, les changements dans la technologie de communication, les changements dans les applications, les systèmes d'exploitation. Ainsi, Sisyphe adopte une approche d'intégration continue et d'automatisation complète pour la création d'ensembles de données. Cette automatisation permet d'identifier les tendances globales des réseaux IoT, par exemple en ce qui concerne la performance de tel ou tel algorithme mis en œuvre dans tel ou tel système d'exploitation. Elle permet ainsi de mieux comprendre l'évolution des réseaux réels.

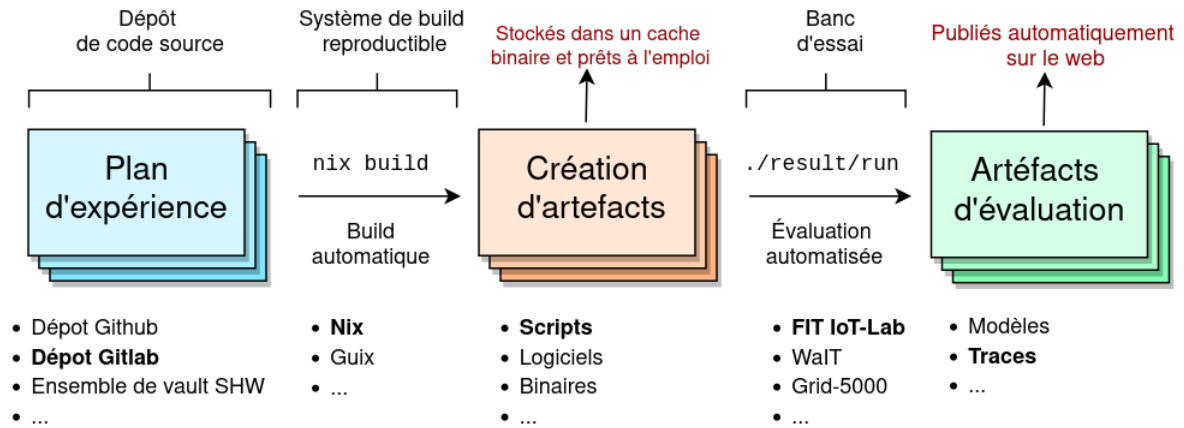


FIGURE 2.5 – Vue d'ensemble de l'architecture de Sisyphus. Les éléments en gras sont utilisés dans ce chapitre.

Plus important encore, l'automatisation complète dissocie l'expérience de l'expérimentateur et met en évidence les dépendances implicites qui pourraient être cachées. En effet, le travail initial d'automatisation est une forme de documentation. Il permet d'observer l'ensemble du processus expérimental et réduit ainsi le risque de *"l'amnésie de l'entreprise"* et facilite les audits.

Architecture

L'architecture globale de Sisyphus est présentée dans la Figure 2.5. Le pipeline de traitement est divisé en deux étapes principales, chacune ayant un niveau de reproductibilité différent. La première étape est l'étape de construction automatisée, qui utilise des plans d'expérience provenant de dépôts de code source pour créer des *artefacts*. Ces artefacts sont ensuite utilisés lors de la deuxième étape, qui est l'étape d'évaluation automatisée. Sisyphus a construit des plans avec des flocons Nix (Nix "flakes" en anglais), qui sont des fonctionnalités expérimentales visant à faciliter la conteneurisation et la composabilité des projets basés sur Nix [78]. Cela suppose que les blueprints sont déjà prêts pour Nix et les flakes. Dans le cas où ils ne le sont pas, un peu de travail peut être nécessaire, comme le montre la suite de ces travaux, et un code d'adaptateur intermédiaire est nécessaire. Sisyphus suppose que l'appel à "nix build" construira les artefacts et exposera un exécutable "run" pour lancer l'expérience. Les résultats de la construction sont publiés en ligne dans un cache binaire, où ils sont identifiés par leurs hachages, ce qui permet de récupérer une partie des artefacts sans avoir à les reconstruire localement. L'étape d'évaluation automatisée est généralement moins reproductible que l'étape de construction automatisée. En effet, pour évaluer une expérience, nous devons faire des appels d'API et interagir avec des ressources réelles, ce qui n'est pas aussi simple que d'exécuter une séquence de programmes dans un environnement contrôlé. Cette étape d'évaluation est automatiquement exécutée sur Sisyphus à intervalles réguliers, par exemple quotidiennement, et les résultats sont également disponibles sur le web. Nous pouvons avoir besoin d'informations secrètes, appelées *secrets*, pour nous connecter à des plateformes telles que Fit IoT-LAB, par exemple des clés privées SSH pour copier des artefacts sur le serveur

[78]: E. DOLSTRA. *Nix Flakes, Part 1: An Introduction and Tutorial*. Tweag

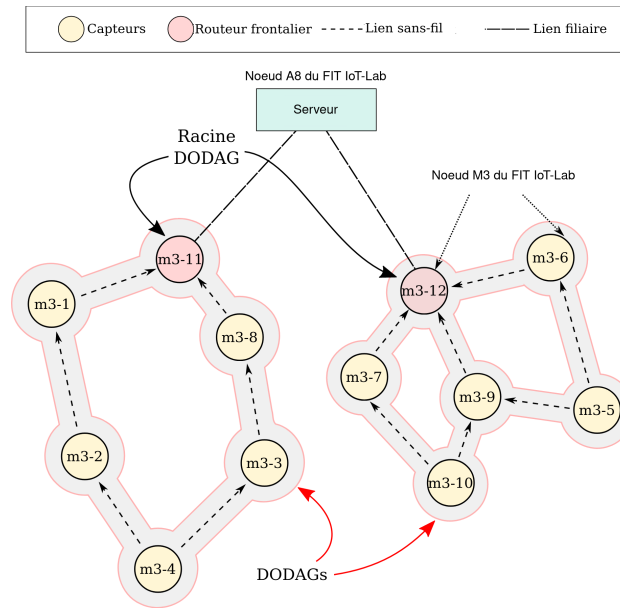


FIGURE 2.6 – Architecture de l'expérimentation.

TABLE 2.2 – Paramètres de l'expérimentation.

Scénario	Nb de capteurs	Nombre de routeurs frontaliers	Taille de la charge utile (octets)	Type de génération de paquets	lambda	Période (s)
Systèmes CVC	100	5	60	Périodique	-	260.0
Éclairage intelligent	100	5	30	Exponentielle	0.00208	-
Intervention d'urgence	40	5	127	Hybride	0.0333	30.0
Réalité virtuelle ou augmentée	10	3	127	Exponentielle	196.74	-
VoIP	10	3	127	Hybride	15.74	.063532

Sisyphe pour les rendre disponibles, des clés API ou des mots de passe. Ces *secrets* sont transmis en tant que variables d'environnement. Les conteneurs sont utilisés pour séparer l'évaluation de chaque expérience, mais nous prévoyons de passer à des machines virtuelles dans de futurs travaux afin d'augmenter le nombre d'expériences.

2.3.2 Données d'applications IoT enrichies

Dans cette section, nous nous concentrons sur la création d'un ensemble de données plus réaliste pour l'évaluation des réseaux. Ce type d'ensemble de données est utile pour comprendre les profils de trafic des applications. Nous commencerons par décrire l'architecture et la configuration de l'expérience. Nous expliquerons également comment nous générons le trafic et les paramètres de l'expérience. Ensuite, nous présenterons les résultats obtenus et les analyserons.

Description de l'expérience

Nous réutilisons les scénarios de la Section 2.2, mais nous les améliorons en tenant compte de différents types de génération de paquets. Comme notre focus est l'évaluation des réseaux, nous présentons une architecture plus avancée, illustrée par la Figure 2.6. Chaque expérience, correspondant à un scénario, implique trois types d'entités (nœuds) :

Serveur : Le serveur est un nœud qui héberge un serveur UDP, recevant des paquets et collectant des informations sur ces paquets. Il est accessible sur Internet via une connexion IPv6. Dans toutes les expériences, il n'y a qu'un seul serveur.

Routeurs frontaliers : Les routeurs frontaliers sont des nœuds qui relient les capteurs à Internet. Ils sont équipés de deux interfaces, l'une connectée à Internet et l'autre au réseau de capteurs, utilisant la technologie 802.15.4 et le protocole de routage RPL. Dans le protocole RPL, ils servent de racine aux DODAG RPL. Ils se trouvent à la périphérie du réseau de capteurs et connectent deux réseaux. Le nombre de routeurs frontaliers reste constant au cours des expériences, mais peut varier d'une expérience à l'autre, comme indiqué dans la Table 2.2.

Capteurs : Les capteurs sont le cœur du réseau. Ils génèrent des données selon une distribution aléatoire et des paramètres de transmission spécifiés, puis les envoient au serveur. Ils sont interconnectés grâce à la technologie IEEE 802.15.4 et utilisent le protocole de routage RPL. Les capteurs maintiennent une table de voisinage et des métriques de coût pour choisir le meilleur parent dans l'arbre de routage, puis relaient les paquets de données le long de l'arbre vers la racine du DODAG.

Nous introduisons un nouveau type de génération de paquets qui va au-delà de la génération de paquets réguliers et périodiques, permettant ainsi une meilleure modélisation des scénarios du monde réel. En effet, dans le scénario CVC, par exemple, les capteurs peuvent envoyer des informations de manière périodique pour contrôler le système. En revanche, dans un scénario d'intervention d'urgence, les capteurs réagissent également lorsqu'ils détectent un danger potentiel, ce qui ne peut pas être pris en compte par une simple génération de trafic périodique. Par conséquent, en plus de la génération périodique précédemment utilisée, où les paquets sont envoyés à des intervalles de temps fixes (appelée "périodique"), nous utilisons également une méthode de génération de paquets basée sur une distribution exponentielle du paramètre de taux λ (appelée "exponentielle"). Cette distribution est générée en utilisant la méthode d'échantillonnage par transformation inverse directement sur le nœud du capteur, en utilisant la génération de nombres aléatoires en virgule flottante du système d'exploitation RIOT. Étant donné que les nœuds FIT IoT-Lab M3 sont basés sur les CPU STM32, qui disposent d'un générateur de nombres aléatoires matériel, nous estimons que la distribution générée est d'une qualité suffisante pour nos cas d'utilisation. De plus, nous avons mis au point une méthode de génération de trafic hybride, qui consiste à générer simultanément des données selon la méthode périodique et la méthode exponentielle. Tous les paramètres sont résumés dans la Table 2.2.

TABLE 2.3 – Caractéristiques brutes des données produites.

Données	Caractéristique	Description
Donnée transmise (UDP)	id_noeud	id du capteur
	horodatage	heure de transmission du paquet
	taille_charge	taille de la charge utile UDP (octets)
	addr_dest	adresse IPv6 de la destination
	port_dest	port de destination
Donnée reçue (Serveur)	charge utile	identifiant hexadécimal du paquet
	horodatage	heure de réception du paquet
	addr_IPv6	adresse IPv6 de la source
	port_recv	port sur lequel le paquet est reçu
	charge utile	identifiant hexadécimal du paquet

```

$ sqlite3 hvac_263685.db3
SQLite version 3.35.2 2021-03-17 19:07:21
Enter ".help" for usage hints.
sqlite> SELECT udp.Timestamp, udp.Node, server.Timestamp FROM server
... > INNER JOIN udp ON server.payload=udp.payload LIMIT 5;
2021-05-06 16:34:22.145909070|ms-377|2021-05-06 16:34:22.170198
2021-05-06 17:09:02.2880198925|ms-367|2021-05-06 17:09:02.308101
2021-05-06 17:22:02.258351087|ms-379|2021-05-06 17:22:02.296746
2021-05-06 18:31:22.448118925|ms-42|2021-05-06 18:31:22.461107
2021-05-06 18:57:22.298322916|ms-362|2021-05-06 18:57:22.324690

```

FIGURE 2.7 – Exemple simple de manipulation des données produites.

[79]: ZSTD compression algorithm

[80]: Sisyphus database schema documentation

2.3.3 Résultats

Les résultats sont collectés soit à partir du serveur, soit à partir des capteurs, par un script exécuté sur les serveurs frontaliers du FIT IoT-Lab. En particulier, les routeurs frontaliers ne collectent aucune statistique pour éviter que la collecte de données n’interfère avec l’expérience, étant donné que leur liaison série est déjà utilisée pour la connexion Ethernet-sur-série (ethos). Les résultats de chaque scénario sont transmis et compressés à l’aide de l’algorithme de compression ZSTD8 [79] directement vers le serveur Sisyphus, où ils peuvent être post-traités après la fin de l’expérience, en particulier pour les formater. Initialement au format texte, les résultats sont transformés en bases de données SQLite3 pour permettre une manipulation et une interrogation aisées. Pour l’instant, nous avons conservé le format de la base de données simple, sans utiliser de relations entre les tables, en utilisant SQLite3 comme un format de fichier binaire. Ces bases de données sont également compressées à l’aide de ZSTD, car leur taille non compressée peut atteindre 100 Go pour une expérience d’une semaine. Toutes les caractéristiques brutes des résultats sont résumées et expliquées dans la Table 2.3. Les bases de données comprennent des tableaux contenant différents niveaux d’informations, depuis les statistiques de haut niveau de la couche 3 jusqu’aux informations de bas niveau de la couche physique, par exemple la puissance de réception moyenne des trames de tel ou tel voisin dans le RPL DODAG. Nous n’entrerons pas dans les détails du schéma de la base de données, qui est documenté dans le référentiel de campagne [80], mais nous présentons dans la Figure 2.7 un exemple d’opération sur les données. Ces opérations nous donnent l’horodatage de la transmission, l’ID du capteur et l’horodatage de la réception pour les 5 premiers paquets reçus de l’ensemble de données.

2.3.4 Analyse statistique des résultats

Dans cette section, nous fournissons une analyse explicative de l’ensemble de données généré ainsi que de ses propriétés. Pour cette analyse, nous nous appuyons sur plusieurs métriques :

Horodatage : Il s’agit du temps associé à chaque paquet dans le réseau. Dans l’ensemble de données brutes la granularité est de l’ordre de la

TABLE 2.4 – Caractéristiques statistiques des ensembles de données sur le débit et la latence pour toutes les applications IoT

QoS		Débit			Latence		
Métriques	Moyenne	S.D	Médiane	Moyenne	S.D	Médiane	
CVC	0.253958	0.204743	0.190762	0.044852	0.058928	0.029333	
VoIP	0.323644	0.108589	0.304021	0.004661	0.001512	0.004562	
Éclairages	0.164938	0.185021	0.094540	0.041879	0.075346	0.011554	
Urgences	0.061513	0.061542	0.043207	0.066475	0.073212	0.031021	
Surveillance	0.337204	0.167844	0.330451	0.380338	4.870949	0.000468	

TABLE 2.5 – Caractéristiques statistiques des ensembles de données sur le PDR et le PLR pour toutes les applications IoT

QoS		PDR			PLR		
Métriques	Moyenne	S.D	Médiane	Moyenne	S.D	Médiane	
CVC	0.331391	0.390711	0.000033	0.232914	0.237775	0.124958	
VoIP	0.532645	0.207119	0.548382	0.501071	0.060243	0.494923	
Éclairages	0.100024	0.226347	0.000000	0.038439	0.101123	0.000000	
Urgences	0.258011	0.226774	0.173908	0.134196	0.127102	0.085104	
Surveillance	0.290765	0.126342	0.294597	0.079721	0.128798	0.035209	

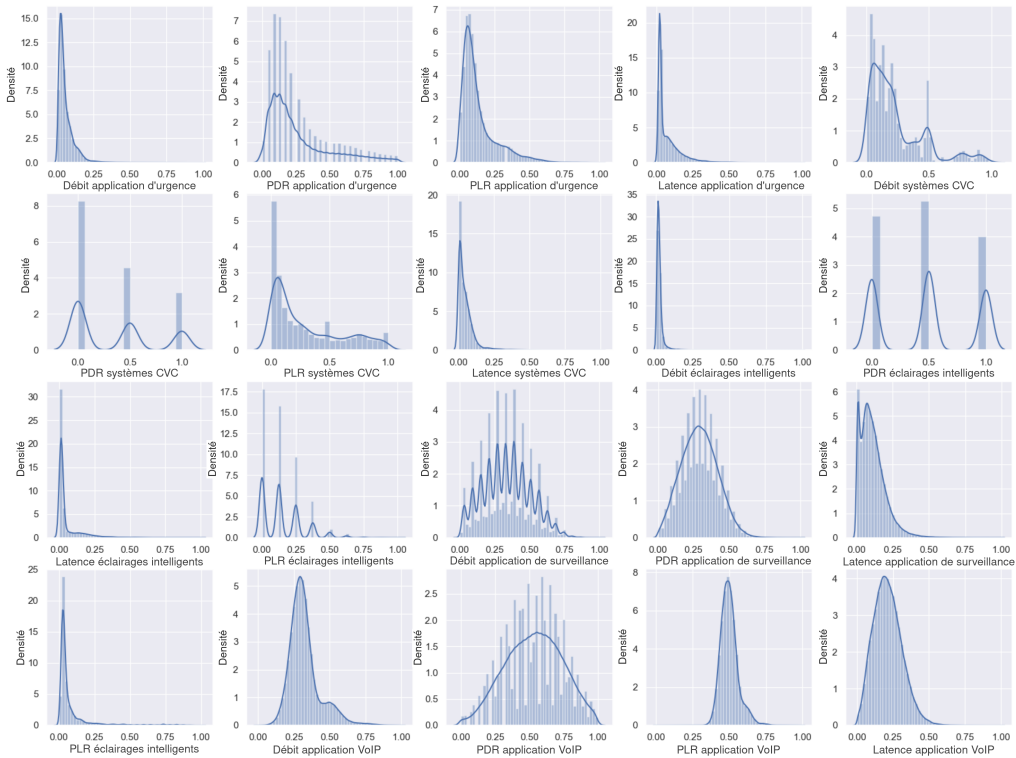


FIGURE 2.8 – Graphiques de distribution de probabilité des données de qualité de service pour toutes les applications IoT.

nanoseconde. Cependant, la granularité est pour l'analyse modifiée à 5 millisecondes, afin de mieux capturer les fluctuations des mesures de qualité de service. Par exemple, il n'a pas toujours été possible de calculer les mesures de qualité de service pour chaque nanoseconde car, dans la plupart des cas, aucun paquet n'a été envoyé ou reçu sur le réseau, ce qui a entraîné la génération de nombreuses valeurs nulles pour les ensembles de données de qualité de service. Ainsi, chacune des caractéristiques décrites ci-dessous est calculée pour un intervalle de temps t de 5 millisecondes sans toutefois perdre d'informations significatives.

temps_{premier paqt} : Il s'agit de l'heure à laquelle le premier paquet est transmis dans un intervalle de temps spécifique de 5 ms.

temps_{dernier paqt} : Il s'agit de l'heure à laquelle le dernier paquet est transmis au serveur dans un intervalle de temps spécifique de 5 ms.

total_{trans paqt} : Il s'agit du nombre total de paquets transmis par un nœud pendant un intervalle de temps spécifique de 5 ms.

total_{rec paqt} : Il s'agit du nombre total de paquets reçus par le serveur pendant un intervalle de temps spécifique de 5 ms.

Débit : Il s'agit du taux du nombre total de paquets reçus (ou de leur taille) sur une période de temps. La formule de débit utilisée par tranche de 5 millisecondes est la suivante :

$$\text{Débit} = \frac{\text{total}_{\text{rec paqt}}}{\text{temps}_{\text{dernier paqt}} - \text{temps}_{\text{premier paqt}}} \quad (2.1)$$

Taux de paquets délivrés (PDR) : Le PDR est le taux de paquets effectivement arrivés à leur destination sur le nombre total de paquets transmis par les nœuds émetteurs. Cette caractéristique est cruciale car elle permet de mesurer si le réseau a des difficultés à acheminer les paquets correctement, ce qui peut résulter, par exemple, d'une saturation du réseau (et donc d'un besoin de ressources supplémentaires). Plus le PDR est élevé, plus le réseau est fiable. Sa formulation mathématique est la suivante :

$$\text{PDR} = \frac{\text{total}_{\text{rec paqt}}}{\text{total}_{\text{trans paqt}}} \times 100 \quad (2.2)$$

Taux de paquets perdus (PLR) : Le PLR est essentiellement l'inverse du PDR, c'est-à-dire le nombre de paquets non arrivés à destination sur le nombre total de paquets transmis. Le PLR est également une caractéristique importante qui mesure la fiabilité du réseau. De manière évidente, plus le PLR est faible, meilleure est la fiabilité du réseau. Sa formulation mathématique est la suivante :

$$\text{PLR} = \frac{\text{total}_{\text{perdu paqt}}}{\text{total}_{\text{rec paqt}}} \times 100 \quad (2.3)$$

Latence de transmission : La latence est le temps moyen de transmission d'un paquet de sa source à sa destination. La latence est une caractéristique

qui indique la qualité du réseau, une latence réduite signifiant une meilleure qualité de service. Sa formule est la suivante :

$$\text{Latence} = \frac{\sum_{t_{\text{paqt}}} (\text{temps}_{\text{rec paqt}} - \text{temps}_{\text{trans paqt}})}{\text{total}_{\text{trans paqt}}} \quad (2.4)$$

Les propriétés statistiques des données pour le débit et la latence, puis le PDR et le PLR sont résumées dans la Table 2.3.4 et la Table 2.3.4 respectivement. Dans la Figure 2.8, les graphiques de densité pour chacune des métriques de la QoS au sein de chaque ensemble de données sont également présentés. Les graphiques de densité sont utilisés pour observer la distribution des ensembles de données sur un intervalle continu. Pour l'application d'urgence, nous avons une distribution positivement asymétrique pour les quatre métriques de la QoS, cela est dû au fait que la moyenne dans les ensembles de données de débit, de PDR, de PLR et de latence est supérieure à leurs valeurs médianes. Pour l'application CVC, le débit, le PLR et la latence présentent également une distribution asymétrique, plus précisément une asymétrie à droite. Cependant, le PDR présente une distribution multimodale avec trois pics différents. Pour l'application d'éclairage, le débit et la latence des deux ensembles de données sont asymétriques à droite, mais le PDR et le PLR sont tous deux des ensembles de données multimodaux. Pour l'application de surveillance, le débit est multimodal avec plus de 12 modes, le PDR présente une distribution normale, le PLR et la latence sont tous deux asymétriques à droite. Enfin, pour l'application VoIP, nous avons une distribution normale pour les trois métriques de la QoS, c'est-à-dire le débit, le PDR et le PLR. Cependant, l'ensemble de données de latence présente une légère asymétrie à droite, car la moyenne des données de latence, c'est-à-dire 0,004661, est légèrement supérieure à la valeur médiane, c'est-à-dire 0,004562.

2.3.5 Conclusion

Dans cette étude, nous avons présenté Sisyphe, un outil permettant la génération automatisée d'artefacts. L'architecture de Sisyphe automatise la création des divers outils nécessaires à l'évaluation expérimentale sur des plateformes telles que FIT IoT-Lab, ainsi que la génération d'artefacts, ce qui aboutit à un processus entièrement reproductible. Les chercheurs utilisant Sisyphe peuvent facilement répéter leurs expériences selon des calendriers personnalisés, ce qui leur permet, par exemple, de prendre en compte la variabilité temporelle de facteurs environnementaux tels que le trafic réseau. Les résultats obtenus grâce à Sisyphe sont plus facilement vérifiables par leurs pairs, ce qui renforce leur crédibilité. Nous avons utilisé Sisyphe pour générer un nouvel ensemble de données enrichies par rapport à celui présenté dans le Chapitre 2.3 et que nous utilisons dans le Chapitre 3.4. Les artefacts que nous avons présentés sont issus d'applications IoT, mais il est important de noter que Sisyphe peut générer des artefacts pour d'autres types de réseaux, en fonction du processus de construction choisi et des plateformes d'évaluation intégrées dans le pipeline Sisyphe. Enfin, nous avons réalisé une analyse statistique des données en utilisant des métriques standard pour l'IoT. Nous mettons à disposition de la communauté le code source et les outils développés

dans le cadre de ce travail. Notre objectif est de rendre la génération de traces accessible tout en promouvant les principes de la science ouverte et de la reproductibilité de la recherche.

2.4 Génération de traces intégrant des ressources mobiles

Nous avons abordé dans les Chapitres 2.2 et 2.3 deux jeux de données et un outil pour générer automatiquement des traces. Dans ces ensembles de données, tous les nœuds du réseau sont fixes. Cependant, l'un de nos objectifs est de proposer des outils pour le déploiement de ressources fixes mais aussi mobiles. C'est donc dans cette optique que nous souhaitons présenter un ensemble de données contenant des points d'accès Edge mobile. Nous tenons à souligner que nous n'avons pas utilisé l'outil du Chapitre 2.3 dans ce cas précis. En effet, nous avons besoin d'une infrastructure sur place, car les robots utilisés n'étaient pas accessibles depuis l'extérieur du FIT IoT-Lab au moment des travaux et n'étaient donc pas utilisables par Sisyphé.

2.4.1 Caractérisation des applications Edge

Pour cette expérimentation, nous avons conservé les applications de VoIP, de vidéosurveillance et d'urgence. Les autres applications n'ont pas été retenues en raison de notre disponibilité limitée en termes de trois points d'accès mobiles. Dans le cadre de cette expérience, les trois applications génèrent des données en utilisant les différents modes de génération et de distribution de données déjà présentés au Chapitre 2.3. Pour rappel, le mode périodique génère des données à chaque instant temporel i , le mode basé sur des événements produit des données selon une loi exponentielle avec un taux d'occurrence de λ , et le mode hybride combine les modes périodique et basé sur des événements. Vous trouverez un aperçu des paramètres utilisés pour ces applications dans la Table 2.6. De plus, chaque application est associée à un robot mobile unique qui reçoit ces données tout en se déplaçant selon un chemin configuré. Ainsi, les trois applications IoT fonctionnent simultanément dans l'expérience, chacune étant associée à un point d'accès mobile unique. Les robots mobiles embarquent des nœuds capteurs M3 qui reçoivent les données des nœuds fixes émetteurs.

TABLE 2.6 – Paramètres de l'expérimentation pour la génération de données mobiles.

Application	Nb de noeuds	Type de génération	Lambda	Période (s)
VoIP	50	Périodique	0	0.0635
Surveillance	25	Exponentielle	196.74	1
Intervention d'urgence	25	Hybride	0.0333	30

TABLE 2.7 – Configurations de réseaux avec différentes complexités

Configurations	Mobilité	tx_puissance	Cannal	Valeurs des canaux
1	Mobile	0dBm	Différents	{11,16,21}
2	Mobile	12dBm	Différents	{11,16,21}
3	Mobile	0dBm	Identique	{11}
4	Mobile	12dBm	Identique	{11}
5	Statique	0dBm	Différents	{11,16,21}
6	Statique	12dBm	Différents	{11,16,21}
7	Statique	0dBm	Identique	{11}
8	Statique	12dBm	Identique	{11}

2.4.2 Dispositif expérimental

Couches matérielle et logicielle

Pour ces données, nous réutilisons les cartes IoT-LAB M3 [69] comme nœuds fixes chargés de la génération de données. Nous programmons ces cartes à l'aide de RIOT2 [14], un système d'exploitation open source conçu pour les nœuds IoT. Les cartes M3 sont statiques et n'ont donc pas besoin de se déplacer. Cependant, pour introduire la mobilité dans l'expérience, nous utilisons des robots TurtleBot2 [81] pour transporter les nœuds capteurs que nous souhaitons rendre mobiles. Les robots TurtleBot2 sont contrôlés via ROS [82], qui fonctionne sur le système d'exploitation Ubuntu. Nous programmons les nœuds statiques à travers l'API du FIT IoT-LAB et collectons également les résultats de la même manière que présenté dans le Chapitre 2.3. Les nœuds communiquent ici encore via le protocole IEEE 802.15.4 en mode broadcast et utilisent le routage RPL.

[14]: Emmanuel BACCELLI, Cenk GUNDOĞAN, Oliver HAHM, Peter KIETZMANN, Martine LENDERS, Hauke PETERSEN, Kaspar SCHLEISER, Thomas C. SCHMIDT et Matthias WAHLISCH. « RIOT : An Open Source Operating System for Low-End Embedded Devices in the IoT ». In : *IEEE Internet of Things Journal* 5.6 (déc. 2018)

[81]: *TurtleBot 2*

[82]: Morgan QUIGLEY, Ken CONLEY, Brian GERKEY, Josh FAUST, Tully FOOTE, Jeremy LEIBS, Rob WHEELER et Andrew Y. NG. « ROS : an open-source Robot Operating System ». In : *ICRA workshop on open source software*. T. 3. 3.2. 2009

Architecture et topologie

Pour l'architecture de l'expérience, comme mentionné précédemment, nous avons utilisé l'API du FIT IoT-LAB pour programmer les nœuds fixes émetteurs et lancer les expériences. En ce qui concerne les TurtleBots, nous n'avons pas encore accès à eux depuis le FIT IoT-LAB, il a donc fallu créer un point d'accès Wi-Fi sur place pour pouvoir programmer

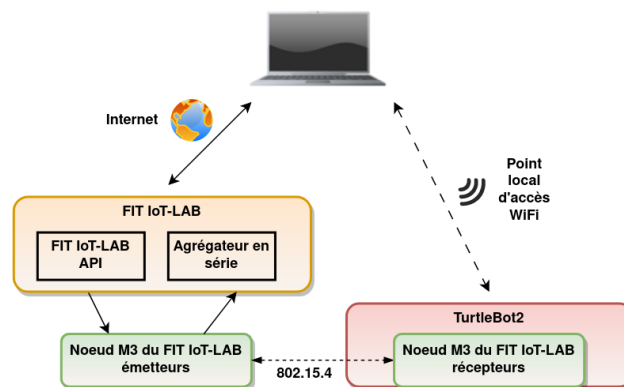


FIGURE 2.10 – Architecture de l'expérimentation avec traces mobiles.

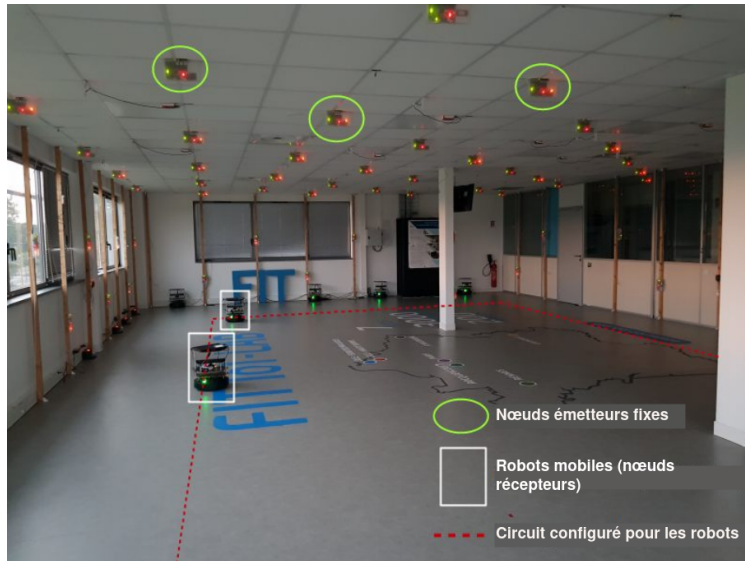


FIGURE 2.11 – Topologie de l'expérimentation pour la génération des données mobiles.



FIGURE 2.9 – Turtlebot 2 [81].

[81] TurtleBot 2

leurs circuits et les cartes M3 embarquées. Les nœuds émetteurs communiquent également avec les nœuds capteurs M3 mobiles via IEEE 802.15.4. L'architecture est présentée dans la Figure 2.10 et la topologie des nœuds dans la Figure 2.11.

2.4.3 Paramètres des expériences

Pour ces données, nous avons exploré plusieurs configurations de réseaux différentes pour créer divers niveaux de dynamique et d'incertitude :

Interférence entre les nœuds capteurs utilisant le même canal de fréquence : À chaque intervalle de temps où une paire de nœuds capteurs utilise le même canal de fréquence, cela crée un niveau d'interférence influencé par la puissance de transmission du nœud capteur, le canal de fréquence utilisé, la distance entre les nœuds capteurs et les caractéristiques du canal sans fil. Dans ces données, nous avons exploré deux techniques d'attribution de canaux différentes : (i) permettre à toutes les applications d'utiliser le même canal de fréquence (interférence inter et intra-application) ou (ii) attribuer à chaque application un canal de fréquence différent (interférence intra-application) pour réduire au moins l'interférence entre deux applications IoT différentes.

Transmission à différentes puissances pour l'envoi de données : Chaque nœud capteur peut être configuré avec une puissance de transmission différente afin de trouver un équilibre entre la portée de transmission et l'interférence. Dans ce cas, les niveaux de puissance de transmission sont de 0 dBm et de 12 dBm.

Mobilité des points d'accès : Les robots agissant en tant que points d'accès mobiles peuvent également introduire un autre niveau d'incertitude dans la communication. Pour évaluer l'impact de la mobilité, nous avons envisagé deux configurations de mobilité : l'une avec des robots statiques et l'autre avec des robots mobiles.

```
timestamp(s);tx_node;msg_id;tx_pow(dBm);channel;rx_robot;success;x(m);y(m);theta(rad);power(W);voltage(V);current(A)
1659009935.204623;m3-197;0;-12;11;robot1;1;29.968557;7.921202;0.565458;0.104061;3.286250;0.031658
1659009935.204852;m3-197;1;-12;11;robot1;0;29.968557;7.921202;0.565458;0.104061;3.286250;0.031658
1659009935.204976;m3-197;2;-12;11;robot1;0;29.968557;7.921202;0.565458;0.104061;3.286250;0.031658
1659009935.208417;m3-188;0;-12;11;robot1;0;29.968557;7.921202;0.565458;0.109679;3.302500;0.033221
1659009935.208640;m3-188;1;-12;11;robot1;0;29.968557;7.921202;0.565458;0.109679;3.302500;0.033221
```

(a) Résultats du côté des émetteurs.

```
timestamp(s);rx_node;robot;msg_id;tx_node;delay(s);rssi(dBm);channel;x(m);y(m);theta(rad);power(W);voltage(V);current(A)
1659009935.203678;m3-101;robot1;0;m3-197;0.000945;-65;11;29.968557;7.921202;0.565458;None;None;None
1659009935.252806;m3-101;robot1;2;m3-203;0.037356;-74;11;29.973518;7.921028;0.462521;None;None;None
1659009935.279235;m3-101;robot1;4;m3-193;0.037687;-72;11;29.973518;7.921028;0.462521;None;None;None
1659009935.289075;m3-101;robot1;4;m3-199;0.032392;-68;11;29.973518;7.921028;0.462521;None;None;None
1659009935.289441;m3-101;robot1;7;m3-194;0.015703;-67;11;29.973518;7.921028;0.462521;None;None;None
```

(b) Résultats du côté des récepteurs.

FIGURE 2.12 – Exemples de résultats pour les données avec points d'accès mobiles.

Hétérogénéité des données : Le mode de génération des données a un impact sur la dynamique du trafic d'une application, qu'il soit basé sur des événements, périodique ou hybride. Par conséquent, lors de la création de l'ensemble de données, nous avons utilisé les trois distributions de génération de données mentionnées précédemment.

En combinant ces différentes variables, nous avons obtenu huit configurations de réseau distinctes, chacune présentant des niveaux variables de puissance de transmission, de canaux de fréquence et de mobilité. Les configurations détaillées sont présentées dans la Table 2.7.

2.4.4 Résultats

La durée des ensembles de données IoT et Edge correspondants est de 6 heures, limitée par la durée de la batterie des robots mobiles. En revanche, pour les configurations réseau avec des points d'accès statiques, la durée des ensembles de données respectifs est de 10 heures. Les résultats obtenus consistent en des données d'échanges de messages entre les nœuds émetteurs et les nœuds récepteurs. Chaque ligne d'information brute contient un horodatage, l'identifiant du nœud récepteur ou émetteur, la puissance de transmission, le canal de transmission, le robot récepteur, la position du robot, et les informations instantanées sur la consommation d'énergie (puissance, tension et courant). Un exemple de données brutes est donnée dans la Figure 2.12.

2.4.5 Conclusion

Nous avons généré des ensembles de données en temps réel dans le banc d'essai FIT IoT-LAB pour huit configurations de réseau différentes, en tenant compte de la mobilité des points d'accès, de la répartition des canaux de fréquence et de la puissance de transmission en tant que paramètres de configuration. Nous mettons à disposition des ensembles de données issus de 20 de ces expériences, correspondant à différentes combinaisons de ces paramètres, pour un total de 68 heures et plus de 47 millions de messages échangés. Tout le code et les données sont disponibles en accès ouvert [83]. Dans le cadre de travaux futurs, nous envisageons d'intégrer ces données dans Sisyphe et de les enrichir en ajoutant, par exemple, davantage de points d'accès mobiles.

[83]: Applications traces

2.5 Conclusion

En conclusion de ce chapitre, nous avons présenté trois ensembles de données IoT qui peuvent être utilisés pour la prédiction et la modélisation des applications IoT. Ces ensembles de données ont été générés à l'aide du FIT IoT-LAB, une plateforme d'essai ouverte dédiée à l'IoT. Nous avons choisi cette plateforme ouverte afin de permettre aux chercheurs d'expérimenter, de reproduire et de générer leurs propres données dans une démarche de science ouverte.

Le premier ensemble de données a été généré de manière "naïve" à l'aide de code disponible sur un dépôt GitLab public. Il reflète la complexité hétérogène des réseaux IoT en incluant plusieurs applications IoT, chacune caractérisée par le nombre de nœuds, le type de messages échangés et la fréquence d'échanges. Nous avons caractérisé chaque application et généré les données associées avec plusieurs métriques pouvant être utilisées pour prédire et modéliser la Qualité de Service.

Le deuxième ensemble de données est une extension du premier, avec l'ajout de profils de génération de trafic différents, tels que la génération périodique ou basée sur des événements. Nous l'avons généré à l'aide d'un outil de génération automatique, reproductible et reproductible de données, que nous mettons à disposition de la communauté scientifique. Cela vise à promouvoir la science ouverte et à aider les chercheurs à gagner du temps tout en produisant des données de haute qualité.

Enfin, nous avons présenté un troisième ensemble de données qui inclut des points d'accès mobiles embarqués sur des robots. Toutes ces données ainsi que les outils de génération sont rendus disponibles publiquement. Ces travaux sont utilisés dans le prochain Chapitre 3 pour la prédiction de la Qualité de Service.

Prédiction de la Qualité de Service

3

Dans le chapitre précédent (Chapitre 2), nous avons discuté de la génération de données IoT et fourni des outils pour cette tâche. Dans ce chapitre-ci, nous utilisons ces données pour la **prédiction de la Qualité de Service (QoS) du réseau**.

3.1 Pourquoi prédire la Qualité de Service ?

L'IoT et en général les communications machine-à-machine (M2M) sont reconnus par les organisations internationales des technologies de communication [84, 85] et par l'industrie [86] comme faisant partie d'un des grands cas d'usage dans les technologies du futur : Communications ultra-fiables et à faible latence (URLLC). Les URLLC sont notamment définis par des exigences fortes et strictes en terme de latence et de fiabilité et donc en général de Qualité de Service (QoS) et Qualité d'Expérience (QoE). Parmi ces exigences en matière de QoS, on retrouve donc la latence mais aussi d'autres métriques comme la probabilité de perte d'un paquet ou encore le débit. En conséquence, il existe un grand intérêt pour classer les applications IoT en fonction de leurs besoins en matière de réseau (par exemple, débit, réseau d'accès) et de leurs exigences fonctionnelles (par exemple, nombre de dispositifs, fréquence de transmission, etc.) en appliquant des mécanismes d'analyse réseau. En particulier, la classification du trafic IoT peut se concentrer sur des applications individuelles, où les caractéristiques du trafic par appareil peuvent être extraites [87], ou elle peut être basée sur l'application elle-même [88]. Néanmoins, ces classifications ne capturent pas efficacement le comportement du réseau et les sources d'incertitude dans le trafic global envoyé au niveau de l'Edge. C'est pourquoi il peut-être nécessaire de prédire les futures fluctuations de la QoS afin d'anticiper le déploiement de ressources supplémentaires, car ce processus peut prendre du temps et n'être pas instantané au moment où nous en avons besoin.

3.2 Prédiction du débit dans des environnements dynamiques

3.2.1 Pourquoi et comment prédire le débit ?

Nous nous sommes tout d'abord concentrés sur la prédiction du débit car il s'agit d'une caractéristique centrale des réseaux IoT, avec des enjeux importants. Le débit mesure la quantité d'information transmise par unité de temps, généralement exprimée en bits par seconde. Dans notre cas d'utilisation, où des paquets de taille fixe sont transmis par une application, nous utilisons la Formule 2.1 par tranche de 5 millisecondes pour obtenir une mesure plus fine. Cette mesure peut facilement être convertie en bits par seconde en remplaçant les paquets par leur taille respective.

3.1 Pourquoi prédire la Qualité de Service ?	35
3.2 Prédiction du débit dans des environnements dynamiques	35
3.2.1 Pourquoi et comment prédire le débit ?	35
3.2.2 Travaux déjà existants sur la prédiction de débit	36
3.2.3 Données	37
3.2.4 Apprentissage automatique pour la prédiction du débit	37
3.2.5 Évaluation des performances des modèles	39
3.2.6 Et maintenant, les résultats !	41
3.2.7 Conclusion	43
3.3 Analyse de modèles d'apprentissage automatique pour la prédiction multi-métriques	43
3.3.1 Une caractéristique c'est bien, plusieurs c'est mieux !	43
3.3.2 Travaux existants sur la prédiction de la QoS avec des réseaux de neurones profonds	43
3.3.3 Un court détour par les fenêtres glissantes	44
3.3.4 Transformateurs temporels	44
3.3.5 Configuration de l'expérimentation	45
3.3.6 Résultats : mode univarié	46
3.3.7 Résultats : mode multivarié	49
3.3.8 Mode univarié ou multivarié ?	50
3.3.9 Conclusion	50
3.4 Conclusion	50

[89]: Adnan AKBAR, Abdullah KHAN, Francois CARREZ et Klaus MOESSNER. « Predictive Analytics for Complex IoT Data Streams ». In : *IEEE Internet of Things J.* 4.5 (2017)

[90]: Ali R. ABDELLAH, Volkov ARTEM, Ammar MUTHANNA, Denis GALLYAMOV et Andrey KOUCHERYAVY. « Deep Learning for IoT Traffic Prediction Based on Edge Computing ». In : *Distributed Computer and Communication Networks : Control, Computation, Communications*. Sous la dir. de V.M. VISHNEVSKIY, K.E. SAMOYLOV et D.V. KOZYREV. T. 1337. Communications in Computer and Information Science. Springer, 2020

[91]: Volkov ARTEM, Abdelhamied A. ATEYA, Ammar MUTHANNA et Andrey KOUCHERYAVY. « Novel AI-Based Scheme for Traffic Detection and Recognition in 5G Based Networks ». In : *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*. 2019. ISBN : 978-3-030-30859-9

[92]: Manuel LOPEZ-MARTIN, Belen CARRO et Antonio SANCHEZ-ESGUEVILLAS. « Neural network architecture based on gradient boosting for IoT traffic prediction ». In : *Future Generation Computer Systems* 100 (2019)

[93]: Ali R. ABDELLAH, Omar ABDULKAREEM MAHMOOD et Andrey KOUCHERYAVY. « Delay prediction in IoT using Machine Learning Approach ». In : *2020 12th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*. 2020

[94]: Muhammad ATEEQ, Farruh ISHMANOV, Muhammad Khalil AFZAL et Muhammad NAEEM. « Predicting Delay in IoT Using Deep Learning : A Multiparametric Approach ». In : *IEEE Access* 7 (2019)

Les appareils IoT ont besoin de communiquer de manière efficace et en temps réel pour garantir une qualité de service adéquate, parfois avec des contraintes de latence critiques. Cependant, dans le contexte de l'IoT, enrichi par le traitement en périphérie (Edge Computing), les communications machine à machine peuvent devenir très nombreuses. Cela, combiné à la nature en couches de ces réseaux, peut charger le réseau et réduire le débit potentiel. De plus, il est prévu que les réseaux futurs seront de plus en plus chargés en appareils, comme nous avons vu dans le Chapitre 1.2. La prédiction du débit est donc cruciale pour pouvoir déployer efficacement des ressources de communication ou de calcul supplémentaires au bon endroit et au bon moment pour soutenir le réseau, ainsi que pour réduire le débit et garantir un meilleur service aux utilisateurs. Par ailleurs, dans le domaine de l'IoT et de l'informatique en périphérie (Edge), nous avons non seulement de nombreux appareils, mais aussi une grande hétérogénéité en termes d'applications produisant des types de trafic variés. Ajoutez à cela la nature très fluctuante des réseaux sans fil, et vous obtenez un problème de prédiction complexe. C'est là que l'apprentissage automatique entre en jeu. En fournissant suffisamment de données historiques (ou données d'entraînement) d'une caractéristique donnée, un modèle d'apprentissage automatique est capable de prédire l'évolution de cette caractéristique en fonction de nouvelles observations. En outre, il existe d'autres méthodes de prédiction, comme les réseaux de neurones, mais ils nécessitent généralement un grand volume de données et sont plus adaptés aux cas non linéaires, ce qui n'est pas toujours le cas dans notre contexte d'utilisation. Nous avons donc privilégié des modèles plus légers et moins gourmands en calculs pour nos prédictions.

3.2.2 Travaux déjà existants sur la prédiction de débit

Akbar et al. [89] se sont concentrés sur le traitement complexe d'événements à l'aide de l'apprentissage automatique. Les auteurs ont proposé un modèle prédictif adaptatif utilisant une technique de régression appelée régression de fenêtre mobile afin de fournir des solutions distribuées et évolutives. Les auteurs de [90] ont effectué la prédiction du trafic du réseau IoT afin de fournir une communication fiable. Pour y parvenir, l'apprentissage profond a été appliqué avec succès en utilisant un LSTM (Long Short Term Memory). Les caractéristiques de l'ensemble de données comprennent l'horodatage, le nombre d'octets et le nombre de paquets. Ce travail a été étendu dans [91] pour les réseaux 5G en utilisant les réseaux de neurones récurrents (RNN) et plus précisément le LSTM. Lopez et al. [92] ont présenté un travail détaillé sur la prévision du volume de trafic IoT basé sur l'algorithme de descente de gradient stochastique et les architectures de réseau neuronal appelées gANET. L'ensemble de données comprennent deux caractéristiques. Les auteurs de [93] ont proposé une méthode de prédiction d'un pas en avant et une méthode de prédiction multi-étapes pour la prédiction des retards dans l'IoT basée sur le réseau neuronal récurrent NARX. Ils ont simulé un environnement IoT et utilisé un ensemble de données simulé. Ateeq et al. [94] ont mis en œuvre une prédiction des retards dans le réseau IEEE 802.15.4 en utilisant une approche multi-paramétrique d'apprentissage profond. Les caractéristiques utilisées par ce travail sont extraites de la couche application, de la couche MAC et de la couche physique du

réseau. Dans [95] l'accent est mis sur la prédiction de la QoS dans les environnements IoT, notamment le temps de réponse du service et le débit. Cependant, l'approche utilisée est basée sur une technique de factorisation de matrices et se limite aux prédictions de valeurs manquantes dans une matrice de données contenant des valeurs pour le temps de réponse et le débit. Le travail dans [96] a également évalué plusieurs approches de factorisation de matrices et a montré comment ces approches peuvent être utilisées pour faire des prédictions de QoS dans un environnement IoT.

On peut noter dans les travaux précédents que la plupart des études utilise un nombre limité de caractéristiques pour la prédiction du trafic IoT et ne fournit pas de caractéristiques détaillées et fines [89-92]. Aussi, malgré leur utilisation de plus en plus répandue, les réseaux de neurones nécessitent un temps d'entraînement intensif [90-92, 94]. Finalement, certaines des études existantes prédisent soit le volume de trafic, soit la QoS pour une seule application et peuvent ne pas fournir des estimations précises en dehors de ces applications [95, 96]. C'est pour ces raisons que nous proposons dans ces travaux une prédiction du débit pour des applications IoT hétérogènes.

3.2.3 Données

Pour les données et leur génération, nous invitons le lecteur à se référer au Chapitre 2.3. Nous utilisons les données produites dans le Chapitre 2.3 et en extrayons le débit par nœud pour chaque application, exprimé en millisecondes. Nous rappelons que la formule du débit est la Formule 2.1.

3.2.4 Apprentissage automatique pour la prédiction du débit

Apprentissage automatique et régression

Nous faisons ici un point technique pour expliquer quelques termes. Commençons par définir ce qu'est la régression, car nous allons utiliser des techniques de régression pour nos prédictions. Dans le vaste domaine de l'apprentissage automatique, on trouve plusieurs techniques, toutes visant à apprendre à un système ou à un modèle à prédire des valeurs futures ou à prendre des décisions. La régression est l'une de ces approches de l'apprentissage automatique. Elle trouve son origine dans l'analyse statistique et vise à modéliser et à capturer les relations entre une variable dépendante (ce que l'on souhaite prédire) et des variables indépendantes (les données d'entrée). Ces dépendances sont modélisées mathématiquement sous la forme d'une équation, qu'elle soit linéaire ou non linéaire. Si nous connaissons l'influence de certaines variables sur une variable cible, nous pouvons alors prédire cette dernière dans le futur. L'un des problèmes bien connus en apprentissage automatique est le surapprentissage (ou *overfitting* en anglais) des données. Pour concevoir un modèle d'apprentissage, nous utilisons à la fois des *données d'entraînement* et des *données de test*. Les données d'entraînement sont utilisées pour que le modèle apprenne, tandis que les données de test

[95]: G. WHITE, A. PALADE, C. CABRERA et S. CLARKE. « IoTPredict : Collaborative QoS Prediction in IoT ». In : *IEEE Int. Conf. on Pervasive Computing and Communications (PerCom)*. 2018

[96]: G. WHITE, A. PALADE, C. CABRERA et S. CLARKE. « Quantitative Evaluation of QoS Prediction in IoT ». In : *IEEE/IFIP Int. Conf. on Dependable Systems and Networks*. 2017

sont réservées à la validation et à l'évaluation du modèle, car elles ne sont pas utilisées pendant l'entraînement. Malheureusement, il peut arriver que le modèle fonctionne *trop bien* sur les données d'entraînement. Il se suradapte alors à ces données, mais il peut avoir du mal à être utilisé sur de nouvelles données ou sur les données de test. Cela se produit lorsque le modèle capture non seulement les dépendances dans les données, mais aussi le bruit ou les fluctuations aléatoires présents dans les données d'entraînement.

Modèles sélectionnés

Maintenant, examinons les différents modèles d'apprentissage automatique que nous avons sélectionnés :

[97]: Mridula BATRA et Rashmi AGRAWAL. « Comparative Analysis of Decision Tree Algorithms ». In : *Nature Inspired Computing*. Sous la dir. de Bijaya Ketan PANIGRAHI, M. N. HODA, Vinod SHARMA et Shivendra GOEL. Singapore : Springer Singapore, 2018. ISBN : 978-981-10-6747-1

Régression par arbre de décision (DTR) : La DTR [97] est un algorithme qui prédit des valeurs continues à l'aide d'arbres de décision. Les caractéristiques d'entrée sont utilisées pour diviser de manière récursive les données d'entraînement en sous-espaces de caractéristiques de plus en plus restreints. Au fur et à mesure que l'on descend de la racine vers les feuilles de l'arbre, la similarité des valeurs cibles augmente. Le parcours de l'arbre est basé sur les divisions en fonction des attributs, et les feuilles de l'arbre fournissent les valeurs prédites. La DTR est facilement interprétable et peu gourmande en ressources, ce qui la rend efficace pour un grand volume de données. Cependant, si le problème de prédiction est trop complexe, l'arbre peut rapidement devenir complexe lui aussi, ce qui entraîne un risque de surapprentissage et le rend incapable de généraliser sur de nouvelles données. La DTR a également du mal à capturer des relations très complexes entre les caractéristiques et est sensible aux fluctuations mineures des données.

[98]: Jerome H. FRIEDMAN. « Greedy Function Approximation : A Gradient Boosting Machine ». In : *Annals of Statistics* 29.5 (2001)

Arbre de régression boostée par gradient (GBRT) : Le GBRT [98] est un ensemble d'arbres de décision utilisés comme estimateurs avec un algorithme de descente de gradient itératif. L'objectif de cet algorithme est de minimiser une fonction de perte dans l'espace des hypothèses. Cette fonction de perte est réduite de manière itérative en ajoutant à chaque étape un nouvel arbre de décision qui réduit la perte tout en maintenant inchangé l'arbre précédent. Le GBRT final est une combinaison linéaire des arbres de décision. Le GBRT est plus résistant au surapprentissage que les arbres de décision classiques et peut modéliser des relations non linéaires complexes entre les variables. Dans la plupart des cas, un grand nombre d'estimateurs conduit à de meilleures performances, ce qui en fait l'un des hyperparamètres du GBRT. De plus, le GBRT peut atténuer le problème des données manquantes et ne nécessite pas la suppression des valeurs aberrantes.

[99]: Alex J. SMOLA et Bernhard SCHOLKOPF. « A Tutorial on Support Vector Regression ». In : *Journal of Machine Learning Research (JMLR)* 3 (2004)

Régression par vecteurs de support (SVR) : Le SVR [99] est un modèle adaptable qui utilise les observations de données pour créer des hyperplans dans un espace de caractéristiques à n dimensions. Il génère des prédictions en utilisant une fonction basée sur une frontière de décision et une marge d'erreur. Le processus d'apprentissage ajuste cette marge pour atteindre une précision acceptable en minimisant les coefficients tout en respectant des contraintes. Les contraintes sont minimisées pour obtenir les coefficients optimaux, et la méthode des multiplicateurs de Lagrange peut résoudre les équations. Cette méthode est efficace pour les

problèmes complexes et est capable de bien généraliser sur de nouvelles données. Cependant, elle demande beaucoup de ressources de calcul et a du mal à s'adapter aux données incomplètes.

Régression des k plus proches voisins (KNNR) : Le KNNR [100] place les données d'entraînement dans un espace en n dimensions et sélectionne les k voisins les plus proches à l'aide d'une mesure de distance prédéfinie. Ainsi, la valeur d'une donnée en entrée peut être prédite en moyennant les résultats de ses voisins, éventuellement en leur attribuant des poids en fonction de leur distance. Le KNNR présente l'avantage de ne pas nécessiter de phase d'entraînement contrairement à d'autres méthodes, car les calculs sont effectués directement lors de la prédiction en recherchant les voisins dans les données historiques. Cependant, cela signifie que les données historiques ne peuvent pas être trop volumineuses ni incomplètes, et que le problème de prédiction ne peut pas être trop complexe.

Régression Linéaire Multiple (MLR) : La MLR [101] est utilisée pour comprendre les relations entre différentes variables. Une équation linéaire est créée en fonction des données d'entraînement, ce qui permet de prédire les résultats. Les coefficients dans l'équation sont ajustés pour minimiser la somme des carrés des résidus, c'est-à-dire les erreurs de prédiction. L'avantage principal de la MLR réside dans son efficacité de calcul à la fois lors de la phase d'entraînement et de prédiction. Cependant, les données doivent présenter des relations linéaires, ne peuvent pas avoir de relations trop complexes entre elles, et le modèle peut souffrir de surapprentissage en présence de données incomplètes ou incohérentes.

3.2.5 Évaluation des performances des modèles

Pour évaluer l'efficacité d'un modèle, il est nécessaire de mesurer dans quelle mesure il présente un écart par rapport à la réalité. En effet, il existe inévitablement des différences entre les prédictions du modèle et les données réelles, mais il est essentiel de minimiser cet écart autant que possible. De plus, il est important de pouvoir comparer différents modèles afin de déterminer lequel est le plus précis. C'est là que les indicateurs d'écart par rapport à la réalité, que nous appelons métriques d'évaluation des modèles, entrent en jeu. Les principales métriques d'évaluation pour les méthodes de régression sont l'erreur absolue moyenne (MAE), l'erreur quadratique moyenne (MSE), et enfin la racine carrée de l'erreur quadratique moyenne (RMSE).

MAE La MAE mesure la moyenne des erreurs absolues entre les valeurs réelles et les valeurs prédites, ce qui permet d'évaluer la précision globale du modèle de régression. Plus la MAE est proche de zéro, meilleure est la capacité du modèle à faire des prédictions précises. La Figure 3.1 permet d'appréhender intuitivement cette notion. La formule pour la MAE est la suivante :

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (3.1)$$

où :

[100]: Christopher M. BISHOP. *Pattern Recognition and Machine Learning*. Springer, 2006

[101]: Jacob COHEN, Patricia COHEN, Stephen G. WEST et Leona S. AIKEN. *Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences*. Routledge, 2013

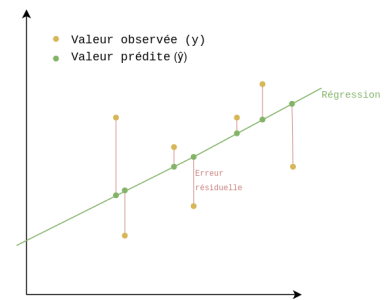


FIGURE 3.1 – Représentation graphique de la MAE.

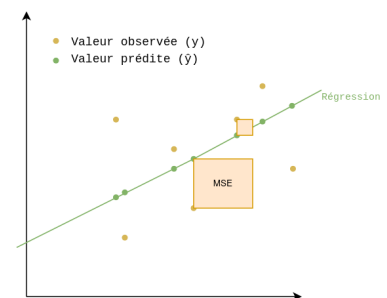


FIGURE 3.2 – Représentation graphique de la MSE.

- n : Nombre d'échantillons
- y_i : Valeur réelle de la cible pour l'échantillon i
- \hat{y}_i : Valeur prédite de la cible pour l'échantillon i
- $\sum_{i=1}^n$: Sommation sur tous les échantillons de 1 à n
- $|\cdot|$: Valeur absolue, qui garantit que toutes les erreurs sont positives
- $\frac{1}{n} \sum_{i=1}^n$: Moyenne des erreurs absolues sur l'ensemble des échantillons

MSE La MSE est similaire à la MAE, sauf qu'ici, on calcule la moyenne du carré des différences entre la prédiction et la valeur réelle. Mettre au carré les erreurs accentue les erreurs aberrantes, c'est-à-dire les valeurs très éloignées de la réalité. Une MSE plus faible indique généralement une meilleure adéquation du modèle aux données. De même, la Figure 3.2 permet d'appréhender intuitivement cette notion. La formule pour la MSE est la suivante :

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (3.2)$$

où :

- n : Nombre d'échantillons
- y_i : Valeur réelle de la cible pour l'échantillon i
- \hat{y}_i : Valeur prédite de la cible pour l'échantillon i
- $\sum_{i=1}^n$: Sommation sur tous les échantillons de 1 à n
- $(\cdot)^2$: Élévation au carré, ce qui garantit que toutes les erreurs sont positives
- $\frac{1}{n} \sum_{i=1}^n$: Moyenne des erreurs quadratiques sur l'ensemble des échantillons

RMSE La RMSE est la racine carrée de la MSE. Prendre la racine carrée de la MSE fournit une valeur dans l'unité d'origine de la cible, ce qui la rend interprétable. La formule pour la RMSE est la suivante :

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (3.3)$$

où :

- n : Nombre d'échantillons
- y_i : Valeur réelle de la cible pour l'échantillon i
- \hat{y}_i : Valeur prédite de la cible pour l'échantillon i
- $\sum_{i=1}^n$: Sommation sur tous les échantillons de 1 à n
- $(\cdot)^2$: Élévation au carré, ce qui garantit que toutes les erreurs sont positives
- $\frac{1}{n} \sum_{i=1}^n$: Moyenne des erreurs quadratiques sur l'ensemble des échantillons (MSE)

- $\sqrt{\cdot}$: Racine carrée, ce qui donne une mesure dans la même unité que la cible d'origine

En outre, il est important de prendre en compte les contraintes de puissance de calcul pour les dispositifs IoT et Edge. Cela signifie qu'il est également essentiel d'évaluer le temps nécessaire à l'entraînement des modèles, qui doit être minimisé. De plus, le temps de prédiction, également appelé temps d'inférence, doit être réduit au minimum pour obtenir des prédictions en temps réel et libérer rapidement les ressources de calcul.

3.2.6 Et maintenant, les résultats!

Nous disposons de nos données et avons extrait les caractéristiques de qualité de service pertinentes. Ensuite, nous avons sélectionné les modèles d'apprentissage automatique et leurs métriques d'évaluation. Les modèles ont été entraînés avec 70% des données applicatives et testés avec les 30% restants. Nous allons donc examiner les résultats et les analyser à l'aide des métriques présentées précédemment.

Comparaison des applications : La Table 3.1 regroupe tous les résultats par application, avec pour chaque application le résultat de chaque métrique d'évaluation par modèle de prédiction. Dans un premier temps, nous pouvons nous intéresser à une comparaison entre les applications. En prenant en compte les trois métriques d'évaluation présentées dans la Table 3.1, on peut observer que les meilleurs résultats sont obtenus sur les applications de système d'urgence, système CVC et d'éclairage intelligent. Les modèles entraînés sur les applications de réalité augmentée, de vidéosurveillance, puis de VoIP obtiennent dans l'ordre les pires résultats. Ces applications ont un nombre de nœuds et une durée d'expérimentation moins importants que dans les autres applications. Pourtant, les modèles de prédiction d'apprentissage automatique ont besoin de beaucoup de données pour fournir des prédictions correctes, ce qui explique donc les faibles résultats sur ces trois applications. De plus, nous avons pu observer que les prédictions présentent un comportement asymétrique, c'est-à-dire que les données ne sont pas bien distribuées autour de la moyenne. Malheureusement, les modèles de régression supposent souvent une distribution asymétrique et ne performant donc pas bien sur les données asymétriques comme ici. En conséquence, nous allons donc nous concentrer sur la comparaison des applications de système d'urgence, système CVC et d'éclairage intelligent.

Analyse de la MAE et la MSE : La Figure 3.3 représente graphiquement la comparaison de la MAE pour les applications et chaque modèle. L'échelle de la MAE est inversée de telle sorte que plus la prédiction est précise, plus le diagramme est élevé sur l'axe des ordonnées. On peut voir que la MLR obtient la meilleure MAE. En effet, on sait que la MLR est très performante pour gérer les dépendances linéaires. De plus, la MLR a une approche paramétrique, ce qui signifie qu'elle prend en compte des relations entre les variables, qu'elles soient dépendantes ou non. Cette approche va donc être plus efficace avec moins de données que les approches non paramétriques si les données suivent la distribution supposée par le modèle, ce qui est le cas ici. On peut aussi voir que la SVR obtient les pires résultats. En effet, la SVR a du mal à s'adapter à un grand

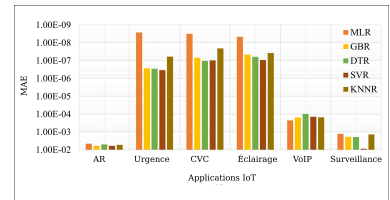


FIGURE 3.3 – Comparaison des résultats MAE pour tous les algorithmes.

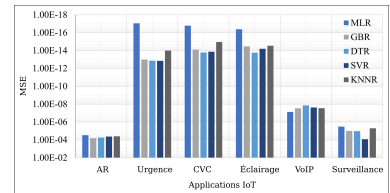


FIGURE 3.4 – Comparaison des résultats MSE pour tous les algorithmes.

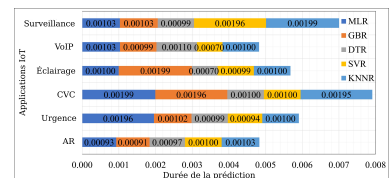


FIGURE 3.5 – Comparaison du temps de prédiction pour tous les algorithmes.

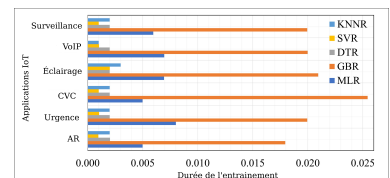


FIGURE 3.6 – Comparaison du temps d'entraînement pour tous les algorithmes.

TABLE 3.1 – Mesures de performance de tous les algorithmes pour les applications IoT.

Application	Algorithme	MAE	MSE	RMSE
Réalité augmentée	MLR	0.0045641	2.9297e-05	0.0054127
	GBR	0.005969	6.5315e-05	0.0080818
	DTR	0.0049392	5.4937e-05	0.0074119
	SVR	0.0058981	4.2406e-05	0.0065121
	KNNR	0.0052350	3.8610e-05	0.0062137
Application d'urgence	MLR	2.6155e-09	8.8448e-18	2.9740e-09
	GBR	2.6301e-07	1.0277e-13	3.2058e-07
	DTR	2.8198e-07	1.3504e-13	3.6747e-07
	SVR	3.3816e-07	1.3879e-13	3.7254e-07
	KNNR	5.9421e-08	1.0169e-14	1.0084e-07
Système CVS	MLR	3.0885e-09	1.6375e-17	4.0467e-09
	GBR	6.8290e-08	7.8697e-15	8.8711e-08
	DTR	1.0306e-07	1.6541e-14	1.2861e-07
	SVR	9.5958e-08	1.3207e-14	1.1492e-07
	KNNR	2.0620e-08	1.1094e-15	3.3308e-08
Éclairage intelligent	MLR	4.5123e-09	4.1255e-17	6.4230e-09
	GBR	4.5025e-08	3.4404e-15	5.8655e-08
	DTR	9.1376e-08	1.7043e-14	1.3055e-07
	SVR	5.9834e-08	6.2366e-15	7.8972e-08
	KNNR	3.7146e-08	2.8440e-15	5.3329e-08
VoIP	MLR	0.0002211	7.2603e-08	0.0002694
	GBR	0.0001546	2.8749e-08	0.0001695
	DTR	9.6463e-05	1.3520e-08	0.0001163
	SVR	0.0001392	2.3750e-08	0.0001541
	KNNR	0.0001483	2.7738e-08	0.0001665
Surveillance vidéo	MLR	0.0012548	3.2583e-06	0.0018050
	GBR	0.0018372	9.6768e-06	0.0031107
	DTR	0.0019151	0.0000103	0.0032104
	SVR	0.0086434	0.0000827	0.0090953
	KNNR	0.0013595	5.0101e-06	0.0022383

ensemble de données comme nous en avons. La Figure 3.4 représente graphiquement la comparaison de la MSE pour les applications et chaque modèle. Comme pour la MSE, l'échelle de la MSE est inversée. Ici encore, la MLR obtient de meilleurs résultats en ce qui concerne la MSE que les autres modèles, suivie de la KNNR mais avec moins d'écart.

Analyse du temps d'entraînement et d'inférence : Les Figures 3.6 et 3.5 présentent le temps d'entraînement et d'inférence respectivement, en secondes. La SVR prend le moins de temps d'entraînement vis-à-vis des autres modèles, cependant, on a vu qu'elle obtient les pires résultats de performance. Le GBR a le temps d'entraînement le plus important dû aux nombreux estimateurs utilisés lors de l'entraînement. Cependant, il obtient des scores de performance corrects, juste après la MLR et la KNNR. Pour ce qui est du temps d'inférence, le meilleur modèle est le SVR pour le système d'urgence et la VoIP. Ensuite, nous avons le GBRT pour la réalité augmentée, et la SVR et la DTR pour les systèmes CVC.

3.2.7 Conclusion

Avec les différentes analyses menées, on peut tirer plusieurs conclusions. Tout d'abord, que le débit est mieux prédit pour les applications de système d'urgence, de système CVC et d'éclairage intelligent en utilisant la MLR en premier lieu, puis la KNNR et le GBRT ensuite. La SVR possède le temps d'entraînement le plus réduit mais produit des résultats peu performants. Enfin, le temps de prédiction est très variable selon l'application et le modèle, dû à la gestion différenciée des modèles de la complexité variables des différents ensembles de données IoT, il est donc difficile de généraliser à ce sujet.

3.3 Analyse de modèles d'apprentissage automatique pour la prédiction multi-métriques

3.3.1 Une caractéristique c'est bien, plusieurs c'est mieux!

Dans la Section précédente, nous avons présenté des méthodes de prédiction du débit pour l'IoT et le Edge. Cependant, bien que le débit soit une caractéristique importante des réseaux sans fil pour la qualité de service, ce n'est pas la seule. Dans ces travaux, nous avons donc cherché à prédire plusieurs métriques de la qualité de service. Pour cela, nous avons réutilisé les données du Chapitre 2.4, qui sont plus riches et nous permettent d'extraire davantage de caractéristiques. Nous avons sélectionné quatre des caractéristiques les plus importantes en IoT pour la qualité de service : le débit, le taux de paquets délivrés (PDR), le taux de paquets perdus (PLR) et la latence de transmission. Nous fournissons une définition et une modélisation de ces caractéristiques dans le Chapitre 2.4, ainsi qu'une analyse des données à travers ces métriques.

3.3.2 Travaux existants sur la prédiction de la QoS avec des réseaux de neurones profonds

Abdellah et al. [93] proposent un réseau de neurones récurrents auto-régressifs non linéaires avec des variables exogènes (NARX) effectuant à la fois des prédictions à un seul pas et à plusieurs pas pour la prédiction du délai. Cependant, les auteurs utilisent un jeu de données simulé qui peut ne pas refléter les conditions réelles des communications sans fil. Le délai est également prédit dans l'article [94] à l'aide d'un réseau de neurones profond simple. Les caractéristiques utilisées dans le cadre de ce travail sont extraites de la couche application, de la couche MAC et de la couche physique du réseau. Les résultats sont prometteurs. Said et Tolba [102] proposent un modèle d'apprentissage profond à trois couches qui prédit le débit, le retard et la perte de paquets d'un système de communication IoT. Ce travail prend en compte divers types d'accès réseaux comme les réseaux de capteurs sans-fil (WSN) ou les réseaux ad-hoc. Dans [103], les auteurs prédisent le débit à l'aide d'un réseau de neurones convolutifs (CNN) avec de la vectorisation cible. Ici aussi, les données utilisées sont simulées et concentrées sur les

[93]: Ali R. ABDELLAH, Omar ABDULKAREEM MAHMOOD et Andrey KOUCHERYAVY. « Delay prediction in IoT using Machine Learning Approach ». In : *2020 12th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*. 2020

[94]: Muhammad ATEEQ, Farruh ISHMANOV, Muhammad Khalil AFZAL et Muhammad NAEEM. « Predicting Delay in IoT Using Deep Learning : A Multiparametric Approach ». In : *IEEE Access* 7 (2019)

[102]: Omar SAID et Amr TOLBA. « Accurate performance prediction of IoT communication systems for smart cities : An efficient deep learning based solution ». In : *Sustainable Cities and Society* 69 (2021)

[103]: Yafei HOU, Kazuto YANO, Norisato SUGA, Julian WEBBER, Eiji NII, Toshihide HIGASHIMORI, Satoshi DENNO et Yoshinori SUZUKI. « A Study of Throughput Prediction using Convolutional Neural Network over Factory Environment ». In : *2021 23rd International Conference on Advanced Communication Technology (ICACT)*. 2021

[104]: Xiaochen FAN, Chaocan XIANG, Liangyi GONG, Xiangjian HE, Chao CHEN et Xiang HUANG. « UrbanEdge : Deep Learning Empowered Edge Computing for Urban IoT Time Series Prediction ». In : *Proceedings of the ACM Turing Celebration Conference - China*. ACM TURC '19. Chengdu, China : Association for Computing Machinery, 2019. ISBN : 9781450371582

[105]: Di WU, He XU, Zhongkai JIANG, Weiren YU, Xuetao WEI et Jiwu LU. « EdgeLSTM : Towards Deep and Sequential Edge Computing for IoT Applications ». In : *IEEE/ACM Transactions on Networking* 29.4 (2021)

[90]: Ali R. ABDELLAH, Volkov ARTEM, Ammar MUTHANNA, Denis GALLYAMOV et Andrey KOUCHERYAVY. « Deep Learning for IoT Traffic Prediction Based on Edge Computing ». In : *Distributed Computer and Communication Networks : Control, Computation, Communications*. Sous la dir. de V.M. VISHNEVSKIY, K.E. SAMOUYLOV et D.V. KOZYREV. T. 1337. Communications in Computer and Information Science. Springer, 2020

[92]: Manuel LOPEZ-MARTIN, Belen CARRO et Antonio SANCHEZ-ESGUEVILLAS. « Neural network architecture based on gradient boosting for IoT traffic prediction ». In : *Future Generation Computer Systems* 100 (2019)

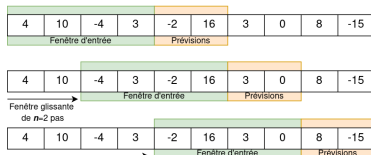


FIGURE 3.7 – Illustration des fenêtres glissantes.

cas d'usage des usines. Fan et al. [104] proposent un modèle de réseau neuronal récurrent (RNN) basé sur l'apprentissage profond qui utilise un mécanisme d'attention pour le traitement des données de l'IdO à la périphérie. Leur architecture UrbanEdge collecte, transmet et traite les données mais nécessite beaucoup de bande passante pour la transmission des données de surveillance. EdgeLSTM est un système présenté dans [105], qui utilise un LSTM (Long Short-Term Memory) en grille associé à une machine à vecteurs de support (SVM). Ce système a pour but la prédiction de données, la maintenance du réseau, la détection d'anomalies et la gestion de la mobilité. Un réseau LSTM est présenté dans [90] pour la prédiction du débit IoT en 5G. Pour finir, Lopez-Martin, Carro et Sanchez-Esguevillas [92] utilisent un algorithme de descente de gradient stochastique et une architecture de réseau neuronal qu'ils appellent gaNET. Ces travaux ne se concentrent pas sur la prédiction multimétrique de la Qualité de Service en fonction du temps, comme proposé dans ce Chapitre.

3.3.3 Un court détour par les fenêtres glissantes

Ici, nous allons introduire des concepts importants pour mieux comprendre la suite de ces travaux. Tout d'abord, pour résumer le concept de série temporelle, il s'agit d'une suite de données indexée par le temps. En d'autres termes, c'est l'évolution d'une ou plusieurs valeurs au cours du temps. Les séries temporelles occupent une place particulière dans le monde de la prédiction, notamment parce qu'elles peuvent présenter des caractéristiques uniques, comme la saisonnalité des données (qui contient des cycles qui se répètent). Il existe, par exemple, des modèles d'apprentissage spécifiquement conçus pour les séries temporelles. Une stratégie très courante utilisée pour la prédiction de séries temporelles est la fenêtre glissante. Si l'on imagine une série temporelle comme une file de données, la fenêtre glissante de taille fixe va glisser d'un certain nombre de pas sur cette file pour sélectionner au fur et à mesure une séquence de données fixe depuis le début jusqu'à la fin. À chaque "pas" de la fenêtre, nous obtenons une séquence de la série qui va chevaucher l'ancienne séquence, et ce de manière continue. La Figure 3.7 illustre cette notion. Cette stratégie permet de bien saisir l'évolution des données dans le temps. Elle est utilisée pour entraîner les modèles et aussi pour assurer la prédiction lorsque de nouvelles données temporelles sont accessibles. Les séries temporelles sont une représentation des données IoT particulièrement intéressante, précisément parce que les applications connaissent des cycles et évoluent avec le temps. Par exemple, une application peut-être plus utilisée à certains horaires que d'autres.

3.3.4 Transformateurs temporels

Pour prédire les caractéristiques présentées plus haut, nous avons utilisé un transformateur temporel. Les transformateurs temporels [106] sont des méthodes très populaires pour l'analyse des séries temporelles, comme les données applicatives de l'IoT que nous avons générées. Les transformateurs temporels reposent sur un mécanisme d'auto-attention qui consiste à indiquer au modèle les parties de la séquence, appelées

TABLE 3.2 – Hyperparamètres utilisés dans toutes les méthodes pour la prédiction univariée du débit dans tous les ensembles de données.

Modèles	Hyperparamètres	Valeur min.	Valeur max.	Meilleure valeur sélectionnée				
				CVC	VoIP	Éclairage	Urgence	Surveillance
MLP	Nombre de neurones	8	512	64	392	288	40	504
	Taux d'abandon	0	0.5	0.1	0.3	0.2	0.1	0.1
	Taux d'apprentissage	1e-2	1e-4	0.01	0.0001	0.001	0.001	0.0001
LSTM empilés	Nombre de neurones	8	128	24	72	104	96	16
	Taux d'abandon	0	0.5	0.4	0.001	0.1	0.4	0.4
	Learning rate	1e-2	1e-4	0.001	0.0001	0.01	0.0001	0.0001
	Nombre de couches	2	6	3	2	4	3	5
LSTM bidirectionnels	Nombre de neurones	8	512	32	16	24	64	352
	Taux d'abandon	0	0.5	0.5	0.4	0.1	0.1	0.2
	Taux d'apprentissage	1e-2	1e-4	0.01	0.01	0.01	0.001	0.001
Transformateur temporel	Taille de la tête	4	256	28	32	128	4	2
	Nombre de têtes	4	32	6	18	24	4	3
	Taux d'abandon	0	0.5	0.2	0.5	0.2	0.5	0.2
	Nombre de blocs transformateurs	4	16	16	8	4	4	2
	Neurones de la couche linéaire	4	128	96	84	52	64	116
	Décrochage linéaire de la couche	0	0.5	0.2	0.5	0.2	0.5	0.2
	Dimension du filtre	4	64	96	28	52	64	8
	Nombre de couches d'attention	1	15	4	5	2	2	3

jetons, qui doivent avoir le plus d'attention car elles sont les plus pertinentes pour la tâche donnée. Les transformateurs temporels possèdent plusieurs *têtes d'attention* qui vont procéder à des calculs d'attention en parallèle différents, comme illustré par la Figure 3.9. Comme les jetons sont traités en parallèle, il faut donc que le modèle puisse reconnaître leur position dans la séquence d'entrée, cette partie étant gérée par un module d'encodage de position avant le mécanisme d'auto-attention. Après être sortis des têtes d'attention, les données sont agrégées et envoyées à un module de réseau à propagation directe qui va pouvoir fournir la prévision. Cette architecture est illustrée dans la Figure 3.8. Pour prédire une unique caractéristique de la qualité de service, nous allons soit donner en entrée uniquement la série temporelle correspondante, ce qui s'appelle une prédiction univariée, soit toutes les autres séries temporelles des autres caractéristiques, ici prédiction multivariée. Par exemple, pour la prédiction du débit, on peut donner soit en entrée du modèle la série temporelle du débit, soit toutes les séries temporelles de toutes les caractéristiques y compris le débit. Cependant, en sortie on aura uniquement la prédiction du débit. Ensuite, pour la prédiction multivariée, chaque tête du mécanisme d'auto-attention va se charger de calculer l'importance entre des caractéristiques deux à deux, ce qui permet de prédire une caractéristique à l'aide des autres. La série temporelle donnée en entrée sera donc introduite sous forme de fenêtre glissante comme vue précédemment.

3.3.5 Configuration de l'expérimentation

Nous disposons de 7 jours de données, que nous avons répartis comme suit : 5 jours de données pour l'entraînement, 1 jour pour la validation et 1 jour pour les tests. La taille de la fenêtre glissante est fixée à 30, c'est-à-dire que l'on prend 30 informations le long de la série temporelle pour la fenêtre d'entrée, comme illustré sur la Figure 3.7. Pour les métriques d'évaluation, nous allons réutiliser la MAE, la MSE et la RMSE, décrites en détail dans le Chapitre 3.3.

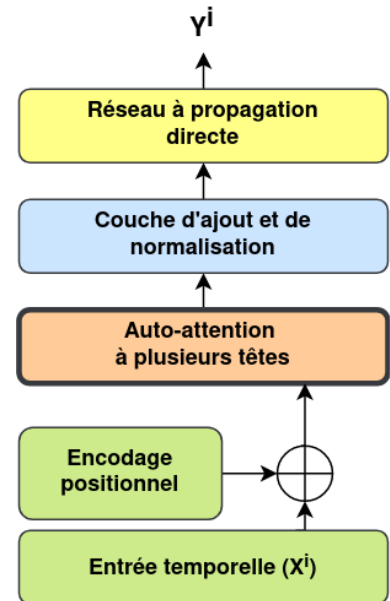


FIGURE 3.8 – Architecture d'un transformateur temporel.

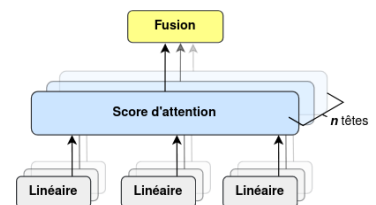


FIGURE 3.9 – Illustration des têtes d'attention.

[106]: Ashish VASWANI, Noam SHAZEER, Niki PARMAR, Jakob USZKOREIT, Llion JONES, Aidan N. GOMEZ, Lukasz KAISER et Illia POLOSUKHIN. « Attention Is All You Need ». In : CoRR abs/1706.03762 (2017). arXiv : 1706.03762

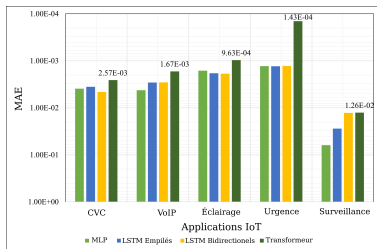


FIGURE 3.10 – MAE de la prédiction univariée du débit pour tous les ensembles de données.

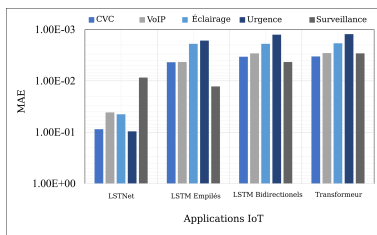


FIGURE 3.11 – MAE de la prédiction multivariée du débit pour tous les ensembles de données.

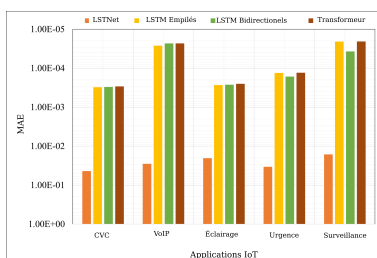


FIGURE 3.12 – MAE de la prédiction PDR multivariée pour tous les ensembles de données.

Ensuite, nous avons besoin de bases de comparaison pour évaluer la performance du modèle. Voici les modèles d'apprentissage profond proposés à la comparaison :

Perceptron multi-couches (MLP) Le perceptron multi-couches est un réseau de neurones avec au moins trois couches de neurones interconnectées : une couche d'entrée, des couches cachées et une couche de sortie. Chaque couche de neurones utilise des fonctions d'activation non linéaires pour modéliser la structure des relations entre les données, et les connexions entre les couches sont pondérées par des poids qui sont ajustés au fur et à mesure de l'apprentissage.

LSTM empilés Comme son nom l'indique, les LSTM empilés superposent des couches LSTM (Long Short-Term Memory). Un LSTM est un réseau de neurones artificiel qui utilise des portes d'informations permettant soit de stocker les informations importantes d'une séquence de données, soit de les oublier, ce qui lui permet de détecter les dépendances à long terme dans les données (ce qui est important pour les séries temporelles longues). Ici, chaque sortie d'un LSTM devient une entrée pour un autre LSTM.

LSTM bidirectionnel Le principe du LSTM bidirectionnel est assez simple : les données sont traitées non seulement dans une "direction" du passé au futur, mais aussi dans l'autre direction, du futur au passé. Cela permet de mieux comprendre une séquence de données et d'extraire des dépendances que l'on ne pourrait pas connaître autrement.

Pour rappel, le transformateur proposé peut avoir en entrée une variable ou plusieurs pour prédire une caractéristique. Pour évaluer les deux modes, nous allons donc comparer le mode univariable avec le MLP, les LSTM empilés et le LSTM bidirectionnel. Pour le mode multivarié, nous comparerons les LSTM empilés, le LSTM bidirectionnel. Les hyperparamètres du modèle proposé et des modèles de comparaison ainsi que l'algorithme complet du transformateur sont détaillés dans la Table 3.2.

3.3.6 Résultats : mode univarié

Débit La Table 3.3 contient les résultats des métriques d'évaluation des modèles de comparaison et du transformateur pour la prédiction du débit. Les MSE et MAE sont représentées graphiquement dans les Figures 3.13 et 3.13, respectivement. Comme pour les travaux précédents, l'échelle de la MSE et de la MAE est inversée : plus la valeur est basse, plus le diagramme est élevé, indiquant ainsi une meilleure performance. Tout d'abord, on peut observer que les applications de système d'urgence et d'éclairage intelligent donnent les meilleurs résultats pour tous les

TABLE 3.3 – Résultats des prévisions univariées pour le débit, les meilleurs résultats sont mis en évidence en gras.

Méthodes	MLP			LSTM empilés			LSTM bidirectionnels			Transformateur temporel		
	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE	MSE	RMSE
CVC	3.91e-3	2.11e-5	4.60e-3	3.56e-3	2e-5	4.48e-3	4.57e-3	2.82e-5	5.31e-3	2.57e-3	1.17e-5	3.42e-3
VoIP	4.20e-3	3.49e-5	5.91e-3	2.89e-3	1.53e-5	3.92e-3	2.87e-3	1.52e-5	3.9e-3	1.67e-3	4.27e-6	2.07e-3
Éclairage	1.62e-3	6.13e-6	2.47e-3	1.83e-3	6.67e-6	2.58e-3	1.87e-3	6.52e-6	2.55e-3	9.63e-4	1.49e-6	1.22e-3
Urgence	1.29e-3	6.86e-6	2.62e-3	1.3e-3	6.88e-6	2.62e-3	1.28e-3	6.87e-6	2.62e-3	1.43e-4	30e-8	1.73e-4
Surveillance	6.22e-2	6.92e-3	8.32e-2	2.76e-2	2.03e-3	4.5e-2	1.28e-2	7.74e-4	2.78e-2	1.26e-2	7.72e-4	2.78e-2

TABLE 3.4 – Résultats des prévisions univariées pour le PDR, les meilleurs résultats sont mis en évidence en gras.

Méthodes	MLP			LSTM empilés			LSTM bidirectionnels			Transformateur temporel		
Métriques	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE	MSE	RMSE
CVC	4.41e-3	2.60e-5	5.10e-3	4.40e-3	2.65e-5	5.15e-3	4.35e-3	2.54e-5	5.04e-3	4.15e-3	2.73e-5	5.23e-3
VoIP	2.71e-5	1.0e-9	3.16e-5	2.31e-5	1.0e-9	2.93e-5	2.43e-5	9.47e-10	3.08e-5	2.31e-5	8.55e-10	2.93e-5
Éclairage	2.47e-4	1.40e-7	3.74e-4	2.64e-4	1.44e-7	3.80e-4	2.65e-4	1.37e-7	3.70e-4	2.63e-4	1.37e-7	3.70e-4
Urgence	1.32e-4	3.0e-8	1.73e-4	1.32e-4	3.0e-8	1.73e-4	1.31e-4	2.9e-8	1.71e-4	8.06e-5	9.0e-9	9.73e-5
Surveillance	3.20e-5	2.0e-9	3.89e-5	2.05e-5	1.0e-9	2.65e-5	3.40e-5	1.96e-9	4.43e-5	2.0e-5	6.96e-10	2.53e-5

TABLE 3.5 – Résultats des prévisions univariées pour le PLR, les meilleurs résultats sont mis en évidence en gras.

Méthodes	MLP			LSTM empilés			LSTM bidirectionnels			Transformateur temporel		
Métriques	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE	MSE	RMSE
CVC	2.76e-5	1.22e-9	3.50e-5	2.36e-5	8.37e-10	2.89e-5	2.72e-5	1.16e-9	3.41e-5	2.27e-5	8.10e-10	2.84e-5
VoIP	2.39e-5	1.0e-9	3.32e-5	2.61e-5	1.30e-9	3.60e-5	1.77e-5	1.0e-9	2.42e-5	1.70e-5	5.82e-10	2.41e-5
Éclairage	2.74e-6	3.74e-11	6.12e-6	3.80e-6	3.37e-11	5.81e-6	3.86e-6	3.32e-11	5.76e-6	3.75e-6	3.32e-11	5.76e-6
Urgence	1.83e-12	5.40e-24	2.32e-12	2.16e-12	7.5e-24	2.74e-12	1.88e-12	5.3e-24	2.32e-12	1.80e-12	5.2e-24	2.30e-12
Surveillance	2.33e-3	5.46e-6	2.34e-3	2.67e-5	1.65e-9	4.06e-5	4.76e-4	2.45e-7	4.96e-4	2.57e-5	1.39e-9	3.73e-5

TABLE 3.6 – Résultats des prévisions univariées pour la latence, les meilleurs résultats sont en gras.

Méthodes	MLP			LSTM empilés			LSTM bidirectionnels			Transformateur temporel		
Métriques	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE	MSE	RMSE
CVC	2.47e-02	9.07e-04	3.01e-02	2.60e-02	9.26e-4	3.04e-02	2.31e-02	7.27e-04	2.70e-02	3.36e-03	1.56e-05	3.95e-03
VoIP	9.69E-01	5.58e+02	2.36e+01	1.34e-03	2.75e-06	1.66e-03	1.42e-03	3.14e-06	1.77e-03	1.27e-03	2.28e-6	1.51e-03
Éclairage	4.45e-02	4.09e-03	6.40e-02	4.44e-02	4.04e-03	6.36e-02	4.4166e-2	4.04e-03	6.36e-02	2.34e-02	1.09e-03	3.30e-02
Urgence	4.63e-02	3.73e-03	6.10e-02	4.90e-02	4.07e-03	6.38e-02	4.84e-02	4.06e-03	6.37e-02	3.86e-02	2.67e-03	5.17e-02
Surveillance	2.74e-04	1.20e-07	3.46e-04	2.76e-04	2.35e-06	1.53e-03	2.63e-04	1.15e-07	3.39e-04	1.54e-04	3.16e-08	1.78e-04

TABLE 3.7 – Résultats des prévisions multivariées pour le débit, les meilleurs résultats sont mis en évidence en gras.

Méthodes	LSTNet			LSTM empilés			LSTM bidirectionnels			Transformateur temporel		
Métriques	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE	MSE	RMSE
CVC	8.74e-2	7.63e-1	8.73e-1	4.34e-3	2.81e-5	5.30e-3	3.38e-3	1.17e-5	3.42e-3	3.35e-3	1.71e-5	4.14e-3
VoIP	4.10e-2	3.91e-1	6.25e-1	4.27e-3	3.32e-5	5.76e-3	2.91e-3	1.55e-5	3.94e-3	2.87e-3	1.48e-5	3.85e-3
Éclairage	4.46e-2	6.88e-1	8.29e-1	1.90e-3	7.18e-6	2.68e-3	1.89e-3	7.17e-6	2.68e-3	1.85e-3	7.10e-6	2.66e-3
Urgence	9.65e-2	1.45e-1	3.81e-1	1.64e-3	7.71e-6	2.78e-3	1.26e-3	6.72e-6	2.59e-3	1.23e-3	6.67e-6	2.50e-3
Surveillance	8.60e-3	9.14e-2	3.02e-1	1.28e-2	7.73e-4	2.78e-2	4.29e-3	3.65e-5	6.04e-3	2.91e-3	1.55e-5	3.93e-3

TABLE 3.8 – Résultats des prévisions multivariées pour le PDR, les meilleurs résultats sont mis en évidence en gras.

Méthodes	LSTNet			LSTM empilés			LSTM bidirectionnels			Transformateur temporel		
Métriques	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE	MSE	RMSE
CVC	4.37e-2	5.07e-1	7.12e-1	3.07e-4	1.16e-7	3.41e-4	3.04e-4	1.15e-7	3.39e-4	2.94e-4	1.31e-7	3.62e-4
VoIP	2.84e-2	3.52e-1	5.93e-1	2.64e-5	1.12e-9	3.35e-5	2.31e-5	8.58e-10	2.92e-5	2.31e-5	8.57e-10	2.90e-5
Éclairage	2.03e-2	6.54e-1	8.09e-1	2.70e-4	1.37e-7	3.70e-4	2.65e-4	1.37e-7	3.70e-4	2.52e-4	1.40e-7	3.74e-4
Urgence	3.39e-2	6.58e-1	8.11e-1	1.32e-4	2.96e-8	1.72e-4	1.64e-4	4.57e-8	2.14e-4	1.31e-4	2.80e-8	1.67e-4
Surveillance	1.63e-2	2.11e-1	4.59e-1	2.08e-5	7.28e-10	2.70e-5	3.72e-5	2.29e-9	4.79e-5	2.06e-5	7.20e-10	2.69e-5

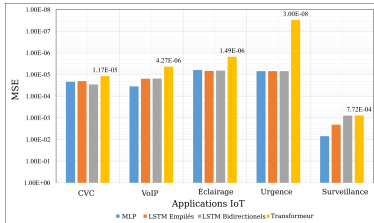
TABLE 3.9 – Résultats des prévisions multivariées pour le DPP, les meilleurs résultats sont mis en évidence en gras.

Méthodes	LSTNet			LSTM empilés			LSTM bidirectionnels			Transformateur temporel		
Métriques	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE	MSE	RMSE
CVC	5.24e-2	4.75e-1	6.89e-1	3.92e-5	2.09e-9	4.57e-5	4.73e-5	4.0e-9	6.33e-5	3.89e-5	2.09e-9	4.57e-5
VoIP	3.77e-2	3.87e-1	6.22e-1	2.21e-5	3.37e-5	1.14e-9	5.04e-5	5.93e-9	7.71e-5	2.20e-5	3.39e-5	1.15e-9
Éclairage	4.03e-2	6.85e-1	8.28e-1	3.94e-6	3.33e-11	5.77e-6	4.14e-6	3.55e-11	5.96e-6	3.84e-6	3.32e-11	5.76e-6
Urgence	2.93e-2	6.27e-1	7.91e-1	1.90e-12	5.68e-24	2.38e-12	1.88e-12	6.27e-24	2.50e-12	1.87e-12	5.64e-24	2.37e-12
Surveillance	8.6e-3	1.06e-1	3.26e-1	2.44e-4	6.07e-7	2.46e-4	1.12e-3	2.0e-6	1.42e-3	1.89e-5	8.34e-10	2.89e-5

modèles. En revanche, les applications de surveillance, de VoIP et de système CVC donnent de moins bons résultats. Cela est dû au fait que

TABLE 3.10 – Résultats des prévisions multivariées pour la latence, les meilleurs résultats sont mis en évidence en gras.

Méthodes	LSTNet			LSTM empilés			LSTM bidirectionnels			Transformateur temporel		
Métriques	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE	MSE	RMSE
CVC	8.40e-2	2.81e0	1.66e0	2.56e-2	9.28e-4	3.05e-2	2.32e-2	7.81e-4	2.79e-2	2.27e-2	7.33e-4	2.71e-2
VoIP	2.41e-2	2.68e-1	5.052e-1	1.35e-3	2.853e-6	1.69e-3	1.35e-3	2.85e-6	1.69e-3	9.24e-4	1.18e-6	1.09e-3
Éclairage	7.13e-2	1.52e0	1.23e0	3.55e-2	2.56e-3	5.06e-2	3.63e-2	2.56e-3	5.06e-2	2.27e-2	1.07e-3	3.27e-2
Urgence	4.18e-2	6.65e-1	8.10e-1	6.35e-2	6.75e-3	8.22e-2	4.77e-02	3.88e-03	6.23e-02	3.88e-02	2.26e-03	4.75e-02
Surveillance	2.41e-02	2.68e-01	5.05e-01	2.53e-04	1.16e-07	3.40e-04	2.49e-04	9.66e-08	3.11e-04	1.56e-04	3.26e-08	1.81e-04

**FIGURE 3.13** – MSE de la prédiction univariée du débit pour tous les ensembles de données.

ces données contiennent ce qu'on appelle des valeurs aberrantes. On sait que les modèles d'apprentissage profond ont plus de mal à apprendre à partir de ces valeurs extrêmes rares, dites aberrantes, ce qui explique ces performances plus basses. La Table 2.3.4 du Chapitre 2.4 montre que, pour ces applications, les écarts-types sont éloignés de la moyenne, ce qui confirme l'existence de ces valeurs aberrantes. En ce qui concerne les applications d'éclairage intelligent et de système d'urgence, il y a également des valeurs extrêmes, mais elles ne sont pas considérées comme aberrantes car elles sont fréquentes. Les modèles sont donc capables d'apprendre à partir de ces valeurs. Les Figures 3.13 et 3.13 montrent clairement que le modèle de transformateurs proposé obtient une erreur plus faible par rapport aux autres modèles. Cela est dû à la capacité des transformateurs temporels à manipuler correctement une longue série temporelle de données, ainsi qu'au mécanisme de têtes d'attention.

Taux de paquets délivrés (PDR) La Table 3.4 contient les résultats pour le PDR. On peut voir que le modèle de transformateurs est le plus efficace dans la majorité des cas, mais pas dans tous. En effet, pour le système CVC, le LSTM bidirectionnel donne la MSE et la RMSE les plus basses en premier. On peut expliquer que le transformateur n'a pas réussi à atteindre ces performances car le modèle a essayé d'apprendre les valeurs aberrantes depuis le module d'attention, ce qui a eu un impact sur les relations entre les données. On note également que ces deux métriques sont plus sensibles aux grandes erreurs. Ensuite, pour l'éclairage intelligent, le MLP obtient le plus bas MAE. Néanmoins, le transformateur obtient quand même la MSE et la RMSE les plus basses.

Taux de paquets perdus (PLR) et latence La Table 3.5 fournit les résultats pour le PLR, et la Table 3.6 pour la latence de transmission. Ici aussi, le transformateur obtient les meilleurs résultats dans la majorité des cas. On note comme cas particulier dans la prédiction du PLR pour l'application d'éclairage intelligent la MAE la plus basse pour le MLP. Cela s'explique par les mêmes raisons que pour le PDR.

En conclusion, le modèle de transformateur temporel obtient la meilleure performance sur 18 des 20 paramètres pour la MAE, sur 19 des 20 paramètres pour la MSE, et sur 19 des 20 paramètres pour la RMSE. Il offre une amélioration significative des performances vis-à-vis des métriques d'évaluation citées, c'est-à-dire la MAE, la MSE et la RMSE. Notamment, il améliore les prédictions du débit de 28% à 89% selon les applications IoT. Pour le PDR, la réduction de l'erreur de la MAE est de 0,43% à 38%, et pour la PLR, la réduction de la MAE est de 2% à 4%. Finalement, l'amélioration de la MAE est de 5% à 96% pour la prédiction de la latence. On note toutefois des réductions de performance

par rapport à d'autres modèles, qui s'expliquent par l'existence de valeurs aberrantes dans les données.

3.3.7 Résultats : mode multivarié

Débit La Table 3.7 contient les résultats des métriques d'évaluation des modèles de comparaison et du transformateur pour la prédiction du débit à plusieurs variables. Les résultats de la MAE pour chaque application et modèle sont représentés graphiquement dans la Figure 3.11, avec toujours une échelle des ordonnées inversée. On peut rapidement voir que le LSTM bidirectionnel présente des performances quasi équivalentes aux transformateurs. On observe une amélioration de la MAE par rapport au LSTM bidirectionnel de 1% pour les applications CVC et VoIP, de 2% pour les applications d'éclairage intelligent et d'urgence, et finalement une amélioration plus importante de 32% pour l'application de surveillance. Cette plus importante amélioration est due aux fluctuations à long terme dans le jeu de données de surveillance que le transformateur est le plus à même de capturer et de prédire.

Taux de paquets délivrés (PDR) La Table 3.8 affiche les résultats de manière similaire pour le PDR. Cette table nous montre que le modèle transformateur obtient la plus faible MAE pour toutes les applications. Néanmoins, le LSTM bidirectionnel obtient la meilleure MSE et RMSE pour l'éclairage intelligent et le CVC. Cela s'explique par la présence de valeurs aberrantes dans ces deux jeux de données et la taille réduite de l'ensemble de données, alors que le transformateur a besoin de plus de données d'entraînement. En effet, comme le LSTM bidirectionnel fonctionne dans les deux sens, il peut être plus précis avec moins de données. La Figure 3.12 présente la MAE par modèle et par application. On voit que le transformateur temporel offre de meilleures performances sur tous les jeux de données, suivis par les LSTM empilés pour les applications de surveillance et d'urgence. Ensuite, nous avons le LSTM bidirectionnel pour les applications CVC, VoIP et d'éclairage. On observe une réduction de la MAE de 1% à 5% par rapport au LSTM bidirectionnel.

Taux de paquets perdus (PLR) Les résultats de la prédiction du PLR sont présentés dans la Table 3.9. On y observe que le transformateur est le modèle le plus performant dans la majorité des cas. Cependant, on note que les LSTM empilés présentent de meilleurs résultats en termes de MSE et RMSE sur l'application VoIP. On peut l'expliquer car les LSTM empilés apprennent plus facilement les dépendances non linéaires complexes, comme expliqué précédemment, entre les différentes séries temporelles. Ces types de dépendances se produisent facilement avec des interférences réseau fréquentes en IoT et une réduction de la bande passante à disposition.

Latence Finalement, le Tableau 3.10 présente les résultats pour la latence. Le modèle proposé surpasse les autres modèles de comparaison sur toutes les applications et toutes les mesures d'évaluation. Ensuite, il y a le LSTM bidirectionnel qui présente de bons résultats sur 4 des 5 applications.

Pour conclure sur le mode à plusieurs variables, le modèle transformateur temporel acquiert la meilleure performance sur 20 des 20 paramètres pour la MAE et sur 16 des 20 paramètres pour la MSE et la RMSE. Il

y a une amélioration de 1% à 92% sur les mesures d'évaluation par le transformateur temporel. On obtient sur la MAE une performance améliorée de 2% à 37%, pour la MSE de 6% à 66% et pour la RMSE de 3% à 42% comparé au deuxième meilleur modèle.

3.3.8 Mode univarié ou multivarié ?

Avec ces différentes analyses, nous avons pu effectivement constater que le modèle démontre de meilleures performances dans le mode univarié que dans le mode multivarié. En effet, le transformateur temporel a obtenu de moins bons résultats dans 4 cas univariés contre 8 cas pour le mode multivarié. Le mode multivarié peut permettre de détecter des dépendances entre les différentes caractéristiques et proposer des prédictions plus fines, mais cela s'accompagne nécessairement d'une complexité accrue. Les valeurs aberrantes vont alors avoir un plus gros impact sur la prédiction dans un contexte multivarié, et les ensembles de données IoT en contiennent beaucoup, ce qui explique les moins bonnes performances. Il faut aussi noter qu'il est bien plus facile de repérer et de contrôler les valeurs aberrantes dans une seule série temporelle que dans plusieurs. On peut en conclure qu'il est préférable d'utiliser le mode univarié pour la prédiction des métriques de la QoS des applications IoT.

3.3.9 Conclusion

Pour ce travail, nous avons étudié la prédiction de la qualité de service (QoS) à partir des données présentées au Chapitre 2.4, formulées sous forme de séries temporelles dans un cadre multivarié et univarié. Les applications IoT hétérogènes sur l'infrastructure des réseaux sans fil ont des besoins stricts en matière de QoS, et ce travail permet donc d'anticiper la baisse de la QoS pour pouvoir y remédier. L'architecture basée sur les transformateurs temporels permet de prédire avec efficacité les métriques importantes de la QoS.

3.4 Conclusion

Dans ce chapitre, nous avons exploré des modèles d'apprentissage automatique basés sur des techniques de régression et de réseaux neuronaux profonds pour la prédiction de séries temporelles de la Qualité de Service dans les réseaux IoT. Tout d'abord, nous avons comparé divers modèles de régression pour prédire le débit. Nous avons conclu que la régression linéaire multiple (MLR) présentait généralement les meilleures performances sur trois des applications IoT incluses dans les données. Ensuite, nous avons développé un modèle de transformateurs temporels pour la prédiction de plusieurs métriques de la Qualité de Service, notamment le débit, le taux de paquets délivrés (PDR), le taux de paquets perdus (PLR) et la latence. Nous avons comparé ce modèle et analysé les résultats, démontrant son efficacité pour la prédiction multimétrique de la QoS.

Ces travaux peuvent être utilisés par les périphériques du réseau pour anticiper le déploiement et l'allocation des ressources. Les données utilisées dans ce chapitre ont été fournies par les travaux du Chapitre 2.

Prédiction de la densité humaine

4.1 Prédiction de la densité humaine

Nous avons présenté nos travaux axés sur la **prédiction de la QoS** dans les Chapitres 2 et 3. Dans ce Chapitre nous traitons d'une autre partie de notre problématique, présentée dans la Section 1.5, c'est-à-dire la **prédiction de la mobilité des foules**.

4.1.1 Pourquoi prédire la mobilité humaine ?

La prédiction de la mobilité humaine peut améliorer la qualité de service (QoS) des réseaux mobiles et IoT en réduisant le temps d'interruption lorsque les utilisateurs se déplacent et changent de stations de base, anticipant ainsi l'allocation des ressources radio. Dans le cas de réseaux mobiles intégrant des ressources locales, notamment le MEC, la prédiction de la mobilité des utilisateurs est encore plus importante, car la migration des services peut subir des interruptions importantes en raison du temps de configuration et de démarrage des machines virtuelles correspondantes, comme déjà vu. De plus, lorsque le MEC intègre des ressources embarquées sur des unités mobiles telles que des robots, des véhicules aériens sans pilote (UAV) ou des véhicules, la prédiction de la mobilité et de la densité humaine aidera à déployer les ressources mobiles au bon moment et au bon endroit et donc anticiper le temps de déploiement. En effet, le schéma de déploiement des ressources MEC mobiles doit savoir à l'avance où les ressources seront nécessaires pour anticiper le temps de déplacement des unités mobiles.

4.1.2 Mais est-ce que la mobilité humaine est prédictible ?

La recherche a montré que la mobilité humaine est définie par une régularité spatiale et temporelle élevée, que ce soit en semaine, le week-end, en ville ou en milieu rural [107], et est donc prévisible. En effet, les individus ont tendance à fréquenter quelques endroits de manière très régulière [108, 109], en partie influencée par les habitudes sociales et les relations humaines [110]. Cette régularité élevée dans la mobilité humaine quotidienne justifie l'intérêt scientifique de prédire la mobilité future en inférant des modèles inhérents [111]. L'explosion récente des techniques d'apprentissage en profondeur et d'apprentissage automatique favorise encore davantage la faisabilité de cette prédiction en raison de leur capacité à apprendre des modèles [112]. De plus, l'augmentation du nombre d'utilisateurs de téléphones mobiles et des systèmes de positionnement modernes offre un bon moyen de collecter de vastes ensembles de données pour alimenter les méthodes d'apprentissage et modéliser la mobilité humaine [113]. Cependant, cette régularité dans la mobilité peut être perturbée par des événements inhabituels ou des

4.1 Prédiction de la densité humaine	53
4.1.1 Pourquoi prédire la mobilité humaine ?	53
4.1.2 Mais est-ce que la mobilité humaine est prédictible ?	53
4.1.3 Prédiction de la mobilité pour le MEC	54
4.2 État de l'art	54
4.2.1 Prédiction de la mobilité à l'échelle de la ville	54
4.2.2 Prédiction de la mobilité dans les réseaux mobiles	55
4.2.3 Le meilleur des deux mondes	56
4.3 Contexte technique	56
4.3.1 Les filtres à particules	56
4.3.2 Filtre à particules pour la prédiction de la densité humaine	57
4.4 Contribution	59
4.4.1 Simulateur de Mobilité Humaine	59
4.4.2 Collecte de données d'observation	59
4.4.3 Diffusion des données	60
4.5 Configuration des expériences	62
4.5.1 Expérience sur la communication	62
4.5.2 Expérience sur la partie calcul du filtre	62
4.6 Résultats	65
4.6.1 Protocole de commérag	65
4.6.2 Prévisions du filtre à particule	71
4.7 Conclusion	73

[112]: Peng XIE, Tianrui LI, Jia LIU, Shengdong DU, Xin YANG et Junbo ZHANG. « Urban flows prediction from spatial-temporal data using machine learning : A survey ». In : *Inf. Fusion* 59 (2019)

[113]: Pierre DEVILLE, Catherine LINARD, Samuel MARTIN, Marius GILBERT, Forrest R. STEVENS, Andrea E. GAUGHAN, Vincent D. BLONDEL et Andrew J. TATEM. « Dynamic population mapping using mobile phone data ». In : *Proceedings of the National Academy of Sciences* 111.45 (2014). eprint : <https://www.pnas.org/doi/pdf/10.1073/pnas.1408439111>

[114]: Qi WANG et John E. TAYLOR. « Patterns and Limitations of Urban Human Mobility Resilience under the Influence of Multiple Types of Natural Disaster ». In : *PLOS* (jan. 2016)

[115]: Xuan SONG, Quanshi ZHANG, Yoshihide SEKIMOTO, Teerayut HORANONT, Satoshi UYAMA et Ryosuke SHIBASAKI. « Modeling and Probabilistic Reasoning of Population Evacuation during Large-Scale Disaster ». In : *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '13. Chicago, Illinois, USA : Association for Computing Machinery, 2013. ISBN : 9781450321747

[116]: Christian SCHRECKENBERGER, Christian BARTELT et Heiner STUCKENSCHMIDT. « Enhancing a Crowd-Based Delivery Network with Mobility Predictions ». In : *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Prediction of Human Mobility*. PredictGIS'19. Chicago, IL, USA : Association for Computing Machinery, 2019. ISBN : 9781450369640

[117]: Akihito SUDO, Takehiro KASHIYAMA, Takahiro YABE, Hiroshi KANASUGI, Xuan SONG, Tomoyuki HIGUCHI, Shin'ya NAKANO, Masaya SAITO et Yoshihide SEKIMOTO. « Particle Filter for Real-Time Human Mobility Prediction Following Unprecedented Disaster ». In : *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. SIGSPACIAL '16. Burlingame, California : Association for Computing Machinery, 2016. ISBN : 9781450345897

[118]: Yu ZHENG, Licia CAPRA, Ouri WOLFSON et Hai YANG. « Urban Computing : Concepts, Methodologies, and Applications ». In : *ACM Trans. Intell. Syst. Technol.* 5 (2014)

[119]: Chao FAN, Xiangqi JIANG et Ali MOSTAFAVI. « Adaptive Reinforcement Learning Model for Simulation of Urban Mobility during Crises ». In : *ArXiv abs/2009.01359* (2020)

catastrophes. Lorsqu'une catastrophe naturelle survient, la mobilité humaine peut montrer de la résilience, mais seulement jusqu'à un certain point, en fonction de la gravité de la catastrophe [114]. Des ensembles de données sur la mobilité pendant les catastrophes existent [115], mais leur réutilisation est limitée car ils dépendent fortement du type et du déroulement de la catastrophe.

4.1.3 Prédiction de la mobilité pour le MEC

Dans le cas de la prédiction de la mobilité humaine pour le déploiement de ressources mobiles du MEC, nous pouvons emprunter deux voies principales : (i) la prédiction axée sur les individus, où nous pouvons prédire leur itinéraire ou leur prochain emplacement [116] (ii) la prédiction d'une population d'individus [117], c'est-à-dire la trajectoire d'une foule. Ces deux approches peuvent être complémentaires, mais lorsqu'il s'agit du déploiement d'unités mobiles, la prédiction de la densité, et donc de la prédiction de la mobilité d'une foule, peut présenter l'avantage d'offrir une vue plus générale pour répondre à la demande en temps réel. En effet, comme nous ne pouvons pas avoir une unité mobile pour chaque utilisateur, nous souhaitons déployer les ressources disponibles là où elles sont nécessaires au lieu de suivre les utilisateurs. De plus, la nature distribuée et dynamique de la MEC activée par des robots soulève certains défis, tels que l'isolement potentiel de certaines unités mobiles. Pour ces raisons, nous proposons ici une prédiction de la mobilité de la foule pour le déploiement d'unités MEC mobiles, à la fois les jours réguliers et les jours irréguliers. Notre contribution principale dans ce travail est de distribuer un modèle de filtre à particules pour la prédiction de la densité humaine, inspiré par le travail de Sudo et al. [117], dans le contexte du calcul multi-accès en périphérie avec des unités mobiles.

4.2 État de l'art

4.2.1 Prédiction de la mobilité à l'échelle de la ville

Plusieurs travaux antérieurs en informatique urbaine [118] visent à prédire la densité de foule ou les comportements humains lors d'événements irréguliers ou de catastrophes. Ces travaux ont pour objectif d'aider les secours à anticiper les flux de population et à organiser les opérations dans les grandes villes. Fan et al. [119] proposent un modèle d'apprentissage par renforcement pour la prédiction de la mobilité urbaine adapté aux scénarios de catastrophe urbaine. Leur modèle apprend d'abord le point de destination en fonction du point d'origine, à partir de données de scénarios réguliers. À partir de la paire origine-destination, un apprentissage par renforcement inverse prédit la trajectoire de l'agent, puis met à jour sa table en ajoutant des informations contextuelles telles que les routes inondées ou les zones fermées sur la carte. Le modèle est très adaptable car nous pouvons définir les informations contextuelles ajoutées en fonction de la situation, et nous n'avons besoin que de jeux de données pour les scénarios réguliers. Cependant, ce modèle pourrait être difficile à utiliser en temps réel en raison du manque d'informations temporelles dans les résultats, du coût en calcul élevé et du temps de

mise en œuvre long, comme mentionné par les auteurs. Fan et al. [120] proposent un modèle à court terme pour prédire la mobilité humaine basée sur les mouvements de foule. Ils déduisent la direction future à court terme des personnes à partir de leurs mouvements récents et des mouvements récents passés de la foule environnante. Cela repose sur l'hypothèse que les foules se rassemblent lentement, ce qui permet de prédire les nouvelles arrivées à partir des précédentes, et sur la probabilité que les gens suivent la même direction que ceux qui ont eu une trajectoire similaire précédente. Jiang et al. [121] proposent un modèle de prédiction de la mobilité humaine pour les régions d'intérêt (ROI) à l'intérieur des villes. Ils définissent un ensemble de ROI pour chaque heure d'une journée à partir de données passées. Ensuite, le modèle prédit, encore une fois pour chaque heure, à l'aide de réseaux de neurones récurrents profonds, si quelqu'un entrera dans une ROI et laquelle. Jiang et al. [122] présentent un modèle de prédiction de la densité de foule et du flux d'entrée-sortie, où le flux d'entrée-sortie représente le nombre de personnes entrant et sortant de la grille de la carte. Ils utilisent un modèle d'apprentissage profond et fournissent deux ensembles de données qu'ils ont générés et utilisés, qui sont à granularité fine et couvrent de vastes zones. Konishi et al. [123] proposent une prédiction de la taille de la population pour des régions spécifiques en utilisant les horaires des utilisateurs à partir de journaux d'application de transport en commun. L'idée est que les gens planifient à l'avance leurs trajectoires régulières et irrégulières via les journaux d'application, ce qui permet des prédictions possibles jusqu'à une semaine à l'avance. Cependant, il ne sera pas en mesure de prédire des irrégularités soudaines telles que des catastrophes. Sudo et al. [117] proposent une prédiction en temps réel de la densité de population pour les catastrophes à l'aide de données de localisation anonymisées. Ils utilisent un filtre à particules qu'ils étendent pour résoudre la grande dimensionnalité du problème. Cependant ce travail est centralisé et n'est pas adapté pour le déploiement d'unités mobiles de manière automatisé.

4.2.2 Prédiction de la mobilité dans les réseaux mobiles

Comme cela peut offrir plusieurs avantages, la prédiction de la mobilité humaine dans les réseaux mobiles est également un sujet largement étudié.

Ozturk et al. [124] proposent une prédiction de transfert basée sur un réseau neuronal profond intégrée dans un modèle de coût de transfert pour l'architecture de réseau cellulaire à séparation de contrôle et de données. Ils montrent que la prédiction de transfert est plus efficace qu'une approche non prédictive uniquement lorsqu'un certain seuil de précision est atteint. Mohamed et al. [125] proposent un modèle de prédiction de la mobilité pour gérer les transferts dans les réseaux cellulaires avec séparation de contrôle et de données. Ils représentent le réseau avec une chaîne de Markov à temps discret et tiennent compte des mouvements aléatoires et moins fréquents. Farooq et al. [126] présentent le framework AURORA de commutation de cellules pour la minimisation de l'énergie dans les réseaux ultra-denses. Le framework utilise une prédiction de mobilité basée sur un modèle semi-markovien qui prédit avec succès le prochain transfert dans l'espace et le temps. Plachy et

[120]: Zipei FAN, Xuan SONG, Ryosuke SHIBASAKI et Ryutaro ADACHI. « City-Momentum : An Online Approach for Crowd Behavior Prediction at a Citywide Level ». In : *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 2015

[121]: Renhe JIANG, Xuan SONG, Zipei FAN, Tianqi XIA, Qianjun CHEN, Qi CHEN et Ryosuke SHIBASAKI. « Deep ROI-Based Modeling for Urban Human Mobility Prediction ». In : *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2 (2018)

[122]: Renhe JIANG, Zekun CAI, Zhaonan WANG, Chuang YANG, Zipei FAN, Qianjun CHEN, Kota TSUBOUCHI, Xuan SONG et Ryosuke SHIBASAKI. « DeepCrowd : A Deep Model for Large-Scale Citywide Crowd Density and Flow Prediction ». In : *IEEE Transactions on Knowledge and Data Engineering* (2021)

[123]: Tatsuya KONISHI, Mikiya MARUYAMA, Kota TSUBOUCHI et Masamichi SHIMOSAKA. « CityProphet : City-Scale Irregularity Prediction Using Transit App Logs ». In : *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 2016

[124]: Metin OZTURK, Mandar GOGATE, Oluwakayode ONIRETI, Ahsan ADEEL, Amir HUSSAIN et Muhammad A. IMRAN. « A novel deep learning driven, low-cost mobility prediction approach for 5G cellular networks : The case of the Control/Data Separation Architecture (CDSA) ». In : *Neurocomputing* 358 (2019)

[125]: Abdelrahim MOHAMED, Oluwakayode ONIRETI, Seyed Amir HOSEINITABATABAEI, Muhammad IMRAN, Ali IMRAN et Rahim TAFAZOLLI. « Mobility prediction for handover management in cellular networks with control/data separation ». In : *2015 IEEE International Conference on Communications (ICC)*. 2015

[126]: Hasan FAROOQ, Ahmad ASGHAR et Ali IMRAN. « Mobility Prediction-Based Autonomous Proactive Energy Saving (AURORA) Framework for Emerging Ultra-Dense Networks ». In : *IEEE Transactions on Green Communications and Networking* 2 (2018)

[127]: Jan PLACHY, Zdenek BECVAR et Emilio Calvanese STRINATI. « Dynamic resource allocation exploiting mobility prediction in mobile edge computing ». In : *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. 2016

[128]: Amnir HADACHI, Oleg BATRASHEV, Artjom LIND, Georg SINGER et Eero VAINIKKO. « Cell phone subscribers mobility prediction using enhanced Markov Chain algorithm ». In : *2014 IEEE Intelligent Vehicles Symposium Proceedings*. 2014

[129]: Andre S. GOMES, Bruno SOUSA, David PALMA, Vitor FONSECA, Zhongliang ZHAO, Edmundo MONTEIRO, Torsten BRAUN, Paulo SIMOES et Luis CORDEIRO. « Edge caching with mobility prediction in virtualized LTE mobile networks ». In : *Future Generation Computer Systems* 70 (2017)

[130]: Hua WEI, Hong LUO et Yan SUN. « Mobility-Aware Service Caching in Mobile Edge Computing for Internet of Things ». In : *Sensors* 20 (2020)

[131]: Teruo HIGASHINO, Hirozumi YAMAGUCHI, Akihito HIROMORI, Akira UCHIYAMA et Keiichi YASUMOTO. « Edge Computing and IoT Based Research for Building Safe Smart Cities Resistant to Disasters ». In : *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. 2017

al. [127] introduisent un algorithme pour les migrations de service de machine virtuelle et la sélection du chemin de communication pour les réseaux de calcul mobile à l'extrémité (MEC) en utilisant une prédiction de mobilité à partir d'un travail précédent [128]. L'algorithme est montré pour réduire la latence de délestage et favoriser l'équité en matière d'énergie et de latence parmi les utilisateurs. Gomes et al. [129] utilisent des travaux précédents sur la prédiction de la mobilité pour permettre une migration de contenu de cache pour le paradigme "Follow-me Cloud". Wei et al. [130] conçoivent un service de mise en cache pour le MEC basé sur un modèle de mobilité à court terme. Le modèle prévoit la position de l'utilisateur en fonction de sa position actuelle, de sa vitesse et de sa direction. Lorsqu'aucune information sur l'utilisateur n'est disponible, il infère les données manquantes à partir de la mobilité d'autres utilisateurs. Le modèle est bien adapté pour un service de cache local avec des ressources fixes, mais ne l'est pas pour de la prédiction avec une vue plus générale pour le déploiement d'unités mobiles.

4.2.3 Le meilleur des deux mondes

Dans le domaine de la prédiction de la mobilité en informatique urbaine, les approches sont centralisées et peuvent être trop lourdes pour être traitées en temps réel, comme c'est nécessaire dans le MEC. De plus, ces travaux ne tiennent pas compte des contraintes et des spécificités des réseaux mobiles, telles que les limitations dans la collecte d'informations, le manque de vue centralisée et la surveillance des zones de données manquantes [131]. Dans le domaine des réseaux mobiles, les travaux précédents se concentrent principalement sur la mobilité individuelle, ce qui convient pour la migration de services ou de caches mais moins pour le déploiement de services où une vue générale est importante pour déployer des ressources rares dans le temps et l'espace. De plus, ils reposent sur des données régulières, ne s'adaptant donc pas à la mobilité irrégulière telle que les grands événements ou les catastrophes et ne tiennent pas compte de scénarios à grande échelle.

Inspirés par les travaux dans ces deux domaines, nous proposons un modèle de prédiction de densité à court terme pour les événements réguliers et irréguliers dans le MEC, en utilisant des serveurs MEC intégrés aux unités mobiles. Pour cela, nous utiliserons une prédiction de la densité de foule sur la carte dans les prochaines étapes de temps pour identifier les futurs regroupements de personnes et anticiper la placement et le déploiement des services réseau.

4.3 Contexte technique

4.3.1 Les filtres à particules

Les méthodes de Monte Carlo sont des techniques populaires qui utilisent l'échantillonnage aléatoire pour résoudre des problèmes complexes. Elles explorent l'espace de résolution dimensionnelle par le biais de simulations plutôt que par une exploration exhaustive, qui serait très chronophage, voire impossible, car l'espace de résolution peut s'avérer

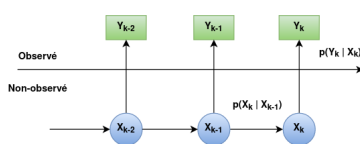


FIGURE 4.1 – Modèle espace-état.

très grand. L'une de ces méthodes est la méthode de Monte Carlo séquentielle, également connue sous le nom de filtre à particules [132]. Les filtres à particules estiment un état caché x_t à chaque instant t d'un système dynamique qui ne peut pas être directement mesuré ou observé. Cette estimation se fait par le biais d'un autre état y_t , qui peut être mesuré et est lié à l'état caché que nous souhaitons estimer, comme illustré par la Figure 4.1. La fonction de densité de probabilité *a posteriori* (appelée la postérieure) $P(x|y_{1:t})$ que nous souhaitons prédire est représentée par un ensemble de particules pondérées x_t^i, w_t^i , où N est la taille de l'ensemble et i représente l'identifiant de la particule individuelle parmi les N particules de l'ensemble. Une particule x_t^i représente une hypothèse et le poids est son importance. Chaque itération du filtre à particules à l'instant t comprend deux étapes : une étape de prédiction et une étape de mise à jour (ou de correction). Dans l'étape de prédiction, nous échantillons les particules x_t^i en fonction d'une distribution de proposition dépendant des estimations passées. Dans l'étape de mise à jour, nous déterminons le poids w_t^i associé aux particules en fonction de la vraisemblance $P(y_t|x_t)$ des observations. Cependant, cela peut conduire à des particules ayant un poids très faible sauf une, ce qui est appelé le problème de dégénérescence [133]. Pour remédier à cela, une étape de rééchantillonnage est ajoutée, où N nouvelles particules sont sélectionnées parmi les anciennes particules avec une probabilité égale à leur poids afin de supprimer probablement les particules de faible poids et de dupliquer les particules de fort poids. Les nouvelles particules sont affectées d'un poids normalisé. Il existe différentes méthodes de rééchantillonnage et cette étape peut être déclenchée en fonction d'un seuil. La Figure 4.2 illustre ce procédé.

4.3.2 Filtre à particules pour la prédiction de la densité humaine

Dans le travail de Sudo et al. [117], un filtre à particules est utilisé pour prévoir la densité de population dans une région géographique donnée. Leur approche repose sur un filtre à particules centralisé, où le vecteur d'état, noté $x_t \in \mathbb{R}^{2N_p + N_\theta}$, comprend deux composantes : la localisation des individus ($a_t \in \mathbb{R}^{2N_p}$) et un vecteur de paramètres comportementaux ($\theta_t \in \mathbb{R}^{N_\theta}$) à l'instant t . Ici, N_p et N_θ correspondent respectivement au nombre d'individus et de paramètres comportementaux. Ce vecteur de paramètres comportementaux, θ_t , est formulé comme un modèle qui met à jour les destinations des individus en fonction de θ_t , qui représentent les probabilités de transition entre les emplacements significatifs prédéfinis. Par exemple, la probabilité de transition entre le domicile d'une personne et son lieu de travail. Les données de distribution de la population ($d_n T \in \mathbb{N}^{N_d}$) sont l'état observé et sont obtenues auprès d'un opérateur mobile à chaque pas de temps discret nT . Les auteurs estiment que la durée de chaque pas de temps T est d'environ 1 heure. Le processus de leur méthode proposée est le suivant : entre les intervalles d'observation, c'est-à-dire lorsque les données d'observation ne sont pas disponibles, la position a_t est mise à jour grâce à un modèle de prédiction, plus précisément un simulateur de mobilité humaine. Lorsque des données d'observation sont fournies, d'abord le modèle de prédiction déplace les particules, puis les particules sont encore déplacées vers des régions

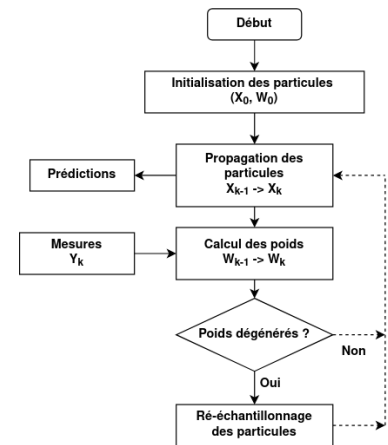


FIGURE 4.2 – Algorithme du filtre à particules.

[132]: Aditya TULSYAN, R. BHUSHAN GOPALUNI et Swanand R. KHARE. « Particle filtering without tears : A primer for beginners ». In : *Computers & Chemical Engineering* 95 (2016)

[133]: A. DOUCET, Simon J. GODSILL et Christophe ANDRIEU. « On sequential Monte Carlo sampling methods for Bayesian filtering ». In : *Statistics and Computing* 10 (2000)

[117]: Akihito SUDO, Takehiro KASHIYAMA, Takahiro YABE, Hiroshi KANASUGI, Xuan SONG, Tomoyuki HIGUCHI, Shin'ya NAKANO, Masaya SAITO et Yoshihide SEKIMOTO. « Particle Filter for Real-Time Human Mobility Prediction Following Unprecedented Disaster ». In : *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. SIGSPACIAL '16*. Burlingame, California : Association for Computing Machinery, 2016. ISBN : 9781450345897

[134]: Yossi RUBNER, Carlo TOMASI et Leonidas J. GUIBAS. « The Earth Mover's Distance as a Metric for Image Retrieval ». In : *International Journal of Computer Vision* 40.2 (nov. 2000)

[135]: Xiaohong SHENG, Yu-Hen HU et Parameswaran RAMANATHAN. « Distributed Particle Filter with GMM Approximation for Multiple Targets Localization and Tracking in Wireless Sensor Network ». In : *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*. IPSN '05. Los Angeles, California : IEEE Press, 2005. ISBN : 0780392027

[136]: Dongbing GU. « Distributed Particle Filter for Target Tracking ». In : *Proceedings 2007 IEEE International Conference on Robotics and Automation*. 2007

[137]: Boris N. ORESHKIN et Mark J. COATES. « Asynchronous distributed particle filter via decentralized evaluation of Gaussian products ». In : *2010 13th International Conference on Information Fusion*. 2010

[138]: Ondrej HLINKA, Ondrej SLUČIAK, Franz HLAWATSCH, Petar M. DJURIĆ et Markus RUPP. « Likelihood Consensus and Its Application to Distributed Particle Filtering ». In : *IEEE Transactions on Signal Processing* 60.8 (2012)

[139]: Deniz ÜSTEBAY, Mark COATES et Michael RABBAT. « Distributed auxiliary particle filters using selective gossip ». In : *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2011

[140]: Claudio J. BORDIN et Marcelo G. S. BRUNO. « Consensus-based distributed particle filtering algorithms for cooperative blind equalization in receiver networks ». In : *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2011

[141]: Dongbing GU, Junxi SUN, Zhen HU et Hongzuo LI. « Consensus based distributed particle filter in sensor networks ». In : *2008 International Conference on Information and Automation*. 2008

[142]: Arash MOHAMMADI et Amir ASIF. « Consensus-based distributed unscented particle filter ». In : *2011 IEEE Statistical Signal Processing Workshop (SSP)*. 2011

[143]: Saptarshi BANDYOPADHYAY et Soon-Jo CHUNG. « Distributed estimation using Bayesian consensus filtering ». In : *2014 American Control Conference* (2014)

avec des probabilités plus élevées en utilisant un solveur EDM (*Earth Mover's Distance*) [134]. Cela est fait pour relever les défis posés par les données d'observation de longue durée et la grande dimensionnalité à la fois des vecteurs d'état et d'observation. Ensuite, la vraisemblance (*likelihood*) est calculée en fonction d'un modèle d'observation, qui repose également sur les données d'observation. Enfin, ils calculent les poids et rééchantillonnent ensuite les particules pour obtenir la densité prévue dans chaque région en agrégeant le vecteur de position a_t de toutes les particules. Il convient de noter que ce processus préserve l'anonymat des individus, car les données observées sont la distribution de la population et les particules ne sont associées à aucun individu.

Filtre à particules distribué

Un filtre à particules peut être distribué sur plusieurs nœuds de calcul au lieu de dépendre d'un serveur central. Il s'agit d'une technique bien établie pour le suivi de cibles au sein de réseaux de capteurs [135-137]. Cette distribution peut avoir lieu à différentes étapes du processus du filtre à particules, en particulier dans une configuration de filtre à particules décentralisé, où chaque nœud calcule indépendamment son propre filtre. Il existe deux catégories principales de méthodes de distribution. La première catégorie implique que les nœuds parviennent à un consensus global sur les calculs statistiques. Grâce au partage itératif et à la fusion d'informations locales entre les nœuds, les agents au sein du réseau distribué parviennent à un consensus sur les informations globales, que ce soit les vraisemblances [138], les poids [139, 140], ou les paramètres postérieurs [141-143], en utilisant des techniques telles que la moyenne, les sommes pondérées ou la fusion bayésienne. Cela se traduit par des statistiques synchronisées à travers le réseau distribué, et les agents obtiennent des filtres à particules locaux identiques. Il convient de noter que parvenir à un consensus sur les paramètres postérieurs, c'est-à-dire les estimations des états cachés, ne nécessite pas de générateurs de nombres aléatoires locaux synchronisés entre les agents. La deuxième catégorie principale implique que les agents partagent directement leurs mesures et les données reçues avec d'autres agents jusqu'à ce que l'information soit *diffusée* dans tout le réseau [144-146]. Cette approche permet à chaque agent de calculer son propre filtre à particules local. Cependant, elle nécessite des générateurs de nombres aléatoires synchronisés pour générer les mêmes particules et obtenir des résultats cohérents. Dans les cas où les ressources de communication sont limitées, comme dans les réseaux sans fil, le partage de mesures peut être gourmand en ressources et surcharger le réseau pendant le processus de diffusion, ce qui peut nécessiter de trouver des moyens de par exemple réduire la taille de l'information en la compressant ou limiter les communications inutiles. Dans le cas d'un réseau MEC impliquant des nœuds mobiles, les capacités de calcul et de communication peuvent être légèrement plus importantes que dans un réseau de capteurs. Cependant, la communication et le calcul demeurent des ressources précieuses et limitées, qui doivent être principalement dédiées à garantir l'exécution efficace des missions de service MEC des nœuds.

4.4 Contribution

Ici, nous proposons la distribution du filtre à particules pour la prédiction de densité, comme décrit dans [117], à travers plusieurs nœuds MEC. Ces nœuds MEC comprennent des dispositifs mobiles tels que des UAVs ou des robots mobiles, chacun équipé de capacités de calcul et communiquant via un réseau ad hoc Wi-Fi. Le filtre à particules est réparti sur l'ensemble de ces nœuds. Dans cette Section, nous allons expliquer les composants de cet algorithme distribué. Nous insistons sur le fait que notre objectif principal n'est pas de répéter l'efficacité du filtre à particules lui-même, mais plutôt de démontrer sa faisabilité dans le contexte d'un réseau distribué. Tous les codes utilisés, ainsi que les données, sont disponibles en accès ouvert [147].

4.4.1 Simulateur de Mobilité Humaine

En tant que simulateur de mobilité humaine, nous utilisons un simulateur basé sur Python. Ce simulateur modélise les déplacements des individus entre les points d'intérêt, les stations et leurs résidences. Les personnes se déplacent à des vitesses correspondant à leur destination et partent d'un point à un autre en suivant des trajectoires euclidiennes. Un simulateur de mobilité humaine de base, tel que celui décrit ici, suffit à nos expérimentations concernant la distribution. De plus, nous prenons en compte des régions plus petites où les piétons peuvent ne pas strictement suivre les routes établies.

4.4.2 Collecte de données d'observation

Le filtre à particules nécessite les mesures avant tout calcul statistique pour permettre au solveur EMD de déplacer les particules près des données observées, nous devons donc effectuer la distribution à travers la diffusion des mesures. Comme les mesures concernent la densité de personnes à travers les régions, nous devons compter ces personnes dans chaque région. Nous disposons de nœuds sur site, ce qui nous permet de compter les personnes directement plutôt que de nous fier aux données GPS agrégées des opérateurs mobiles collectées à de longs intervalles. Pour effectuer le comptage, les nœuds peuvent utiliser des méthodes non visuelles, telles que LiDAR [148] et Kinect, le comptage basé sur la vision [149-151], les techniques basées sur la radio [152], ou même une combinaison de capteurs. Ils peuvent également exploiter d'autres nœuds ou des capteurs sans puissance de calcul, tels que les caméras des villes. En ce qui concerne le partage d'informations, les robots disséminent leurs propres comptages de densité dans leurs régions respectives, avec les identifiants de région et la couverture du comptage dans la région. Les identifiants de région sont prépartagés avec la carte de la région cible où ils sont déployés. Ensuite, chaque robot peut reconstituer le comptage global de la densité sur l'ensemble de la carte une fois qu'ils ont obtenu toutes les informations nécessaires grâce à la diffusion des mesures.

[144]: Matthew ROSENCRANTZ, Geoffrey J. GORDON et Sebastian THRUN. « Decentralized Sensor Fusion with Distributed Particle Filters ». In : *Conference on Uncertainty in Artificial Intelligence*. 2002

[145]: Petar M. DJURIĆ, Jonathan BEAUDEAU et Mónica F. BUGALLO. « Non-centralized target tracking with mobile agents ». In : *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2011

[146]: Chang LIU, Shengbo Eben LI et J. Karl HEDRICK. « Measurement Dissemination-Based Distributed Bayesian Filter Using the Latest-In-and-Full-Out Exchange Protocol for Networked Unmanned Vehicles ». In : *IEEE Transactions on Industrial Electronics* 64.11 (2017)

[147]: *Dépot prédiction densité*. <https://gitlab.inria.fr/nsanti/mobility-prediction>

[148]: Sylvia T. KOUYOUMDJIEVA, Peter DANIELIS et Gunnar KARLSSON. « Survey of Non-Image-Based Approaches for Counting People ». In : *IEEE Communications Surveys & Tutorials* 22.2 (2020)

[149]: H. Hakan CETINKAYA et Muammer AKCAY. « People Counting at Campuses ». In : *Procedia - Social and Behavioral Sciences* 182 (2015). 4th WORLD CONFERENCE on EDUCATIONAL TECHNOLOGY RESEARCHES (WCETR-2014)

[150]: Adal ALASHBAN, Alhanouf ALSADAN, Norah F. ALHUSSAINAN et Ridha OUNI. « Single Convolutional Neural Network With Three Layers Model for Crowd Density Estimation ». In : *IEEE Access* 10 (2022)

[151]: Aravinda S. RAO, Jayavardhana GUBBI, Slaven MARUSIC, Paul STANLEY et Marimuthu PALANISWAMI. « Crowd density estimation based on optical flow and hierarchical clustering ». In : *2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. 2013

[152]: Kyosuke SHIBATA et Hiroshi YAMAMOTO. « People Crowd Density Estimation System using Deep Learning for Radio Wave Sensing of Cellular Communication ». In : *2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIIC)*. 2019

4.4.3 Diffusion des données

Une fois que les nœuds ont terminé la tâche de comptage des personnes dans leurs régions respectives, l'étape suivante consiste à diffuser ces données à travers le réseau. Pour ce faire, nous utilisons un protocole de diffusion d'informations basé sur le principe du protocole de propagation de rumeurs (*gossip protocol*). Les protocoles de propagation de rumeurs, que l'on appellera par la suite protocole de commérage, offrent plusieurs avantages, notamment la résilience aux défaillances de nœuds ou de liaisons, la robustesse dans les topologies de réseau dynamiques typiques des réseaux ad hoc avec des nœuds mobiles, et la scalabilité [153]. Le principe sous-jacent de ce protocole est simple et s'inspire à la fois de la diffusion de rumeurs et de la contamination épidémique. Dans cette approche, chaque nœud partage ses informations avec les nœuds voisins, qui partagent ensuite ces informations avec leurs propres voisins, et ainsi de suite, jusqu'à ce que l'information ait été diffusée à chaque nœud du réseau.

[153]: Ondrej HLINKA, FRANZ HLAWSCH et Petar M. DJURIC. « Distributed particle filtering in agent networks : A survey, classification, and comparison ». In : *IEEE Signal Processing Magazine* 30.1 (2013)

Il existe plusieurs configurations possibles pour un protocole de commérage, et dans notre cas, nous mettons en œuvre le protocole de diffusion Susceptible, Infecté, Retiré (SIR). Dans ce protocole, les nœuds peuvent se trouver dans l'un des trois états suivants : susceptible, infecté ou retiré. Les nœuds dans l'état susceptible sont ouverts à la réception de nouvelles informations. Ils sont prêts à accepter et à diffuser des données fraîches. Les nœuds dans l'état infecté possèdent des informations à diffuser à d'autres. Cet état indique qu'ils ont des données à partager. Les nœuds dans l'état retiré sont temporairement exclus du protocole car ils n'ont plus de nouvelles informations à distribuer. De plus, étant donné que nous avons plusieurs informations à diffuser, chaque information reçue par un nœud peut également présenter son propre état infecté (pour les données prêtes à être partagées) et un état retiré (pour les données devenues obsolètes). Dans le protocole de diffusion SIR, chaque fois qu'un nœud reçoit une information qu'il a déjà reçue auparavant, il existe une possibilité qu'il marque cette information comme retirée en fonction d'une probabilité prédéfinie appelée **obsolescence**. Si toutes les informations reçues par un robot sont dans l'état retiré, il existe également une possibilité que le nœud lui-même soit marqué comme retiré. Cependant, si un nœud retiré reçoit de nouvelles informations, il passe à l'état infecté, prêt à diffuser les données fraîches.

La diffusion des informations entre les nœuds se fait à travers trois modes principaux. Dans le mode *push*, un nœud envoie périodiquement un ensemble aléatoire d'informations à ses voisins jusqu'à ce qu'il ne reste plus de nouvelles informations à distribuer, ce qui signifie que toutes les informations sont passées à l'état retiré. Dans le mode *pull*, les nœuds demandent activement de nouvelles informations à leurs voisins jusqu'à ce qu'ils soient retirés du protocole eux-mêmes, ce qui se produit lorsqu'ils ne reçoivent plus de nouvelles informations. Dans le mode hybride, une combinaison des modes *push* et *pull* est utilisée. Cela signifie que les nœuds peuvent à la fois envoyer des informations à leurs voisins et demander des informations à ces derniers. Chaque mode a ses avantages et ses inconvénients uniques [154]. Le protocole détaillé est présenté dans l'Algorithme 1.

[154]: Everaldo LEME, Naghmeh IVAKI, Nuno LARANJEIRO et Regina MORAES. « Analyzing Gossip Protocols for Reliable MANET Applications ». In : *2017 IEEE International Conference on Edge Computing (EDGE)*. 2017

Algorithme 1 : Algorithme SIR

```

si push est Vrai alors
  planifierMiseAJour(intervalleMiseAJour)
fin si
si pull est Vrai alors
  planifierRequête(intervalleRequête)
fin si

```

```

Fonction requête() :
voisins = sélectionnerVoisinsAléatoires(maxVoisinsAléatoires)
pour chaque n parmi les voisins faire
  envoyerRequête(infos, n)
fin pour

```

```

Fonction miseÀJour() :
voisins = sélectionnerVoisinsAléatoires(maxVoisinsAléatoires)
ragots = sélectionnerRagotsInfectésAléatoires(maxRagotsAléatoires)
pour chaque n parmi les voisins faire
  envoyer(ragots, n)
fin pour

```

```

Fonction gérerRequête() : # demande d'information reçue
si pull alors
  envoyerRagots(dest)
fin si

```

```

Fonction gérerReçu(ragot) : # nouvelles informations reçues
si le ragot n'est pas dans la listeRagots alors
  ragot.état = infecté
  listeRagots.ajouter(ragot)
sinon
  # ragot déjà connu et a une chance d'être supprimé
  si aléatoire < obsolètescence alors
    ragot.état = supprimé {ce ragot ne sera plus envoyé}
  fin si
fin si

```

```

Fonction mettreÀJourÉtat() :
si nœud.état == supprimé alors
  # nouvelles informations
  nœud.état = infecté
sinon si nœud.état == infecté et infosInfectées.vide() alors
  # pas de nouvelles informations à diffuser
  si aléatoire < obsolètescence alors
    nœud.état = supprimé
  fin si
fin si

```

4.5 Configuration des expériences

Pour les expériences nous proposons d'évaluer deux parties de la distribution du filtre à particules. La première concerne la partie calcul, c'est-à-dire si le filtre à particules peut obtenir de bons résultats lorsqu'il est distribué sur plusieurs nœuds de calcul et s'il performe bien dans un scénario où certains nœuds sont isolés puis retrouvent les autres nœuds de calcul. L'autre partie concerne la communication, où l'agent mobile doit diffuser les mesures de la densité.

4.5.1 Expérience sur la communication

Pour la communication et évaluer le protocole de comméragage, nous utilisons ns-3, un simulateur de réseau open-source populaire qui propose des capacités de simulation réseau étendues, permettant la simulation de divers scénarios complexes. Dans cette section, nous fournissons une description des paramètres utilisés, des scénarios de simulation avec une explication des métriques utilisées pour l'évaluation. Pour les paramètres de mise en réseau, nous considérons un réseau Wi-Fi ad-hoc avec des agents mobiles. Nous définissons la couche physique du réseau avec un gain de -10 dB pour la sensibilité, le type de liaison de données pour la capture de paquets est défini par le standard IEEE 802.11. Un canal Wi-Fi est créé avec un modèle de retard de propagation à vitesse constante. Pour la perte de propagation en fonction de la distance nous utilisons le modèle de Friis [155] qui prend en compte la fréquence du signal ainsi que d'autres paramètres. Nous y rajoutons le modèle de perte de propagation Nakagami qui prend en compte les effets de l'affaiblissement du signal en raison de la distance et de l'atténuation provoquée par la présence d'obstacles dans le chemin de transmission. C'est un modèle réaliste pour simuler des environnement urbain [156]. Pour la couche de liaison de données, elle est définie par la norme 802.11n, et le type MAC Wi-Fi est spécifié en mode "AdhocWifiMac". Pour enregistrer et analyser les résultats de la simulation et le comportement du réseau, nous activons le traceur ASCII pour tracer tous les événements et les paramètres, ainsi que PCAP pour capturer le trafic réseau. Pour le paramétrage de la mobilité dans la simulation, nous configurons des agents mobiles avec des positions X et Y aléatoires à l'intérieur d'une zone délimitée. Les nœuds mobiles se déplacent de manière aléatoire à l'intérieur de la zone rectangulaire définie avec une vitesse de 15 mètres par seconde et une pause de 5 secondes.

[155]: H.T. FRIIS. « A Note on a Simple Transmission Formula ». In : *Proceedings of the IRE* 34.5 (1946)

[156]: Hanene BRAHMIA et Cherif TOLBA. « Nakagami fading impact on the performances of VANET routing protocols in a Realistic Urban Area setting ». In : *International Journal of Advanced Networking and Applications* 11.4 (2020)

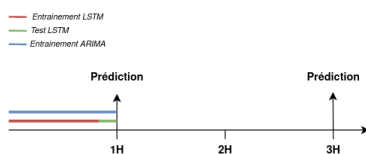


FIGURE 4.3 – Expérimentation pour les performances de prédiction.

[157]: O. MICHEL. « Webots : Professional Mobile Robot Simulation ». In : *Journal of Advanced Robotics Systems* 1.1 (2004)

4.5.2 Expérience sur la partie calcul du filtre

Simulation et scénario d'expérimentation

Pour la composante calcul du filtre à particules, nous testons le code dans Webots, un simulateur de robot open source [157]. Nous configurons trois robots pour utiliser le filtre à particules comme leurs contrôleurs. Nous générons 3 heures de données synthétiques et effectuons des tests avec les robots. Nous évaluons différents intervalles d'observation et leur impact sur les résultats. Nous commençons par tester des intervalles

d'observation de 5, 10, 15, 20 et 30 minutes, puis comparons les résultats entre eux. De plus, nous les comparons après 1 heure et 3 heures en utilisant deux méthodes de référence (que nous avons entraînées avec des données de 1 heure). Cette expérience est illustrée dans la Figure 4.3. Nous synchronisons le générateur de nombres aléatoires avec une graine prédéfinie.

Données

Nous générons des données synthétiques à l'aide d'un simulateur de mobilité humaine. Initialement, nous avons tenté d'utiliser des données de mobilité en libre accès, telles que les données Geolife [158], mais elles ne fournissaient pas les informations de vérité terrain nécessaires pour évaluer la distribution dont nous avons besoin. De plus, elles couvraient une zone géographique plus large que notre cas d'utilisation spécifique, qui implique le déploiement d'unités mobiles dans une région limitée. Par conséquent, nous avons choisi d'utiliser le générateur de mobilité humaine pour générer des données de mobilité sur une 3 heure de temps et avons défini la taille de l'ensemble, c'est-à-dire le nombre d'individus, à 500 sur une carte de 500 mètres par 500 mètres. Les paramètres comportementaux sont définis de manière aléatoire selon une distribution normale avec une moyenne de 0,4 et une variance de 0,2.

[158]: Yu ZHENG, Hao FU, Xing XIE, Wei-Ying MA et Quannan LI. *Geolife GPS trajectory dataset - User Guide*. Juill. 2011

Méthodes de références

Pour les deux méthodes de référence, nous commençons par utiliser ARIMA, qui signifie Auto-Regressive Integrated Moving Average. ARIMA est une technique de régression qui capture les structures temporelles dans les données de séries chronologiques pour des prévisions à court terme [159]. ARIMA nécessite trois paramètres pour définir son comportement : p (ordre de retard), qui représente le nombre d'observations passées incluses dans le modèle ; d (ordre intégré), indiquant le nombre de différences nécessaires pour rendre la série chronologique stationnaire ; et q (ordre de la moyenne mobile), spécifiant la taille de la fenêtre de moyenne mobile. Bien que nous puissions également inclure des ordres saisonniers pour capturer des motifs saisonniers, notre jeu de données est trop limité pour cela. Nous automatisons le calcul des paramètres pour obtenir un meilleur ajustement aux données. En tant que deuxième méthode de référence, nous utilisons un LSTM (*Long Short-Term Memory*). Un LSTM est un réseau de neurones artificiel qui utilise des portes d'information pour soit stocker des informations importantes d'une séquence de données, soit les oublier, ce qui lui permet de détecter les dépendances à long terme dans les données, un aspect crucial pour la gestion de séries chronologiques longues. Nous utilisons une couche LSTM avec 50 unités et une activation ReLU, suivie d'une couche de sortie dense. Le modèle est compilé en utilisant la fonction de perte de l'erreur quadratique moyenne (MSE) et l'optimiseur ADAM. Il subit 50 époques d'entraînement avec une taille de lot de 16, produisant une sortie détaillée. Les données sont organisées sous la forme d'une série temporelle de 15 'pas', comme nous l'avons déjà vu dans la Figure 3.7. Nous utilisons 50 minutes de données pour l'entraînement du modèle, puis

[159]: George EP BOX, Gwilym M JENKINS, Gregory C REINSEL et Greta M LJUNG. *Time series analysis : forecasting and control*. John Wiley & Sons, 2015

[160]: Bing LIU. « Lifelong Machine Learning : A Paradigm for Continuous Learning ». In : *Frontiers of Computer Science* 11.3 (juin 2017)

[161]: Quang PHAM, Chenghao LIU, Doyen SAHOO et Steven C. H. HOI. *Learning Fast and Slow for Online Time Series Forecasting*. 2022. arXiv : 2202.11672 [cs.LG]

TABLE 4.1 – Paramètres du filtre à particules.

Paramètre	Valeur
K'	0.9
γ	0.7
κ	1
λ	0.1
$\hat{\epsilon}a$	$0.1I^{N_{2p}}$
$\hat{\epsilon}\theta$	$0.01I^{N_{\theta}}$ height

les 10 minutes de données suivantes pour les tests. Nous n'appliquons pas d'expériences dans lesquelles nous alimentons le LSTM avec de nouvelles données au fur et à mesure de leur arrivée, et ce pour plusieurs raisons. Tout d'abord, le modèle peut rapidement perdre la capacité de généralisation acquise lors de la première phase d'entraînement. Cette approche pourrait être envisagée en cas de changements brusques et fondamentaux dans les données, mais sinon, le modèle risquerait de perdre les connaissances précieuses acquises sur les motifs saisonniers de la série temporelle [160]. On pourrait argumenter que l'arrivée d'un événement d'urgence constitue un changement brusque, mais dans notre environnement automatisé, il est difficile de savoir quand cet événement se produit et s'il est judicieux d'incorporer les nouvelles données au modèle. Cela pourrait même détériorer considérablement les prédictions. De plus, nous ne sommes pas certains que la situation soit permanente, et elle pourrait évoluer de nouveau vers une "normalité". L'adaptation des LSTM dans des environnements non stationnaires constitue une branche de la recherche à part entière qui n'entre pas dans le cadre de ces travaux [161]. Une solution potentielle consisterait à ré-entraîner entièrement le modèle avec les anciennes données en y ajoutant les nouvelles. Cependant, cette approche s'avère extrêmement coûteuse en termes de calculs.

Paramètres

S'agissant des paramètres du modèle d'observation, nous réglons κ à 1 car nous ne dépendons pas d'un opérateur mobile. En ce qui concerne γ , étant donné que nous ne disposons pas d'un ratio d'utilisation des données par les utilisateurs, nous choisissons d'utiliser le ratio de précision lors du comptage des personnes dans une zone avec la méthode choisie, que nous fixons à 0,7. De plus, nous maintenons la même valeur pour λ , qui représente le ratio d'observations incorrectes. Nous conservons également les mêmes valeurs pour $\hat{\epsilon}a$ et $\hat{\epsilon}\theta$. Ces paramètres sont résumés dans la Table 4.1. Pour le modèle de transition de destination, nous considérons les mêmes quatre catégories qu'auparavant, la seule différence étant que nous remplaçons les magasins placés tout les 500 mètres par des points d'intérêt (que l'on retrouve dans la littérature sous le nom de PoI - *Point of Interest*), ce qui donne les catégories suivantes : domicile, stations de bus, points d'intérêt et position d'attente. Compte tenu des données synthétiques, nous répartissons uniformément les stations de bus, les points d'intérêt et les domiciles sur une carte de 1000 mètres sur 1000 mètres, peuplée de 1000 individus, 50 points d'intérêt et 200 stations. La vitesse des piétons est fixée à 1,42 mètre par seconde. La taille de l'ensemble est fixée à 1500. Nous réutilisons l'ensemble initial de probabilités de transition pour chaque particule utilisée dans [117], à l'exception du remplacement des magasins par les points d'intérêt.

Métriques d'évaluation

Pour les métriques d'évaluation, nous réutilisons les métriques présentées à la Section 3.2, à savoir la MAE (Erreur Absolue Moyenne) et la RMSE (Racine Carrée de l'Erreur Quadratique Moyenne). Étant donné que notre contexte concerne la prédiction de la densité de la population, la MSE

(Erreur Quadratique Moyenne) a très peu de pertinence, car elle implique une mise au carré des erreurs. Nous préférons donc nous concentrer sur des métriques qui peuvent être plus facilement interprétées.

4.6 Résultats

4.6.1 Protocole de comméragé

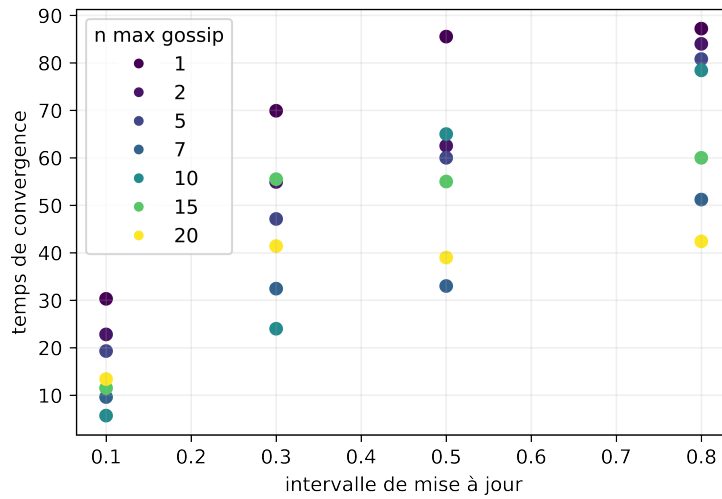
Dans cette section, nous examinons donc la faisabilité du protocole de communication pour la diffusion d'informations entre les nœuds mobiles MEC au sein d'un réseau ad-hoc Wi-Fi, dans le contexte de la mise en place d'un filtre à particules distribué pour prédire la densité humaine en milieu urbain en extérieur.

Étude sur le nombre de gossips échangés

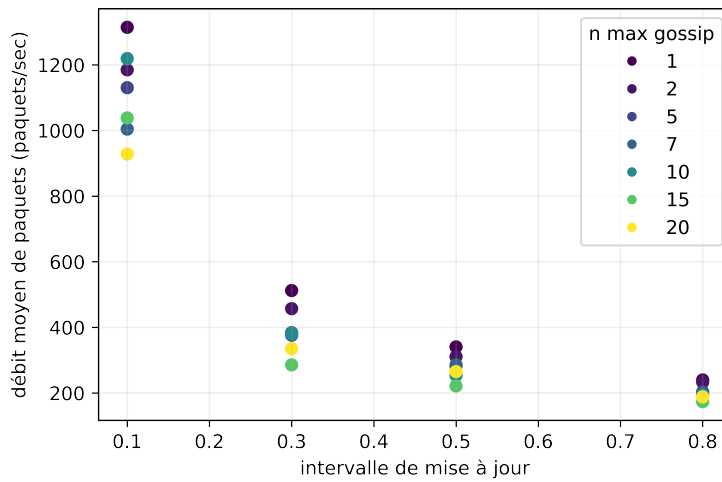
TABLE 4.2 – Performances par le nombre de gossip envoyés et interval de mise à jour pour le mode *push*.

Intervalle de mise à jour	$n_{\max \text{ gossip}}$	Temps de convergence	Débit moyen de paquets (paquets/s)	Débit moyen en kilobits (Kbits/s)	Débit en pourcentage
0.1	1	30.300431	1314.50	615.54315	0.41%
0.1	2	22.804892	1184.92	576.66464	0.38%
0.1	5	19.303035	1130.50	568.56155	0.38%
0.1	7	9.616020	1004.23	544.76215	0.36%
0.1	10	5.704958	1219.18	679.66294	0.45%
0.1	15	11.503102	1038.16	551.06229	0.37%
0.1	20	13.403710	928.19	501.09964	0.33%
0.3	1	69.907550	512.07	241.61460	0.16%
0.3	2	54.913891	456.88	223.20252	0.15%
0.3	5	47.124702	376.03	193.11089	0.13%
0.3	7	32.442154	384.03	202.51337	0.14%
0.3	10	24.002979	379.79	206.20004	0.14%
0.3	15	55.500513	285.42	154.02163	0.10%
0.3	20	41.408925	334.95	178.98803	0.12%
0.5	1	85.529055	340.31	161.81555	0.11%
0.5	2	62.522933	310.49	153.09077	0.10%
0.5	5	60.001526	282.26	144.18893	0.10%
0.5	7	33.000642	285.63	153.51883	0.10%
0.5	10	65.000923	253.63	132.45278	0.09%
0.5	15	55.004493	221.33	119.24135	0.08%
0.5	20	39.000356	265.02	140.85780	0.09%
0.8	1	87.205985	239.44	114.52037	0.08%
0.8	2	84.000480	232.99	114.90532	0.08%
0.8	5	80.802651	203.11	104.31407	0.07%
0.8	7	51.205526	197.57	105.48411	0.07%
0.8	10	78.421846	193.61	100.87952	0.07%
0.8	15	60.000486	173.35	93.84520	0.06%
0.8	20	42.401117	188.05	103.28319	0.07%

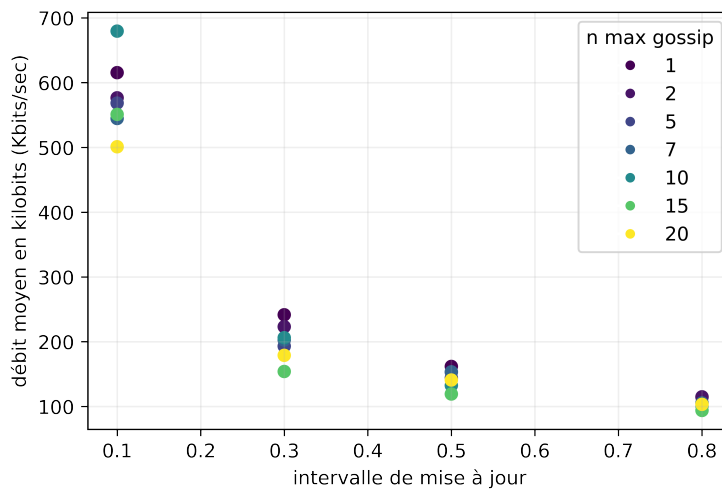
Les Figures 4.4 et 4.5 présentent les résultats des expériences pour le mode *push* et *pull*, respectivement, concernant le nombre de *gossips* envoyés par intervalle de mise à jour (ou de requête), c'est-à-dire la fréquence à laquelle les nœuds envoient des *gossips* à leurs nœuds voisins (ou une demande de *gossips*). Les Figures 4.4a et 4.5a affichent les résultats en termes de temps de convergence, tandis que les Figures 4.4b et 4.5b présentent le taux moyen de paquets échangés par seconde, et enfin, les Figures 4.4c et 4.5c montrent le débit binaire moyen en kilobits par seconde. Le nombre de nœuds est fixé à 30 sur une zone de 200 mètres par 200 mètres. Les nœuds envoient leurs requêtes ou mises à jour à un seul voisin sélectionné aléatoirement pour cette étude et aussi les suivantes. Les Tables 4.2 et 4.3 résument les résultats obtenus pour les modes *push* et *pull*, respectivement, en mettant en évidence les meilleurs



(a) Temps de convergence par nombre de gossips envoyés et d’intervalles de mise à jour.

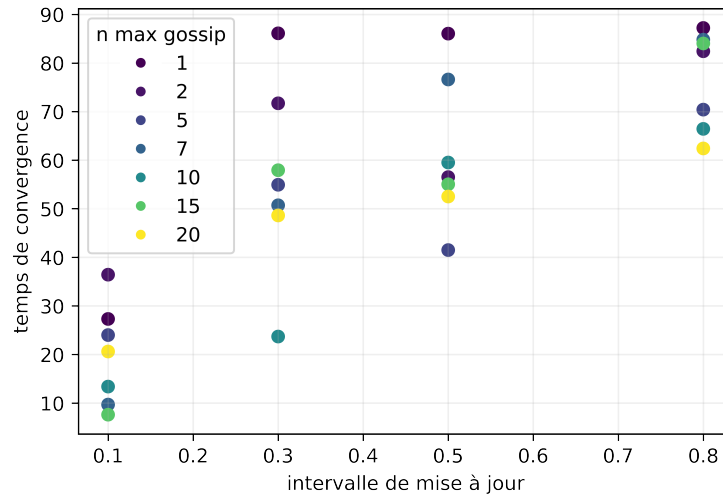


(b) Nombre moyen de paquets envoyés toutes les secondes par nombre de gossips envoyés et d’intervalles de mise à jour.

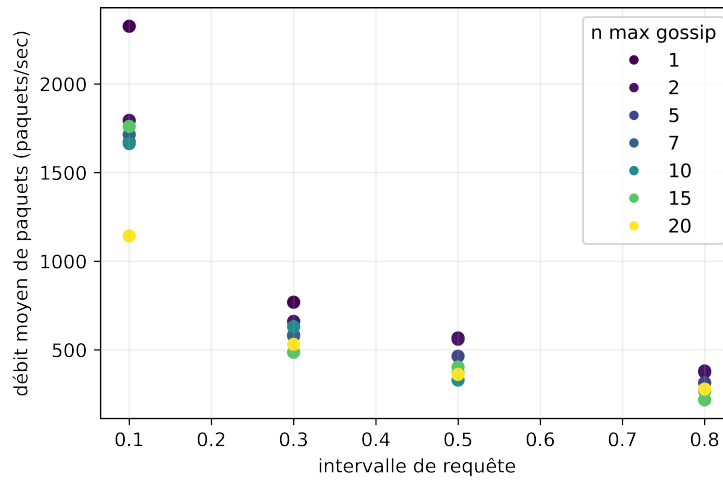


(c) Débit binaire moyen en kilobits toutes les secondes par nombre de gossips envoyés et d’intervalles de mise à jour.

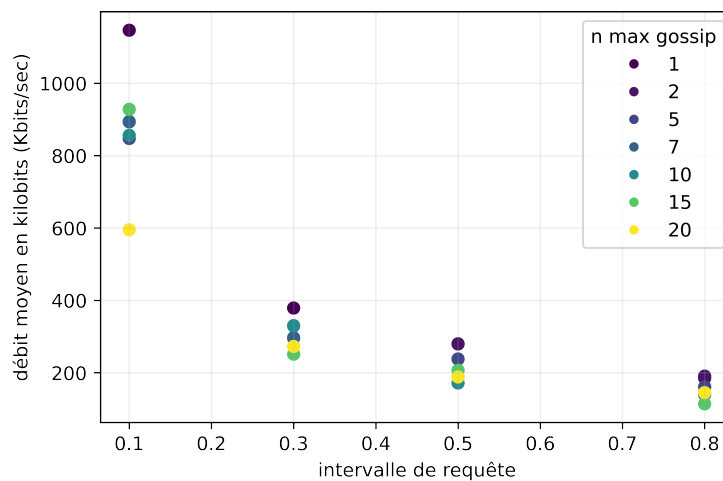
FIGURE 4.4 – Comparaison du nombre de gossips envoyés par intervalles de mise à jour pour le mode *push*.



(a) Temps de convergence par nombre de gossips envoyés et d'intervalles de requête.



(b) Nombre moyen de paquets envoyés toutes les secondes par nombre de gossips envoyés et d'intervalles de requête.



(c) Débit binaire moyen en kilobits toutes les secondes par nombre de gossips envoyés et d'intervalles de requête.

FIGURE 4.5 – Comparaison du nombre de gossips envoyés par intervalles de requêtes pour le mode *pull*.

TABLE 4.3 – Performances par le nombre de gossip envoyés et interval de requête pour le mode *pull*.

Intervalle de requête	n_{\max} gossip	Temps de convergence	Débit moyen de paquets (paquets/s)	Débit moyen en kilobits (Kbits/s)	Débit en pourcentage
0.1	1	27.323061	2325.64	1146.51588	0.76%
0.1	2	36.434818	1794.45	893.57112	0.60%
0.1	5	24.009254	1672.28	847.51961	0.57%
0.1	7	9.702794	1714.26	894.18336	0.60%
0.1	10	13.403443	1663.31	856.08688	0.57%
0.1	15	7.623158	1760.22	928.01060	0.62%
0.1	20	20.624673	1142.76	595.19797	0.40%
0.3	1	86.106415	768.20	378.97029	0.25%
0.3	2	71.703989	660.11	330.10664	0.22%
0.3	5	54.936913	582.81	296.95012	0.20%
0.3	7	50.717854	577.73	295.72447	0.20%
0.3	10	23.702956	630.31	329.46831	0.22%
0.3	15	57.933923	485.85	251.50051	0.17%
0.3	20	48.624551	530.36	272.93576	0.18%
0.5	1	86.042825	566.29	279.71844	0.19%
0.5	2	56.514734	559.63	280.16256	0.19%
0.5	5	41.499818	463.48	238.15885	0.16%
0.5	7	76.607795	367.91	189.41983	0.13%
0.5	10	59.515987	328.80	171.82168	0.11%
0.5	15	55.041331	402.35	206.88460	0.14%
0.5	20	52.521830	360.90	188.48784	0.13%
0.8	1	87.209023	374.68	185.75900	0.12%
0.8	2	82.416905	380.07	190.85571	0.13%
0.8	5	70.401899	313.86	161.29612	0.11%
0.8	7	84.804751	274.13	141.19521	0.09%
0.8	10	66.439687	269.28	139.98871	0.09%
0.8	15	84.021606	217.38	114.23875	0.08%
0.8	20	62.413904	277.92	145.15691	0.10%

résultats obtenus par intervalle de mise à jour (ou de requête) en vert, tandis que les pires résultats sont marqués en rouge.

Mode *push* Tout d’abord, il est observable sur la Figure 4.4a que, en général, à mesure que l’intervalle de mise à jour augmente, le temps de convergence augmente également. C’est cohérent avec une réduction du nombre d’échanges d’informations. En ce qui concerne le nombre de messages *gossips* échangés à chaque intervalle, on pourrait logiquement s’attendre à ce que, en général, un plus grand nombre de ces messages entraîne un temps de convergence plus court et donc optimal, et inversement. Cependant, il est intéressant de noter que ce n’est pas strictement le cas pour les intervalles de mise à jour courts de 0.1 et 0.3 secondes. Par exemple, avec un intervalle de mise à jour de 0.3, le temps de convergence optimal est obtenu avec un nombre de *gossips* échangés de 10, plutôt que 20 ou 15. Cela est dû au fait que les nœuds se retrouvent plus rapidement surchargés d’informations, ce qui signifie qu’ils ont plus de données à traiter en un court laps de temps. Pour des intervalles plus longs de 0.5 et 0.8, on observe une relation plus linéaire, car le réseau est moins encombré, bien que des exceptions subsistent, comme un nombre de *gossip* de 7 qui s’avère être le plus performant pour l’intervalle de mise à jour de 0.5 secondes.

En ce qui concerne le débit moyen de paquets et binaire par seconde, on constate clairement sur les Figures 4.4c et 4.4b qu’il existe une relation directe entre l’augmentation de l’intervalle de mise à jour et la réduction du débit pour les deux métriques, ce qui semble logique. Cependant, il est important de noter une forte diminution entre les intervalles de 0.1 et de 0.3 secondes.

La Table 4.2 révèle clairement qu’il existe un compromis entre le débit et le temps de convergence. Par exemple, avec un maximum de 10 *gossips* et un intervalle de mise à jour de 0.1 secondes, le temps de convergence est d’environ 6 secondes, mais le débit moyen en kilobits par seconde

est d'environ 680. En revanche, pour un intervalle de 0.3 secondes avec le même maximum de 10 *gossips*, le temps de convergence est de 24 secondes, mais le débit binaire est réduit de près de 70%.

Il est donc essentiel de prendre en considération ces aspects lors de la conception du protocole de communication dans notre cas d'utilisation. Les nœuds dépendent des données d'observation pour prédire les mouvements des foules, mais ils doivent également optimiser l'utilisation de leurs canaux de communication pour mener à bien leur mission.

Mode *pull* En ce qui concerne le temps de convergence, nous pouvons tirer des conclusions similaires à celles de l'analyse précédente pour le mode *push* à partir de la Figure 4.5a. Cependant, il est à noter que, en comparaison, ce mode prend généralement plus de temps pour converger. Cela est dû au fait que les nœuds doivent d'abord envoyer une requête avant de recevoir une réponse, ce qui augmente le temps d'échange.

Quant au débit moyen de paquets et binaire, représenté dans les Figures 4.5b et 4.5c respectivement, nous pouvons également tirer des conclusions similaires à celles du mode précédent. En comparant les deux modes, le mode *pull* obtient des débits plus élevés sur tous les intervalles et avec tous les nombres maximum de *gossips* échangés, comme le confirme la Table 4.3. Cette augmentation du débit s'explique, comme précédemment mentionné, par un nombre plus élevé de messages échangés dans ce mode.

Étude sur le mode

TABLE 4.4 – Données de performance par mode.

Mode	Temps de convergence	Débit moyen de paquets (paquets/sec)	Débit moyen en kilobits (Kbits/sec)	Débit en pourcentage
<i>Pull</i>	42.623478	1892.79	961.38616	0.64%
<i>Push</i>	41.720388	1310.37	624.74084	0.42%
Hybride	14.747196	1627.12	809.24912	0.54%
<i>Flooding</i>	9.397160	3517.19	1787.70540	1.19%

Dans cette section, nous allons comparer les trois modes différents de l'algorithme de commérage, à savoir le mode *pull*, le mode *push* et le mode hybride qui combine les deux. De plus, nous avons ajouté un protocole appelé *flooding*. Dans ce protocole de *flooding* tout les *gossips* connus sont échangés avec tout les voisins connus. Le nombre de noeuds est fixé à 30 sur une zone de 200 mètres par 200 mètres. Les intervalles de mise à jour et de requête ont tous été fixés à 0,3 secondes (même pour le *flooding*), et le nombre maximum de *gossips* échangés est de 10, car il s'agit du paramétrage qui semble offrir le meilleur compromis, comme montré dans les résultats précédents. Les résultats sont présentés dans le Tableau 4.4. La colonne "Débit en pourcentage" représente le pourcentage du débit binaire par seconde produit par le protocole par rapport à un débit binaire théorique de la norme IEEE 802.11n qui est de 150 Mbit/s. Il est intéressant de noter que le *flooding* offre le temps de convergence le plus rapide, cependant, il obtient le plus important débit de paquets et binaire. Cette observation s'explique aisément par la nature du protocole, qui inonde littéralement les canaux de communication d'informations. Le mode hybride se révèle être la meilleure option, car bien qu'il augmente le débit d'environ 19,23% par rapport au mode *push*, il permet une réduction significative du temps de convergence, d'environ

65%. De plus, il réduit le débit par rapport au *flooding* de 54,62%, tout en maintenant un temps de convergence d'environ 15 secondes, ce qui est tout à fait raisonnable.

Étude sur la scalabilité

TABLE 4.5 – Données de performance par nombre de noeuds en mode hybride.

Nombre de nœuds	Temps de convergence	Débit moyen de paquets (paquets/sec)	Débit moyen en kilobits (Kbits/sec)	Débit en pourcentage
10	7.808300	237.80	131.09219	0.09%
30	27.913096	785.61	419.74122	0.28%
50	50.708449	1688.93	942.72670	0.63%
100	83.115058	4856.52	2973.78203	1.98%

TABLE 4.6 – Données de performance par nombre de noeuds pour le *flooding*.

Nombre de nœuds	Temps de convergence	Débit moyen de paquets (paquets/sec)	Débit moyen en kilobits (Kbits/sec)	Débit en pourcentage
10	1.411411	1690.02	880.79740	0.59%
30	77.842099	2883.13	1497.52306	1.0%
50	82.823951	3604.45	1940.95096	1.29%
100	79.778647	9822.36	5362.12297	3.57%

Nous allons étudier le temps de convergence, le débit de paquets et le débit binaire en fonction du nombre de nœuds participant au protocole, comme présenté dans la Table 4.5. Ces résultats concernent le mode hybride, avec un nombre maximal d'échanges de *gossips* correspondant à environ un tiers du nombre total de nœuds, et un intervalle de requête et de mise à jour de 0,3 secondes. Il est clairement visible que plus le nombre de nœuds augmente, plus le temps de convergence, le nombre de paquets échangés et le débit binaire augmentent. Lorsqu'il y a un grand nombre de nœuds, le temps de convergence et le débit commencent à devenir significatifs, il peut donc être intéressant de mettre en place des stratégies pour réduire ce temps de convergence, telles que la sélection des voisins basée sur leur vitesse de propagation de l'information ou la qualité de la liaison. Dans la Table 4.6, les résultats présentent le temps de convergence, le débit de paquets et le débit binaire en fonction du nombre de nœuds participant, mais cette fois pour le protocole de *flooding*. Si l'on compare avec la Table 4.5, on constate qu'avec un nombre de nœuds équivalent à 10, le temps de convergence est meilleur pour le *flooding*, cependant, le débit augmente considérablement, atteignant une augmentation du débit binaire de près de 572%. Pour 30 et 50 nœuds, le temps de convergence pour le *flooding* est significativement plus élevé, tout comme le débit. Cela s'explique par une saturation d'informations pour les nœuds, ce qui les oblige à traiter un grand volume d'informations qui inondent le réseau. Le temps de convergence reste sensiblement le même pour 100 nœuds, mais le débit binaire augmente de manière significative, atteignant une augmentation de 80,28%.

Conclusion

L'étude sur le nombre de *gossips* échangés révèle des relations complexes entre l'intervalle de mise à jour, le nombre de *gossips* à envoyer, le temps de convergence, et le débit. Il existe un compromis entre le temps de convergence et le débit, soulignant l'importance de concevoir un protocole de communication adapté aux besoins du réseau tout en optimisant l'utilisation des canaux de communication. Le mode hybride

se démarque comme la meilleure option, combinant une réduction significative du temps de convergence avec une augmentation du débit par rapport au mode *push*, tout en maintenant un débit binaire raisonnable. En ce qui concerne l'étude sur la scalabilité, il est clair que le nombre de nœuds a un impact sur le temps de convergence, le nombre de paquets échangés et le débit binaire. Des stratégies de sélection de voisins basées sur la vitesse ou la qualité du lien peuvent être nécessaires pour gérer efficacement un grand nombre de nœuds. En conclusion, le protocole de comméragage se révèle être une solution adaptée et efficace pour la diffusion d'informations entre les nœuds mobiles MEC au sein d'un réseau ad-hoc Wi-Fi en milieu urbain en extérieur, notamment pour l'exécution d'un filtre à particules distribué. Des pistes d'amélioration incluent la sélection des nœuds voisins en fonction de critères définis (comme la vitesse) plutôt que de manière aléatoire, la binarisation et la compression des données, ainsi qu'une optimisation plus précise des paramètres du protocole.

4.6.2 Prévisions du filtre à particule

Dans cette section, nous examinons les différents paramètres et les performances du filtre à particules sur les données générées synthétiquement.

Étude sur l'intervalle d'observation des données réelles

La Table 4.7 présente les résultats des trois métriques pour les intervalles d'observation expérimentés. Ces métriques sont collectées toutes les 2,5 minutes et résumées par la valeur maximale, la valeur minimale, et la moyenne sur les 3 heures de données. Les résultats pour la RMSE sont mis en évidence en gras, car la RMSE implique d'abord un carré des écarts, mettant ainsi en avant les résultats aberrants tout en restant interprétables par la réduction au carré. On peut observer une très légère, voire quasi inexistante, amélioration de la RMSE lorsque l'intervalle d'observation est réduit. Ce résultat n'est pas surprenant, étant donné que le filtre à particules est initialement conçu pour fonctionner avec de longs intervalles d'observation d'une heure [117]. En conséquence, il pourrait être tentant de conclure que cette amélioration n'est pas suffisamment significative pour justifier une réduction importante de l'intervalle d'observation, qui entraîne une utilisation plus intense des ressources de communication précieuses des unités mobiles. Cependant, dans notre cas d'utilisation, caractérisé par une grande dynamique où les robots peuvent facilement se retrouver isolés, il est essentiel de prendre en compte le fait que certains cycles d'observation pourraient être manqués. Par conséquent, maintenir un intervalle trop long pourrait finalement nuire à la qualité des prédictions dans ces situations, où des observations moins fréquentes pourraient s'avérer contre-productives.

Comparaison des prédictions avec les modèles de références

Pour évaluer les performances du filtre par rapport aux modèles de référence, nous comparons les prédictions après une heure de données, puis nous effectuons une deuxième prédiction pour la troisième heure et

TABLE 4.7 – Performances selon l'intervalle d'observation. Les résultats pour la moyenne de la RMSE sont en gras.

Intervalle d'observation (minutes)	Métrique	Max	Min	Moyenne
5	MAE	5.28	2.16	3.34
	MSE	38.48	7.92	18.74
	RMSE	6.20	2.81	4.25
10	MAE	5.28	2.32	3.56
	MSE	46.40	8.88	20.49
	RMSE	6.81	2.97	4.47
15	MAE	4.88	2.00	3.48
	MSE	38.88	5.84	19.56
	RMSE	6.24	2.41	4.36
20	MAE	4.96	2.40	3.64
	MSE	40.40	8.72	21.51
	RMSE	6.36	2.95	4.59
30	MAE	5.12	2.40	3.55
	MSE	46.24	8.32	20.65
	RMSE	6.80	2.88	4.49
40	MAE	5.84	1.92	3.76
	MSE	49.52	6.88	22.95
	RMSE	7.04	2.62	4.71
50	MAE	5.20	2.64	3.79
	MSE	39.28	10.96	23.61
	RMSE	6.27	3.31	4.79

TABLE 4.8 – Performances des prévisions des modèles.

Modèle	Prévision au bout d'1 heure		Prévision au bout de 3 heures	
	MAE	RMSE	MAE	RMSE
LSTM	2.5	3.26	2.41	2.97
ARIMA	3.83	4.37	3.12	4.19
DPF	2.64	3.1	2.72	3.37

comparons les résultats. Nous n'utilisons pas la MSE comme métrique, car elle manque d'interprétabilité. Il est important de rappeler que les modèles LSTM et ARIMA sont entraînés sur une heure de données, et l'intervalle d'observation pour le filtre à particules est fixé à 18 minutes. Les résultats sont présentés dans la Table 4.8. Nous pouvons constater que le LSTM est le modèle qui performe le mieux, à l'exception d'une légère exception sur la RMSE pour la première prédiction. Cependant, il est essentiel de noter que le filtre à particules reste très proche du LSTM, en particulier lorsque l'on considère que l'erreur calculée ici concerne le nombre de personnes par région. De plus, notre objectif ici n'est pas nécessairement d'obtenir la meilleure prédiction absolue, mais plutôt une prédiction raisonnable. Cette approche distribuée présente l'avantage par rapport aux modèles d'apprentissage profond de ne pas nécessiter de temps d'entraînement, elle ne dépend pas de données historiques et est plus facilement distribuable.

Conclusion

Nous avons pu constater que le filtre à particules offre des performances similaires à un modèle LSTM, tout en ayant l'avantage d'être plus léger, car il ne nécessite pas de temps d'entraînement ni de données historiques. De plus, il se révèle plus performant que le modèle ARIMA. Par ailleurs,

nous avons observé que l'augmentation de l'intervalle d'observation n'a pas entraîné une réduction significative des performances. Cela implique que les robots peuvent rester isolés sur de longues périodes de temps sans subir de détérioration des prédictions. Néanmoins, ces intervalles d'observation doivent rester assez courts pour éviter que les unités mobiles isolées ne manquent des cycles d'observations importants. Une autre piste à explorer pourrait consister à déclencher le partage des mesures de manière opportuniste, lorsque les unités ne sont plus isolées. Toutefois, cela nécessite une étude approfondie et un réglage précis pour garantir que les cycles d'observations soient adéquats sans surcharger le réseau.

4.7 Conclusion

Nous avons examiné la faisabilité de la distribution d'un filtre à particules pour prédire la densité humaine dans un contexte d'unités mobiles MEC. Nous avons mis en évidence la complexité des relations entre l'intervalle de mise à jour, le nombre de *gossips* à envoyer, le temps de convergence et le débit dans un contexte de réseau ad-hoc Wi-Fi en milieu urbain extérieur. Le mode hybride s'est révélé être une option prometteuse, offrant une réduction significative du temps de convergence tout en maintenant un débit binaire raisonnable par rapport au mode *push*. De plus, nous avons discuté de la scalabilité du protocole de comméragage, montrant que des stratégies de sélection de voisins basées sur des critères spécifiques peuvent s'avérer nécessaires pour gérer efficacement un grand nombre de nœuds. En résumé, nous pouvons conclure que le protocole de comméragage constitue une solution adaptée pour la diffusion d'informations entre les nœuds mobiles MEC, en particulier pour l'application d'un filtre à particules distribué. Cependant, des améliorations sont envisageables, telles que la sélection de voisins basée sur des critères spécifiques, la binarisation et la compression des données, ainsi qu'une optimisation plus précise des paramètres du protocole. De plus, le filtre à particules se révèle être une alternative prometteuse aux modèles plus complexes, offrant des avantages en termes de légèreté et de performances. Nous avons également souligné l'importance de l'intervalle d'observation, tout en suggérant la possibilité d'un partage opportuniste des mesures dans un contexte de réseau mobile. Cependant, il convient de noter que cette approche nécessiterait une étude approfondie et un ajustement précis pour maintenir des cycles d'observations adéquats sans surcharger le réseau.

5.1 Gestion des ressources MEC : allocation et déploiement

Dans ce Chapitre, nous fournissons un état de l'art détaillé des méthodes d'allocation des ressources MEC fixes mais aussi des ressources MEC mobiles. Dans le contexte de cette thèse, nous présentons cet état de l'art afin de faciliter la conception de nouveaux schémas qui tirent parti des outils de prédiction présentés dans les chapitres précédents, à savoir les Chapitres 3 et 4. Comme dit en introduction au Chapitre 1.3, les serveurs MEC sont limités en ressources de communication et de calculs et nécessitent donc une gestion précise de celles-ci. Les ressources mobiles ont également besoin d'un schéma de déploiement intelligent pour fournir leurs services de manière efficace. Dans cette étude, nous examinons l'allocation des ressources et les techniques de déploiement des ressources mobiles dans le contexte de la MEC. Nous allons dans un premier temps voir les schémas d'allocation des ressources pour la MEC. De la même manière, nous passons ensuite en revue le déploiement des ressources mobiles de la MEC.

5.2 Allocation de ressources

Dans ce Chapitre, nous commençons par présenter les différents aspects à prendre en compte pour modéliser les tâches des appareils demandeurs. Ensuite, nous exposons les différents types de ressources MEC que nous pouvons allouer, qu'il s'agisse de ressources informatiques ou de communication. Nous discutons ensuite, des différents objectifs d'optimisation des méthodes d'allocation de ressources. Enfin, nous passons en revue les différentes méthodes qui présentent leur propre compromis entre précision et vitesse. Nous discutons de ces méthodes en fonction de l'échelle du système et des exigences. En ce qui concerne l'architecture, les articles examinés tiennent compte du nombre de serveurs de périphérie, du nombre d'utilisateurs, du nombre de points d'accès (AP) et peuvent intégrer le Cloud dans le système. Leurs objectifs sont principalement de réduire la consommation d'énergie du système ou la latence. Il existe également d'autres objectifs tels que la réduction des différents coûts du réseau, l'assurance de sa fiabilité ou la maximisation de la couverture des tâches comme par exemple le nombre de tâches traitées. Les ressources allouées sont principalement des ressources informatiques, avec ou sans ressources de communication, et le placement des tâches comme par exemple sur quel serveur une tâche sera traitée. Les ressources de communication sont rarement allouées seules. Selon les objectifs et le système, les articles examinés tiennent compte de différentes contraintes, telles que les capacités de communication des serveurs MEC, la batterie des appareils, et ainsi de suite. Enfin, chaque article examiné peut résoudre, en plus de l'allocation de ressources, la décision de déchargement, c'est-à-dire la

5.1	Gestion des ressources MEC : allocation et déploiement	75
5.2	Allocation de ressources	75
5.2.1	Modélisations des tâches	76
5.2.2	Quelles ressources peut-on allouer ?	76
5.2.3	Allouer des ressources, mais pour quels objectifs ?	78
5.2.4	Méthodes d'allocation des ressources	80
5.3	Déploiement de ressources mobiles	84
5.3.1	Comment modéliser un système avec des ressources mobiles	85
5.3.2	Méthodes pour le déploiement	85
5.4	Conclusion	87

[162]: Jienan CHEN, Siyu CHEN, Qi WANG, Bin CAO, Gang FENG et Jianhao HU. « iRAF : A Deep Reinforcement Learning Approach for Collaborative Mobile Edge Computing IoT Networks ». In : *IEEE Internet of Things Journal* 6.4 (2019)

[163]: Pengfei WANG, Zijie ZHENG, Boya DI et Lingyang SONG. « HetMEC : Latency-Optimal Task Assignment and Resource Allocation for Heterogeneous Multi-Layer Mobile Edge Computing ». In : *IEEE Transactions on Wireless Communications* 18.10 (2019)

[164]: Jiao ZHANG, Xiping HU, Zhaolong NING, Edith C.-H. NGAI, Li ZHOU, Jibo WEI, Jun CHENG, Bin HU et Victor C. M. LEUNG. « Joint Resource Allocation for Latency-Sensitive Services Over Mobile Edge Computing Networks With Caching ». In : *IEEE Internet of Things Journal* 6.3 (2019)

[165]: Lei YANG, Jiannong CAO, Hui CHENG et Yusheng JI. « Multi-User Computation Partitioning for Latency Sensitive Mobile Cloud Applications ». In : *IEEE Transactions on Computers* 64.8 (2015)

[166]: Linpei LI, Xiangming WEN, Zhaoming LU, Qi PAN, Wenpeng JING et Zhiqun HU. « Energy-Efficient UAV-Enabled MEC System : Bits Allocation Optimization and Trajectory Design ». In : *Sensors (Basel, Switzerland)* 19 (2019)

[167]: Tuyen X. TRAN et Dario POMPILI. « Joint Task Offloading and Resource Allocation for Multi-Server Mobile-Edge Computing Networks ». In : *IEEE Transactions on Vehicular Technology* 68.1 (2019)

[167]: Tuyen X. TRAN et Dario POMPILI. « Joint Task Offloading and Resource Allocation for Multi-Server Mobile-Edge Computing Networks ». In : *IEEE Transactions on Vehicular Technology* 68.1 (2019)

[168]: Stefania SARDELLITI, Mattia MERLUZZI et Sergio BARBAROSSA. « Optimal Association of Mobile Users to Multi-Access Edge Computing Resources ». In : *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*. 2018

[169]: Rania ISLAMBOULI et Sanaa SHARAFEDDINE. « Optimized 3D Deployment of UAV-Mounted Cloudlets to Support Latency-Sensitive Services in IoT Networks ». In : *IEEE Access* 7 (2019)

décision de traiter les tâches localement sur les appareils ou à distance sur les serveurs.

5.2.1 Modélisations des tâches

La modélisation du système est cruciale dans un schéma d'allocation de ressources efficace. Elle permet à l'algorithme de prendre en compte les aspects critiques du système et de fournir des résultats adaptés. Un aspect important de la modélisation du système est la modélisation des tâches. Une tâche est couramment modélisée par (S, C, L) , où S représente les données associées (données d'entrée et code), C les cycles CPU nécessaires pour accomplir la tâche et L le délai de la tâche, c'est-à-dire la latence maximale tolérée pour la tâche [162-164]. S et C peuvent être déduits grâce au profilage du code [165-167]. Certains travaux n'incluent pas le délai de la tâche dans leur modélisation [167, 168]. Cependant, ce paramètre est essentiel dans les applications sensibles à la latence pour des raisons évidentes. Il permet au schéma de respecter les délais des tâches, de prioriser les tâches avec des délais proches et peut être utilisé pour abandonner certaines tâches obsolètes qui surchargent les tampons. De plus, les travaux peuvent supposer la division des tâches pour allouer des parties avec différentes ressources afin d'accélérer le traitement. Cependant, si certaines parties des tâches ont des dépendances fortes, cela peut surcharger le réseau car une ressource doit attendre que les autres traitent sa partie. Certains travaux ne tiennent pas compte de cette division pour simplifier le schéma [166]. La vitesse de génération des tâches [163] ou leur distribution [169] ont un impact sur la charge de travail sur le réseau et peuvent aider à prévenir les goulets d'étranglement.

5.2.2 Quelles ressources peut-on allouer ?

Divers types de ressources peuvent être alloués, principalement des ressources de communication et des ressources de calcul. De plus, nous pouvons les considérer conjointement, ce qui conduit à des schémas plus efficaces.

Les ressources pour la communication...

Les ressources de communication sont généralement liées à la bande passante. Elle est allouée aux dispositifs avec un pourcentage de la largeur de bande totale disponible [164, 170] ou en fonction de la quantité de bande passante radio [162]. Cela peut également dépendre de la méthode d'accès au canal considérée dans le système. Pour les systèmes utilisant TDMA, certains travaux allouent du temps dans chaque créneau horaire pour chaque dispositif proportionnellement aux données qu'ils doivent télécharger [163, 171]. Pour l'OFDMA, des sous-porteuses sont allouées [172, 173]. La méthode prometteuse d'accès multiple non orthogonal (NOMA), adaptée à la 5G, permet de partager les sous-porteuses entre plusieurs utilisateurs plutôt qu'à un seul, comme dans l'OFDMA. Ainsi, si le système utilise NOMA, le schéma d'allocation de ressources doit attribuer les sous-porteuses à plusieurs utilisateurs [174, 175]. Certains travaux modélisent également de manière abstraite les ressources de communication

pour les appliquer à différents types de systèmes. Un point à prendre en compte lors de l'allocation de ressources de communication est l'interférence. L'interférence intra-cellulaire est généralement ignorée lors de l'attribution de sous-canaux [167, 176]. L'interférence inter-cellulaire rend l'allocation de ressources plus complexe car elle ajoute une dépendance entre les taux de téléchargement des utilisateurs [167]. Certains travaux ignorent cette interférence inter-cellulaire en supposant que les cellules sont suffisamment éloignées les unes des autres ou qu'elles disposent d'une allocation de bande passante orthogonale [176]. Cependant, il peut être intéressant de la prendre en compte pour les réseaux où ces interférences sont très probables, comme les réseaux MEC ultra-denses. De plus, étant donné que la MEC peut disposer de technologies d'accès radio (RAT) hétérogènes, cela pose le problème intéressant de choisir le bon RAT pour servir un dispositif à un moment donné pour une tâche donnée. Par exemple, Hsu et al. [177] considèrent la 5G sous licence et la NR-U sans licence. En effet, chaque RAT a ses propres caractéristiques, telles que la couverture, le support de la mobilité, le débit de données, etc. Tout cela peut influencer le délai, la consommation d'énergie et la qualité de service. Ils peuvent également entraîner des coûts supplémentaires, comme la 5G au lieu du Wi-Fi généralement gratuit. Enfin, les communications d'urgence de nouvelle génération [178] et les communications de sécurité publique [179] sont des défis à prendre en compte dans l'allocation des ressources de la MEC.

...pour le calcul...

Contrairement au Cloud, les systèmes MEC disposent de ressources informatiques de calcul limitées. Par conséquent, il s'agit de ressources critiques que nous devons allouer efficacement. Cela est d'autant plus important lorsque de nombreux utilisateurs mobiles sont présents ou que les ressources de périphérie sont extrêmement limitées, comme c'est souvent le cas dans les applications critiques. Si les ressources de calcul sont mal allouées, des tâches importantes des dispositifs peuvent ne pas être traitées en temps voulu, et l'ensemble du système peut être congestionné. Les ressources de calcul peuvent être mesurées en cycles CPU par seconde [164, 167, 180] ou en cœurs CPU [181]. Pour les MEC basées sur les UAV, certains travaux allouent le nombre de bits déchargés à l'UAV [166, 182]. Cependant, des travaux récents envisagent d'allouer plutôt les fréquences CPU, car cela semble réduire encore davantage la consommation d'énergie [183-185]. Pour réduire encore plus la latence et aider les serveurs MEC, certains travaux envisagent également d'utiliser une partie de la puissance de calcul disponible sur certains dispositifs pouvant l'utiliser pour aider d'autres dispositifs. Les dispositifs disposant de ressources suffisantes communiquent directement avec les dispositifs demandeurs, ce qui est appelé la communication de dispositif à dispositif (D2D) [186] ou la communication de machine à machine (M2M) [174].

...et les deux conjointement!

L'allocation de ressources de communication et de ressources de calcul conjointes s'influence mutuellement. Peu importe la quantité de ressources de communication allouée à une tâche, si elle ne dispose pas de

[164]: Jiao ZHANG, Xiping HU, Zhaolong NING, Edith C.-H. NGAI, Li ZHOU, Jibo WEI, Jun CHENG, Bin HU et Victor C. M. LEUNG. « Joint Resource Allocation for Latency-Sensitive Services Over Mobile Edge Computing Networks With Caching ». In : *IEEE Internet of Things Journal* 6.3 (2019)

[170]: Sladana JOSILO et Gyorgy DAN. « Joint Allocation of Computing and Wireless Resources to Autonomous Devices in Mobile Edge Computing ». In : *Proceedings of the 2018 Workshop on Mobile Edge Communications. MECOMM'18*. Budapest, Hungary : Association for Computing Machinery, 2018. ISBN : 9781450359061

[172]: Ata KHALILI, Sheyda ZARANDI et Mehdi RASTI. « Joint Resource Allocation and Offloading Decision in Mobile Edge Computing ». In : *IEEE Communications Letters* 23.4 (2019)

[173]: Jiao ZHANG, Li ZHOU, Fuhui ZHOU, Boon-Chong SEET, Haijun ZHANG, Zhiping CAI et Jibo WEI. « Computation-Efficient Offloading and Trajectory Scheduling for Multi-UAV Assisted Mobile Edge Computing ». In : *IEEE Transactions on Vehicular Technology* 69.2 (2020)

[174]: Nengyu ZHU, Xiaodong XU, Shujun HAN et Suyu LV. « Sleep-Scheduling and Joint Computation-Communication Resource Allocation in MEC Networks for 5G IIoT ». In : *2021 IEEE Wireless Communications and Networking Conference (WCNC)*. 2021

[175]: Mohammad Arif HOSSAIN et Nirwan ANSARI. « Energy Aware Latency Minimization for Network Slicing Enabled Edge Computing ». In : *IEEE Transactions on Green Communications and Networking* 5.4 (2021)

[176]: Quoc-Viet PHAM, Long Bao LE, Sang-Hwa CHUNG et Won-Joo HWANG. « Mobile Edge Computing With Wireless Backhaul : Joint Task Offloading and Resource Allocation ». In : *IEEE Access* 7 (2019)

[177]: Che-Wei HSU, Yung-Lin HSU et Hung-Yu WEI. « Energy-Efficient Edge Offloading in Heterogeneous Industrial IoT Networks for Factory of Future ». In : *IEEE Access* 8 (2020)

[181]: Tianyu YANG, Yulin HU, M. CenK GURSOY, Anke SCHMEINK et Rudolf MATHAR. « Deep Reinforcement Learning based Resource Allocation in Low Latency Edge Computing Networks ». In : *2018 15th International Symposium on Wireless Communication Systems (ISWCS)*. 2018

[186]: Rong CHAI, Junliang LIN, Minglong CHEN et Qianbin CHEN. « Task Execution Cost Minimization-Based Joint Computation Offloading and Resource Allocation for Cellular D2D MEC Systems ». In : *IEEE Systems Journal* 13.4 (2019)

[162]: Jienan CHEN, Siyu CHEN, Qi WANG, Bin CAO, Gang FENG et Jianhao HU. « iRAF : A Deep Reinforcement Learning Approach for Collaborative Mobile Edge Computing IoT Networks ». In : *IEEE Internet of Things Journal* 6.4 (2019)

[187]: Qiang FAN et Nirwan ANSARI. « Application Aware Workload Allocation for Edge Computing-Based IoT ». In : *IEEE Internet of Things Journal* 5.3 (2018)

suffisamment de ressources de calcul, la tâche ne sera pas traitée plus rapidement, et inversement [163]. Ainsi, ces deux types de ressources sont impliqués dans les exigences de qualité de service (QoS), telles que la latence et la consommation d'énergie, et leur allocation conjointe conduit à des résultats plus efficaces [167].

Mais aussi la sélection de servers

La sélection du serveur pour une tâche peut également être considérée comme une allocation de ressources. L'association de nœuds MEC avec des tâches est pertinente car les nœuds MEC peuvent avoir des capacités hétérogènes, tant en quantité qu'en qualité, et mieux correspondre aux besoins d'une tâche [162, 187]. De plus, il existe un compromis à prendre en compte entre le temps de calcul et le délai du réseau, en fonction des charges de travail des serveurs, de leur distance par rapport aux dispositifs et de la qualité de leur canal. Par exemple, il peut être préférable d'attribuer une tâche à un serveur plus éloigné mais moins occupé [187].

5.2.3 Allouer des ressources, mais pour quels objectifs ?

Les objectifs du schéma d'allocation de ressources dépendent des cas d'utilisation et des applications considérées par le travail. Ils peuvent être plus adaptés aux applications sensibles à la latence en minimisant le temps de réalisation des tâches, ou correspondre aux systèmes MEC avec des dispositifs alimentés par batterie en minimisant leur consommation d'énergie. L'objectif peut être ajusté avec un poids dans la fonction objective. Il peut viser à prioriser certains dispositifs [167] ou certains aspects d'un problème multi-objectif [176], comme donner plus de poids à la consommation d'énergie plutôt qu'à la latence.

Paramètres prenant en compte l'énergie

Dans les réseaux de calcul en périphérie mobile (MEC), les appareils mobiles et les objets de l'IoT sont majoritairement sur batterie. Par conséquent, minimiser leur consommation d'énergie est crucial pour maintenir la qualité de l'expérience de l'utilisateur [163] et préserver la batterie des appareils autonomes afin de leur permettre de terminer leurs tâches. Certains travaux considèrent la consommation d'énergie globale, par exemple, depuis le calcul local jusqu'au calcul déporté [172]. D'autres travaux ne tiennent compte que de la consommation d'énergie de l'appareil, car on suppose que les serveurs MEC disposent de sources d'alimentation fiables [176, 188]. Cependant, dans les applications critiques, les serveurs peuvent être alimentés par batterie, comme ceux embarqués sur les drones (UAV) ou dans les bus. Ils peuvent avoir plus d'énergie à leur disposition que les appareils finaux, néanmoins, leur budget énergétique est limité. De plus, la consommation d'énergie de l'ensemble du système est toujours importante pour minimiser l'impact énergétique de l'application. La consommation d'énergie pour une tâche est souvent calculée comme $E = \kappa F \beta \cdot c$, où F représente les capacités de calcul de l'appareil en cycles CPU par seconde, c représente le nombre de

cycles CPU nécessaires pour terminer la tâche, κ et β sont des constantes qui dépendent de l'architecture de la puce de l'appareil [167, 189]. κ est souvent de l'ordre de 10^{-26} ou 10^{-27} et β est de 2 [162, 176, 190, 191]. Ainsi, les capacités de calcul influencent le temps de traitement de la tâche mais aussi la consommation d'énergie. Par conséquent, il existe un compromis entre la consommation d'énergie et le temps d'exécution à prendre en compte. Ce compromis peut être ajusté avec un facteur de poids dans l'objectif d'optimisation pour répondre aux besoins de l'application, en ayant une faible consommation d'énergie ou une latence réduite [190, 192]. Il peut également inclure des facteurs externes tels que la batterie résiduelle de l'appareil [190] et s'ajuster ainsi aux besoins des appareils en temps réel. Lorsque les serveurs MEC sont des drones UAV, le temps de vol stationnaire doit être inclus dans le modèle énergétique sous la forme : $E_h = P \cdot T$, où P est la puissance nécessaire pour le vol stationnaire et T le temps de vol stationnaire [192]. En plus du temps de vol stationnaire, les UAV consomment de l'énergie pour voler, en fonction de leur vitesse et de leur poids [166]. Leurs accélérations ont également un impact significatif sur l'énergie [183]. Nous pouvons ignorer certains points de consommation d'énergie dans l'optimisation, qu'il s'agisse de la consommation d'énergie au repos que nous ne pouvons pas contrôler. C'est le cas de la consommation d'énergie au repos du serveur ou de la consommation d'énergie des liaisons indépendantes du trafic [166, 193]. De plus, certaines actions sont négligeables par rapport à d'autres dans le système, comme la consommation d'énergie liée au téléchargement [191].

Réduction de la latence

Dans les applications utilisant le MEC la latence est un point essentiel, la minimisation de la latence est donc un objectif clé pour pouvoir assurer une QoS et une QoE optimal. De plus, pour les missions critiques la latence peut avoir des conséquences vitales, comme les opérations de recherche et de sauvetage. La latence d'une tâche comprend le temps de traitement et le temps de transmission nécessaires depuis l'appareil jusqu'au point d'extrémité (Edge) et éventuellement vers le Cloud [163, 187, 189, 194]. Certains travaux [171] ajoutent à cela le temps de compression, présent dans les systèmes avec des tâches lourdes comme le traitement vidéo. Nous pouvons également ajouter le délai d'attente de calcul local ou distant [180, 195] en raison de la génération continue de tâches, même lorsque d'autres tâches sont traitées. Le temps de traitement dépend des cycles CPU nécessaires pour terminer la tâche et des capacités de calcul, par exemple, des cycles CPU par seconde, alloués à la tâche [163, 168]. La latence est également affectée par la vitesse de génération des données. Lorsque la génération est supérieure aux capacités de traitement du système, les données s'accumulent dans les tampons et les nœuds ne traitent pas les tâches en temps réel. Wang et al. [196] le désignent comme un état de blocage et proposent d'adapter le schéma d'allocation des ressources en fonction de l'état du système, qu'il soit en état de blocage ou en état de non-blocage. De plus, la génération de données est généralement non uniforme dans le système, ce qui entraîne des charges de travail variables entre les serveurs, certains pouvant être surchargés tandis que d'autres sont exempts de tâches. Il est donc intéressant de prendre en compte l'équilibre dans l'allocation des ressources [197],

[192]: Songtao GUO, Jiadi LIU, Yuan-yuan YANG, Bin XIAO et Zhetao LI. « Energy-Efficient Dynamic Computation Offloading and Cooperative Task Scheduling in Mobile Cloud Computing ». In : *IEEE Transactions on Mobile Computing* 18.2 (2019)

[183]: Yuan LIU, Ke XIONG, Qiang NI, Pingyi FAN et Khaled Ben LETAIEF. « UAV-Assisted Wireless Powered Cooperative Mobile Edge Computing : Joint Offloading, CPU Control, and Trajectory Optimization ». In : *IEEE Internet of Things Journal* 7.4 (2020)

[180]: Yue WANG, Xiaofeng TAO, Y. Thomas HOU et Ping ZHANG. « Effective Capacity-Based Resource Allocation in Mobile Edge Computing With Two-Stage Tandem Queues ». In : *IEEE Transactions on Communications* 67.9 (2019)

[195]: Chen-Feng LIU, Mehdi BENNIS, Mérouane DEBBAH et H. Vincent POOR. « Dynamic Task Offloading and Resource Allocation for Ultra-Reliable Low-Latency Edge Computing ». In : *IEEE Transactions on Communications* 67.6 (2019)

[187]: Qiang FAN et Nirwan ANSARI. « Application Aware Workload Allocation for Edge Computing-Based IoT ». In : *IEEE Internet of Things Journal* 5.3 (2018)

[198]: Lei FENG, Yu ZHOU, Tang LIU, Xiaoyu QUE, Peng YU, Tao HONG et Xuesong QIU. « Energy-Efficient Offloading for Mission-Critical IoT Services Using EVT-Embedded Intelligent Learning ». In : *IEEE Transactions on Green Communications and Networking* 5.3 (2021)

[199]: Jianhui LIU et Qi ZHANG. « Adaptive Task Partitioning at Local Device or Remote Edge Server for Offloading in MEC ». In : *2020 IEEE Wireless Communications and Networking Conference (WCNC)*. 2020

[200]: Mohammed S. ELBAMBY, Cristina PERFECTO, Chen-Feng LIU, Jihong PARK, Sumudu SAMARAKOON, Xianfu CHEN et Mehdi BENNIS. « Wireless Edge Computing With Latency and Reliability Guarantees ». In : *Proceedings of the IEEE* 107.8 (2019)

[201]: David OHMANN, Meryem SIMSEK et Gerhard P. FETTWEIS. « Achieving high availability in wireless networks by an optimal number of Rayleigh-fading links ». In : *2014 IEEE Globecom Workshops (GC Wkshps)*. 2014

[202]: Muhammad Adeel MAHMOOD, Winston K.G. SEAH et Ian WELCH. « Reliability in wireless sensor networks : A survey and challenges ahead ». In : *Computer Networks* 79 (2015)

[203]: Elie El HABER, Hyame ASSEM ALAMEDDINE, Chadi ASSI et Sanaa SHARAFEDDINE. « UAV-Aided Ultra-Reliable Low-Latency Computation Offloading in Future IoT Networks ». In : *IEEE Transactions on Communications* 69.10 (2021)

[204]: Mehdi BENNIS, Mérouane DEBBAH et H. Vincent POOR. « Ultrareliable and Low-Latency Wireless Communication : Tail, Risk, and Scale ». In : *Proceedings of the IEEE* 106.10 (2018)

[195]: Chen-Feng LIU, Mehdi BENNIS, Mérouane DEBBAH et H. Vincent POOR. « Dynamic Task Offloading and Resource Allocation for Ultra-Reliable Low-Latency Edge Computing ». In : *IEEE Transactions on Communications* 67.6 (2019)

ainsi que le compromis entre le temps de calcul et le temps de transmission lors du déplacement d'une tâche vers un nœud moins chargé mais plus éloigné [187]. De plus, certains appareils peuvent traiter des tâches critiques ou occuper un rôle central dans le système, ce qui signifie qu'ils ont besoin de priorité dans leur traitement. Une solution proposée dans [171] consiste à minimiser la somme pondérée des retards de tous les appareils, le poids reflétant l'importance des appareils dans le système. En alternative, [198] propose de mesurer la priorité de chaque tâche en fonction des exigences en matière de délai et de fiabilité. En règle générale, le temps de téléchargement depuis le serveur vers les appareils est ignoré, car les données résultantes sont plus petites et les liaisons descendantes ont des débits plus élevés [166, 167]. De plus, le temps de transmission entre une station de base et son serveur MEC associé est ignoré [164]. Enfin, comme vu précédemment, la répartition des tâches peut considérablement réduire le temps de traitement en parallélisant le traitement. [199] montre que la placement dynamique de la décision de partitionnement des tâches, c'est-à-dire la décision de traiter sur quel nœud chaque partie de la tâche, peut réduire la latence. En effet, si la décision est prise sur le nœud demandeur, il peut utiliser ses ressources limitées. Mais si la décision est prise sur un serveur MEC distant, cela peut prendre plus de temps pour atteindre d'autres serveurs MEC, en fonction de leur emplacement par rapport à l'appareil.

Garantie de fiabilité

Certaines tâches d'applications critiques peuvent être d'une grande importance, comme des applications de surveillance ou pour la santé. Par conséquent, le MEC doit avoir un certain niveau de fiabilité pour garantir le traitement de ces tâches. Dans les réseaux sans fil, la fiabilité est considérée comme la probabilité de transférer avec succès des données dans un délai [200]. Un premier défi dans les réseaux MEC est la défaillance des nœuds. La redondance des tâches est une solution pertinente pour atténuer cet effet [201, 202]. Cependant, elle peut surcharger le réseau si la redondance nécessite plus de ressources de calcul ou de communication que nécessaire. Une mesure de la défaillance des nœuds aide à garantir la fiabilité minimale des tâches, évitant la surutilisation des ressources [203]. Un autre défi est la survenue d'événements extrêmes dans les files d'attente de traitement des serveurs et des appareils mobiles (UE). Lorsque les files d'attente sont surchargées, elles peuvent abandonner certaines tâches critiques, et assurer un délai d'attente moyen n'est pas suffisant pour empêcher cela [204]. Ainsi, le travail [195] utilise les statistiques de la longueur extrême de la file d'attente pour garantir la fiabilité.

5.2.4 Méthodes d'allocation des ressources

La méthode choisie pour l'allocation des ressources doit proposer un compromis satisfaisant entre la précision, la complexité de calcul et la scalabilité en fonction du problème et de son contexte. Certaines méthodes peuvent être incapables de résoudre un problème [181] ou de satisfaire les exigences du système. De plus, la méthode doit s'adapter à l'échelle du système, ne pas être trop complexe pour les systèmes à

grande échelle, et répondre à ses besoins, par exemple si des résultats suboptimaux sont suffisants.

Méthode d'optimisation

Les méthodes classiques d'optimisation mathématique visent à résoudre les problèmes de manière optimale. Cao et al. [188] résolvent de manière optimale l'allocation des ressources dans un réseau à trois nœuds pour minimiser la consommation d'énergie des appareils avec la méthode de dualité de Lagrange. Chen et al. [194] proposent un schéma d'allocation de ressources et de placement de tâches dans les réseaux ultra-denses pour minimiser le temps de réalisation des tâches. Ils résolvent la partie allocation de ressources computationnelles du problème avec les conditions de Karush–Kuhn–Tucker (KKT). Ren et al. [171] exploitent les conditions KKT pour allouer les ressources d'un serveur MEC aux utilisateurs tout en minimisant le délai, où les données sont compressées localement par l'utilisateur avant envoi. Bien que les méthodes classiques d'optimisation mathématique permettent d'obtenir des résultats optimaux, elles présentent une complexité significative. Par conséquent, elles sont adaptées aux systèmes à petite échelle avec peu de paramètres. Elles ne conviennent pas aux systèmes à grande échelle où la complexité est trop élevée pour être gérée, et elles ne seront soit pas en mesure de résoudre le problème, soit demanderont une quantité de temps irréalisable.

[188]: Xiaowen CAO, Feng WANG, Jie XU, Rui ZHANG et Shuguang CUI. « Joint Computation and Communication Cooperation for Energy-Efficient Mobile Edge Computing ». In : *IEEE Internet of Things Journal* 6.3 (2019)

Techniques de décomposition

Les techniques de décomposition sont utilisées pour des problèmes difficiles où des solutions optimales sont peu probables. Elles décomposent le problème initial en sous-problèmes plus faciles à résoudre. De plus, elles sont employées à la place des solutions optimales pour réduire la complexité. Une complexité réduite est importante dans les systèmes MEC où les ressources sont limitées, contrairement au cloud. Elles peuvent constituer un bon compromis entre l'efficacité et les résultats, notamment dans les systèmes où une précision moyenne est suffisante. De plus, nous pouvons les mettre en œuvre plus facilement et de manière distribuée.

Algorithmes itératifs : Les algorithmes itératifs sont utilisés lorsque l'on considère plusieurs problèmes conjoints. On peut découpler les sous-problèmes et les résoudre individuellement, par exemple en séparant la décision de déchargement (offloading) de l'allocation des ressources. Pour maintenir la connexion entre les sous-problèmes, nous les résolvons à l'aide d'un algorithme itératif. Chaque itération prend la sortie de l'itération précédente en entrée pour mettre à jour la solution jusqu'à convergence vers la solution optimale. Cela permet de réduire la complexité, mais au prix de la précision de la solution. Dans les algorithmes itératifs, il est important de surveiller les propriétés de convergence et le nombre d'itérations nécessaires. Par exemple, Li et al. [205] proposent une heuristique en deux étapes résolvant de manière itérative la décision de déchargement et l'allocation de la fréquence CPU dans le but de minimiser la consommation d'énergie des appareils mobiles. Pham et al. [176] proposent l'algorithme itératif JOBCA pour résoudre le problème d'allocation des ressources et de déchargement pour les réseaux de liaison sans fil. Les réseaux avec liaison sans fil peuvent être utilisés pour les

[206]: Qiuping LI, Junhui ZHAO et Yi GONG. « Computation offloading and resource allocation for mobile edge computing with multiple access points ». In : *IET Communications* 13.17 (2019). eprint : <https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/iet-com.2019.0446>

[192]: Songtao GUO, Jiadi LIU, Yuan-yuan YANG, Bin XIAO et Zhetao LI. « Energy-Efficient Dynamic Computation Offloading and Cooperative Task Scheduling in Mobile Cloud Computing ». In : *IEEE Transactions on Mobile Computing* 18.2 (2019)

[180]: Yue WANG, Xiaofeng TAO, Y. Thomas HOU et Ping ZHANG. « Effective Capacity-Based Resource Allocation in Mobile Edge Computing With Two-Stage Tandem Queues ». In : *IEEE Transactions on Communications* 67.9 (2019)

[207]: Stephen BOYD, Neal PARIKH, Eric CHU, Borja PELEATO et Jonathan ECKSTEIN. « Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers ». In : *Foundations and Trends® in Machine Learning* 3.1 (2011)

[208]: Lin WANG, Lei JIAO, Jun LI, Julien GEDEON et Max MÜHLHÄUSER. « MOERA : Mobility-Agnostic Online Resource Allocation for Edge Computing ». In : *IEEE Transactions on Mobile Computing* 18.8 (2019)

zones rurales ou les services d'urgence où la liaison filaire est coûteuse et contraignante. Li et al. [206] introduisent un schéma de déchargement et d'allocation des ressources pour plusieurs points d'accès sans fil afin de minimiser les coûts monétaires et énergétiques. Tran et Pompili [167] proposent un schéma d'allocation de ressources pour plusieurs serveurs dans des réseaux ultra-denses afin de minimiser la somme pondérée du temps de réalisation des tâches avec la consommation d'énergie des appareils. Pour cela, ils introduisent un algorithme heuristique itératif pour résoudre le problème initial en temps polynomial. Fan et Ansari [187] traitent de l'allocation de charge pour les cloudlets, en prenant en compte le compromis entre l'envoi de tâches à un cloudlet proche mais surchargé ou à un cloudlet éloigné mais moins occupé. Pour simplifier le problème initial, ils proposent un algorithme itératif résolvant l'attribution des tâches et l'allocation des ressources de calcul. Zhang et al. [190] cherchent à trouver le compromis entre la latence et la consommation d'énergie. Ils examinent un scénario avec une petite cellule et un autre avec plusieurs petites cellules, et proposent un algorithme de recherche itératif pour les scénarios avec plusieurs cellules. Zhu et al. [174] introduisent un schéma d'allocation de ressources pour l'Internet industriel des objets (IIoT) 5G. Dans ce schéma, ils incluent des appareils disposant de suffisamment de ressources de calcul pour aider d'autres appareils dans la communication machine à machine.

Décomposition mathématique : Des solutions mathématiques existent pour transformer le problème en sous-problèmes plus simples. Ensuite, nous pouvons les résoudre avec des méthodes d'optimisation classiques. Ji et Guo [192] proposent une allocation de ressources pour deux utilisateurs, l'un loin et l'autre près du serveur MEC. En mode relais, où l'utilisateur le plus proche sert de relais entre le serveur MEC et l'utilisateur distant, ils utilisent la méthode de Dinkelbach pour transformer le problème non convexe en un problème convexe. Ensuite, ils le résolvent avec des méthodes classiques d'optimisation convexe. Wang et al. [180] proposent une stratégie d'allocation de ressources avec des files d'attente en tandem à deux étages pour maximiser les revenus du réseau. La première file d'attente est destinée à la transmission de paquets via la station de base et la deuxième au traitement informatique au serveur MEC. Le problème NP-difficile initial est décomposé en sous-problèmes convexes à l'aide d'un algorithme basé sur la méthode des multiplicateurs de Lagrange alternés (ADMM). L'ADMM est un algorithme permettant de résoudre des problèmes avec une fonction objective fractionnable. Il est adapté aux systèmes décentralisés en raison de sa décomposabilité et nécessite quelques itérations pour converger avec une précision modérée [207]. Cependant, il est lent à converger pour une grande précision. Yang et al. [193] traitent le problème d'allocation de tâches pour les cloudlets avec une décomposition de Bender pour minimiser la consommation énergétique globale. Zhang et al. [164] utilisent une décomposition de Benders généralisée modifiée pour les services sensibles à la latence avec mise en cache pour minimiser la latence globale. Ils résolvent également le problème avec une méthode de branchement et de bornage qui a une complexité de calcul exponentielle. Wang et al. [208] introduisent MOERA, un algorithme d'allocation de ressources en ligne visant à minimiser les coûts opérationnels arbitraires et les coûts qui réduisent la qualité de service (par exemple, le délai) et tiennent compte de la mobilité de l'utilisateur sans leur connaissance préalable. Ils utilisent une

technique de régularisation [209] pour diviser en sous-problèmes. Wang et al. [196] envisagent des systèmes MEC ayant un état non bloquant et un état bloquant lorsque trop de données s'accumulent dans les tampons d'un serveur. Pour l'état non bloquant, ils divisent les problèmes en affectation de tâches et en allocation de ressources avec l'inégalité de Cauchy-Schwarz. Pour l'état bloquant, ils cherchent à rétablir rapidement l'état non bloquant en équilibrant les ressources de transmission et de calcul entre les couches. Lyu et al. [189] abordent l'admission de tâches et l'allocation de ressources en minimisant la consommation d'énergie avec leur schéma EROS. Le problème initial est simplifié en un problème de programmation entière en pré-admettant des tâches qui doivent être déchargées pour respecter leurs délais. Ensuite, il est résolu avec un algorithme de programmation dynamique quantifié. Haber et al. [203] fournissent un schéma d'allocation de ressources pour le MEC assisté par UAV, en tenant compte de la position et de la fiabilité de l'UAV.

Théorie des jeux

Les méthodes de la théorie des jeux sont adaptées aux systèmes où chaque nœud a ses propres intérêts individuels. Par exemple, lorsqu'il y a un fournisseur de services cherchant à maximiser ses revenus et des appareils autonomes, chacun souhaitant accomplir ses tâches le plus rapidement possible. Ces méthodes peuvent proposer un consensus dans de tels systèmes de manière décentralisée. Josilo et Dán [170] fournissent un modèle d'allocation de ressources où les fournisseurs de services de bord et les appareils interagissent comme un jeu de Stackelberg. Les appareils sont les leaders et souhaitent minimiser le temps de réalisation de leurs tâches en choisissant à quel serveur de bord ils déchargent leurs tâches et par quel point d'accès. Sardellitti et al. [168] utilisent la théorie des appariements pour attribuer des utilisateurs à un serveur MEC et à leurs ressources de communication et de calcul, en fonction des préférences des utilisateurs.

Méthodes d'apprentissage automatiques

Les méthodes d'apprentissage apprennent à partir du passé et/ou de l'environnement. Elles sont plus rapides que les méthodes classiques, mais peuvent être moins précises. Chacune possède ses propres avantages ou inconvénients.

Calcul évolutif (EA) : L'EA s'inspire de la biologie. De nombreux algorithmes existent sous l'EA et sont plus ou moins adaptés à certains problèmes, avec leurs propres avantages et inconvénients. Par exemple, les algorithmes génétiques ont tendance à ne pas rester piégés dans des optima locaux [210, 211], mais ils peuvent être difficiles à ajuster pour résoudre des problèmes spécifiques. Ainsi, Wan et al. [211] proposent une utilisation différente de l'EA pour l'assignation de ressources basée sur les tâches, y compris l'hybridation de différents algorithmes EA. Li et al. [210] utilisent un algorithme génétique pour minimiser le temps de réalisation des tâches pour les appareils mobiles et un serveur de bord.

Apprentissage par renforcement : Les schémas d'allocation de ressources peuvent utiliser une méthode d'apprentissage par renforcement, plus

[209]: Niv BUCHBINDER, Shahar CHEN et Joseph (Seffi) NAOR. « Competitive Analysis via Regularization ». In : *Proceedings of the 2014 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. eprint : <https://epubs.siam.org/doi/pdf/10.1137/1.9781611973402.32>

[196]: Pengfei WANG, Chao YAO, Zijie ZHENG, Guangyu SUN et Lingyang SONG. « Joint Task Assignment, Transmission, and Computing Resource Allocation in Multilayer Mobile Edge Computing Systems ». In : *IEEE Internet of Things Journal* 6.2 (2019)

[189]: Xinchun LYU, Hui TIAN, Wei NI, Yan ZHANG, Ping ZHANG et Ren Ping LIU. « Energy-Efficient Admission of Delay-Sensitive Tasks for Mobile Edge Computing ». In : *IEEE Transactions on Communications* 66.6 (2018)

[203]: Elie El HABER, Hyame Assem ALAMEDDINE, Chadi ASSI et Sanaa SHARAFEDDINE. « UAV-Aided Ultra-Reliable Low-Latency Computation Offloading in Future IoT Networks ». In : *IEEE Transactions on Communications* 69.10 (2021)

[170]: Sladana JOSILO et Gyorgy DAN. « Joint Allocation of Computing and Wireless Resources to Autonomous Devices in Mobile Edge Computing ». In : *Proceedings of the 2018 Workshop on Mobile Edge Communications. MECOMM'18*. Budapest, Hungary : Association for Computing Machinery, 2018. ISBN : 9781450359061

[168]: Stefania SARDELLITTI, Mattia MERLUZZI et Sergio BARBAROSSA. « Optimal Association of Mobile Users to Multi-Access Edge Computing Resources ». In : *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*. 2018

[210]: Zhi LI et Qi ZHU. « Genetic Algorithm-Based Optimization of Offloading and Resource Allocation in Mobile-Edge Computing ». In : *Information* 11.2 (fév. 2020)

[211]: Liangtian WAN, Lu SUN, Xiangjie KONG, Yuyuan YUAN, Ke SUN et Feng XIA. « Task-Driven Resource Assignment in Mobile Edge Computing Exploiting Evolutionary Computation ». In : *IEEE Wireless Communications* 26.6 (2019)

[212]: Sihua WANG, Mingzhe CHEN, Xuanlin LIU, Changchuan YIN, Shuguang CUI et H. VINCENT POOR. « A Machine Learning Approach for Task and Resource Allocation in Mobile-Edge Computing-Based Networks ». In : *IEEE Internet of Things Journal* 8.3 (2021)

[166]: Linpei LI, Xiangming WEN, Zhaoming LU, Qi PAN, Wenpeng JING et Zhiqun HU. « Energy-Efficient UAV-Enabled MEC System : Bits Allocation Optimization and Trajectory Design ». In : *Sensors (Basel, Switzerland)* 19 (2019)

[183]: Yuan LIU, Ke XIONG, Qiang NI, Pingyi FAN et Khaled Ben LETAIEF. « UAV-Assisted Wireless Powered Cooperative Mobile Edge Computing : Joint Offloading, CPU Control, and Trajectory Optimization ». In : *IEEE Internet of Things Journal* 7.4 (2020)

[185]: Tiankui ZHANG, Yu XU, Jonathan LOO, Dingcheng YANG et Lin XIAO. « Joint Computation and Communication Design for UAV-Assisted Mobile Edge Computing in IoT ». In : *IEEE Transactions on Industrial Informatics* 16.8 (2020)

spécifiquement une méthode de *Q-learning*. Le *Q-learning* a l'avantage d'être sans modèle et de s'adapter à un environnement stochastique, ce qui en fait une solution pour les contextes dynamiques. De plus, on peut le régler pour prendre des décisions à court ou à long terme. Wang et al. [212] proposent un algorithme d'apprentissage par renforcement à plusieurs piles pour l'allocation de ressources dans le calcul mobile en périphérie. Ils utilisent plusieurs piles pour tirer parti d'un schéma d'allocation de ressources historique et éviter d'apprendre le même schéma. Cependant, un inconvénient de l'apprentissage par renforcement est la table Q. Elle deviendra excessivement grande pour les systèmes à grande échelle en raison de nombreux états possibles différents, rendant son stockage et la recherche de la valeur Q complexe [191, 197]. En alternative, on peut utiliser une méthode d'apprentissage par renforcement profond, avec un réseau de neurones profond pour estimer la valeur Q pour une action et un état donné. Mais on perd les propriétés "sans modèle" du *Q-learning*, et il faut former un modèle. Chen et al. [162] proposent un apprentissage par renforcement profond pour le réseau CoMEC, où des serveurs de périphérie collaboratifs sont connectés. Li et al. [191] utilisent un apprentissage par renforcement profond pour l'allocation de ressources computationnelles d'un serveur MEC à des appareils mobiles en minimisant le délai d'exécution et la consommation d'énergie. Wang et al. [197] introduisent un algorithme d'allocation de ressources basé sur l'apprentissage par renforcement profond pour minimiser le délai de calcul et de routage dans les réseaux de périphérie. Ils tiennent également compte de l'équilibrage de l'allocation de ressources pour réduire la pression localisée sur le réseau et améliorer les délais.

Réseau de neurones profonds : Li et Lv [191] utilisent un réseau de neurones profonds (DNN) pour l'allocation de ressources afin de minimiser la consommation d'énergie du réseau. Ils forment des DNN pour simuler le comportement d'un algorithme de programmation quadratique séquentielle. Ils forment un DNN avec un nombre fixe d'appareils dans l'ensemble de données et un autre avec un nombre aléatoire d'appareils, ce qui rend le second plus flexible que le premier. Ainsi, le DNN mettra moins de temps à résoudre le problème avec une approximation du résultat optimal. Cependant, l'environnement est très dynamique et comporte de nombreuses incertitudes. Un ensemble de formation peut être sous-représentatif du système complexe et le DNN formé n'est pas suffisamment flexible pour faire face à certaines situations, car il n'adapte pas en cours d'exécution [197]. De plus, il peut être difficile de trouver de bonnes données à l'avance.

5.3 Déploiement de ressources mobiles

Pour le déploiement d'une ressource mobile, nous devons optimiser sa trajectoire entre un point de départ et un point d'arrivée pour desservir les appareils mobiles en minimisant le délai [213] ou la consommation d'énergie du système [166, 183, 185]. Pour le déploiement de plusieurs ressources mobiles, nous devons optimiser leur nombre, c'est-à-dire minimiser leur nombre tout en satisfaisant l'objectif, leurs emplacements et les associer aux utilisateurs mobiles. En effet, avec plusieurs ressources mobiles, nous n'avons pas de point de départ et d'arrivée, nous ne

pouvons donc pas planifier l'ensemble de la trajectoire, mais plutôt calculer le prochain point de localisation. Les objectifs peuvent être la minimisation de la consommation d'énergie [214], la minimisation du nombre de nœuds déployés [169] ou l'équilibrage de la charge de travail entre les ressources [215]. De plus, le schéma de déploiement est souvent lié à un autre problème : la planification des tâches, la décision de déchargement, l'optimisation du processeur, c'est-à-dire la quantité de CPU dont une tâche a besoin, l'allocation des ressources ou des bits.

5.3.1 Comment modéliser un système avec des ressources mobiles

Déploiement d'un UAV : Généralement, lorsqu'on considère le déploiement d'un seul UAV, on suppose qu'il commence et termine sa trajectoire à des emplacements prédéfinis. Ainsi, l'UAV effectue des cycles au cours desquels les appareils peuvent décharger leurs tâches [182, 183, 185, 216, 217]. Le problème consiste alors à étudier la planification de la trajectoire dans ces cycles. Le cycle est divisé en créneaux horaires, où l'UAV est considéré comme statique, de même que les appareils lorsqu'ils sont mobiles [182, 183, 185]. En général, dans ces systèmes, la zone couverte n'est pas très étendue [166], et donc ces travaux sont adaptés au déploiement à court terme et aux applications à petite échelle, ou pour aider les serveurs fixes dans de courtes distances.

Déploiement de plusieurs UAVs : Le déploiement de plusieurs UAVs peut couvrir de vastes zones et être utilisé dans des applications à grande échelle. C'est un défi complexe qui est fortement lié au schéma d'allocation de ressources car ils dépendent les uns des autres. Les recherches précédentes examinent différents scénarios pour le déploiement de plusieurs UAVs. [218] et [219] supposent un système MEC à trois couches, avec une couche d'appareils, une couche d'UAVs et des serveurs MEC terrestres fixes. Islambouli et Sharafeddine [169] étudient le déploiement d'essaims d'UAVs, certains UAVs agissant comme relais pour le déchargement multi-saut lorsque la puissance de transmission est trop faible. D'autres travaux considèrent le déploiement des UAVs conjointement avec d'autres problèmes comme la planification des tâches [214], l'association des utilisateurs et l'allocation des ressources [173].

5.3.2 Méthodes pour le déploiement

Comme l'allocation de ressources, le déploiement des UAV est souvent associé à un problème conjoint. Dans ces cas, le problème devient trop complexe pour être résolu directement. Par conséquent, les travaux ont tendance à décomposer le problème initial en sous-problèmes et à les résoudre de manière itérative, où la partie déploiement est résolue avec les résultats des problèmes conjoints précédemment résolus [173, 183, 216]. Dans les sous-sections suivantes, lorsque les auteurs utilisent des algorithmes itératifs, nous nous concentrerons sur la partie déploiement.

[182]: Jingyu XIONG, Hongzhi GUO et Jiajia LIU. « Task Offloading in UAV-Aided Edge Computing : Bit Allocation and Trajectory Optimization ». In : *IEEE Communications Letters* 23.3 (2019)

[183]: Yuan LIU, Ke XIONG, Qiang Ni, Pingyi FAN et Khaled Ben LETAIEF. « UAV-Assisted Wireless Powered Cooperative Mobile Edge Computing : Joint Offloading, CPU Control, and Trajectory Optimization ». In : *IEEE Internet of Things Journal* 7.4 (2020)

[185]: Tiankui ZHANG, Yu XU, Jonathan LOO, Dingcheng YANG et Lin XIAO. « Joint Computation and Communication Design for UAV-Assisted Mobile Edge Computing in IoT ». In : *IEEE Transactions on Industrial Informatics* 16.8 (2020)

[216]: Mushu LI, Nan CHENG, Jie GAO, Yinlu WANG, Lian ZHAO et Xuemin SHEN. « Energy-Efficient UAV-Assisted Mobile Edge Computing : Resource Allocation and Trajectory Optimization ». In : *IEEE Transactions on Vehicular Technology* 69.3 (2020)

[217]: Yuwen QIAN, Feifei WANG, Jun LI, Long SHI, Kui CAI et Feng SHU. « User Association and Path Planning for UAV-Aided Mobile Edge Computing With Energy Restriction ». In : *IEEE Wireless Communications Letters* 8.5 (2019)

[218]: Shuo WAN, Jiaxun LU, Pingyi FAN et Khaled B. LETAIEF. « Toward Big Data Processing in IoT : Path Planning and Resource Management of UAV Base Stations in Mobile-Edge Computing System ». In : *IEEE Internet of Things Journal* 7.7 (2020)

[173]: Jiao ZHANG, Li ZHOU, Fuhui ZHOU, Boon-Chong SEET, Haijun ZHANG, Zhiping CAI et Jibo WEI. « Computation-Efficient Offloading and Trajectory Scheduling for Multi-UAV Assisted Mobile Edge Computing ». In : *IEEE Transactions on Vehicular Technology* 69.2 (2020)

[182]: Jingyu XIONG, Hongzhi GUO et Jiajia LIU. « Task Offloading in UAV-Aided Edge Computing : Bit Allocation and Trajectory Optimization ». In : *IEEE Communications Letters* 23.3 (2019)

[220]: Meisam RAZAVIYAYN, Mingyi HONG, Zhi-Quan Tom LUO et J. S. PANG. « Parallel Successive Convex Approximation for Nonsmooth Nonconvex Optimization ». In : *NIPS*. 2014

[173]: Jiao ZHANG, Li ZHOU, Fuhui ZHOU, Boon-Chong SEET, Haijun ZHANG, Zhiping CAI et Jibo WEI. « Computation-Efficient Offloading and Trajectory Scheduling for Multi-UAV Assisted Mobile Edge Computing ». In : *IEEE Transactions on Vehicular Technology* 69.2 (2020)

[183]: Yuan LIU, Ke XIONG, Qiang NI, Pingyi FAN et Khaled Ben LETAIEF. « UAV-Assisted Wireless Powered Cooperative Mobile Edge Computing : Joint Offloading, CPU Control, and Trajectory Optimization ». In : *IEEE Internet of Things Journal* 7.4 (2020)

[185]: Tiankui ZHANG, Yu XU, Jonathan LOO, Dingcheng YANG et Lin XIAO. « Joint Computation and Communication Design for UAV-Assisted Mobile Edge Computing in IoT ». In : *IEEE Transactions on Industrial Informatics* 16.8 (2020)

[221]: Fuhui ZHOU, Yongpeng WU, Rose Qingyang HU et Yi QIAN. « Computation Rate Maximization in UAV-Enabled Wireless-Powered Mobile-Edge Computing Systems ». In : *IEEE Journal on Selected Areas in Communications* 36.9 (2018)

[222]: Dorit S. HOCHBAUM et Anu PATHRIA. « Analysis of the greedy approach in problems of maximum k-coverage ». In : *Naval Research Logistics (NRL)* 45.6 (1998). eprint : <https://onlinelibrary.wiley.com/doi/pdf/10.1002/%28SICI%291520-6750%28199809%2945%3A6%3C615%3A%3AAID-NAV5%3E3.0.CO%3B2-5>

Optimisation convexe

L'optimisation convexe permet de trouver une solution optimale à un problème relativement simple. Elle peut être suffisante pour un problème centralisé impliquant un seul UAV, mais pas pour un problème plus complexe et distribué impliquant plusieurs UAVs. Xiong et al. [182] utilisent le solveur CVX pour résoudre une trajectoire d'UAV ainsi que l'allocation de ressources et d'octets. Li et al. [166] proposent également une fonction convexe pouvant être résolue par un solveur CVX dans leur algorithme alternatif à deux étapes pour la trajectoire des UAVs et l'allocation de bits.

Optimisation convexe successive (SCA)

Les problèmes d'optimisation non convexe sont fréquents dans le contexte de la MEC activée par des UAV en raison de nombreuses contraintes et paramètres. Ainsi, la méthode d'Optimisation Convexe Successive (SCA) résout ces problèmes en les approximant de manière itérative sous forme de problèmes convexes [220]. Cette méthode produira une solution optimale locale de manière parallèle et distribuée. Certains travaux utilisent la méthode SCA pour résoudre des problèmes de trajectoire d'UAV [173, 183, 185, 221] et de position d'UAV [184, 203]. Cependant, l'optimiseur résultant peut avoir une complexité de calcul élevée et ne répond pas aux exigences en temps réel du système [183].

Algorithmes gloutons

Les algorithmes gloutons sont des solutions heuristiques bien connues pour les problèmes de couverture [222], comme dans le déploiement d'UAV. Ils proposent une bonne estimation de la solution optimale globale pour des problèmes complexes. Chen et al. [223] utilisent un algorithme glouton pour déployer des UAVs à des emplacements et associer les tâches de leurs appareils afin de maximiser les tâches déchargées. Wang et al. [224] utilisent un algorithme glouton pour dispatcher des UAVs, en tenant compte des points chauds des utilisateurs, afin de maximiser le nombre de tâches traitées.

Méta-heuristiques basées sur la population

Les méta-heuristiques basées sur la population recherchent les meilleures solutions parmi un ensemble de solutions candidates. Elles commencent avec une population aléatoire de solutions, puis fusionnent, conservent ou éliminent chaque solution à chaque itération pour obtenir la plus adaptée. Elles ont l'avantage d'éviter les optima locaux [225] au prix d'une complexité plus élevée qu'une méthode d'optimisation classique. Par conséquent, il peut être difficile de les utiliser pour des solutions en ligne. De plus, chaque algorithme possède ses propres avantages et inconvénients.

Calcul évolutif : Wang et al. [214] utilisent un algorithme de Differential Evolution (DE) pour décider de l'emplacement des UAVs. Leur problème comporte des variables de décision mixtes et est de longueur variable,

ce qui pose un problème pour utiliser efficacement un algorithme DE. Ils proposent donc un nouvel encodage où chaque UAV est un individu et la population est une solution de déploiement. Yang et al. [215] utilisent également un DE pour déployer des UAVs à un emplacement afin d'équilibrer la charge de travail entre eux et d'éviter les goulots d'étranglement dans le réseau.

Optimisation du mouvement des ions : Islambouli et Sharafeddine [169] utilisent l'optimisation du mouvement des ions [225] pour choisir le nombre d'UAVs et leurs positions, ainsi que les associations d'appareils et les allocations de calcul. L'algorithme modélise la population de solutions possibles comme des anions et des cations et choisit de manière itérative une solution efficace. Le travail [225] montre que l'optimisation du mouvement des ions a tendance à éviter les optima locaux et nécessite peu de paramètres d'ajustement par rapport à d'autres algorithmes basés sur la population.

[225]: Behzad JAVIDY, Abdolreza HATAMLOU et Seyedali MIRJALILI. « Ions motion algorithm for solving optimization problems ». In : *Applied Soft Computing* 32 (2015)

5.4 Conclusion

Nous avons examiné les travaux sur l'allocation de ressources en MEC, en mettant en évidence leur modélisation, leurs objectifs et leurs méthodes, et nous avons fait de même pour les schémas de déploiement de ressources mobiles. Avec ce travail, nous espérons aider les chercheurs à concevoir des schémas de gestion des ressources en MEC qui conviennent aux applications hautement dynamiques, telles que les applications critiques et sensibles au temps, et à exploiter pleinement le potentiel de l'informatique en périphérie. De plus, nous avons noté un manque d'utilisation d'outils de prédiction dans ces travaux. Les différents outils proposés dans cette thèse peuvent donc pleinement servir à un schéma de gestion de ressources MEC qui prédit en avance la charge de trafic changeante et les conditions du canal pour anticiper la chute de la QoS. Pour le déploiement des ressources mobiles, la prédiction de la mobilité pourra aider à anticiper la demande en temps et en espace.

Conclusion générale

6

6.1 Récapitulatif des travaux

Dans le Chapitre 1, nous avons exploré l'informatique en périphérie multiaccès en tant que complément à l'informatique en nuage, qui offre des services proches des utilisateurs et des objets, tout en réduisant la latence. Cependant, cette proximité implique que les ressources ne résident pas dans de vastes centres de données, mais sont distribuées sur des nœuds plus dispersés, ce qui entraîne des ressources de calcul et de communication considérablement plus limitées. Cette situation pose de nouveaux défis pour optimiser l'utilisation de ces ressources. Nous avons également exploré comment cette proximité permet l'utilisation d'unités mobiles telles que les UAV ou les robots terrestres pour transporter les ressources MEC, offrant ainsi des services plus flexibles. Le déploiement optimal de ces ressources MEC mobiles nécessite une attention particulière.

Ces défis peuvent être abordés en prévoyant les besoins et l'activité du réseau afin d'anticiper le déploiement des ressources. La prédiction de la qualité de service (QoS) permet d'optimiser l'utilisation des ressources en évitant à la fois la surutilisation et la sous-utilisation. De plus, la mobilité des foules revêt une importance cruciale pour déployer les ressources mobiles de manière optimale.

Outils pour la génération des données

C'est pourquoi, dans cette thèse, nous avons abordé la question des données dans le Chapitre 2, car elles constituent un élément essentiel des algorithmes de prédiction. Nous avons constaté qu'il n'existait pas de données correspondant à nos besoins pour les algorithmes de prédiction. Par conséquent, nous avons commencé par générer nos propres données IoT, qui représentent la nature incertaine, hétérogène et dynamique des réseaux IoT. Dans un souci de promouvoir la science ouverte, ces données et les méthodes de leur génération sont mises à disposition du public. Cette démarche nous a permis de constater que la génération de données est un élément central dans le domaine des réseaux sans fil, et qu'il n'existe pas d'outil complet pour générer automatiquement et de manière reproductible de telles données. Nous proposons donc un tel outil, basé sur des principes de conception garantissant la reproductibilité des traces. Nous présentons également des données générées, enrichies par rapport aux données précédentes, grâce à cet outil, introduisant ainsi des niveaux de complexité supplémentaires. Enfin, nous présentons un troisième jeu de données incluant des points d'accès mobiles, et mis à disposition publiquement comme les autres données et outils.

6.1	Récapitulatif des travaux	89
6.2	Discussion des résultats	90
6.3	Perspectives et travaux futurs	92
6.3.1	Travaux futurs	92
6.3.2	Les perspectives de re- cherche dans les réseaux MEC avec unités mobiles	93

Prédiction des besoins

Ces données ont été générées dans le but de développer des algorithmes de **prédiction de la QoS**, présentés dans le Chapitre 3. Dans un premier temps, nous avons comparé des algorithmes de prédiction du débit dans les réseaux IoT en utilisant la régression. Cependant, d'autres métriques peuvent indiquer la QoS d'un réseau, c'est pourquoi nous proposons ensuite un modèle de transformateurs temporels pour la prédiction multi-métrique. Nous évaluons ces performances en le comparant à d'autres modèles d'apprentissage profond. Un autre aspect de la prédiction des besoins concerne la **prédiction de la mobilité des foules**, plus précisément la densité humaine, afin d'aider à la prise de décision pour le déploiement de ressources mobiles. Dans le Chapitre 4, nous présentons un algorithme de prédiction distribué sur des nœuds mobiles à l'aide d'un filtre à particules. Nous évaluons deux aspects de cette distribution : la communication des données et comment la distribution affecte les différents aspects du filtre à particules.

Étude sur les méthodes de déploiement

Pour conclure, en tant que contribution finale de cette thèse, nous avons réalisé une étude sur les travaux concernant l'allocation de ressources MEC, afin de permettre aux chercheurs de concevoir de tels algorithmes d'allocation en utilisant les algorithmes de prédiction présentés précédemment.

6.2 Discussion des résultats

Nous allons dans cette Section approfondir la discussion sur les contributions apportées à cette thèse, en mettant en avant leurs avantages et leurs apports, tout en soulignant également leurs limitations.

Génération de données

Nous avons généré toutes nos données dans une optique de science ouverte, ce qui nous permet de les rendre vérifiables et réutilisables par la communauté. Notre contribution est donc double, car elles bénéficient non seulement à nos propres recherches, mais aussi à d'autres travaux. Par exemple, les jeux de données peuvent être facilement enrichis pour d'autres expériences avec des objectifs différents. Toutes les données ont été générées à partir de bancs d'essai. Les bancs d'essai ont l'avantage d'offrir un environnement contrôlé pour la génération, ce qui permet d'étudier les caractéristiques souhaitées des réseaux. La cohérence des données peut être facilement vérifiée grâce à la reproductibilité des expérimentations et à la génération automatisée des traces.

Cependant, il convient de noter que les données issues de tels bancs d'essai peuvent ne pas refléter la réalité dans toute sa complexité, ce qui peut être plus ou moins gênant selon les objectifs visés. Elles dépendent donc fortement des paramètres expérimentaux. De plus, l'utilisation des bancs d'essai ajoute une couche de complexité supplémentaire

en termes d'apprentissage, car en plus de la programmation, il faut également maîtriser l'utilisation de la plateforme. Toutefois, ce coût initial est compensé par la suite par les possibilités d'automatisation et de simplification de la récupération des résultats. Cette critique s'applique également à l'outil que nous proposons, nommé *Sisyphé*.

Prédiction de la qualité de service

Dans nos premières recherches sur la prédiction de la QoS, nous avons comparé différents modèles de régression pour prédire le débit. Les modèles de régression sont relativement légers et présentent une complexité plutôt simple, ce qui les rend particulièrement adaptés dans le contexte d'un réseau MEC où les ressources sont limitées et précieuses. Nous avons vu que ces modèles présentent de bonnes performances. Cependant, notre analyse s'est concentrée exclusivement sur le débit en tant que métrique de QoS du réseau. La prédiction du débit seul peut ne pas fournir suffisamment d'indications concernant les besoins en ressources du réseau. De plus, les modèles de régression sont sensibles aux données aberrantes, qui peuvent être plus fréquentes dans des conditions réelles, contrairement aux données que nous avons générées et utilisées pour l'apprentissage des modèles.

Ensuite, nous avons présenté un modèle de transformateur temporel pour prédire plusieurs métriques de la QoS et l'avons comparé à d'autres modèles d'apprentissage en profondeur. Les transformateurs temporels, et l'apprentissage en profondeur en général, sont des modèles très performants qui conviennent particulièrement aux séries temporelles, comme c'est le cas dans la prédiction de la QoS. De plus, les transformateurs temporels sont parallélisables, ce qui accélère le traitement des données. Nous avons ainsi pu prédire plusieurs métriques de la QoS, ce qui nous permet d'anticiper plus précisément les besoins en ressources. Un inconvénient de ces modèles d'apprentissage en profondeur est la nécessité d'une grande quantité de données, qui peut être rare, en particulier dans le domaine des réseaux sans fil. Nous avons contourné cette limitation en générant nos propres données en quantité suffisante. De plus, ces modèles sont très complexes et demandent d'importantes ressources de calcul, ce qui peut être adapté pour des serveurs MEC fixes, mais moins pour des ressources plus limitées, voire des ressources mobiles.

Prédiction de la densité humaine

Une telle approche distribuée de la prédiction de la densité sur les nœuds mobiles permet à ces derniers d'être complètement autonomes, même lorsqu'ils sont isolés. L'approche adoptée avec le filtre à particules présente l'avantage de ne pas nécessiter de données historiques sur la mobilité des personnes. De plus, les données restent anonymes, car nous n'avons pas besoin de suivre les individus, mais seulement de prédire leur densité dans une région donnée. Nous obtenons de bonnes performances, suffisantes pour obtenir une vue d'ensemble et ainsi anticiper le déploiement des unités mobiles.

Cependant, notre approche expérimentale est limitée par l'utilisation de données synthétiques et d'un simulateur de mobilité "naïf", sans routes,

à partir duquel les données sont générées, bien que cela soit suffisant pour notre objectif, qui était de tester la distribution du filtre à particules plutôt que sa performance en soi. En ce qui concerne l'algorithme lui-même, la partie communication manque de raffinement lorsqu'il est utilisé dans le cadre d'un réseau MEC ad-hoc. En effet, la sélection des voisins est aléatoire et ne tient pas compte de paramètres qui seraient plus pertinents pour cette sélection, tels que la qualité de la liaison. De plus, nous évaluons séparément les deux aspects de la distribution. Nous avons fait ce choix pour avoir des simulateurs complets et fiables, afin d'étudier initialement la faisabilité de la distribution. Cependant, il sera nécessaire à terme de tester l'algorithme dans son intégralité au sein d'un simulateur robotique qui inclut également la simulation de la communication.

6.3 Perspectives et travaux futurs

6.3.1 Travaux futurs

Les travaux que nous avons proposés peuvent non seulement être approfondis, mais aussi servir de base pour de nouvelles recherches à l'avenir.

En effet, une perspective naturelle de ces recherches est de concevoir des mécanismes d'allocation de ressources qui utilisent les prédictions établies dans cette thèse. Nous nous intéressons particulièrement à un modèle de déploiement adaptatif et autonome d'unités mobiles dans le cadre du MEC. En nous appuyant sur notre étude du déploiement d'unités mobiles, nous constatons qu'il existe peu de travaux proposant un déploiement à grande échelle de plusieurs unités mobiles, et à notre connaissance, aucun d'entre eux n'utilise des modèles de prévision des besoins. Nous pensons que le Q-learning est une piste prometteuse pour ce type de déploiement, car il s'adapte efficacement à un système distribué et à un environnement changeant. De plus, il présente l'avantage d'être léger en termes de calculs.

Nous avons également pour ambition de développer et enrichir davantage l'outil *Sisyphé* afin de le rendre plus simple d'utilisation pour les expérimentateurs et de réduire au maximum la courbe d'apprentissage. Cela implique notamment l'ajout de nouvelles données provenant d'autres bancs d'essai, tels que Grid-5000, pour élargir l'utilisation de l'outil à d'autres communautés, au-delà de la seule communauté des réseaux sans fil.

Un autre aspect qui suscite fortement notre intérêt est la conception d'un protocole de comméragé adapté aux réseaux ad-hoc MEC. Les protocoles de comméragé ont été initialement conçus pour les réseaux filaires et n'ont pas tenu compte des caractéristiques mobiles et de la dynamique des réseaux, caractérisés par des liaisons de communication incertaines. Il existe déjà des travaux sur les algorithmes de communication dans les réseaux mobiles ad hoc (MANET) mais qui ne s'appliquent pas au contexte particulier du MEC. Nous envisageons donc d'approfondir cette dimension de recherche.

6.3.2 Les perspectives de recherche dans les réseaux MEC avec unités mobiles

Déploiement avec plusieurs unités mobiles et plusieurs utilisateurs

Les stratégies de déploiement et de gestion de multiples ressources mobiles demeurent un défi majeur au sein d'un environnement comportant de nombreux utilisateurs mobiles. Jusqu'à présent, la recherche s'est principalement concentrée sur le déploiement d'une seule unité ou un scénario avec un seul utilisateur, mais les déploiements impliquant plusieurs unités mobiles et utilisateurs présentent des défis bien plus complexes, comme la gestion des délais de déplacement des unités mobiles, qui peuvent être imprévisibles et influencer considérablement la performance du réseau. Les trajectoires de ces unités mobiles doivent être prises en compte pour optimiser leur utilisation tout en minimisant les interférences et en garantissant une Qualité de Service adéquate. La durée de vie des batteries des unités mobiles est un autre facteur critique. De plus, la mobilité très dynamique d'un grand nombre d'utilisateurs complique davantage la gestion du réseau. Dans ce contexte, l'optimisation devient un défi multidimensionnel, car il faut souvent jongler avec des objectifs contradictoires. Par exemple, maximiser la couverture pour offrir un service de qualité aux utilisateurs tout en préservant la durée de vie des batteries des unités mobiles est un dilemme fréquent. Ces objectifs contradictoires conduisent à des problèmes d'optimisation non convexes, ce qui signifie que les solutions ne peuvent pas être trouvées par des méthodes mathématiques classiques, rendant nécessaire le développement de techniques d'optimisation avancées.

Sécurité et confidentialité des données

La mobilité des unités MEC les rend très vulnérables aux attaques, en particulier les attaques pour épuiser leurs batteries et les empêcher d'accomplir leurs missions. Ces unités doivent se déplacer au sein d'un environnement vaste et non contrôlé, avec la présence potentielle de unités mobiles "malveillantes" qui usurpent l'identité (*spoofing*) d'un drone du réseau ou encore d'un centre de données MEC où le drone souhaite décharger ses données ou ses tâches. Les communications sans fil sont intrinsèquement plus exposées aux attaques, et les unités mobiles peuvent passer beaucoup de temps à communiquer pendant leurs missions, ce qui augmente les opportunités d'écoutes clandestines (*eavesdropping*) par des attaquants. La confidentialité des données doit également retenir l'attention des chercheurs, car cela peut poser problème lorsque les unités mobiles peuvent couvrir de vastes zones et ainsi collecter différents types de données sensibles. Ces données peuvent potentiellement être échangées ou déchargées à plusieurs intervalles dans le réseau à l'aide de connexions sans-fil, augmentant ainsi les risques d'attaques. Cela représente d'autant plus un défi que ces unités sont limitées en ressources et ne peuvent pas recourir à des solutions cryptographiques, de sécurité et de confidentialité lourdes. Les protocoles de sécurité et de confidentialité doivent tenir compte de la nature très hétérogène de ces réseaux MEC, avec des unités mobiles composées de couches logicielles et matérielles pouvant devenir très complexes.

Intégration aux réseaux et technologies du futur

Avec l'avènement de la 5G et déjà les préparatifs pour la 6G, l'intégration des réseaux MEC avec des unités mobiles devient cruciale pour suivre les avancées des objets intelligents et/ou communicants, ainsi que des réseaux MEC fixes. Il est également important de tenir compte de l'intégration de plusieurs technologies émergentes pour relever les défis des réseaux MEC mobiles. La communication par lumière visible, en complément de la communication par radiofréquence, représente une voie très prometteuse, car elle ne dépend pas des ressources limitées du spectre et offre des débits élevés. Des recherches sont déjà en cours pour exploiter cette forme de communication tout en palliant ses défauts. NOMA (*Non-Orthogonal Multiple Access*) est une technologie d'avenir bénéfique pour les réseaux MEC avec des unités mobiles, améliorant l'efficacité spectrale, la gestion de l'accès multiple et la qualité de service. Cela permet une meilleure utilisation des ressources et la prise en charge de l'IoT dans des environnements exigeants. Les réseaux radio cognitifs, proposant une optimisation de l'utilisation du spectre de fréquences radio disponible, suscitent un grand intérêt au sein de la communauté de la recherche. Leur intégration dans les réseaux MEC avec des unités mobiles est essentielle pour permettre la communication avec les opérateurs mobiles de demain. En résumé, l'évolution vers la 5G et au-delà, la diversification des technologies de communication, telles que la communication par lumière visible et NOMA, ainsi que l'intégration des réseaux radio cognitifs, joueront un rôle essentiel dans l'amélioration de l'efficacité des réseaux MEC mobiles et la prise en charge des besoins de demain.

En conclusion, les réseaux MEC avec des unités mobiles sont très complexes, et les défis sont nombreux en raison de leur caractère récent. Cette technologie est en constante évolution en tandem avec les innovations technologiques, qu'elles soient matérielles ou logicielles, à travers toutes les couches des réseaux. Les perspectives d'évolution sont également considérables et exigent une attention soutenue de la recherche en vue de la rendre parfaitement mature pour des cas d'utilisation à grande échelle, tels que la gestion des catastrophes, tout en assurant une intégration parfaite dans son environnement.

Bibliographie

- [1] John R. BARRY. *Wireless Infrared Communications*. 1^{re} éd. The Springer International Series in Engineering and Computer Science. New York, NY : Springer, 1994, p. X, 181 (cf. p. 1).
- [2] T. KOMINE et M. NAKAGAWA. « Fundamental analysis for visible-light communication system using LED lights ». In : *IEEE Transactions on Consumer Electronics* 50.1 (2004), p. 100-107. DOI : [10.1109/TCE.2004.1277847](https://doi.org/10.1109/TCE.2004.1277847) (cf. p. 1).
- [3] International Telecommunication UNION. *ITU-T Recommendation X.200 : Information Technology - Open Systems Interconnection - Basic Reference Model : The Basic Model*. Recommendation X.200. ITU-T, 1994 (cf. p. 1).
- [4] Li Da XU, Wu HE et Shancang LI. « Internet of Things in Industries : A Survey ». In : *IEEE Transactions on Industrial Informatics* 10.4 (2014), p. 2233-2243 (cf. p. 1).
- [5] Kevin ASHTON. « Internet of Things ». In : *RFID Journal* (juin 2009) (cf. p. 1).
- [6] I.F. AKYILDIZ et al. « Wireless sensor networks : a survey ». In : *Computer Networks* 38.4 (2002), p. 393-422. DOI : [https://doi.org/10.1016/S1389-1286\(01\)00302-4](https://doi.org/10.1016/S1389-1286(01)00302-4) (cf. p. 1).
- [7] Shibo HE et al. « Energy provisioning in wireless rechargeable sensor networks ». In : *2011 Proceedings IEEE INFOCOM*. 2011, p. 2006-2014. DOI : [10.1109/INFCOM.2011.5935007](https://doi.org/10.1109/INFCOM.2011.5935007) (cf. p. 2).
- [8] Khan MUHAMMAD et al. « Secure Surveillance Framework for IoT Systems Using Probabilistic Image Encryption ». In : *IEEE Transactions on Industrial Informatics* 14 (2018), p. 3679-3689 (cf. p. 2).
- [9] Muhammad AYAZ et al. « Internet-of-Things (IoT)-Based Smart Agriculture : Toward Making the Fields Talk ». In : *IEEE Access* 7 (2019), p. 129551-129583 (cf. p. 2).
- [10] Hugh BOYES et al. « The industrial internet of things (IIoT) : An analysis framework ». In : *Computers in Industry* 101 (2018), p. 1-12. DOI : <https://doi.org/10.1016/j.compind.2018.04.015> (cf. p. 2).
- [11] Ammar GHARAIBEH et al. « Smart Cities : A Survey on Data Management, Security, and Enabling Technologies ». In : *IEEE Communications Surveys Tutorials* 19.4 (2017), p. 2456-2501. DOI : [10.1109/COMST.2017.2736886](https://doi.org/10.1109/COMST.2017.2736886) (cf. p. 2).
- [12] Paolo BARONTI et al. « Wireless sensor networks : A survey on the state of the art and the 802.15.4 and Zig-Bee standards ». In : *Computer Communications* 30.7 (2007). *Wired/Wireless Internet Communications*, p. 1655-1695. DOI : <https://doi.org/10.1016/j.comcom.2006.12.020> (cf. p. 2).
- [13] Rashmi Sharan SINHA, Yiqiao WEI et Seung-Hoon HWANG. « A survey on LPWA technology : LoRa and NB-IoT ». In : *ICT Express* 3.1 (2017), p. 14-21. DOI : <https://doi.org/10.1016/j.icte.2017.03.004> (cf. p. 2).
- [14] Emmanuel BACCELLI et al. « RIOT : An Open Source Operating System for Low-End Embedded Devices in the IoT ». In : *IEEE Internet of Things Journal* 5.6 (déc. 2018), p. 4428-4440 (cf. p. 2, 31).
- [15] Urs HUNKELER, Hong Linh TRUONG et Andy STANFORD-CLARK. « MQTT-S — A publish/subscribe protocol for Wireless Sensor Networks ». In : *2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE '08)*. 2008, p. 791-798. DOI : [10.1109/COMSWA.2008.4554519](https://doi.org/10.1109/COMSWA.2008.4554519) (cf. p. 2).
- [16] Pradipta GHOSH et al. « Robotic Wireless Sensor Networks ». In : *Mission-Oriented Sensor Networks and Systems*. 2017 (cf. p. 2).
- [17] L. ROMEO et al. « Automated Deployment of IoT Networks in Outdoor Scenarios using an Unmanned Ground Vehicle ». In : *2020 IEEE International Conference on Industrial Technology (ICIT)*. 2020, p. 369-374. DOI : [10.1109/ICIT45562.2020.9067099](https://doi.org/10.1109/ICIT45562.2020.9067099) (cf. p. 2).
- [18] Rajesh M, Anu GEORGE et Sudarshan T.S.B. « Energy efficient deployment of Wireless Sensor Network by multiple mobile robots ». In : *2015 International Conference on Computing and Network Communications (CoCoNet)*. 2015, p. 72-78. DOI : [10.1109/CoCoNet.2015.7411169](https://doi.org/10.1109/CoCoNet.2015.7411169) (cf. p. 2).

- [19] Xuxun LIU et al. « Quick Convex Hull-Based Rendezvous Planning for Delay-Harsh Mobile Data Gathering in Disjoint Sensor Networks ». In : *IEEE Transactions on Systems, Man, and Cybernetics : Systems* 51.6 (2021), p. 3844-3854. DOI : [10.1109/TSMC.2019.2938790](https://doi.org/10.1109/TSMC.2019.2938790) (cf. p. 2, 3).
- [20] Bala Gangadhara GUTAM et al. « Optimal rendezvous points selection and mobile sink trajectory construction for data collection in WSNs ». In : *Journal of Ambient Intelligence and Humanized Computing* 14 (2021), p. 7147-7158 (cf. p. 2, 3).
- [21] Xiaoguo YE et Weifa LIANG. « Charging utility maximization in wireless rechargeable sensor networks ». In : *Wireless Networks* 23.7 (oct. 2017), p. 2069-2081. DOI : [10.1007/s11276-016-1271-6](https://doi.org/10.1007/s11276-016-1271-6) (cf. p. 2, 3).
- [22] Dionisis KANDRIS et al. « Applications of Wireless Sensor Networks : An Up-to-Date Survey ». In : *Applied System Innovation* (2020) (cf. p. 2, 3).
- [23] Peter MELL et Timothy GRANCE. *The NIST Definition of Cloud Computing*. Rapp. tech. National Institute of Standards et Technology, 2011 (cf. p. 2, 3).
- [24] Motahareh NAZARI JAHANTIGH et al. « Integration of Internet of Things and cloud computing : a systematic survey ». In : *IET Communications* 14.2 (2020), p. 165-176. DOI : <https://doi.org/10.1049/iet-com.2019.0537> (cf. p. 2, 4).
- [25] Christos STERGIU et al. « Secure integration of IoT and Cloud Computing ». In : *Future Generation Computer Systems* 78 (2018), p. 964-975. DOI : <https://doi.org/10.1016/j.future.2016.11.031> (cf. p. 2, 4).
- [26] Alessio BOTTA et al. « Integration of Cloud computing and Internet of Things : A survey ». In : *Future Generation Computer Systems* 56 (2016), p. 684-700. DOI : <https://doi.org/10.1016/j.future.2015.09.021> (cf. p. 2, 4).
- [27] Weisong SHI et al. « Edge Computing : Vision and Challenges ». In : *IEEE Internet of Things Journal* 3.5 (2016), p. 637-646. DOI : [10.1109/JIOT.2016.2579198](https://doi.org/10.1109/JIOT.2016.2579198) (cf. p. 3-5).
- [28] CISCO. *Cisco Annual Internet Report (2018–2023)*. White Paper. Cisco Systems, 2020 (cf. p. 3, 4).
- [29] Yuyi MAO et al. « A Survey on Mobile Edge Computing : The Communication Perspective ». In : *IEEE Communications Surveys Tutorials* 19.4 (2017), p. 2322-2358. DOI : [10.1109/COMST.2017.2745201](https://doi.org/10.1109/COMST.2017.2745201) (cf. p. 3, 4).
- [30] Mahadev SATYANARAYANAN. « The Emergence of Edge Computing ». In : *Computer* 50.1 (2017), p. 30-39. DOI : [10.1109/MC.2017.9](https://doi.org/10.1109/MC.2017.9) (cf. p. 3, 4).
- [31] Ashkan YOUSEFPOUR et al. « All One Needs to Know about Fog Computing and Related Edge Computing Paradigms : A Complete Survey ». In : *ArXiv abs/1808.05283* (2018) (cf. p. 4).
- [32] Mahadev SATYANARAYANAN et al. « The Case for VM-Based Cloudlets in Mobile Computing ». In : *IEEE Pervasive Computing* 8.4 (2009), p. 14-23. DOI : [10.1109/MPRV.2009.82](https://doi.org/10.1109/MPRV.2009.82) (cf. p. 4).
- [33] V. BAHL. *Emergence of Micro Datacenter (Cloudlets/Edges) for Mobile Computing*. 2015. URL : <http://www.microsoft.com/en-us/research/wp-content/uploads/2016/11/Micro-Data-Centers-mDCs-for-Mobile-Computing-1.pdf> (cf. p. 4).
- [34] Y. C. HU et al. *Mobile Edge Computing - A Key Technology Towards 5G*. Rapp. tech. White Paper No. 11. ETSI (European Telecommunications Standards Institute), 2015 (cf. p. 4).
- [35] F. GIUST et al. *MEC Deployments in 4G and Evolution Towards 5G*. Rapp. tech. White Paper No. 24. ETSI (European Telecommunications Standards Institute), 2018 (cf. p. 4, 5).
- [36] Dimitrios DECHOUNIOTIS et al. « Edge Computing Resource Allocation for Dynamic Networks : The DRUID-NET Vision and Perspective ». In : *Sensors* 20.8 (2020). DOI : [10.3390/s20082191](https://doi.org/10.3390/s20082191) (cf. p. 4-6).
- [37] M. ULEMA et al. *Public Safety Technology Gaps and Opportunities*. White Paper. 2021 (cf. p. 5, 6).
- [38] Tarik TALEB et al. « On Multi-Access Edge Computing : A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration ». In : *IEEE Communications Surveys Tutorials* 19.3 (2017), p. 1657-1681. DOI : [10.1109/COMST.2017.2705720](https://doi.org/10.1109/COMST.2017.2705720) (cf. p. 5, 6).
- [39] Muhammad ABRAR et al. « Energy Efficient UAV-Enabled Mobile Edge Computing for IoT Devices : A Review ». In : *IEEE Access* 9 (2021), p. 127779-127798. DOI : [10.1109/ACCESS.2021.3112104](https://doi.org/10.1109/ACCESS.2021.3112104) (cf. p. 5, 6).

- [40] CHIST-ERA consortium. <https://www.chistera.eu/projects/druid-net> (cf. p. 6).
- [41] Jan PLACHY, Zdenek BECVAR et Emilio Calvanese STRINATI. « Dynamic resource allocation exploiting mobility prediction in mobile edge computing ». In : *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. 2016, p. 1-6. DOI : [10.1109/PIMRC.2016.7794955](https://doi.org/10.1109/PIMRC.2016.7794955) (cf. p. 8).
- [42] Nina SANTI et Nathalie MITTON. « A resource management survey for mission critical and time critical applications in multi access edge computing ». In : *ITU Journal on Future and Evolving Technologies*. Volume 2 (2021), Issue 2. 2.2 (nov. 2021) (cf. p. 10).
- [43] Firdose SAEIK et al. « Task Offloading in Edge and Cloud Computing : A Survey on Mathematical, Artificial Intelligence and Control Theory Solutions ». In : *Computer Networks* 195 (août 2021). DOI : [10.1016/j.comnet.2021.108177](https://doi.org/10.1016/j.comnet.2021.108177) (cf. p. 10).
- [44] Nina SANTI et al. « Automated and Reproducible Application Traces Generation for IoT Applications ». In : *Q2SWinet 2021 - 17th ACM Symposium on QoS and Security for Wireless and Mobile Networks*. Alicante, Spain : ACM, nov. 2021, p. 1-8. DOI : [10.1145/3479242.3487321](https://doi.org/10.1145/3479242.3487321) (cf. p. 10).
- [45] Aroosa HAMEED et al. « A Machine Learning Regression approach for Throughput Estimation in an IoT Environment ». In : *iThings-2021 : The 14th IEEE International Conference on Internet of Things*. Melbourne, Australia, déc. 2021 (cf. p. 10).
- [46] Kevin JIOKENG, Nathalie MITTON et Nina SANTI. *Generating Mobility-Aware Traces for IoT Applications*. In : *IEEE Global Communications Conference 2022, Demo sessions*. <https://hal.science/hal-03851704>. 2022. URL : <https://hal.science/hal-03851704> (cf. p. 10).
- [47] Nina SANTI, Brandon FOUBERT et Nathalie MITTON. « Comment générer des traces applicatives avec FIT IoT-LAB pour la science ouverte ». In : *CORES 2021 – 6ème Rencontres Francophones sur la Conception de Protocoles, l'Évaluation de Performance et l'Expérimentation des Réseaux de Communication*. La Rochelle, France, sept. 2021 (cf. p. 10).
- [48] Elike HODO et al. « Threat analysis of IoT networks using artificial neural network intrusion detection system ». In : *2016 International Symposium on Networks, Computers and Communications (ISNCC)*. 2016, p. 1-6. DOI : [10.1109/ISNCC.2016.7746067](https://doi.org/10.1109/ISNCC.2016.7746067) (cf. p. 11).
- [49] A. VERMA et V. RANGA. « Machine Learning Based Intrusion Detection Systems for IoT Applications ». In : *Wireless Personal Communications* 111 (2020), p. 2287-2310. DOI : [10.1007/s11277-019-06986-8](https://doi.org/10.1007/s11277-019-06986-8) (cf. p. 11).
- [50] Abebe Abeshu DIRO et Naveen CHILAMKURTI. « Distributed attack detection scheme using deep learning approach for Internet of Things ». In : *Future Generation Computer Systems* 82 (2018), p. 761-768. DOI : <https://doi.org/10.1016/j.future.2017.08.043> (cf. p. 11).
- [51] Aroosa HAMEED et Aris LEIVADEAS. « IoT Traffic Multi-Classification Using Network and Statistical Features in a Smart Environment ». In : *2020 IEEE 25th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*. 2020, p. 1-7. DOI : [10.1109/CAMAD50429.2020.9209311](https://doi.org/10.1109/CAMAD50429.2020.9209311) (cf. p. 11).
- [52] Tamas PFLANZNER, Zoltan FEHER et Attila KERTESZ. « A Crawling Approach to Facilitate Open IoT Data Archiving and Reuse ». In : *2019 Sixth International Conference on Internet of Things : Systems, Management and Security (IOTSMS)*. 2019, p. 235-242. DOI : [10.1109/IOTSMS48152.2019.8939248](https://doi.org/10.1109/IOTSMS48152.2019.8939248) (cf. p. 12, 21).
- [53] M. KUZMAN et al. « A Testbed and an Experimental Public Dataset for Energy-Harvested IoT Solutions ». In : *IEEE Int. Conference on Industrial Informatics (INDIN)*. 2019 (cf. p. 12, 21).
- [54] Z. XIAO et al. « TrajData : On Vehicle Trajectory Collection With Commodity Plug-and-Play OBU Devices ». In : *IEEE Internet of Things Journal* 7 (2020) (cf. p. 12, 21).
- [55] R. JACOB et al. « Dataset : Wireless Link Quality Estimation on FlockLab - and Beyond ». In : *Proceedings of the 2nd Workshop on Data Acquisition To Analysis*. 2019 (cf. p. 12, 21).
- [56] Nickolaos KORONIOTIS et al. « Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics : Bot-IoT dataset ». In : *Future Generation Computer Systems* 100 (2019), p. 779-796. DOI : <https://doi.org/10.1016/j.future.2019.05.041> (cf. p. 12).

- [57] Abdullah ALSAEDI et al. « TON_IoT Telemetry Dataset : A New Generation Dataset of IoT and IIoT for Data-Driven Intrusion Detection Systems ». In : *IEEE Access* 8 (2020), p. 165130-165150. DOI : [10.1109/ACCESS.2020.3022862](https://doi.org/10.1109/ACCESS.2020.3022862) (cf. p. 12).
- [58] Ivan VACCARI et al. « MQTTset, a New Dataset for Machine Learning Techniques on MQTT ». In : *Sensors* 20.22 (2020). DOI : [10.3390/s20226578](https://doi.org/10.3390/s20226578) (cf. p. 12).
- [59] Mohamed Amine FERRAG et al. « Edge-IIoTset : A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications for Centralized and Federated Learning ». In : *IEEE Access* 10 (2022), p. 40281-40306. DOI : [10.1109/ACCESS.2022.3165809](https://doi.org/10.1109/ACCESS.2022.3165809) (cf. p. 12).
- [60] Sajjad DADKHAH et al. « Towards the Development of a Realistic Multidimensional IoT Profiling Dataset ». In : *2022 19th Annual International Conference on Privacy, Security Trust (PST)*. 2022, p. 1-11. DOI : [10.1109/PST55820.2022.9851966](https://doi.org/10.1109/PST55820.2022.9851966) (cf. p. 12).
- [61] Mohanad SARHAN et al. « NetFlow Datasets for Machine Learning-Based Network Intrusion Detection Systems ». In : (2021), p. 117-135. DOI : [10.1007/978-3-030-72802-1_9](https://doi.org/10.1007/978-3-030-72802-1_9) (cf. p. 12, 13).
- [62] Ninareh MEHRABI et al. « A Survey on Bias and Fairness in Machine Learning ». In : 54.6 (2021). DOI : [10.1145/3457607](https://doi.org/10.1145/3457607) (cf. p. 13).
- [63] *Artifact Review and Badging Version 1.1*. Août 2020. URL : <https://www.acm.org/publications/policies/artifact-review-and-badging-current> (cf. p. 13).
- [64] Cedric ADJIH et al. « FIT IoT-LAB : A Large Scale Open Experimental IoT Testbed ». In : Milan, Italy, déc. 2015 (cf. p. 14, 15).
- [65] *Applications traces*. <https://gitlab.inria.fr/druidnet/application-traces> (cf. p. 14).
- [66] Tanweer ALAM. « Cloud-Based IoT Applications and Their Roles in Smart Cities ». In : *Smart Cities* 4.3 (2021), p. 1196-1219. DOI : [10.3390/smartcities4030064](https://doi.org/10.3390/smartcities4030064) (cf. p. 14).
- [67] Philipp SCHULZ et al. « Latency Critical IoT Applications in 5G : Perspective on the Design of Radio Interface and Network Architecture ». In : *IEEE Communications Magazine* 55.2 (2017), p. 70-78. DOI : [10.1109/MCOM.2017.1600435CM](https://doi.org/10.1109/MCOM.2017.1600435CM) (cf. p. 14).
- [68] Amit Kumar SIKDER et al. « IoT-enabled smart lighting systems for smart cities ». In : *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*. 2018, p. 639-645. DOI : [10.1109/CCWC.2018.8301744](https://doi.org/10.1109/CCWC.2018.8301744) (cf. p. 14).
- [69] *IoT-LAB M3*. <https://www.iot-lab.info/docs/boards/iot-lab-m3/> (cf. p. 16, 31).
- [70] E. CALLAWAY et al. « Home networking with IEEE 802.15.4 : a developing standard for low-rate wireless personal area networks ». In : *IEEE Communications Magazine* 40.8 (2002), p. 70-77. DOI : [10.1109/MCOM.2002.1024418](https://doi.org/10.1109/MCOM.2002.1024418) (cf. p. 16).
- [71] A. DUNKELS, B. GRONVALL et T. VOIGT. « Contiki - a lightweight and flexible operating system for tiny networked sensors ». In : *29th Annual IEEE International Conference on Local Computer Networks*. 2004, p. 455-462. DOI : [10.1109/LCN.2004.38](https://doi.org/10.1109/LCN.2004.38) (cf. p. 16).
- [72] Christian COLLBERG et Todd A. PROEBSTING. « Repeatability in Computer Systems Research ». In : *Commun. ACM* 59.3 (fév. 2016), p. 62-69. DOI : [10.1145/2812803](https://doi.org/10.1145/2812803) (cf. p. 20).
- [73] Roberto Di COSMO, Morane GRUENPETER et Stefano ZACCHIROLI. « Referencing Source Code Artifacts : a Separate Concern in Software Citation ». In : *Computing in Science and Engineering* (déc. 2019), p. 1-9. DOI : [10.1109/MCSE.2019.2963148](https://doi.org/10.1109/MCSE.2019.2963148) (cf. p. 21).
- [74] Roberto Di COSMO et Stefano ZACCHIROLI. « Software Heritage : Why and How to Preserve Software Source Code ». In : *iPRES 2017 - 14th International Conference on Digital Preservation*. Kyoto, Japan, sept. 2017, p. 1-10 (cf. p. 21).
- [75] J. M. PERKEL. « Challenge to Scientists : Does Your Ten-Year-Old Code Still Run ? » In : *Nature* 584.7822 (2020) (cf. p. 22).
- [76] E. DOLSTRA. « The Purely Functional Software Deployment Model ». Thèse de doct. Utrecht University, 2006 (cf. p. 22).

- [77] Pierre BRUNISHOLZ et al. « WalT : A reproducible testbed for reproducible network experiments ». In : *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 2016, p. 146-151. DOI : [10.1109/INFOCOMW.2016.7562062](https://doi.org/10.1109/INFOCOMW.2016.7562062) (cf. p. 22).
- [78] E. DOLSTRA. *Nix Flakes, Part 1 : An Introduction and Tutorial*. Tweag. URL : <https://www.tweag.io/blog/2020-05-25-flakes/> (cf. p. 23).
- [79] *ZSTD compression algorithm*. URL : <https://github.com/facebook/zstd> (cf. p. 26).
- [80] *Sisyphé database schema documentation*. URL : https://github.com/sisyphé-re/riot_rpl_udp_scenarios (cf. p. 26).
- [81] *TurtleBot 2*. URL : <https://www.turtlebot.com/turtlebot2/> (cf. p. 31, 32).
- [82] Morgan QUIGLEY et al. « ROS : an open-source Robot Operating System ». In : *ICRA workshop on open source software*. T. 3. 3.2. 2009, p. 5 (cf. p. 31).
- [83] *Applications traces*. URL : <https://gitlab.inria.fr/druidnet/iot-mobility-traces/> (cf. p. 33).
- [84] ITU-R. *IMT vision - Framework and overall objectives of the future development of IMT for 2020 and beyond*. 2015. URL : <http://www.itu.int/rec/R-REC-M.2083-0-201509-I> (cf. p. 35).
- [85] 3GPP. *Study on enhancement of Ultra-Reliable Low-Latency Communication (URLLC) support in the 5G Core network (5GC)*. Technical report (TR) 23.725. 3rd Generation Partnership Project (3GPP), 2018 (cf. p. 35).
- [86] NOKIA. *5G use cases and requirements*. White paper. 2016 (cf. p. 35).
- [87] AROOSA HAMEED et Aris LEIVADEAS. « IoT Traffic Multi-Classification Using Network and Statistical Features in a Smart Environment ». In : *2020 IEEE 25th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*. 2020, p. 1-7. DOI : [10.1109/CAMAD50429.2020.9209311](https://doi.org/10.1109/CAMAD50429.2020.9209311) (cf. p. 35).
- [88] Adrian PEKAR et al. « Application Domain-Based Overview of IoT Network Traffic Characteristics ». In : *ACM Comput. Surv.* 53.4 (juill. 2020). DOI : [10.1145/3399669](https://doi.org/10.1145/3399669) (cf. p. 35).
- [89] Adnan AKBAR et al. « Predictive Analytics for Complex IoT Data Streams ». In : *IEEE Internet of Things J.* 4.5 (2017), p. 1571-1582. DOI : [10.1109/jiot.2017.2712672](https://doi.org/10.1109/jiot.2017.2712672) (cf. p. 36, 37).
- [90] Ali R. ABDELALAH et al. « Deep Learning for IoT Traffic Prediction Based on Edge Computing ». In : *Distributed Computer and Communication Networks : Control, Computation, Communications*. Sous la dir. de V.M. VISHNEVSKIY, K.E. SAMOUYLOV et D.V. KOZYREV. T. 1337. Communications in Computer and Information Science. Springer, 2020, p. 2-15. DOI : [10.1007/978-3-030-66242-4_2](https://doi.org/10.1007/978-3-030-66242-4_2) (cf. p. 36, 37, 44).
- [91] Volkov ARTEM et al. « Novel AI-Based Scheme for Traffic Detection and Recognition in 5G Based Networks ». In : *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*. 2019, p. 243-255 (cf. p. 36, 37).
- [92] Manuel LOPEZ-MARTIN, Belen CARRO et Antonio SANCHEZ-ESGUEVILLAS. « Neural network architecture based on gradient boosting for IoT traffic prediction ». In : *Future Generation Computer Systems* 100 (2019), p. 656-673. DOI : <https://doi.org/10.1016/j.future.2019.05.060> (cf. p. 36, 37, 44).
- [93] Ali R. ABDELALAH, Omar ABDULKAREEM MAHMOOD et Andrey KOUCHERYAVY. « Delay prediction in IoT using Machine Learning Approach ». In : *2020 12th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*. 2020, p. 275-279. DOI : [10.1109/ICUMT51630.2020.9222245](https://doi.org/10.1109/ICUMT51630.2020.9222245) (cf. p. 36, 43).
- [94] Muhammad ATEEQ et al. « Predicting Delay in IoT Using Deep Learning : A Multiparametric Approach ». In : *IEEE Access* 7 (2019), p. 62022-62031. DOI : [10.1109/ACCESS.2019.2915958](https://doi.org/10.1109/ACCESS.2019.2915958) (cf. p. 36, 37, 43).
- [95] G. WHITE et al. « IoTredict : Collaborative QoS Prediction in IoT ». In : *IEEE Int. Conf. on Pervasive Computing and Communications (PerCom)*. 2018, p. 1-10. DOI : [10.1109/PERCOM.2018.8444598](https://doi.org/10.1109/PERCOM.2018.8444598) (cf. p. 37).
- [96] G. WHITE et al. « Quantitative Evaluation of QoS Prediction in IoT ». In : *IEEE/IFIP Int. Conf. on Dependable Systems and Networks*. 2017, p. 61-66. DOI : [10.1109/DSN-W.2017.26](https://doi.org/10.1109/DSN-W.2017.26) (cf. p. 37).

- [97] Mridula BATRA et Rashmi AGRAWAL. « Comparative Analysis of Decision Tree Algorithms ». In : *Nature Inspired Computing*. Sous la dir. de Bijaya Ketan PANIGRAHI et al. Singapore : Springer Singapore, 2018, p. 31-36 (cf. p. 38).
- [98] Jerome H. FRIEDMAN. « Greedy Function Approximation : A Gradient Boosting Machine ». In : *Annals of Statistics* 29.5 (2001), p. 1189-1232 (cf. p. 38).
- [99] Alex J. SMOLA et Bernhard SCHOLKOPF. « A Tutorial on Support Vector Regression ». In : *Journal of Machine Learning Research (JMLR)* 3 (2004), p. 951-965 (cf. p. 38).
- [100] Christopher M. BISHOP. *Pattern Recognition and Machine Learning*. Springer, 2006 (cf. p. 39).
- [101] Jacob COHEN et al. *Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences*. Routledge, 2013 (cf. p. 39).
- [102] Omar SAID et Amr TOLBA. « Accurate performance prediction of IoT communication systems for smart cities : An efficient deep learning based solution ». In : *Sustainable Cities and Society* 69 (2021), p. 102830. DOI : <https://doi.org/10.1016/j.scs.2021.102830> (cf. p. 43).
- [103] Yafei HOU et al. « A Study of Throughput Prediction using Convolutional Neural Network over Factory Environment ». In : *2021 23rd International Conference on Advanced Communication Technology (ICACT)*. 2021, p. 429-434. DOI : [10.23919/ICACT51234.2021.9370905](https://doi.org/10.23919/ICACT51234.2021.9370905) (cf. p. 43).
- [104] Xiaochen FAN et al. « UrbanEdge : Deep Learning Empowered Edge Computing for Urban IoT Time Series Prediction ». In : *Proceedings of the ACM Turing Celebration Conference - China*. ACM TURC '19. Chengdu, China : Association for Computing Machinery, 2019. DOI : [10.1145/3321408.3323089](https://doi.org/10.1145/3321408.3323089) (cf. p. 44).
- [105] Di WU et al. « EdgeLSTM : Towards Deep and Sequential Edge Computing for IoT Applications ». In : *IEEE/ACM Transactions on Networking* 29.4 (2021), p. 1895-1908. DOI : [10.1109/TNET.2021.3075468](https://doi.org/10.1109/TNET.2021.3075468) (cf. p. 44).
- [106] Ashish VASWANI et al. « Attention Is All You Need ». In : *CoRR* abs/1706.03762 (2017) (cf. p. 44, 46).
- [107] Chaoming SONG et al. « Limits of Predictability in Human Mobility ». In : *Science* 327.5968 (2010), p. 1018-1021. DOI : [10.1126/science.1177170](https://doi.org/10.1126/science.1177170) (cf. p. 53).
- [108] Marta C. GONZÁLEZ, César A. HIDALGO et Albert-László BARABÁSI. « Understanding individual human mobility patterns ». In : *Nature* 453 (7196 juin 2008), p. 779-782. DOI : [10.1038/nature06958](https://doi.org/10.1038/nature06958) (cf. p. 53).
- [109] Andres SEVTSUK et Carlo RATTI. « Does Urban Mobility Have a Daily Routine? Learning from the Aggregate Data of Mobile Networks ». In : *Journal of Urban Technology* 17.1 (2010), p. 41-60. DOI : [10.1080/10630731003597322](https://doi.org/10.1080/10630731003597322) (cf. p. 53).
- [110] Clodoveu Augusto DAVIS et al. « Inferring the Location of Twitter Messages Based on User Relationships ». In : *Transactions in GIS* 15 (2011) (cf. p. 53).
- [111] Jiangchuan ZHENG et Lionel M. NI. « An Unsupervised Framework for Sensing Individual and Cluster Behavior Patterns from Human Mobile Data ». In : *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. UbiComp '12. Pittsburgh, Pennsylvania : Association for Computing Machinery, 2012, p. 153-162. DOI : [10.1145/2370216.2370241](https://doi.org/10.1145/2370216.2370241) (cf. p. 53).
- [112] Peng XIE et al. « Urban flows prediction from spatial-temporal data using machine learning : A survey ». In : *Inf. Fusion* 59 (2019), p. 1-12 (cf. p. 53).
- [113] Pierre DEVILLE et al. « Dynamic population mapping using mobile phone data ». In : *Proceedings of the National Academy of Sciences* 111.45 (2014), p. 15888-15893. DOI : [10.1073/pnas.1408439111](https://doi.org/10.1073/pnas.1408439111) (cf. p. 53).
- [114] Qi WANG et John E. TAYLOR. « Patterns and Limitations of Urban Human Mobility Resilience under the Influence of Multiple Types of Natural Disaster ». In : *PLOS* (jan. 2016). DOI : [10.1371/journal.pone.0147299](https://doi.org/10.1371/journal.pone.0147299) (cf. p. 54).
- [115] Xuan SONG et al. « Modeling and Probabilistic Reasoning of Population Evacuation during Large-Scale Disaster ». In : *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '13. Chicago, Illinois, USA : Association for Computing Machinery, 2013, p. 1231-1239. DOI : [10.1145/2487575.2488189](https://doi.org/10.1145/2487575.2488189) (cf. p. 54).

- [116] Christian SCHRECKENBERGER, Christian BARTELT et Heiner STUCKENSCHMIDT. « Enhancing a Crowd-Based Delivery Network with Mobility Predictions ». In : *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Prediction of Human Mobility*. PredictGIS'19. Chicago, IL, USA : Association for Computing Machinery, 2019, p. 66-75. DOI : [10.1145/3356995.3364542](https://doi.org/10.1145/3356995.3364542) (cf. p. 54).
- [117] Akihito SUDO et al. « Particle Filter for Real-Time Human Mobility Prediction Following Unprecedented Disaster ». In : *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. SIGSPACIAL '16. Burlingame, California : Association for Computing Machinery, 2016. DOI : [10.1145/2996913.2997000](https://doi.org/10.1145/2996913.2997000) (cf. p. 54, 55, 57, 59, 64, 71).
- [118] Yu ZHENG et al. « Urban Computing : Concepts, Methodologies, and Applications ». In : *ACM Trans. Intell. Syst. Technol.* 5 (2014) (cf. p. 54).
- [119] Chao FAN, Xiangqi JIANG et Ali MOSTAFAVI. « Adaptive Reinforcement Learning Model for Simulation of Urban Mobility during Crises ». In : *ArXiv abs/2009.01359* (2020) (cf. p. 54).
- [120] Zipei FAN et al. « CityMomentum : An Online Approach for Crowd Behavior Prediction at a Citywide Level ». In : *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 2015 (cf. p. 55).
- [121] Renhe JIANG et al. « Deep ROI-Based Modeling for Urban Human Mobility Prediction ». In : *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2 (2018) (cf. p. 55).
- [122] Renhe JIANG et al. « DeepCrowd : A Deep Model for Large-Scale Citywide Crowd Density and Flow Prediction ». In : *IEEE Transactions on Knowledge and Data Engineering* (2021) (cf. p. 55).
- [123] Tatsuya KONISHI et al. « CityProphet : City-Scale Irregularity Prediction Using Transit App Logs ». In : *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 2016 (cf. p. 55).
- [124] Metin OZTURK et al. « A novel deep learning driven, low-cost mobility prediction approach for 5G cellular networks : The case of the Control/Data Separation Architecture (CDSA) ». In : *Neurocomputing* 358 (2019) (cf. p. 55).
- [125] Abdelrahim MOHAMED et al. « Mobility prediction for handover management in cellular networks with control/data separation ». In : *2015 IEEE International Conference on Communications (ICC)*. 2015 (cf. p. 55).
- [126] Hasan FAROOQ, Ahmad ASGHAR et Ali IMRAN. « Mobility Prediction-Based Autonomous Proactive Energy Saving (AURORA) Framework for Emerging Ultra-Dense Networks ». In : *IEEE Transactions on Green Communications and Networking* 2 (2018) (cf. p. 55).
- [127] Jan PLACHY, Zdenek BECVAR et Emilio Calvanese STRINATI. « Dynamic resource allocation exploiting mobility prediction in mobile edge computing ». In : *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. 2016 (cf. p. 56).
- [128] Amnir HADACHI et al. « Cell phone subscribers mobility prediction using enhanced Markov Chain algorithm ». In : *2014 IEEE Intelligent Vehicles Symposium Proceedings*. 2014 (cf. p. 56).
- [129] Andre S. GOMES et al. « Edge caching with mobility prediction in virtualized LTE mobile networks ». In : *Future Generation Computer Systems* 70 (2017) (cf. p. 56).
- [130] Hua WEI, Hong LUO et Yan SUN. « Mobility-Aware Service Caching in Mobile Edge Computing for Internet of Things ». In : *Sensors* 20 (2020) (cf. p. 56).
- [131] Teruo HIGASHINO et al. « Edge Computing and IoT Based Research for Building Safe Smart Cities Resistant to Disasters ». In : *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. 2017 (cf. p. 56).
- [132] Aditya TULSYAN, R. BHUSHAN GOPALUNI et Swanand R. KHARE. « Particle filtering without tears : A primer for beginners ». In : *Computers & Chemical Engineering* 95 (2016), p. 130-145. DOI : <https://doi.org/10.1016/j.compchemeng.2016.08.015> (cf. p. 57).
- [133] A. DOUCET, Simon J. GODSILL et Christophe ANDRIEU. « On sequential Monte Carlo sampling methods for Bayesian filtering ». In : *Statistics and Computing* 10 (2000), p. 197-208 (cf. p. 57).

- [134] Yossi RUBNER, Carlo TOMASI et Leonidas J. GUIBAS. « The Earth Mover's Distance as a Metric for Image Retrieval ». In : *International Journal of Computer Vision* 40.2 (nov. 2000), p. 99-121. DOI : [10.1023/A:1026543900054](https://doi.org/10.1023/A:1026543900054) (cf. p. 58).
- [135] Xiaohong SHENG, Yu-Hen HU et Parameswaran RAMANATHAN. « Distributed Particle Filter with GMM Approximation for Multiple Targets Localization and Tracking in Wireless Sensor Network ». In : *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*. IPSN '05. Los Angeles, California : IEEE Press, 2005, 24-es (cf. p. 58).
- [136] Dongbing GU. « Distributed Particle Filter for Target Tracking ». In : *Proceedings 2007 IEEE International Conference on Robotics and Automation*. 2007, p. 3856-3861. DOI : [10.1109/ROBOT.2007.364070](https://doi.org/10.1109/ROBOT.2007.364070) (cf. p. 58).
- [137] Boris N. ORESHKIN et Mark J. COATES. « Asynchronous distributed particle filter via decentralized evaluation of Gaussian products ». In : *2010 13th International Conference on Information Fusion*. 2010, p. 1-8. DOI : [10.1109/ICIF.2010.5712070](https://doi.org/10.1109/ICIF.2010.5712070) (cf. p. 58).
- [138] Ondrej HLINKA et al. « Likelihood Consensus and Its Application to Distributed Particle Filtering ». In : *IEEE Transactions on Signal Processing* 60.8 (2012), p. 4334-4349. DOI : [10.1109/TSP.2012.2196697](https://doi.org/10.1109/TSP.2012.2196697) (cf. p. 58).
- [139] Deniz ÜSTEBAY, Mark COATES et Michael RABBAT. « Distributed auxiliary particle filters using selective gossip ». In : *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2011, p. 3296-3299. DOI : [10.1109/ICASSP.2011.5946726](https://doi.org/10.1109/ICASSP.2011.5946726) (cf. p. 58).
- [140] Claudio J. BORDIN et Marcelo G. S. BRUNO. « Consensus-based distributed particle filtering algorithms for cooperative blind equalization in receiver networks ». In : *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2011, p. 3968-3971. DOI : [10.1109/ICASSP.2011.5947221](https://doi.org/10.1109/ICASSP.2011.5947221) (cf. p. 58).
- [141] Dongbing Gu et al. « Consensus based distributed particle filter in sensor networks ». In : *2008 International Conference on Information and Automation*. 2008, p. 302-307. DOI : [10.1109/ICINFA.2008.4608015](https://doi.org/10.1109/ICINFA.2008.4608015) (cf. p. 58).
- [142] Arash MOHAMMADI et Amir ASIF. « Consensus-based distributed unscented particle filter ». In : *2011 IEEE Statistical Signal Processing Workshop (SSP)*. 2011, p. 237-240. DOI : [10.1109/SSP.2011.5967669](https://doi.org/10.1109/SSP.2011.5967669) (cf. p. 58).
- [143] Saptarshi BANDYOPADHYAY et Soon-Jo CHUNG. « Distributed estimation using Bayesian consensus filtering ». In : *2014 American Control Conference (2014)*, p. 634-641 (cf. p. 58).
- [144] Matthew ROSENCRANTZ, Geoffrey J. GORDON et Sebastian THRUN. « Decentralized Sensor Fusion with Distributed Particle Filters ». In : *Conference on Uncertainty in Artificial Intelligence*. 2002 (cf. p. 58, 59).
- [145] Petar M. DJURIĆ, Jonathan BEAUDEAU et Mónica F. BUGALLO. « Non-centralized target tracking with mobile agents ». In : *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2011, p. 5928-5931. DOI : [10.1109/ICASSP.2011.5947711](https://doi.org/10.1109/ICASSP.2011.5947711) (cf. p. 58, 59).
- [146] Chang LIU, Shengbo Eben LI et J. Karl HEDRICK. « Measurement Dissemination-Based Distributed Bayesian Filter Using the Latest-In-and-Full-Out Exchange Protocol for Networked Unmanned Vehicles ». In : *IEEE Transactions on Industrial Electronics* 64.11 (2017), p. 8756-8766. DOI : [10.1109/TIE.2017.2698375](https://doi.org/10.1109/TIE.2017.2698375) (cf. p. 58, 59).
- [147] *Dépot prédiction densité*. <https://gitlab.inria.fr/nsanti/mobility-prediction> (cf. p. 59).
- [148] Sylvia T. KOUYOUMDJIEVA, Peter DANIELIS et Gunnar KARLSSON. « Survey of Non-Image-Based Approaches for Counting People ». In : *IEEE Communications Surveys & Tutorials* 22.2 (2020), p. 1305-1336. DOI : [10.1109/COMST.2019.2902824](https://doi.org/10.1109/COMST.2019.2902824) (cf. p. 59).
- [149] H. Hakan CETINKAYA et Muammer AKCAY. « People Counting at Campuses ». In : *Procedia - Social and Behavioral Sciences* 182 (2015). 4th WORLD CONFERENCE on EDUCATIONAL TECHNOLOGY RESEARCHES (WCETR-2014), p. 732-736. DOI : <https://doi.org/10.1016/j.sbspro.2015.04.821> (cf. p. 59).
- [150] Adal ALASHBAN et al. « Single Convolutional Neural Network With Three Layers Model for Crowd Density Estimation ». In : *IEEE Access* 10 (2022), p. 63823-63833. DOI : [10.1109/ACCESS.2022.3180738](https://doi.org/10.1109/ACCESS.2022.3180738) (cf. p. 59).

- [151] Aravinda S. RAO et al. « Crowd density estimation based on optical flow and hierarchical clustering ». In : *2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. 2013, p. 494-499. DOI : [10.1109/ICACCI.2013.6637221](https://doi.org/10.1109/ICACCI.2013.6637221) (cf. p. 59).
- [152] Kyosuke SHIBATA et Hiroshi YAMAMOTO. « People Crowd Density Estimation System using Deep Learning for Radio Wave Sensing of Cellular Communication ». In : *2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIIC)*. 2019, p. 143-148. DOI : [10.1109/ICAIIIC.2019.8669071](https://doi.org/10.1109/ICAIIIC.2019.8669071) (cf. p. 59).
- [153] Ondrej HLINKA, Franz HLAWATSCH et Petar M. DJURIC. « Distributed particle filtering in agent networks : A survey, classification, and comparison ». In : *IEEE Signal Processing Magazine* 30.1 (2013), p. 61-81. DOI : [10.1109/MSP.2012.2219652](https://doi.org/10.1109/MSP.2012.2219652) (cf. p. 60).
- [154] Everaldo LEME et al. « Analyzing Gossip Protocols for Reliable MANET Applications ». In : *2017 IEEE International Conference on Edge Computing (EDGE)*. 2017, p. 98-105. DOI : [10.1109/IEEE.EDGE.2017.21](https://doi.org/10.1109/IEEE.EDGE.2017.21) (cf. p. 60).
- [155] H.T. FRIIS. « A Note on a Simple Transmission Formula ». In : *Proceedings of the IRE* 34.5 (1946), p. 254-256. DOI : [10.1109/JRPR0C.1946.234568](https://doi.org/10.1109/JRPR0C.1946.234568) (cf. p. 62).
- [156] Hanene BRAHMIA et Cherif TOLBA. « Nakagami fading impact on the performances of VANET routing protocols in a Realistic Urban Area setting ». In : *International Journal of Advanced Networking and Applications* 11.4 (2020), p. 4330-4335 (cf. p. 62).
- [157] O. MICHEL. « Webots : Professional Mobile Robot Simulation ». In : *Journal of Advanced Robotics Systems* 1.1 (2004), p. 39-42 (cf. p. 62).
- [158] Yu ZHENG et al. *Geolife GPS trajectory dataset - User Guide*. Geolife GPS trajectories 1.1. Juill. 2011 (cf. p. 63).
- [159] George EP Box et al. *Time series analysis : forecasting and control*. John Wiley & Sons, 2015 (cf. p. 63).
- [160] Bing LIU. « Lifelong Machine Learning : A Paradigm for Continuous Learning ». In : *Frontiers of Computer Science* 11.3 (juin 2017), p. 359-361. DOI : [10.1007/s11704-016-6903-6](https://doi.org/10.1007/s11704-016-6903-6) (cf. p. 64).
- [161] Quang PHAM et al. *Learning Fast and Slow for Online Time Series Forecasting*. 2022 (cf. p. 64).
- [162] Jienan CHEN et al. « iRAF : A Deep Reinforcement Learning Approach for Collaborative Mobile Edge Computing IoT Networks ». In : *IEEE Internet of Things Journal* 6.4 (2019), p. 7011-7024. DOI : [10.1109/JIOT.2019.2913162](https://doi.org/10.1109/JIOT.2019.2913162) (cf. p. 76, 78, 79, 84).
- [163] Pengfei WANG et al. « HetMEC : Latency-Optimal Task Assignment and Resource Allocation for Heterogeneous Multi-Layer Mobile Edge Computing ». In : *IEEE Transactions on Wireless Communications* 18.10 (2019), p. 4942-4956. DOI : [10.1109/TWC.2019.2931315](https://doi.org/10.1109/TWC.2019.2931315) (cf. p. 76, 78, 79).
- [164] Jiao ZHANG et al. « Joint Resource Allocation for Latency-Sensitive Services Over Mobile Edge Computing Networks With Caching ». In : *IEEE Internet of Things Journal* 6.3 (2019), p. 4283-4294. DOI : [10.1109/JIOT.2018.2875917](https://doi.org/10.1109/JIOT.2018.2875917) (cf. p. 76, 77, 80, 82).
- [165] Lei YANG et al. « Multi-User Computation Partitioning for Latency Sensitive Mobile Cloud Applications ». In : *IEEE Transactions on Computers* 64.8 (2015), p. 2253-2266. DOI : [10.1109/TC.2014.2366735](https://doi.org/10.1109/TC.2014.2366735) (cf. p. 76).
- [166] Linpei LI et al. « Energy-Efficient UAV-Enabled MEC System : Bits Allocation Optimization and Trajectory Design ». In : *Sensors (Basel, Switzerland)* 19 (2019) (cf. p. 76, 77, 79, 80, 84-86).
- [167] Tuyen X. TRAN et Dario POMPILI. « Joint Task Offloading and Resource Allocation for Multi-Server Mobile-Edge Computing Networks ». In : *IEEE Transactions on Vehicular Technology* 68.1 (2019), p. 856-868. DOI : [10.1109/TVT.2018.2881191](https://doi.org/10.1109/TVT.2018.2881191) (cf. p. 76-80, 82).
- [168] Stefania SARDELLITI, Mattia MERLUZZI et Sergio BARBAROSSA. « Optimal Association of Mobile Users to Multi-Access Edge Computing Resources ». In : *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*. 2018, p. 1-6. DOI : [10.1109/ICCW.2018.8403594](https://doi.org/10.1109/ICCW.2018.8403594) (cf. p. 76, 79, 83).
- [169] Rania ISLAMBOULI et Sanaa SHARAFEDDINE. « Optimized 3D Deployment of UAV-Mounted Cloudlets to Support Latency-Sensitive Services in IoT Networks ». In : *IEEE Access* 7 (2019), p. 172860-172870. DOI : [10.1109/ACCESS.2019.2956150](https://doi.org/10.1109/ACCESS.2019.2956150) (cf. p. 76, 85, 87).

- [170] Sladana JOSILO et Gyorgy DAN. « Joint Allocation of Computing and Wireless Resources to Autonomous Devices in Mobile Edge Computing ». In : *Proceedings of the 2018 Workshop on Mobile Edge Communications. MECOMM'18*. Budapest, Hungary : Association for Computing Machinery, 2018, p. 13-18. DOI : [10.1145/3229556.3229559](https://doi.org/10.1145/3229556.3229559) (cf. p. 76, 77, 83).
- [171] Jinke REN et al. « Latency Optimization for Resource Allocation in Mobile-Edge Computation Offloading ». In : *IEEE Transactions on Wireless Communications* 17.8 (2018), p. 5506-5519. DOI : [10.1109/TWC.2018.2845360](https://doi.org/10.1109/TWC.2018.2845360) (cf. p. 76, 79-81).
- [172] Ata KHALILI, Sheyda ZARANDI et Mehdi RASTI. « Joint Resource Allocation and Offloading Decision in Mobile Edge Computing ». In : *IEEE Communications Letters* 23.4 (2019), p. 684-687. DOI : [10.1109/LCOMM.2019.2897008](https://doi.org/10.1109/LCOMM.2019.2897008) (cf. p. 76-78).
- [173] Jiao ZHANG et al. « Computation-Efficient Offloading and Trajectory Scheduling for Multi-UAV Assisted Mobile Edge Computing ». In : *IEEE Transactions on Vehicular Technology* 69.2 (2020), p. 2114-2125. DOI : [10.1109/TVT.2019.2960103](https://doi.org/10.1109/TVT.2019.2960103) (cf. p. 76, 77, 85, 86).
- [174] Nengyu ZHU et al. « Sleep-Scheduling and Joint Computation-Communication Resource Allocation in MEC Networks for 5G IIoT ». In : *2021 IEEE Wireless Communications and Networking Conference (WCNC)*. 2021, p. 1-7. DOI : [10.1109/WCNC49053.2021.9417322](https://doi.org/10.1109/WCNC49053.2021.9417322) (cf. p. 76, 77, 82).
- [175] Mohammad Arif HOSSAIN et Nirwan ANSARI. « Energy Aware Latency Minimization for Network Slicing Enabled Edge Computing ». In : *IEEE Transactions on Green Communications and Networking* 5.4 (2021), p. 2150-2159. DOI : [10.1109/TGCN.2021.3083153](https://doi.org/10.1109/TGCN.2021.3083153) (cf. p. 76, 77).
- [176] Quoc-Viet PHAM et al. « Mobile Edge Computing With Wireless Backhaul : Joint Task Offloading and Resource Allocation ». In : *IEEE Access* 7 (2019), p. 16444-16459. DOI : [10.1109/ACCESS.2018.2883692](https://doi.org/10.1109/ACCESS.2018.2883692) (cf. p. 77-79, 81).
- [177] Che-Wei Hsu, Yung-Lin Hsu et Hung-Yu WEI. « Energy-Efficient Edge Offloading in Heterogeneous Industrial IoT Networks for Factory of Future ». In : *IEEE Access* 8 (2020), p. 183035-183050. DOI : [10.1109/ACCESS.2020.3029253](https://doi.org/10.1109/ACCESS.2020.3029253) (cf. p. 77).
- [178] Evangelos K. MARKAKIS et al. « Efficient Next Generation Emergency Communications over Multi-Access Edge Computing ». In : *IEEE Communications Magazine* 55.11 (2017), p. 92-97. DOI : [10.1109/MCOM.2017.1700345](https://doi.org/10.1109/MCOM.2017.1700345) (cf. p. 77).
- [179] Marco MEZZAVILLA et al. « Public Safety Communications above 6 GHz : Challenges and Opportunities ». In : *IEEE Access* 6 (2018), p. 316-329. DOI : [10.1109/ACCESS.2017.2762471](https://doi.org/10.1109/ACCESS.2017.2762471) (cf. p. 77).
- [180] Yue WANG et al. « Effective Capacity-Based Resource Allocation in Mobile Edge Computing With Two-Stage Tandem Queues ». In : *IEEE Transactions on Communications* 67.9 (2019), p. 6221-6233. DOI : [10.1109/TCOMM.2019.2920835](https://doi.org/10.1109/TCOMM.2019.2920835) (cf. p. 77, 79, 82).
- [181] Tianyu YANG et al. « Deep Reinforcement Learning based Resource Allocation in Low Latency Edge Computing Networks ». In : *2018 15th International Symposium on Wireless Communication Systems (ISWCS)*. 2018, p. 1-5. DOI : [10.1109/ISWCS.2018.8491089](https://doi.org/10.1109/ISWCS.2018.8491089) (cf. p. 77, 78, 80).
- [182] Jingyu XIONG, Hongzhi GUO et Jiajia LIU. « Task Offloading in UAV-Aided Edge Computing : Bit Allocation and Trajectory Optimization ». In : *IEEE Communications Letters* 23.3 (2019), p. 538-541. DOI : [10.1109/LCOMM.2019.2891662](https://doi.org/10.1109/LCOMM.2019.2891662) (cf. p. 77, 85, 86).
- [183] Yuan LIU et al. « UAV-Assisted Wireless Powered Cooperative Mobile Edge Computing : Joint Offloading, CPU Control, and Trajectory Optimization ». In : *IEEE Internet of Things Journal* 7.4 (2020), p. 2777-2790. DOI : [10.1109/JIOT.2019.2958975](https://doi.org/10.1109/JIOT.2019.2958975) (cf. p. 77, 79, 84-86).
- [184] Zhe YU et al. « Joint Task Offloading and Resource Allocation in UAV-Enabled Mobile Edge Computing ». In : *IEEE Internet of Things Journal* 7.4 (2020), p. 3147-3159. DOI : [10.1109/JIOT.2020.2965898](https://doi.org/10.1109/JIOT.2020.2965898) (cf. p. 77, 86).
- [185] Tiankui ZHANG et al. « Joint Computation and Communication Design for UAV-Assisted Mobile Edge Computing in IIoT ». In : *IEEE Transactions on Industrial Informatics* 16.8 (2020), p. 5505-5516. DOI : [10.1109/TII.2019.2948406](https://doi.org/10.1109/TII.2019.2948406) (cf. p. 77, 84-86).

- [186] Rong CHAI et al. « Task Execution Cost Minimization-Based Joint Computation Offloading and Resource Allocation for Cellular D2D MEC Systems ». In : *IEEE Systems Journal* 13.4 (2019), p. 4110-4121. DOI : [10.1109/JSYST.2019.2921115](https://doi.org/10.1109/JSYST.2019.2921115) (cf. p. 77, 78).
- [187] Qiang FAN et Nirwan ANSARI. « Application Aware Workload Allocation for Edge Computing-Based IoT ». In : *IEEE Internet of Things Journal* 5.3 (2018), p. 2146-2153. DOI : [10.1109/JIOT.2018.2826006](https://doi.org/10.1109/JIOT.2018.2826006) (cf. p. 78-80, 82).
- [188] Xiaowen CAO et al. « Joint Computation and Communication Cooperation for Energy-Efficient Mobile Edge Computing ». In : *IEEE Internet of Things Journal* 6.3 (2019), p. 4188-4200. DOI : [10.1109/JIOT.2018.2875246](https://doi.org/10.1109/JIOT.2018.2875246) (cf. p. 78, 81).
- [189] Xinchen LYU et al. « Energy-Efficient Admission of Delay-Sensitive Tasks for Mobile Edge Computing ». In : *IEEE Transactions on Communications* 66.6 (2018), p. 2603-2616. DOI : [10.1109/TCOMM.2018.2799937](https://doi.org/10.1109/TCOMM.2018.2799937) (cf. p. 79, 83).
- [190] Jiao ZHANG et al. « Energy-Latency Tradeoff for Energy-Aware Offloading in Mobile Edge Computing Networks ». In : *IEEE Internet of Things Journal* 5.4 (2018), p. 2633-2645. DOI : [10.1109/JIOT.2017.2786343](https://doi.org/10.1109/JIOT.2017.2786343) (cf. p. 79, 82).
- [191] Ji LI et al. « Deep reinforcement learning based computation offloading and resource allocation for MEC ». In : *2018 IEEE Wireless Communications and Networking Conference (WCNC)*. 2018, p. 1-6. DOI : [10.1109/WCNC.2018.8377343](https://doi.org/10.1109/WCNC.2018.8377343) (cf. p. 79, 84).
- [192] Songtao GUO et al. « Energy-Efficient Dynamic Computation Offloading and Cooperative Task Scheduling in Mobile Cloud Computing ». In : *IEEE Transactions on Mobile Computing* 18.2 (2019), p. 319-333. DOI : [10.1109/TMC.2018.2831230](https://doi.org/10.1109/TMC.2018.2831230) (cf. p. 79, 82).
- [193] Song YANG et al. « Cloudlet Placement and Task Allocation in Mobile Edge Computing ». In : *IEEE Internet of Things Journal* 6.3 (2019), p. 5853-5863. DOI : [10.1109/JIOT.2019.2907605](https://doi.org/10.1109/JIOT.2019.2907605) (cf. p. 79, 82).
- [194] Min CHEN et Yixue HAO. « Task Offloading for Mobile Edge Computing in Software Defined Ultra-Dense Network ». In : *IEEE Journal on Selected Areas in Communications* 36.3 (2018), p. 587-597. DOI : [10.1109/JSAC.2018.2815360](https://doi.org/10.1109/JSAC.2018.2815360) (cf. p. 79, 81).
- [195] Chen-Feng LIU et al. « Dynamic Task Offloading and Resource Allocation for Ultra-Reliable Low-Latency Edge Computing ». In : *IEEE Transactions on Communications* 67.6 (2019), p. 4132-4150. DOI : [10.1109/TCOMM.2019.2898573](https://doi.org/10.1109/TCOMM.2019.2898573) (cf. p. 79, 80).
- [196] Pengfei WANG et al. « Joint Task Assignment, Transmission, and Computing Resource Allocation in Multilayer Mobile Edge Computing Systems ». In : *IEEE Internet of Things Journal* 6.2 (2019), p. 2872-2884. DOI : [10.1109/JIOT.2018.2876198](https://doi.org/10.1109/JIOT.2018.2876198) (cf. p. 79, 83).
- [197] Jiadai WANG et al. « Smart Resource Allocation for Mobile Edge Computing : A Deep Reinforcement Learning Approach ». In : *IEEE Transactions on Emerging Topics in Computing* 9.3 (2021), p. 1529-1541. DOI : [10.1109/TETC.2019.2902661](https://doi.org/10.1109/TETC.2019.2902661) (cf. p. 79, 84).
- [198] Lei FENG et al. « Energy-Efficient Offloading for Mission-Critical IoT Services Using EVT-Embedded Intelligent Learning ». In : *IEEE Transactions on Green Communications and Networking* 5.3 (2021), p. 1179-1190. DOI : [10.1109/TGCN.2021.3072140](https://doi.org/10.1109/TGCN.2021.3072140) (cf. p. 80).
- [199] Jianhui LIU et Qi ZHANG. « Adaptive Task Partitioning at Local Device or Remote Edge Server for Offloading in MEC ». In : *2020 IEEE Wireless Communications and Networking Conference (WCNC)*. 2020, p. 1-6. DOI : [10.1109/WCNC45663.2020.9120484](https://doi.org/10.1109/WCNC45663.2020.9120484) (cf. p. 80).
- [200] Mohammed S. ELBAMBY et al. « Wireless Edge Computing With Latency and Reliability Guarantees ». In : *Proceedings of the IEEE* 107.8 (2019), p. 1717-1737. DOI : [10.1109/JPR0C.2019.2917084](https://doi.org/10.1109/JPR0C.2019.2917084) (cf. p. 80).
- [201] David OHMANN, Meryem SIMSEK et Gerhard P. FETTWEIS. « Achieving high availability in wireless networks by an optimal number of Rayleigh-fading links ». In : *2014 IEEE Globecom Workshops (GC Wkshps)*. 2014, p. 1402-1407. DOI : [10.1109/GL0COMM.2014.7063630](https://doi.org/10.1109/GL0COMM.2014.7063630) (cf. p. 80).
- [202] Muhammad Adeel MAHMOOD, Winston K.G. SEAH et Ian WELCH. « Reliability in wireless sensor networks : A survey and challenges ahead ». In : *Computer Networks* 79 (2015), p. 166-187. DOI : <https://doi.org/10.1016/j.comnet.2014.12.016> (cf. p. 80).

- [203] Elie El HABER et al. « UAV-Aided Ultra-Reliable Low-Latency Computation Offloading in Future IoT Networks ». In : *IEEE Transactions on Communications* 69.10 (2021), p. 6838-6851. DOI : [10.1109/TCOMM.2021.3096559](https://doi.org/10.1109/TCOMM.2021.3096559) (cf. p. 80, 83, 86).
- [204] Mehdi BENNIS, Mérouane DEBBAH et H. Vincent POOR. « Ultrareliable and Low-Latency Wireless Communication : Tail, Risk, and Scale ». In : *Proceedings of the IEEE* 106.10 (2018), p. 1834-1853. DOI : [10.1109/JPROC.2018.2867029](https://doi.org/10.1109/JPROC.2018.2867029) (cf. p. 80).
- [205] Huilin LI et al. « Joint Optimization Strategy of Computation Offloading and Resource Allocation in Multi-Access Edge Computing Environment ». In : *IEEE Transactions on Vehicular Technology* 69.9 (2020), p. 10214-10226. DOI : [10.1109/TVT.2020.3003898](https://doi.org/10.1109/TVT.2020.3003898) (cf. p. 81).
- [206] Qiuping LI, Junhui ZHAO et Yi GONG. « Computation offloading and resource allocation for mobile edge computing with multiple access points ». In : *IET Communications* 13.17 (2019), p. 2668-2677. DOI : <https://doi.org/10.1049/iet-com.2019.0446> (cf. p. 82).
- [207] Stephen BOYD et al. « Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers ». In : *Foundations and Trends® in Machine Learning* 3.1 (2011), p. 1-122. DOI : [10.1561/22000000016](https://doi.org/10.1561/22000000016) (cf. p. 82).
- [208] Lin WANG et al. « MOERA : Mobility-Agnostic Online Resource Allocation for Edge Computing ». In : *IEEE Transactions on Mobile Computing* 18.8 (2019), p. 1843-1856. DOI : [10.1109/TMC.2018.2867520](https://doi.org/10.1109/TMC.2018.2867520) (cf. p. 82).
- [209] Niv BUCHBINDER, Shahar CHEN et Joseph (Seffi) NAOR. « Competitive Analysis via Regularization ». In : *Proceedings of the 2014 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, p. 436-444. DOI : [10.1137/1.9781611973402.32](https://doi.org/10.1137/1.9781611973402.32) (cf. p. 83).
- [210] Zhi LI et Qi ZHU. « Genetic Algorithm-Based Optimization of Offloading and Resource Allocation in Mobile-Edge Computing ». In : *Information* 11.2 (fév. 2020), p. 83. DOI : [10.3390/info11020083](https://doi.org/10.3390/info11020083) (cf. p. 83).
- [211] Liangtian WAN et al. « Task-Driven Resource Assignment in Mobile Edge Computing Exploiting Evolutionary Computation ». In : *IEEE Wireless Communications* 26.6 (2019), p. 94-101. DOI : [10.1109/MWC.001.1800582](https://doi.org/10.1109/MWC.001.1800582) (cf. p. 83).
- [212] Sihua WANG et al. « A Machine Learning Approach for Task and Resource Allocation in Mobile-Edge Computing-Based Networks ». In : *IEEE Internet of Things Journal* 8.3 (2021), p. 1358-1372. DOI : [10.1109/JIOT.2020.3011286](https://doi.org/10.1109/JIOT.2020.3011286) (cf. p. 84).
- [213] Qiyu HU et al. « Joint Offloading and Trajectory Design for UAV-Enabled Mobile Edge Computing Systems ». In : *IEEE Internet of Things Journal* 6.2 (2019), p. 1879-1892. DOI : [10.1109/JIOT.2018.2878876](https://doi.org/10.1109/JIOT.2018.2878876) (cf. p. 84).
- [214] Yong WANG et al. « Joint Deployment and Task Scheduling Optimization for Large-Scale Mobile Users in Multi-UAV-Enabled Mobile Edge Computing ». In : *IEEE Transactions on Cybernetics* 50.9 (2020), p. 3984-3997. DOI : [10.1109/TCYB.2019.2935466](https://doi.org/10.1109/TCYB.2019.2935466) (cf. p. 85, 86).
- [215] Lei YANG et al. « Multi-UAV-Enabled Load-Balance Mobile-Edge Computing for IoT Networks ». In : *IEEE Internet of Things Journal* 7.8 (2020), p. 6898-6908. DOI : [10.1109/JIOT.2020.2971645](https://doi.org/10.1109/JIOT.2020.2971645) (cf. p. 85, 87).
- [216] Mushu LI et al. « Energy-Efficient UAV-Assisted Mobile Edge Computing : Resource Allocation and Trajectory Optimization ». In : *IEEE Transactions on Vehicular Technology* 69.3 (2020), p. 3424-3438. DOI : [10.1109/TVT.2020.2968343](https://doi.org/10.1109/TVT.2020.2968343) (cf. p. 85).
- [217] Yuwen QIAN et al. « User Association and Path Planning for UAV-Aided Mobile Edge Computing With Energy Restriction ». In : *IEEE Wireless Communications Letters* 8.5 (2019), p. 1312-1315. DOI : [10.1109/LWC.2019.2913843](https://doi.org/10.1109/LWC.2019.2913843) (cf. p. 85).
- [218] Shuo WAN et al. « Toward Big Data Processing in IoT : Path Planning and Resource Management of UAV Base Stations in Mobile-Edge Computing System ». In : *IEEE Internet of Things Journal* 7.7 (2020), p. 5995-6009. DOI : [10.1109/JIOT.2019.2954825](https://doi.org/10.1109/JIOT.2019.2954825) (cf. p. 85).

- [219] Gaoxiang WU et al. « Energy efficient for UAV-enabled mobile edge computing networks : Intelligent task prediction and offloading ». In : *Computer Communications* 150 (2020), p. 556-562. DOI : <https://doi.org/10.1016/j.comcom.2019.11.037> (cf. p. 85).
- [220] Meisam RAZAVIYAYN et al. « Parallel Successive Convex Approximation for Nonsmooth Nonconvex Optimization ». In : *NIPS*. 2014 (cf. p. 86).
- [221] Fuhui ZHOU et al. « Computation Rate Maximization in UAV-Enabled Wireless-Powered Mobile-Edge Computing Systems ». In : *IEEE Journal on Selected Areas in Communications* 36.9 (2018), p. 1927-1941. DOI : [10.1109/JSAC.2018.2864426](https://doi.org/10.1109/JSAC.2018.2864426) (cf. p. 86).
- [222] Dorit S. HOCHBAUM et Anu PATHRIA. « Analysis of the greedy approach in problems of maximum k-coverage ». In : *Naval Research Logistics (NRL)* 45.6 (1998), p. 615-627. DOI : [https://doi.org/10.1002/\(SICI\)1520-6750\(199809\)45:6<615::AID-NAV5>3.0.CO;2-5](https://doi.org/10.1002/(SICI)1520-6750(199809)45:6<615::AID-NAV5>3.0.CO;2-5) (cf. p. 86).
- [223] Yuqing CHEN et Zhaohui ZHENG. « Joint Deployment and Task Computation of UAVs in UAV-assisted Edge Computing Network ». In : *2020 21st Asia-Pacific Network Operations and Management Symposium (APNOMS)*. 2020, p. 413-416. DOI : [10.23919/APNOMS50412.2020.9237019](https://doi.org/10.23919/APNOMS50412.2020.9237019) (cf. p. 86).
- [224] Jingrong WANG, Kaiyang LIU et Jianping PAN. « Online UAV-Mounted Edge Server Dispatching for Mobile-to-Mobile Edge Computing ». In : *IEEE Internet of Things Journal* 7.2 (2020), p. 1375-1386. DOI : [10.1109/JIOT.2019.2954798](https://doi.org/10.1109/JIOT.2019.2954798) (cf. p. 86).
- [225] Behzad JAVIDY, Abdolreza HATAMLOU et Seyedali MIRJALILI. « Ions motion algorithm for solving optimization problems ». In : *Applied Soft Computing* 32 (2015), p. 72-79. DOI : <https://doi.org/10.1016/j.asoc.2015.03.035> (cf. p. 86, 87).

